Review of ArcticBeach v1.0: A physics-based parameterization of pan-Arctic coastline erosion

by Rebecca Rolph et al.

Reviewed by Jennifer M. Frederick, Sandia National Laboratories, jennifer.frederick@sandia.gov

Summary

This manuscript describes a model for Arctic coastal erosion that is based on a simplified physical erosion model of a partially frozen cliff and beach, coupled to a storm surge model. It is presented as a first step toward a parameterization of pan-Arctic shoreline erosion at a coarse spatial scale for capturing erosion rates on the order of years to decades. It uses physical data as boundary conditions, such as wind speeds and directions, wave period and height, and sea surface temperature, as well as accounting for sea ice cover. The authors claim the new model provides a promising starting point to project the retreat of Arctic shorelines, or to evaluate historical retreat in places that have had few observations.

General Summary of Comments

I do not recommend publication in its current form. The model presented (ArcticBeachv1.0) is under-developed and the authors have not shown that this model has any predictive skill that outperforms a random number generator (proof described in detail in my review). For transparency, I have also included the Python script which performs this analysis. My suggestion to the authors is further development of the model and resubmission for publication at a later date and after further collaboration and consultation with peers in this research field. One benefit of the model presented is its low computational cost. If the low computational cost can be maintained while improving its ability to robustly predict coastal retreat rates, this would represent a ground-breaking advance in the field!

The results summarized in Figure 4 show the modeled annual and cumulative retreat at Mamontovy Khayata (MK) and Drew Point (DP) vs observations at each site. At first glance, the modeled retreat looks poor, but an error analysis was not provided to quantify model performance. For any predictive model, a thorough analysis of model predictive skill is required to evaluate its performance and ability to make reliable, robust predictions. One of the simplest routines is to test model predictions against a random prediction. If the model has good predictive skill, it should outperform a prediction generated at random within a plausible range of possible outcomes. This is essentially like posing the null hypothesis and showing that the model can disprove the null hypothesis. In this case, the null hypothesis states that, 'ArcticBeachv1.0 cannot predict the annual erosion rate any better than a random number generator can.' If the ArcticBeachv1.0 model can predict annual erosion rate statistically significantly better than a random number generator, then it can rightfully claim predictive skill. My concern here for both locations is that, while there are a few years where modeled erosion matched observed erosion fairly well, there are also many years in this time series where the erosion is far outside of the running average. In these years, a model with high predictive skill should be able to reproduce the trend, if it has captured the correct physics. However, the ArcticBeachv1.0 model predictions end up under- or over-estimating the retreat, in the OPPOSITE direction just as many times as they estimate the retreat in the CORRECT direction (above or below the mean retreat).

The conclusion from the analysis for predictive skill (described in full detail below) shows that the ArcticBeachv1.0 model has no predictive skill at the DP location, and has inverse predictive skill at the MK location. Based on the error analysis, I disagree with the authors, as stated in the abstract, that the ArcticBeachv1.0 model provides a promising starting point to project the retreat of Arctic shorelines, or to evaluate historical retreat in places that have had few observations. The results of this analysis at both locations indicate that the model in its current form is under-developed, and cannot be relied upon to provide robust and skillful predictions for coastal retreat rates in the Arctic more than a randomly generated number can (in the case of the DP location) nor can be relied to provide a prediction in the correct trend direction (in the case of the MK location).

Detailed Analysis

I performed an analysis on the modeled retreat vs the observed retreat to quantify the error. I used the mean squared error (MSE) of the annual retreat predictions as the performance metric. The MSE will penalize large differences between predicted and observed values more so than small differences, which is appropriate in this case because swings in retreat far outside the normal or average erosion behavior signify major disruptions in erosion drivers, which is what we want to capture with a robust, skillful predictive model. It is also thought that future conditions will become more extreme as climate changes in the Arctic, and thus erosion may continue to behave erratically. The mean squared error is defined as

$$MSE = \frac{1}{N}\sum_{j=1}^{N}(M_j - O_j)^2$$

where N is the number of retreat predictions with available retreat observations, M is a modeled retreat prediction, and O is a corresponding retreat observation for the model prediction. For this test, the MSEs for each location for ArcticBeachv1.0 vs observations are shown in Table 1. For the MK location, the MSE of the annual retreat between 1995 - 2018 was 125.48 m^2, and for the Drew Point location, the MSE of the annual retreat between 2007 - 2016 was 61.55 m^2. A perfect prediction for every year would yield an MSE of 0 m^2 at both locations.

Table 1 Mean squared error and cumulative erosion error statistics for ArcticBeachv1.0 and a randomly generated model.

