

# Project 3: Comparison of Different Statistical Models to predict Surface Mass Balance over Valley Glaciers

Britt Haanen, Pragay Shourya Moudgil

December 12, 2024

## Abstract

The retreat of glaciers has received considerable attention due to its effects on water availability and hydropower generation, raising important concerns for both the environment and society. Therefore, understanding the impact of climate on glacier evolution has become crucial. The present report explores the application of various statistical models, namely Linear Regression, Ridge Regression, Neural Networks, XG Boost, and Random Forest, to predict surface mass balance. Our analysis includes a broad range of meteorological, topographical, and radiational input variables. The performance of these models is evaluated based on different combinations of input variables to determine their impact on prediction accuracy. The Random Forest model consistently demonstrated superior performance, achieving the highest accuracy across most combinations of input variables. When all variables were included, XGBoost showed superior performance with an  $R^2$  score of 0.84. It was closely followed by the Random Forest model with an  $R^2$  score of 0.83. Moreover, the Mean Absolute Error (MAE) further underscored these findings. However, Linear model such as Ordinary Least Square and Ridge Regression underperformed for all combinations. These findings emphasize the importance of feature selection in improving model performance and indicate the possible usefulness of ensemble methods to support gradient boosting in this context.

## 1 Introduction

The retreat of glaciers, amongst others in European Alps, is one of the most stringent symptoms of climate change. The loss of mass and subsequent retreat of glaciers has been accelerating since the 1980s because of human activity [49, 32] and is projected to continue in the future [35]. This phenomenon carries with it a substantial set of environmental and societal consequences [8]. An illustrative example is the sea level rise: the estimated contribution of glaciers to sea level rise is reported between  $0.77 \pm 0.31$  [23] and  $0.92 \pm 0.39$  [48] mm sea-level equivalent per year between 2006 and 2015, roughly constituting 25% of the observed sea-level rise. It is thus desirable, among climate scientists, to have accurate models to study glacier evolution.

Glacier mass is dictated by a balance between surface accumulation (through solid precipitation, refreezing, wind-blown snow, and avalanches) and surface ablation (mainly through surface melt and sublimation) [5]. Thus, glacier mass is dictated by a variety of parameters. The surface mass balance (SMB) combines local ablation and accumulation processes into a relatively simple metric for a specific point on a glacier. Thus, it allows us to study the local impact of glacier change through accumulation and ablation [45, 44, 26]. By using a time series of SMB values for a particular site, numerical models may also be trained to simulate glacier evolution. SMB can be measured through direct field measurements [6, 2, 28] or by using empirical or physically-based models [30, 27], though the required variables for the latter are generally unavailable.

Recently, machine learning (ML) methods have become of interest to model glacier evolution. Steiner et. al. [39] used Artificial Neural Networks (ANN) to simulate mass balances, focusing on the Grosse Aletsch glacier in Switzerland. Bolibar et. al. [7] proposed the Alpine Parameterized Glacier Model (ALPGM), which employs an AI-based approach to simulate annual glacier-wide surface mass balances and the evolution of glacier volume and surface area over various time scales. A recent study by Van der Meer et. al. [41] applies a ML model based on decision trees and XGBoost to estimate SMB from meteorological data on precipitation and temperature. The method was benchmarked against a gold standard numerical model, and upon comparison of both models to the actual SMB data, it was

