KATHOLISCHE UNIVERSITÄT
EICHSTÄTT-INGOLSTADT

Katholische Universität
Eichstätt-Ingolstadt

Parameter and its uncertainty Estimation

submitted by
Elshan Dashtiyev
Matriculation number: 297547

Faustino Vazquez Gabino
Matriculation number: 203961

Supervised by
Dr. Tijana Janjic
Dr. Maryam Ramezani Ziarani
WFI - Ingolstadt School of Management

Submission date:
July 18, 2024

Abstract

Shallow water equations (SWE) modeling is the process of using mathematical and computational methods to simulate the behavior of shallow water bodies, such as rivers, lakes, and coastal areas. This technique is widely used in various fields, including environmental science, civil engineering, and oceanography, to name a few. The goal of SWE modeling is to understand the dynamics of water flow and predict future states based on initial conditions and external influences. There are three primary approaches to SWE modeling: analytical, numerical, and hybrid models. Each approach has its strengths and weaknesses, and the choice of approach depends on the specific characteristics of the water body and the modeling problem. Accurate SWE modeling can have a significant impact on environmental management, disaster prevention, and infrastructure planning. It enables organizations to anticipate flooding events, optimize water resource management, design effective flood control measures, and assess the impact of environmental changes on water bodies.

The dataset we worked on consists of different parameters that include phic, hrain and alpha and other variables, for evaluating the performance of different SWE models. For this project, we will be focusing on first creating a proper dataset with uniformly randomly distributing the needed parameters to get results for our rain, height and velocity. Furthermore, to evaluate the performance of each model, we will use a range of metrics, including (MAPE) and (MASE). Our goal is to develop a model that can capture the patterns and correlations from the dataset that we created and use it to predict the initial parameters we used.

Contents

1 Introduction

1.1 Background Objective

Our project aims to develop a model for simulating the behavior of shallow water bodies using the modified shallow water equations (SWE). This model will help us understand and predict water flow dynamics in various environments.

The modified SWE model includes three key variables: velocity (u), height (h), and rain (r). Here's a brief overview:

Velocity (or wind) - u: It impacts the momentum and the diffusion of rain and height in the system. Height - h: This signifies fluid height level h fields Rain - r: This denotes the mass of rain in the system. It influences the geopotential and the momentum of the fluid.

The one-dimensional modified shallow-water model consists of following equations:

Equations

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{\partial(\phi + c^2 r)}{\partial x} = \beta_u + D_u \frac{\partial^2 u}{\partial x^2} \tag{1}$$

$$\phi = \begin{cases} \phi_c & \text{if } h > h_c \\ gh & \text{else} \end{cases} \tag{2}$$

$$\frac{\partial r}{\partial t} + u\frac{\partial r}{\partial x} = D_r \frac{\partial^2 r}{\partial x^2} - \alpha r - \begin{cases} \delta \frac{\partial u}{\partial x} & \text{if } h > h_r \text{ and } \frac{\partial u}{\partial x} < 0 \\ 0 & \text{else} \end{cases} \tag{3}$$

$$\frac{\partial h}{\partial t} + \frac{\partial(uh)}{\partial x} = D_h \frac{\partial^2 h}{\partial x^2} \tag{4}$$

These equations describe how the velocity, geopotential, rain, and height of the fluid change over time.

Our goal is to use these parameters (velocity, height, and rain) and use Machine Learning models to analyze their behavior to predict the target variables phic, alpha, and hrain.

1.2 Research Question

In the context of the modified shallow water equations model, is it possible to accurately predict the target variables constant (phic), (alpha), and height based on the observed behavior and interactions of the feature variables, specifically, velocity, rain, and height?

1.3 Materials and methods

In this project, we'll use Python and machine learning techniques to predict our target variables. We'll employ regression models, specifically Linear Regression, Random Forest, and Neural Networks, to understand the relationship between our features (height, rain, and velocity) and our target variables (phic, hrain and alpha)

2 Database

2.1 Requirements

The database should be composed of 100,000 samples. Each sample should contain six distinct entries. These entries include three target variables (phic, alpha, and hrain), each uniformly distributed. Additionally, there are three features (height, rain, and velocity), each represented by 250 grid points. This structure ensures a comprehensive and diverse dataset for our modeling purposes.

Challenges During the creation of the database, we encountered several challenges. First accessing the main code, then generating the data was significantly hard, as was obtaining the same values for the target variables for each run. Additionally, time constraints was a problem, with each run taking an average of 8 seconds. Given these difficulties, our advisor, Maryam, helped us <3. With her help, we were able to create a database consisting of 16,035 instances. Unfortunately, this is almost seven times smaller than our initial goal of 100,000 samples.

2.2 Target Variables

The target variables in our model are alpha, phic, and hrain, each with a specific role and uniformly distributed within certain bounds for each run:

alpha: This represents the half-life of the influence of rain, which is roughly 1 hour.

phic: This is the geopotential constant value above the first threshold that allows for unstable convection.

hrain: This is the height threshold for rain.

Variable	**Lower Bound**	**Upper Bound**
α	0.0003	0.001
ϕ_c	889.7	889.9
h_{rain}	90.15	90.25

Table 1: Uniform Distribution Bounds for Target Variables

2.3 Features

Velocity or wind (u): This feature signifies the speed and direction of the fluid flow. Height (h): This feature represents the fluid height level h fields in our model. Rain (r): This feature denotes the mass of rain in the system.

Understanding these features and their interactions is key to accurately predicting the target variables in our model.