	Mamontovy Khayata	Drew Point
Mean Squared Error (MSE) Annual Erosion, ArcticBeachv1.0 Model	125.48 m^2	61.55 m^2
Average MSE Annual Erosion, Randomly Generated Model	16.36 m^2	53.85 m^2
Standard Deviation of MSE Annual Erosion, Randomly Generated Model	+/- 3.73 m^2	+/- 19.12 m^2
Error Cumulative Erosion, ArcticBeachv1.0 Model	48.77 m	3.42 m
Average Error Cumulative Erosion, Randomly Generated Model	39.19 m	20.28 m

Standard Deviation of Average Error Cumulative Erosion, Randomly Generated Model	+/- 15.10 m	+/- 13.99 m

Next, the model predictions are tested against a random number generator to judge predictive skill and give meaning to the MSE values calculated for ArcticBeachv1.0. For the MK location, a random number from within a plausible range of retreat was generated for each year using Python's random package. The range in retreat was calculated as the minimum and maximum of the observed retreat data with a 10% envelope (e.g. 1.18 m – 12.04 m). The MSE for the randomly chosen annual retreat was calculated against the observations. This numerical experiment was performed 5,000 times, and a histogram of results was created to obtain statistical behavior (shown in Figure 1). As reported in Table 1, the average MSE of the annual retreat from the randomly generated model was 16.36 m^2, as compared to the ArcticBeachv1.0 model value of 125.48 m^2 (shown as the red line superimposed on the histogram). The error is much larger for the ArcticBeachv1.0 model than the randomly generated model, while also lying *significantly* outside of the 1^{st} standard deviation of the randomly generated model's "predictions" (predictions in quotations because they are not truly predictions but random numbers). This suggests that the ArcticBeachv1.0 model has predictive skill, but its predictive skill is *opposite* of the observations (in the direction of larger error or in the opposite direction from mean annual retreat). This is clearly seen by inspection of Figure 4a in the manuscript, where large mismatches in the opposite direction from the mean annual retreat rates are predicted by the ArcticBeachv1.0 model, especially between years 2002 - 2018.