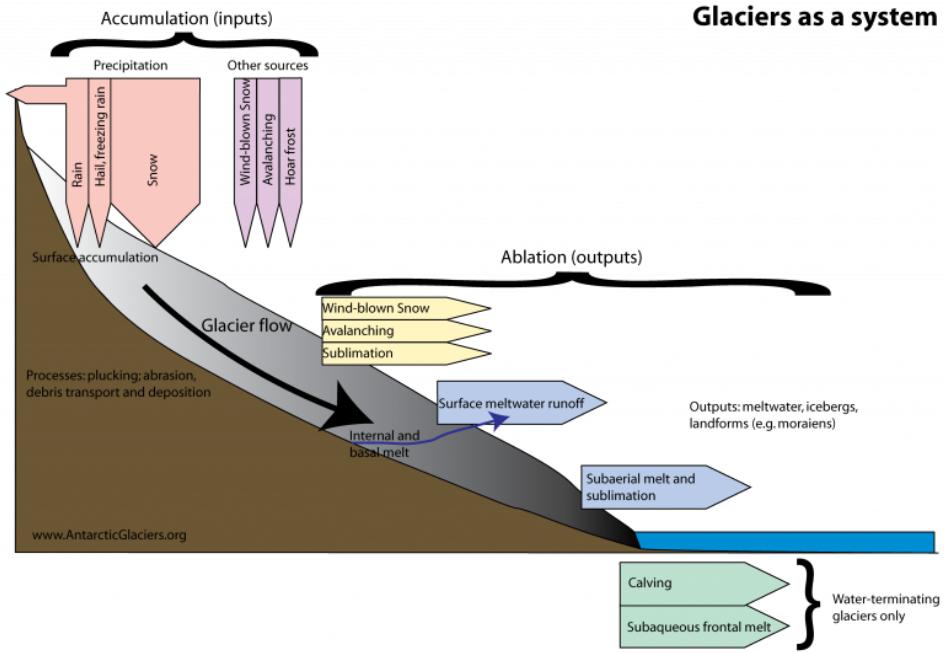


Figure 1: The inputs to a glacier system (accumulation) are precipitation and wind-blown snow and avalanches. The glacier loses mass (ablation) mainly from calving and surface melt. (*source: [www.antarcticglaciers.org](http://www.antarcticglaciers.org)*)

found that the XGBoost model was highly accurate in modelling SMB. Van der Meer et. al. reported an MAE of 417 mm.

Despite recent efforts to better understand the applicability of Machine Learning/Deep Learning models in Surface Mass Balance predictions, the literature remains limited. This project aims to bridge the current gap by applying some of the methods learned in this course to the same data. Specifically, we have applied Linear regression (both OLS and Ridge), a Neural Network, XGBoost, and Random Forest models to the data set. The underlying theory of the applied methods will first be described in Section 2. Next, the technical implementation of these methods will be described in Section 3. The results obtained with our models will be presented in Section 4, and directions for future study will be discussed in Section 5.

## 2 Theory

This section will describe the relevant theoretical concepts for understanding this report. Firstly, we will discuss the Surface Mass Balance, which is the data that we wish to model. Next, we will describe the theoretical background behind the implemented models. Lastly, we will examine the metrics that were used to evaluate model performance.

### 2.1 Surface Mass Balance

The gain and loss of ice in glacier systems is known as mass balance. It refers to a change in the mass of all or part of a glacier over some specified period. It can also be defined as the total sum of all the accumulation (e.g., snow, ice, freezing rain) and melt or ice loss (e.g., calving icebergs, melting, sublimation) across the entire glacier [5]. If glaciers have an equilibrium mass balance with climate, they remain the same size and do not grow or shrink. But if the glacier is not in equilibrium, the glacier will grow with a positive mass balance or shrink with a negative mass balance.

$$b = c + a \quad (1)$$

Where b is Mass Balance, c is the accumulation, and a is ablation.

Surface mass balance (SMB) depends on different climatic and topographical features. One of the most commonly used models in mass balance utilizes air temperature and precipitation as atmospheric forcing variables [22]. Despite its simplicity, this model outperforms various other models. However, even though air temperature and precipitation can explain most of the variance in the data, other factors such as the location of glaciers, their aspect, and slope also impact glacier melting. The energy balance of the glacier ultimately governs the mass balance. Therefore, it is important to analyze these variables with respect to mass balance predictions.

## 2.2 Linear regression models

Linear regression models may be the simplest form of ML models. The aim is to fit a continuous function  $f(x)$  to our data, where this function has been parametrised in terms of the parameters  $\beta$ . In order to find the optimal parameters (i.e. those values of  $\beta$  that minimise our cost function) one solves analytical expressions for these optimal model parameters [18, 21].