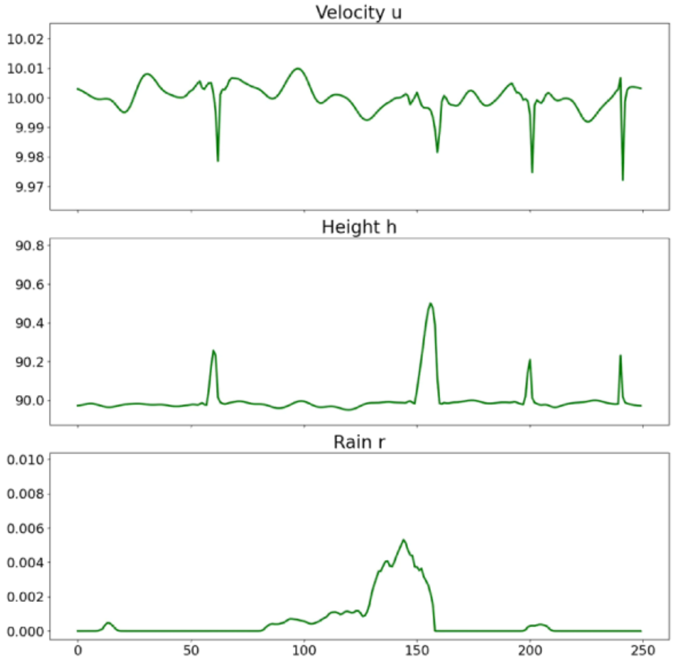


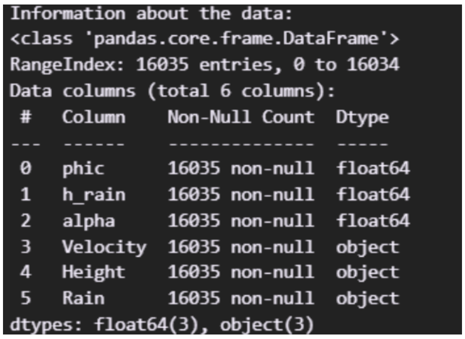
Figure 1: Example Features over 250 gridpoints(x-axis

3 Exploratory Data Analysis

3.1 Data Overview

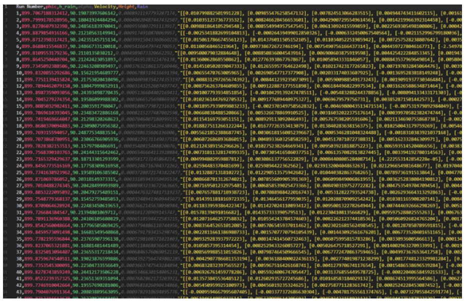
The data consists of 6 columns:

3 Target Variables : phic, alpha and hrain
3 Parameter Features : Velocity, height and rain



```
Information about the data:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16035 entries, 0 to 16034
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   phic      16035 non-null  float64
 1   h_rain    16035 non-null  float64
 2   alpha     16035 non-null  float64
 3   Velocity  16035 non-null  object
 4   Height    16035 non-null  object
 5   Rain      16035 non-null  object
dtypes: float64(3), object(3)
```

Figure 2: General Information about the data



```
Run Number,phic,h_rain,alpha,Velocity,Height,Rain
```

Figure 3: Preview of the data

The database is a Pandas Data Frame consisting of 6 columns, the first three are the target variables, and the values were generated uniformly distributed depending on the values from Table 1 1. The last three columns represent the features, each one of them is a numpy array consisting of 250 values (each value represents a grid point) as we can in the example 1

3.2 Uni-variate Analysis

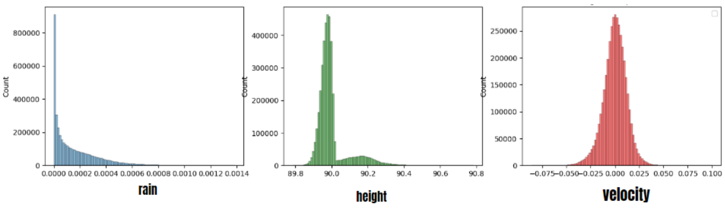


Figure 4: Histograms Features

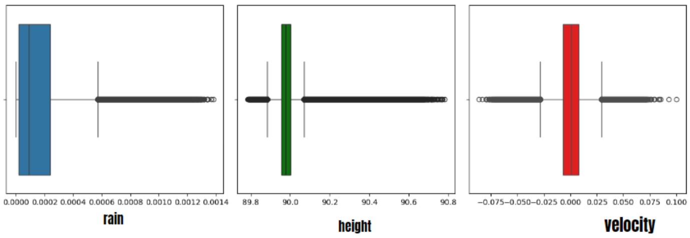


Figure 5: Box-plots Features

Distributions Features We can check the distribution of the data from the features in the plots of the histogram 4 and the box-plot5.

It is interesting to see that only for velocity the data is uniformly distributed and presents a great amount of values between 0, For rain and height the distribution is weird since most of them have more amount of values on the first amount of values, there is a clear sign that this can be related with the fact that when the height of the cloud increase there is a higher likelihood of rain. For this case specifically, since each instance is different and can affect the target variables we decide to don't drop outliers.

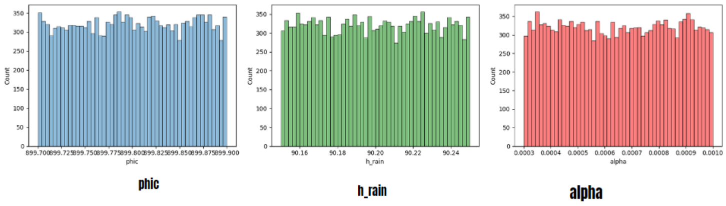


Figure 6: Histograms Target Variables

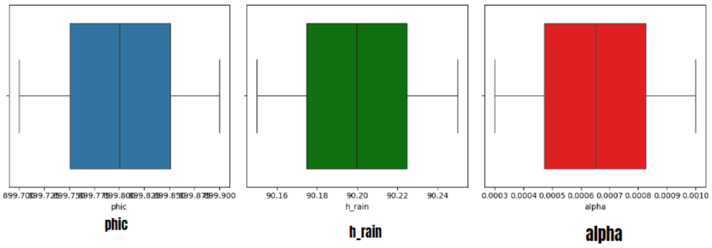


Figure 7: Box-plots Target Variables

Distributions Target Variables As we can see in the histogram of the target variables 6 and in the box-plot of the target variables 7, the data is uniformly distributed with the values shown in Table 1 1. It doesn't show any sign of outliers.