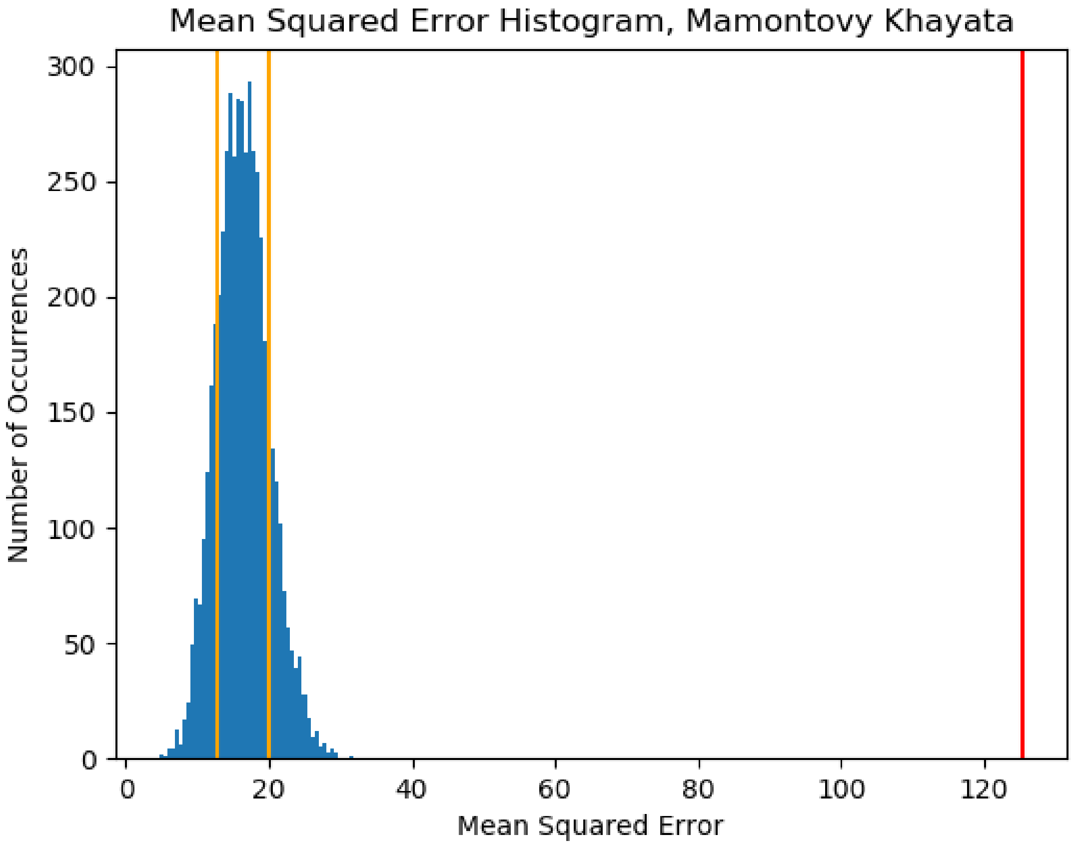

Figure 1. Histogram of the mean squared error for a randomly generated model for the MK location. The orange lines show +/- 1 standard deviation from the mean, while the red line shows the mean squared error for the ArcticBeachv1.0 model.

The test was repeated for the Drew Point location. For the DP location, a random number between a plausible range of retreat was generated for each year using Python's random package. The range in retreat was calculated as the minimum and maximum of the observed retreat data with a 10% envelope (e.g. 5.94 m – 24.83 m). The MSE for the randomly chosen annual retreat was calculated against the observations. This numerical experiment was performed 5,000 times, and a histogram of results was created to obtain statistical behavior (shown in Figure 2). As reported in Table 1, the average MSE of the annual retreat from the randomly generated model was 53.85 m^2, as compared to the ArcticBeachv1.0 model value of 61.55 m^2 (shown as the red line superimposed on the histogram). In this case, the ArcticBeachv1.0 model performed slightly worse than the randomly generated model (since the MSE for the ArcticBeachv1.0 model was higher than the mean MSE for the randomly generated model). Additionally, the MSE for the ArcticBeachv1.0 model sits within the 1st standard deviation of the MSE for the randomly generated model. This suggests that the ArcticBeachv1.0 model does not predict erosion rates *significantly* different than a randomly generated number. If it did, then the MSE would be well below the 1st standard deviation of the randomly generated model. The performance can also be seen by inspection of Figure 4b, where the ArcticBeachv1.0 model predictions end up under- or over-estimating the retreat at DP, in the OPPOSITE direction just as many times as they estimate the retreat in the CORRECT direction (above or below the mean retreat over the time period).