3.3 Multivariate Analysis

Pearson Correlation Heatmap As we can in the correlation plot 8, if we focus on the 9 values from the lower left corner those are the values we care about since is the correlation between the target variables and the features, we can see that the is no relation between the values, since all of them not even reach 0.1 and most of them keep values of order 10^{-3}, the highest value is negative correlation and hrain and height but still not good enough. So we can say now that the predictions can be not really good.

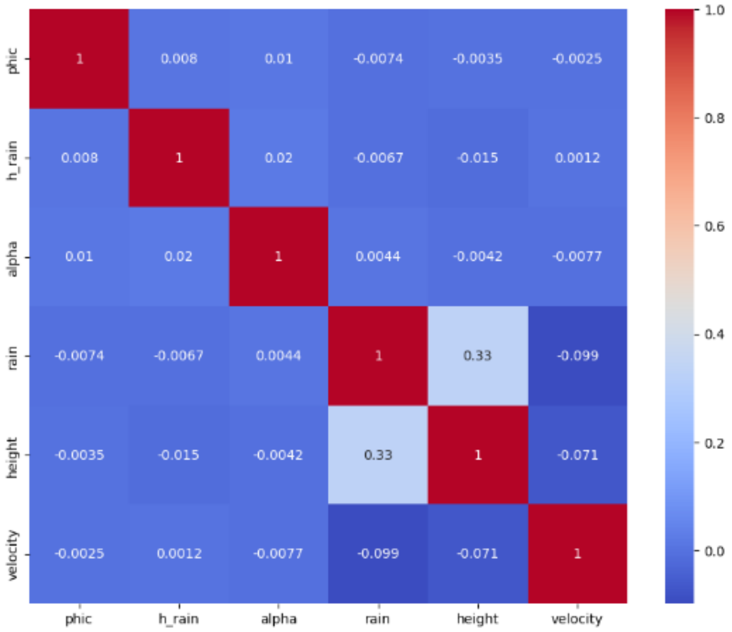


Figure 8: Heat map Variables(Pearson Correlation)

3.4 Metrics

It was hard to choose the metrics for the analysis and comparison of the models since we thought at the beginning of the project used common metrics like Root mean square error (RMSE) and R2 score (R2) nevertheless the values were close to 0 so it was hard to see a difference and a proper analysis of different values, so we decided to look for metrics whose values were easier to understand and compare, so we choose mean absolute percentage error(MAPE). mean absolute scaled error(MASE).

Mean Absolute Percentage Error(MAPE)

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{5}$$

Where:
n is the total number of data points.
y_i is the actual value for the ith data point.
$\hat{y}_i$ is the predicted value for the ith data point.

Mean Absolute Scaled Error(MASE)

$$MASE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{\frac{1}{n-1}\sum_{i=2}^{n}|y_i - y_{i-1}|} \tag{6}$$

Where:
n is the total number of data points.
y_i is the actual value for the ith data point.
$\hat{y}_i$ is the predicted value for the ith data point.
$y_i - 1$ isn't from a previous time step. Instead, it's our simple guess based on the last known value. This guess is what we call a naive forecast.

3.5 Conclusions

During this Exploratory Data Analysis, it is clear we are going to have many challenges to face during the implementation of Machine Learning models since there is no clear correlation as we could see in the Heat map Plot 8 between the target variables and the features. Still, we hope that the selected metrics can help us to analyze the behavior of the different prediction results and the same way we expect to choose the right Machine Learning for this specific problem.

4 Models Implementation

For this project we know that we are working with a Supervised Problem, specifically a Regression, for that reason we decided to implement Machine Learning models suitable for regression such as Linear Regression, Random Forest, Neural Network, and Bayesian Neural Network.

4.1 Model Selection

In this part of our project, we have decided to try different models. Such as Linear Regression, Random Forest and Neural Network

4.2 Linear Regression

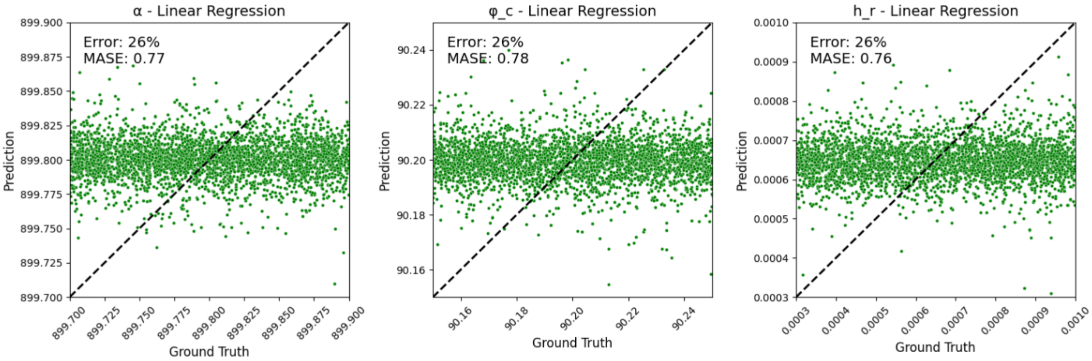


Figure 9: Linear Regression Original Prediction

Original Data Prediction The original data prediction we obtained shows that the Linear Regression model is doing quite badly in predicting our parameters. Interestingly, for linear regression, it mainly predicts between specific values (ex: 0.0005-0.0008).