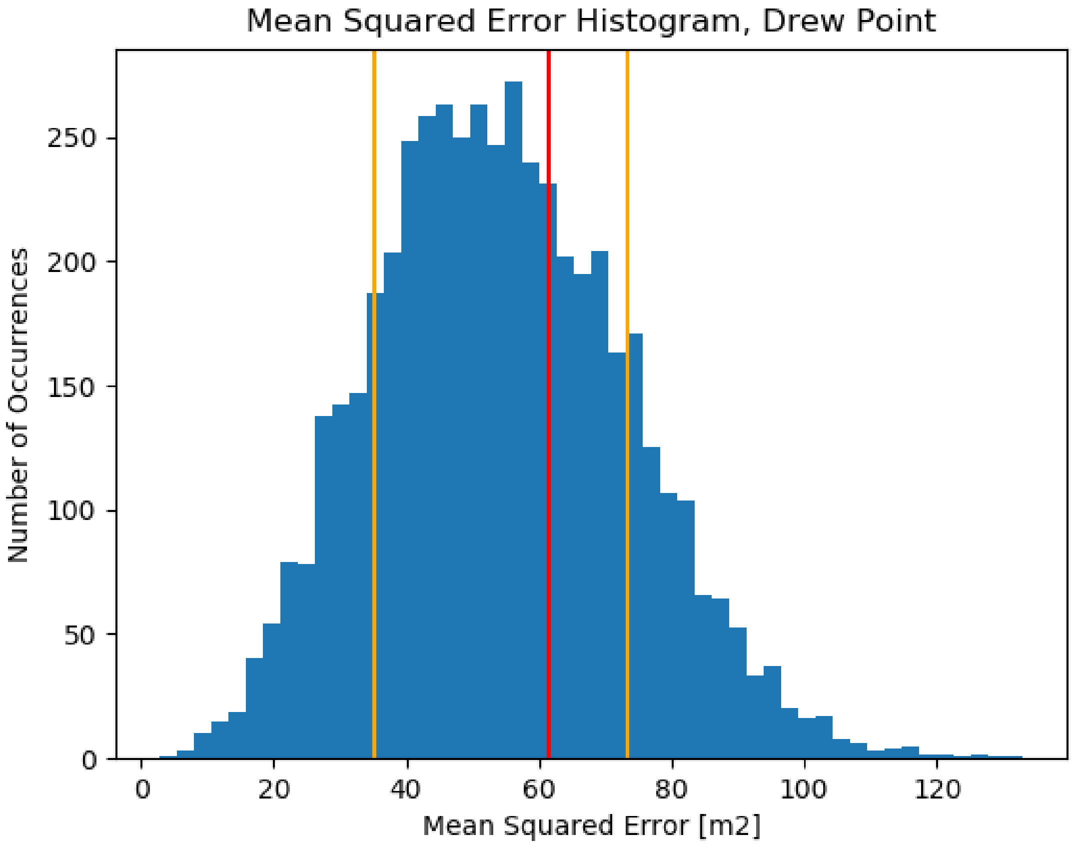

Figure 2 Histogram of the mean squared error for a randomly generated model for the DP location. The orange lines show +/- 1 standard deviation from the mean, while the red line shows the mean squared error for the ArcticBeachv1.0 model.

Furthermore, the analysis was extended to quantify the error in the cumulative erosion. The cumulative erosion error was calculated as the difference between the sum of the observed annual retreat values and the sum of the modeled annual retreat values. As reported in Table 1, the cumulative retreat error was 48.77 m (reported as "roughly 40 m" in the manuscript text, line 215) for the MK location, and 3.42 m (reported as "within a few meters" in the manuscript text, line 215) for the DP location.

Similarly to the random model numerical experiments presented for the annual retreat predictions, the same procedure is repeated for the cumulative erosion error. For each year, using the same set of random numbers that were generated for annual retreat, the cumulative retreat was calculated by summing the random annual retreat values for each numerical experiment. A histogram was created for each location, shown in Figure 3 (MK) and Figure 4 (DP).

For the MK location, the mean cumulative erosion error for randomly generated model was 39.19 m (see Table 1), as compared to the ArcticBeachv1.0 model value of 48.77 m^2 (shown as the red line superimposed on the histogram). In this case, the ArcticBeachv1.0 model performed slightly worse than the randomly generated model (since the cumulative erosion error for the ArcticBeachv1.0 model was higher than the error in the randomly generated model). Moreover, the mean cumulative erosion error for the ArcticBeachv1.0 model sits within the first standard deviation of the cumulative erosion error for the randomly generated model. This suggests that the ArcticBeachv1.0 model does not predict cumulative erosion *significantly* different than a randomly generated number at the MK location. Interestingly, while the annual retreat predictions were skillful (albeit in the opposite direction), the cumulative retreat might as well have been generated at random.