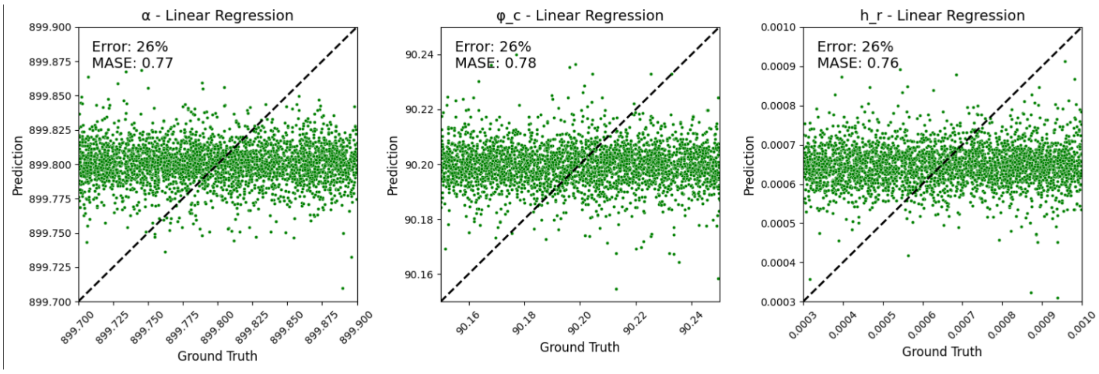


Figure 10: Linear Regression Standart Scaled

Standard Scaled Data Prediction There was a slight improvement in Linear Regression, however, no change was seen on NN for standard scaler

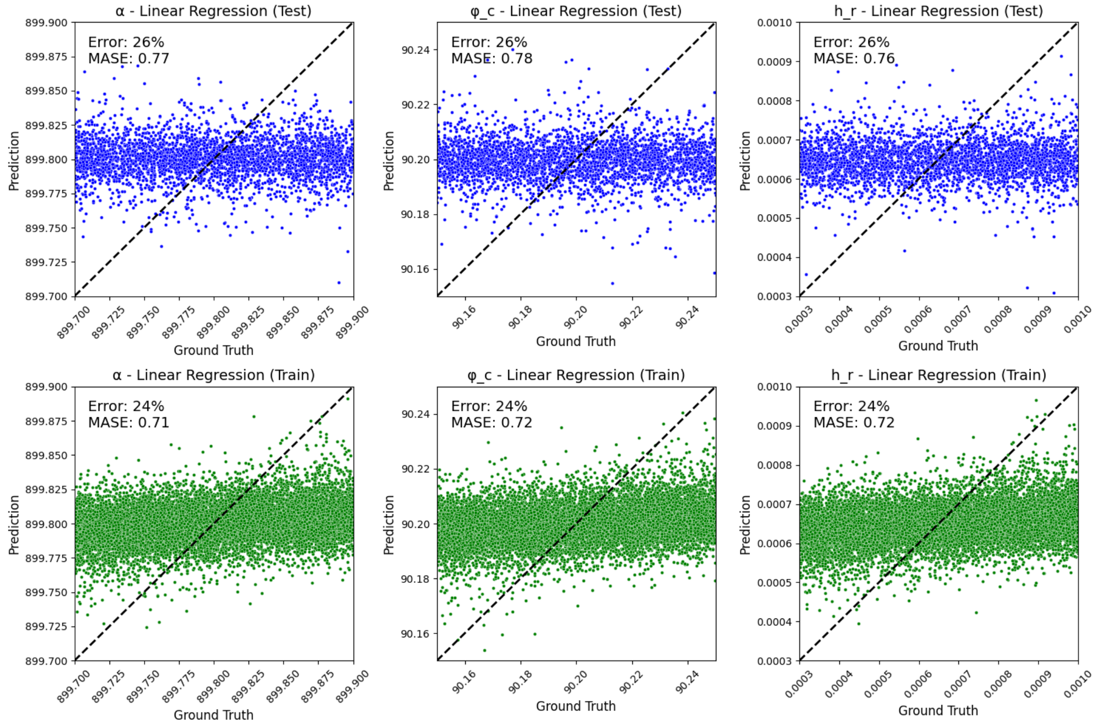


Figure 11: Linear Regression MinMax Scaled

MinMax Scaled Data Prediction Even though no overfitting was recorded, we can see that there was a slight improvement in metrics

PCA + Polynomial Features Data Prediction

4.3 Random Forest

Next, we tried with a decision tree model called Random Forest which works with independent trees

From here we can see that Random Forest did well on the train set and was able to find some of the needed patterns, however, it did fail to predict on the test set, showing a clear signs of overfitting

GridSearch Because of the clear sign of overfitting, we decided to do a Gridsearch to find the best hyperparameters we can use to poentially make the test set better

However, instead of improving the test set, gridsearch decreased it on the train set, we can see this from the chart too

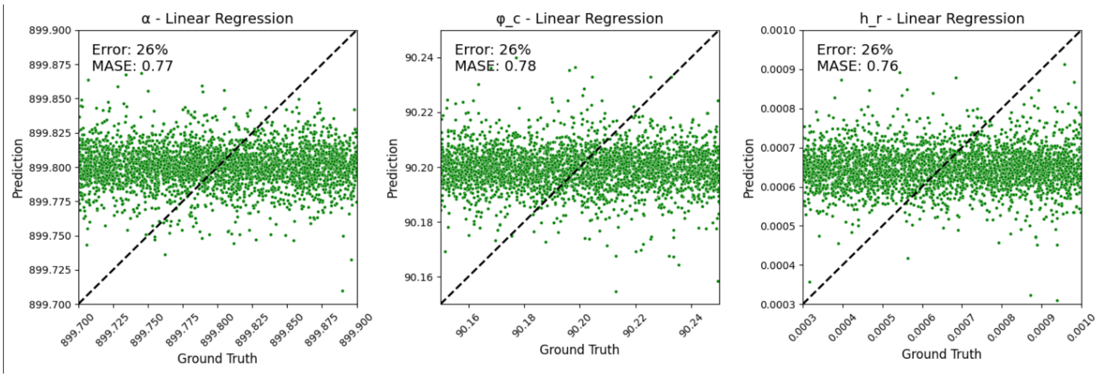


Figure 12: Linear Regression PCA and Scaled

4.4 Neural Network

Original Data Prediction The NN model here consists of a sequential architect with three hidden layers, each using ReLU activation, and an output layer with a linear activation to produce continuing values. The model is trained to minimize mean squared error using the Adam optimizer.