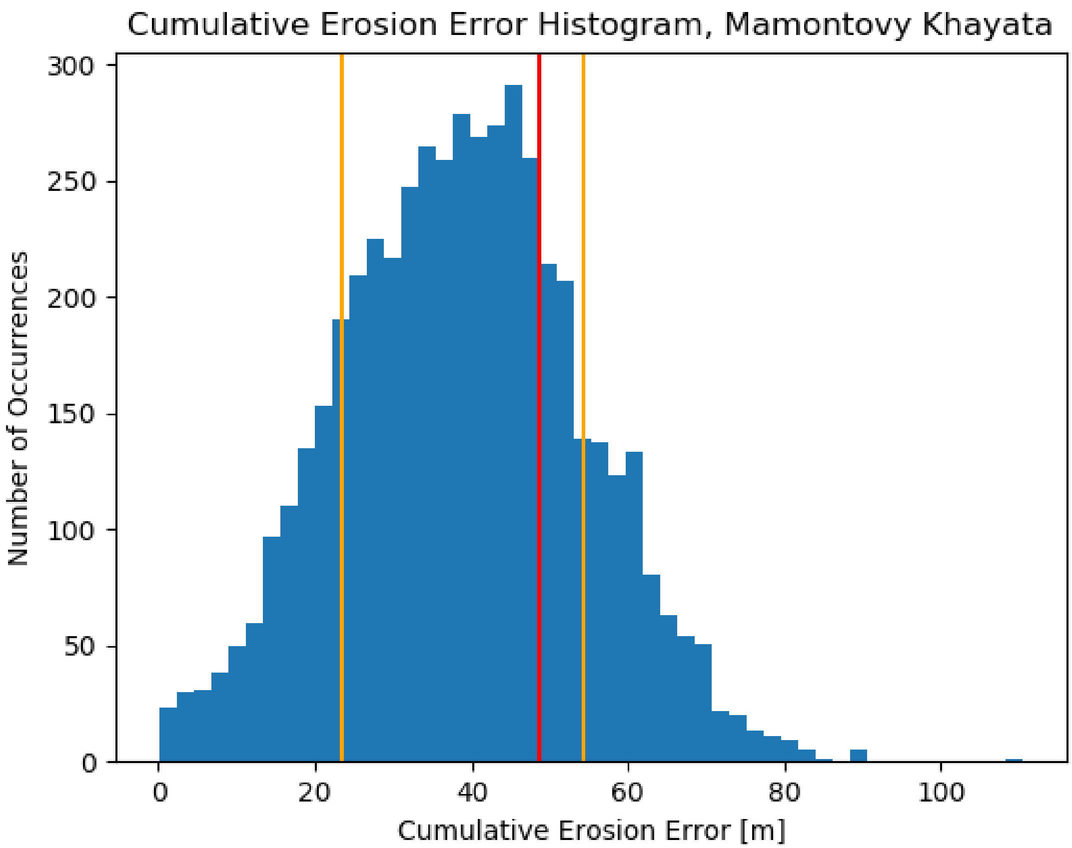

Figure 3 Histogram of the cumulative erosion error for a randomly generated model for the MK location. The orange lines show +/- 1 standard deviation from the mean, while the red line shows the cumulative erosion error for the ArcticBeachv1.0 model.

At the DP location, the mean cumulative erosion error for randomly generated model was 20.28 m (see Table 1), as compared to the ArcticBeachv1.0 model value of 3.42 m^2 (shown as the red line superimposed on the histogram). In this case, the ArcticBeachv1.0 model performed significantly better than the randomly generated model (since the cumulative erosion error for the ArcticBeachv1.0 model was lower than the error in the randomly generated model and it was positioned outside of the 1st standard deviation of the randomly generated model error). This makes sense because the ArcticBeachv1.0 model did a decent job predicting the erosion rate at Drew Point for years in which the erosion was relatively average, but happened to over- or under- estimate the erosion for anomalous years at roughly equal magnitudes, and as a result summing to roughly zero, thus providing little contribution to the cumulative retreat error metric.

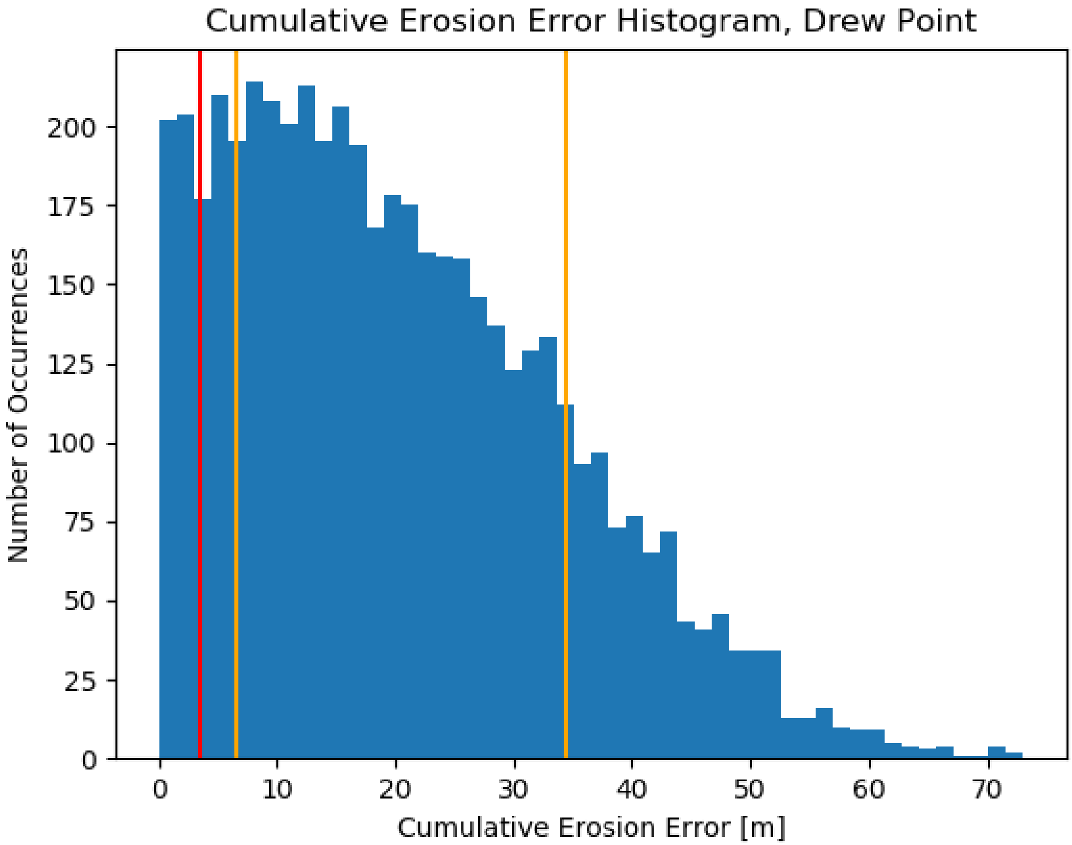

Figure 4 Histogram of the cumulative erosion error for a randomly generated model for the DP location. The orange lines show +/- 1 standard deviation from the mean, while the red line shows the cumulative erosion error for the ArcticBeachv1.0 model.