Here we can see that the NN model is doing quite badly in predicting our parameters. Interestingly, NN is scattered around the whole graph and is not consistent with the original set.

Standard Scaled Data Prediction No change was seen on NN for standard scaler

MinMax Scaled Data Prediction We can see that there was a slight improvement in metrics however it came at the cost of NN model being more thinly ranged between a few data points, instead of being scattered better

PCA + Standart Scaler Data Prediction

4.5 Bayesian Neural Network

Challenges We were unable to run this model due to problems in TensorFlow library

5 Analysis

```
Original Prediction:

Neural Network Errors (Train): {'phic':
    12.605628455701273, 'h_rain': 12.658154387503407, '
    alpha': 12.889857127262314}
Neural Network Errors (Test): {'phic':
    30.603384758647557, 'h_rain': 30.338436667232276, '
    alpha': 30.43518798485399}
```

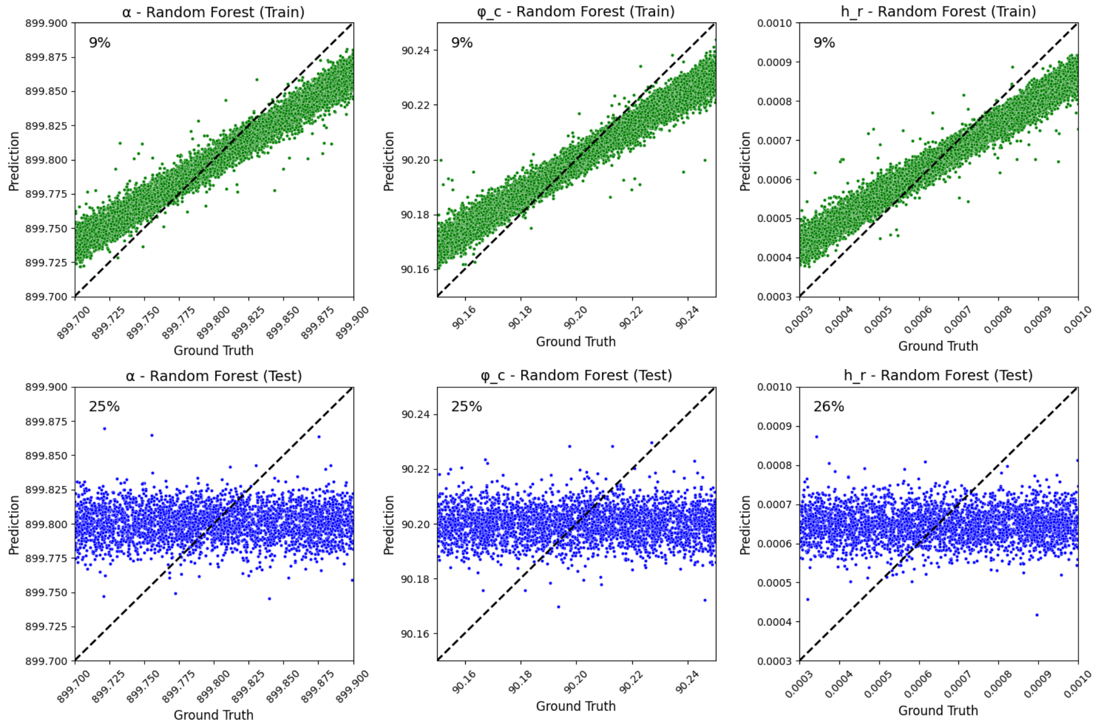


Figure 13: Random Forest Scaled and PCA

```
Linear Regression Errors (Train): {'phic':
    23.91715841326829, 'h_rain': 24.077858915850907, '
    alpha': 24.180259572647238}
Linear Regression Errors (Test): {'phic':
    25.649218596490662, 'h_rain': 25.7463600069922, '
    alpha': 25.58922486609742}
Neural Network MASE (Train): [0.3763319491039043,
    0.37757761364342646, 0.3856400762017765]
Neural Network MASE (Test): [0.9194383389995237,
    0.9229734648117783, 0.9031402797145143]
Linear Regression MASE (Train): [0.714029520568744,
    0.7182137484486134, 0.7234275021134338]
Linear Regression MASE (Test): [0.7705969496177801,
    0.783270653085793, 0.7593401333596429]

Standart Scaled:

Neural Network Errors (Train): {'phic':
    11.81220474218926, 'h_rain': 11.660333584316156, '
    alpha': 11.37657797950784}
Neural Network Errors (Test): {'phic':
    31.4195055714978, 'h_rain': 31.323282336401938, '
    alpha': 31.331917532867614}
```