The conclusion from the analysis for predictive skill shows that the ArcticBeachv1.0 model has no predictive skill at the DP location, and has inverse predictive skill at the MK location. Based on the error analysis, I disagree with the authors, as stated in the abstract, that the ArcticBeachv1.0 model provides a promising starting point to project the retreat of Arctic shorelines, or to evaluate historical retreat in places that have had few observations. The results of this analysis at both locations indicate that the model in its current form is under-developed, and cannot be relied upon to provide robust and skillful predictions for coastal retreat rates in the Arctic more than a randomly generated number can (in the case of the DP location) nor can be relied to provide a prediction in the correct trend direction (in the case of the MK location). As disheartening as this error analysis seems, the MK location does show promise because of its ability to capture opposite trends. I suggest to the authors to investigate this behavior more closely, as it probably indicates some physical behavior captured in the model that may be relevant for erosion rates, but in the opposite sense.

Table 1: From what I understand, the two study locations have identical material properties, but they differ in geometry only with cliff height. Is this an adequate demonstration of the model's ability to provide a "physics-based numerical model that can be applied across all partially frozen shorelines"? (Quote from lines 36-37) I was expecting more diversity between demonstration sites.

```
import random
import numpy as np
import matplotlib.pyplot as plt

# This script performs an error analysis of the ArcticBeachv1.0 model
# as presented in the manuscript by R. Rolph et al. submitted to
# EGU journal Geoscientific Model Development.

# Abbreviations throughout this script:
# DP = Drew Point
# MK = Mamontovy Khayata
# mae = mean absolute error
# mse = mean squared error
# cee = cumulative erosion error
# diff = difference
# obs = observation
# ABv1p0, AB = ArcticBeachv1.0
# cumu = cumulative

# I used https://automeris.io/WebPlotDigitizer/ to pull numbers from
# Figure 4b in the manuscript.
year_DP = [2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016]
ABv1p0_DP = [4.783878951, 9.017355894, 19.55784951, 18.78027211, 17.3115148, 28.58638707, 12.90524288, 13.07803786, 13.85561526, 30.14154187]
obs_DP = [22.15835141, 15.84598698, 19.32754881, 6.605206074, 16.95227766, 22.5813449, 13.34056399, 16.39913232, 16.20390456, 22.02819957]

diff = []
mae = []
diff2 = []
mse = []
cumu_DP = 0
cumu_ABDP = 0
for i in range(len(obs_DP)):
    cumu_ABDP = cumu_ABDP + ABv1p0_DP[i]
    cumu_DP = cumu_DP + obs_DP[i]
    diff.append(abs(ABv1p0_DP[i]-obs_DP[i]))
    diff2.append(diff[i]*diff[i])
    mae.append(diff[i]*(1/len(obs_DP)))
    mse.append(diff2[i]*(1/len(obs_DP)))
mae_DP = sum(mae)
print('Drew Point MAE = '+str(mae_DP))
mse_DP = sum(mse)
print('Drew Point MSE = '+str(mse_DP))
print('Drew Point Cumulative Erosion = '+str(cumu_DP))
print('ArcticBeach Cumulative Erosion = '+str(cumu_ABDP))
cee_DP = abs(cumu_DP-cumu_ABDP)
print('Drew Point Cumu. Erosion Error = '+str(cee_DP))

# I used https://automeris.io/WebPlotDigitizer/ to pull numbers from
# Figure 4a in the manuscript.
year_MK = [1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018]
ABv1p0_MK = [9.504391468, 1.066499373, 3.513174404, 3.732747804, 3.324968632, 3.074027604, 4.422835634, 0.752823087, 1.819322459, 0.846925972,