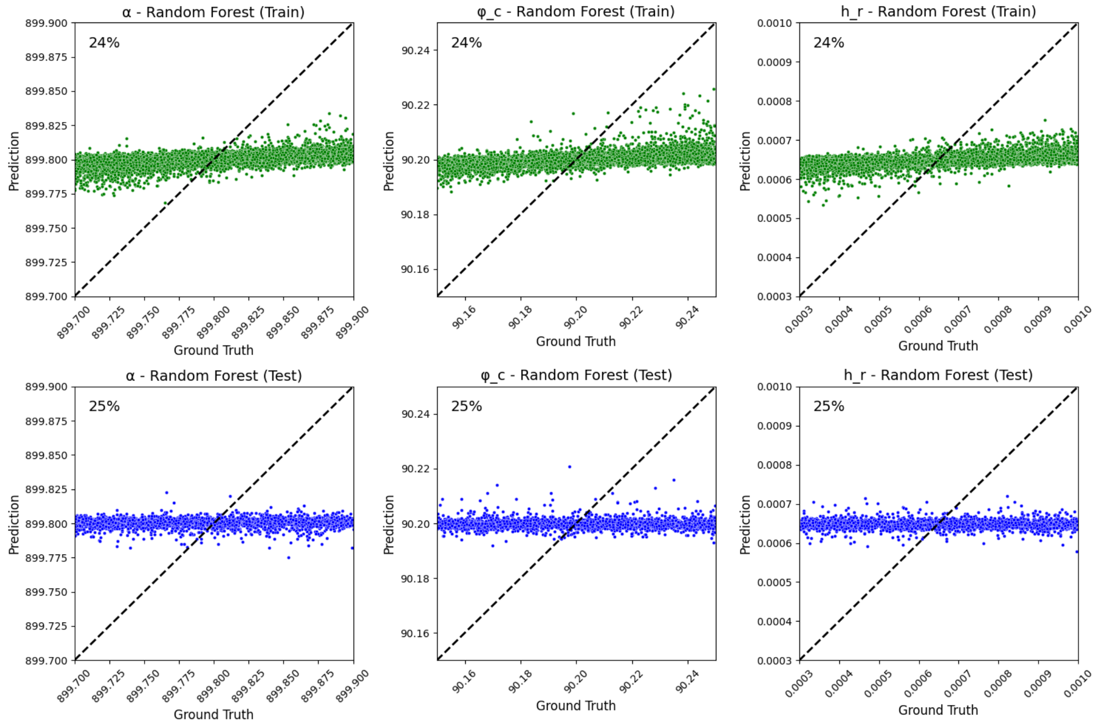


Figure 14: Neural Network Original Prediction

```
Linear Regression Errors (Train): {’phic’:
   23.917158413268265, ’h_rain’: 24.0778589158509, ’
   alpha’: 24.180259572647238}
Linear Regression Errors (Test): {’phic’:
   25.649218596490662, ’h_rain’: 25.746360006992163, ’
   alpha’: 25.589224866097414}
Neural Network MASE (Train): [0.3526448561818384,
   0.3478138118933738 5, 0.3403656344377705]
Neural Network MASE (Test): [0.9439576126193457,
   0.9529350092891233, 0.929750024172946]
Linear Regression MASE (Train): [0.7140295205687434,
   0.7182137484486133, 0.7234275021134335]
Linear Regression MASE (Test): [0.7705969496177801,
   0.7832706530857918, 0.7593401333596426]

MinMax Scaled:

Neural Network Errors (Train): {’phic’:
   24.346112038009178, ’h_rain’: 24.504316948472464, ’
   alpha’: 24.442620204451345}
Neural Network Errors (Test): {’phic’:
   25.338765043165008, ’h_rain’: 25.3316915674618, ’
   alpha’: 25.53891629589814}
```

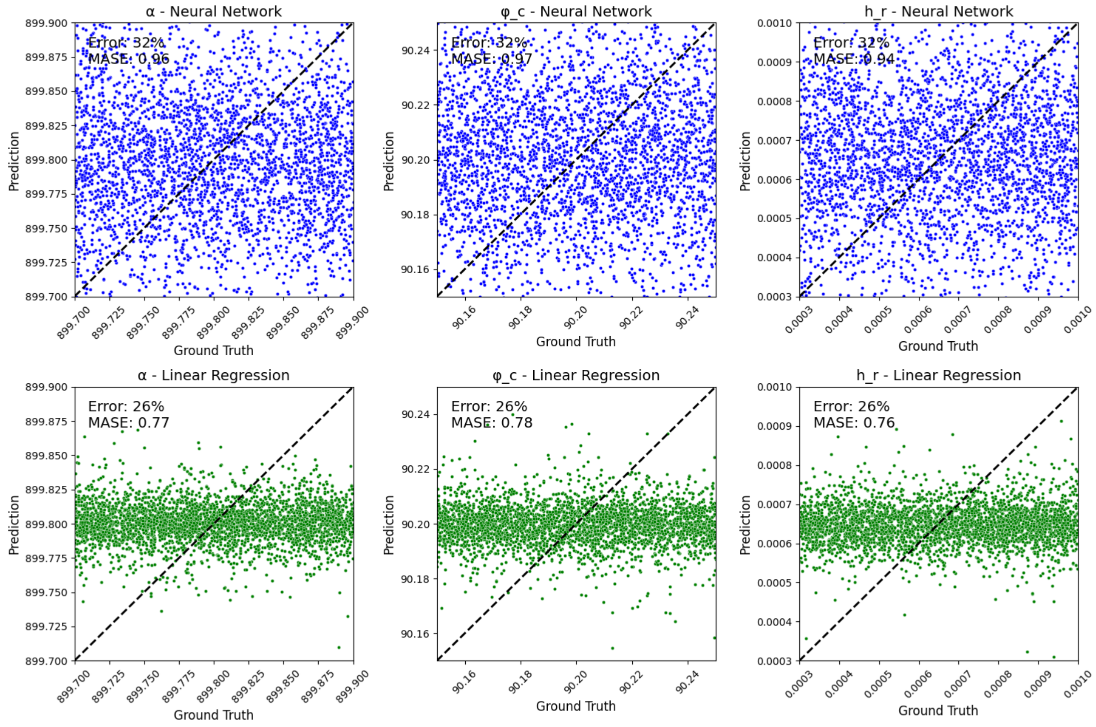


Figure 15: Neural Network Original Prediction

```
Neural Network MASE (Train): [0.7268356217672061,
   0.7309344817677838, 0.7312768345802221]
Neural Network MASE (Test): [0.761269781997029,
   0.7706553700183288, 0.7578472660842894]
Linear Regression Errors (Train): {'phic':
   23.917158413267988, 'h_rain': 24.077858915850932, '
   alpha': 24.180259572647063}
Linear Regression Errors (Test): {'phic':
   25.64921859649079, 'h_rain': 25.746360006991935, '
   alpha': 25.58922486609759}
Linear Regression MASE (Train): [0.714029520568735,
   0.7182137484486141, 0.7234275021134285]
Linear Regression MASE (Test): [0.7705969496177839,
   0.7832706530857847, 0.759340133359648]

PCA+Scaled:

Neural Network Errors (Train) with PCA: {'phic':
   24.854382545708344, 'h_rain': 25.01990667913588, '
   alpha': 25.174372349465653}
Neural Network Errors (Test) with PCA: {'phic':
   25.099976949959423, 'h_rain': 25.087569011184353, '
   alpha': 25.251960812683937}
```

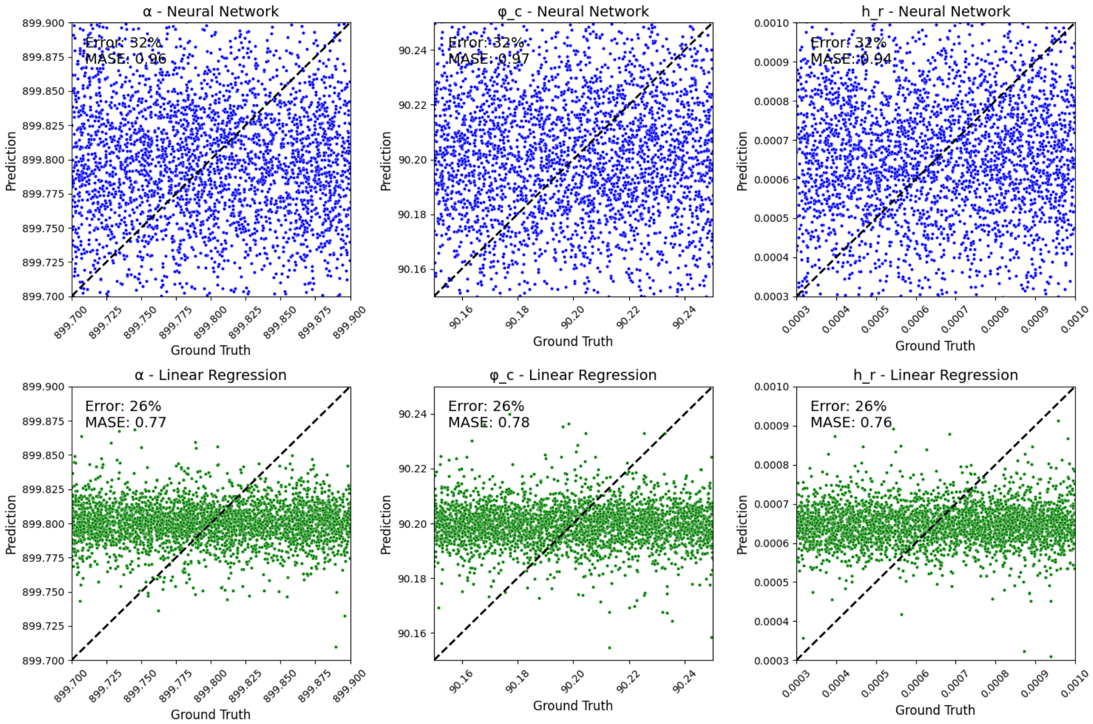


Figure 16: Neural Network Scaled Prediction

```
Linear Regression Errors (Train): {'phic':
    23.91715841326829, 'h_rain': 24.077858915850907, '
    alpha': 24.180259572647238}
Linear Regression Errors (Test): {'phic':
    25.649218596490662, 'h_rain': 25.7463600069922, '
    alpha': 25.58922486609742}
Neural Network MASE (Train) with PCA:
    [0.7420096713202805, 0.7463139070902574,
    0.753169470796281]
Neural Network MASE (Test) with PCA:
    [0.7540957086217707, 0.7632285324367647,
    0.7493320876827414]
Linear Regression MASE (Train): [0.714029520568744,
    0.7182137484486134, 0.7234275021134338]
Linear Regression MASE (Test): [0.7705969496177801,
    0.783270653085793, 0.7593401333596429]

RandomForest:

Random Forest Errors (Train): {'phic':
    9.688384707929853, 'h_rain': 9.784855948075439, '
    alpha': 9.806060994716589}
Random Forest MASE (Train): 0.29012475229376455
```