1.191969887, 1.411543287, 2.383939774, 1.442910916, 8.814303639, 13.80175659, 8.061480552, 14.33500627, 8.814303639, 14.33500627, 9.91217064,
22.45922208, 13.08030113, 16.49937265]
obs_MK = [5.740276035, 4.485570891, 3.795483061, 3.795483061, 3.136762861, 3.732747804, 3.889585947, 6.179422836, 6.587202008, 6.493099122,
5.207026349, 5.39523212, 6.775407779, 6.900878294, 4.987452949, 5.740276035, 10.94730238, 4.203262233, 2.321204517, 4.861982434, 7.402760351,
1.317440402, 3.199498118, 2.728983689]

diff = []
mae = []
diff2 = []
mse = []
cumu_MK = 0
cumu_ABMK = 0
for i in range(len(obs_MK)):
    cumu_ABMK = cumu_ABMK + ABv1p0_MK[i]
    cumu_MK = cumu_MK + obs_MK[i]
    diff.append(abs(ABv1p0_MK[i]-obs_MK[i]))
    diff2.append(diff[i]*diff[i])
    mae.append(diff[i]*(1/len(obs_DP)))
    mse.append(diff2[i]*(1/len(obs_DP)))
mae_MK = sum(mae)
print('Mamontovy Khayata MAE = '+str(mae_MK))
mse_MK = sum(mse)
print('Mamontovy Khayata MSE = '+str(mse_MK))
print('Mamontovy Khayata Cumulative Erosion = '+str(cumu_MK))
```

```
print('ArcticBeach Cumulative Erosion = '+str(cumu_ABMK))
cee_MK = abs(cumu_MK-cumu_ABMK)
print('Mamontovy Khayata Cumu. Erosion Error = '+str(cee_MK))

mae_vector_DP = []
mse_vector_DP = []
cee_vector_DP = []

mae_vector_MK = []
mse_vector_MK = []
cee_vector_MK = []

print(min(obs_DP)*0.90); print(max(obs_DP)*1.10)
print(min(obs_MK)*0.90); print(max(obs_MK)*1.10)

for k in range(5000):
   randomlist_DP = []
   diff = []
   mae = []
   diff2 = []
   mse = []

   for i in range(len(obs_DP)):
      randomlist_DP.append(random.uniform(min(obs_DP)*0.90,max(obs_DP)*1.10))
      diff.append(abs(randomlist_DP[i]-obs_DP[i]))
      diff2.append(diff[i]*diff[i])
      mae.append(diff[i]*(1/len(obs_DP)))
      mse.append(diff2[i]*(1/len(obs_DP)))

   mae_vector_DP.append(sum(mae))
   mse_vector_DP.append(sum(mse))
   cee_vector_DP.append(abs(sum(randomlist_DP)-cumu_DP))

for k in range(5000):
   randomlist_MK = []
   diff = []
   mae = []
   diff2 = []
   mse = []

   for i in range(len(obs_MK)):
      randomlist_MK.append(random.uniform(min(obs_MK)*0.90,max(obs_MK)*1.10))
      diff.append(abs(randomlist_MK[i]-obs_MK[i]))
      diff2.append(diff[i]*diff[i])
      mae.append(diff[i]*(1/len(obs_MK)))
      mse.append(diff2[i]*(1/len(obs_MK)))

   mae_vector_MK.append(sum(mae))
   mse_vector_MK.append(sum(mse))
   cee_vector_MK.append(abs(sum(randomlist_MK)-cumu_MK))


print(' '); print('Drew Point Location:')
print('average MAE = '+str(np.mean(mae_vector_DP)))
print('minimum MAE = '+str(min(mae_vector_DP)))
print('maximum MAE = '+str(max(mae_vector_DP)))
print('st.dev  MAE = '+str(np.std(mae_vector_DP)))
print('confidence = ');print(str(np.mean(mae_vector_DP)-np.std(mae_vector_DP)) + ' - ' + str(np.mean(mae_vector_DP)+np.std(mae_vector_DP)))
plt.hist(mae_vector_DP, 50)
plt.axvline(mae_DP, color='red')
plt.axvline(np.mean(mae_vector_DP)-np.std(mae_vector_DP), color='orange')
plt.axvline(np.mean(mae_vector_DP)+np.std(mae_vector_DP), color='orange')
plt.title('Mean Absolute Error Histogram, Drew Point')
plt.ylabel('Number of Occurrences')
plt.xlabel('Mean Absolute Error [m]')
plt.show()
print(' '); print('Drew Point Location:')
print('average MSE = '+str(np.mean(mse_vector_DP)))
print('minimum MSE = '+str(min(mse_vector_DP)))
print('maximum MSE = '+str(max(mse_vector_DP)))
print('st.dev  MSE = '+str(np.std(mse_vector_DP)))
print('confidence = '); print(str(np.mean(mse_vector_DP)-np.std(mse_vector_DP)) + ' - ' + str(np.mean(mse_vector_DP)+np.std(mse_vector_DP)))
plt.hist(mse_vector_DP, 50)
plt.axvline(mse_DP, color='red')
plt.axvline(np.mean(mse_vector_DP)-np.std(mse_vector_DP), color='orange')
```

```
plt.axvline(np.mean(mse_vector_DP)+np.std(mse_vector_DP), color='orange')
plt.title('Mean Squared Error Histogram, Drew Point')
plt.ylabel('Number of Occurrences')
plt.xlabel('Mean Squared Error [m2]')
plt.show()
print(' '); print('Drew Point Location:')
print('average CEE = '+str(np.mean(cee_vector_DP)))
print('minimum CEE = '+str(min(cee_vector_DP)))
print('maximum CEE = '+str(max(cee_vector_DP)))
print('st.dev  CEE = '+str(np.std(cee_vector_DP)))
print('confidence = '); print(str(np.mean(cee_vector_DP)-np.std(cee_vector_DP)) + ' - ' + str(np.mean(cee_vector_DP)+np.std(cee_vector_DP)))
plt.hist(cee_vector_DP, 50)
plt.axvline(cee_DP, color='red')
plt.axvline(np.mean(cee_vector_DP)-np.std(cee_vector_DP), color='orange')
plt.axvline(np.mean(cee_vector_DP)+np.std(cee_vector_DP), color='orange')
plt.title('Cumulative Erosion Error Histogram, Drew Point')
plt.ylabel('Number of Occurrences')
plt.xlabel('Cumulative Erosion Error [m]')
plt.show()

print(' '); print('Mamontovy Khayata Location:')
print('average MAE = '+str(np.mean(mae_vector_MK)))
print('minimum MAE = '+str(min(mae_vector_MK)))
print('maximum MAE = '+str(max(mae_vector_MK)))
print('st.dev  MAE = '+str(np.std(mae_vector_MK)))
print('confidence = '); print(str(np.mean(mae_vector_MK)-np.std(mae_vector_MK)) + ' - ' + str(np.mean(mae_vector_MK)+np.std(mae_vector_MK)))
plt.hist(mae_vector_MK, 50)
plt.axvline(mae_MK, color='red')
plt.axvline(np.mean(mae_vector_MK)-np.std(mae_vector_MK), color='orange')
plt.axvline(np.mean(mae_vector_MK)+np.std(mae_vector_MK), color='orange')
plt.title('Mean Absolute Error Histogram, Mamontovy Khayata')
plt.ylabel('Number of Occurrences')
plt.xlabel('Mean Absolute Error')
plt.show()
print(' '); print('Mamontovy Khayata Location:')
print('average MSE = '+str(np.mean(mse_vector_MK)))
print('minimum MSE = '+str(min(mse_vector_MK)))
print('maximum MSE = '+str(max(mse_vector_MK)))
print('st.dev  MSE = '+str(np.std(mse_vector_MK)))
print('confidence = '); print(str(np.mean(mse_vector_MK)-np.std(mse_vector_MK)) + ' - ' + str(np.mean(mse_vector_MK)+np.std(mse_vector_MK)))
plt.hist(mse_vector_MK, 50)
plt.axvline(mse_MK, color='red')
plt.axvline(np.mean(mse_vector_MK)-np.std(mse_vector_MK), color='orange')
plt.axvline(np.mean(mse_vector_MK)+np.std(mse_vector_MK), color='orange')
plt.title('Mean Squared Error Histogram, Mamontovy Khayata')
plt.ylabel('Number of Occurrences')
plt.xlabel('Mean Squared Error')
plt.show()
print(' '); print('Mamontovy Khayata Location:')
print('average CEE = '+str(np.mean(cee_vector_MK)))
print('minimum CEE = '+str(min(cee_vector_MK)))
print('maximum CEE = '+str(max(cee_vector_MK)))
print('st.dev  CEE = '+str(np.std(cee_vector_MK)))
print('confidence = '); print(str(np.mean(cee_vector_MK)-np.std(cee_vector_MK)) + ' - ' + str(np.mean(cee_vector_MK)+np.std(cee_vector_MK)))
plt.hist(cee_vector_MK, 50)
plt.axvline(cee_MK, color='red')
plt.axvline(np.mean(cee_vector_MK)-np.std(cee_vector_MK), color='orange')
plt.axvline(np.mean(cee_vector_MK)+np.std(cee_vector_MK), color='orange')
plt.title('Cumulative Erosion Error Histogram, Mamontovy Khayata')
plt.ylabel('Number of Occurrences')
plt.xlabel('Cumulative Erosion Error [m]')
plt.show()
```