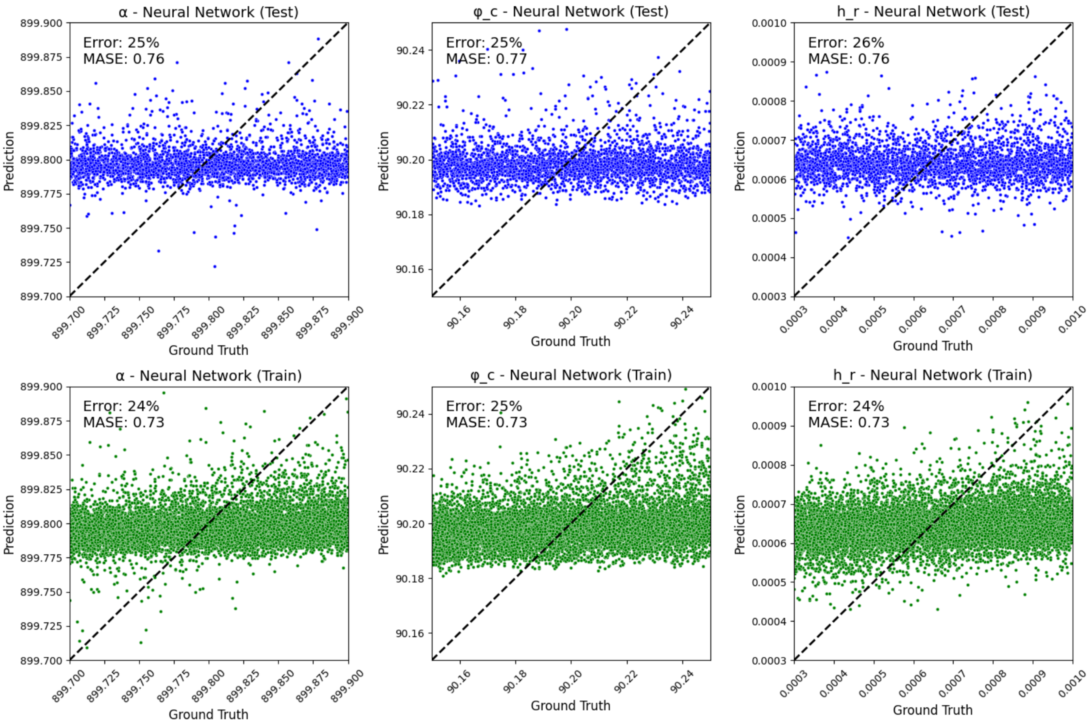


Figure 17: Neural Network MinMax Scaled Prediction

```
Random Forest Errors (Test): {'phic':
    26.497797528111676, 'h_rain': 26.09863419154153, '
    alpha': 26.6479409704088}
Random Forest MASE (Test): 0.7953850373686746
MASE for a (Train   RF): 0.29
MASE for a (Test - RF): 0.80
MASE for f_c (Train - RF): 0.29
MASE for f_c (Test - RF): 0.80
MASE for h_r (Train - RF): 0.29
MASE for h_r (Test - RF): 0.80

RandomForest with GridSearch:

Random Forest Errors (Train): {'phic':
    24.131026371471723, 'h_rain': 24.325490312639424, '
    alpha': 24.47078441650804}
Random Forest MASE (Train): 0.7221671782650554
Random Forest Errors (Test): {'phic':
    25.072560473049155, 'h_rain': 25.08914560926009, '
    alpha': 25.263236116020504}
Random Forest MASE (Test): 0.7565626799983757
MASE for a (Train - RF): 0.72
MASE for a (Test - RF): 0.76
```

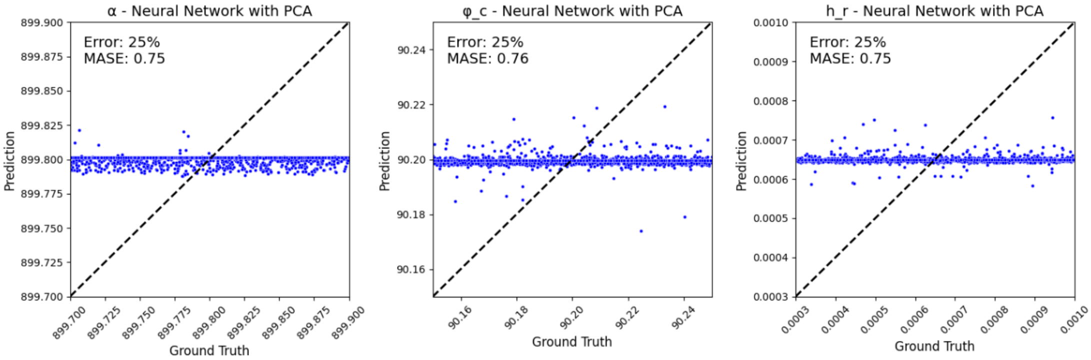
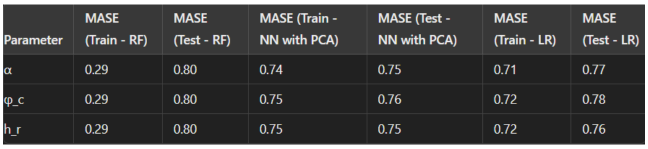


Figure 18: Neural Network Pca and Scaled Prediction

```
MASE for f_c (Train - RF): 0.72
MASE for f_c (Test - RF): 0.76
MASE for h_r (Train - RF): 0.72
MASE for h_r (Test - RF): 0.76
```

5.1 Comparison Models

Linear Regression and Neural Network did not see any significant changes with any of the scalers. However, the plotting of scatter plots was worse on MinMax for NN since it is assigning them to 0 and 1. PCA helped reduce the running time however did not help much with performance. When Comparing all 3 models, Neural Network and Linear Regression did not do well, even with different features, although we can see that the best performing one was Random Forest with Standart Scaler and PCA, however, it was also overfitting since only the train set it was scoring well.



Parameter	MASE (Train - RF)	MASE (Test - RF)	MASE (Train - NN with PCA)	MASE (Test - NN with PCA)	MASE (Train - LR)	MASE (Test - LR)
α	0.29	0.80	0.74	0.75	0.71	0.77
φ_c	0.29	0.80	0.75	0.76	0.72	0.78
h_r	0.29	0.80	0.75	0.75	0.72	0.76

Figure 19: Mase Comparison

6 Conclusions

In conclusion, the evaluation of the models and the final analysis have led us to identify several key challenges that we will need to address as we proceed with our project:

1. **Database Size**: Our current database may not be sufficiently large for the complexity of the problem at hand since its almost 7 times smaller than the desired one. We consider a larger dataset could potentially improve the performance of our machine learning models.

2. **Correlation Strength**: The correlations between our features and target variables are not particularly strong. This could make it challenging for our models to accurately predict the target variables based on the features.

3. **Feature Quantity**: The number of features in our dataset is relatively small. While this can simplify the modeling process, it may also limit the complexity and richness of the patterns that our models can learn. The implementation of Polynomial Features also showed us that even with higher degree values, the models' performance did not increase considerably enough.

Despite these challenges, we remain committed to our goal of developing a decent and useful database as well as robust and accurate models for predicting the behavior of shallow water. We believe that with careful method selection and proper fine-tuning, and possibly augmenting our dataset, we can get over these obstacles and achieve our project objectives in the future.

7 References

1. WavestoWeather SWM Large Ensemble: https://github.com/wavestoweather/SWM$_largeensemble$

2. Keras Bayesian Neural Networks Example: https://keras.io/examples/keras$_recipes/bayesian_neural_net$

3. Difference Between StandardScaler and MinMaxScaler on Stack Overflow: https://stackoverflow.com/questions/51237635/difference-between-standard-scaler-and-minmaxscaler

4. MAE, MAPE, MASE, and the Scaled RMSE Tutorial: https://www.pmorgan.com.au/tutorials/mae,-mape,-mase-and-the-scaled-rmse/