### 2.2.1 Ordinary Least Squares

Ordinary Least Squares (OLS) presents the most simple form of linear regression. It aims to minimise the sum of squares between the predicted values  $\tilde{y}_i$  and the real data  $y_i$

$$C(\beta) = \frac{1}{2n} \sum_{i=0}^{n-1} (y_i - \tilde{y}_i)^2 \quad (2)$$

We also recall that the optimal parameters are given by:

$$\hat{\beta}_{OLS} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (3)$$

where  $\mathbf{X}$  is the design matrix containing the training data and  $\mathbf{y}$  are the corresponding real data values [18]. For a more elaborate discussion of OLS regression and some mathematical proofs regarding the expectation values and variance of the model and its parameters we refer the reader to Project 1 [13].

### 2.2.2 Ridge regression

Ridge regression adds to the cost function defined by Equation 2 by adding a term with hyperparameter  $\lambda$  as follows [18]

$$C(\beta) = \frac{1}{2n} \sum_{i=0}^{n-1} (y_i - \tilde{y}_i)^2 + \lambda \beta^T \beta \quad (4)$$

leading to the equation for optimal parameters [18]

$$\hat{\beta}_{Ridge} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y} \quad (5)$$

The hyperparameter  $\lambda$  adds a penalisation for larger values of  $\beta$ , thus it may prevent overfit [36]. Ridge therefore presents a slightly more refined approach than OLS, though it does require tuning of an extra parameter. Again, more details on Ridge regression may be found in Project 1 [13].

## 2.3 Machine Learning and Deep Learning models

### 2.3.1 Neural networks

Neural networks are models which produce statistical predictions based on input data. These predictions are created by feeding data through the model layers, where each layer is associated with a set of parameters (weights  $w$  and biases  $b$ ) that we wish to adjust to arrive at the best possible predictions [40]. Figure 2 shows the concept of a neural network which has an input layer, a hidden layer, and an output layer. Each layer consists of an arbitrary number of nodes, represented in the figure by a circle. The steps of predicting and then adjusting the weights, which also requires the step backpropagation, has been discussed in detail in Project 2 [14].

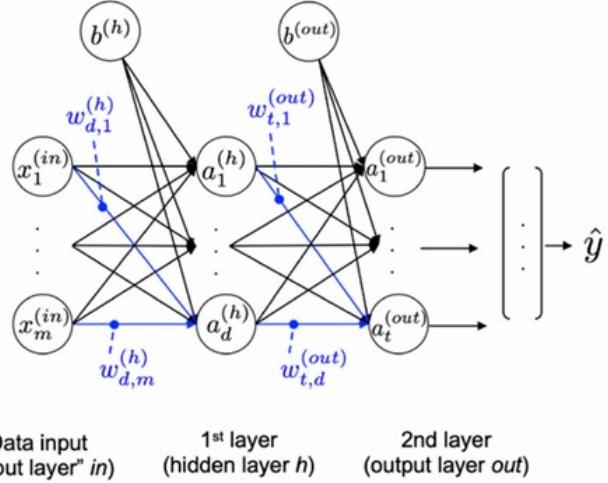


Figure 2: A conceptual illustration of a neural network, taken from Raschka et. al. [36].

The NN makes predictions based on the input data  $x$  during its feed forward pass. First, data input is performed in the input layer. Then, the inputs are propagated to the hidden layer by mathematical operations with the weights  $w^{(h)}$  and the biases  $b^{(h)}$ , so that the input in the model is transformed to the input for a node. Nodes apply an activation function and then propagate the result to the next layer using the same mathematical operations with  $w$  and  $b$ . In Figure 2, this next layer would be the output, but a NN can also contain any additional number of hidden layers before the output. In the output layer, the resulting values are transformed to predicted values  $\hat{y}$  for the input data  $x$  using a final activation function [36, 40].

In order to arrive at the best possible predictions, the network will be trained by updating its parameters so that:

$$\begin{aligned} w &= w - \eta \Delta w \\ b &= b - \eta \Delta w \end{aligned}$$

The updates to the parameters are performed using gradient descent, so that the parameter updates  $\Delta w$  and  $\Delta b$  are informed by the gradient of the cost function. Thus, the parameters are updated in such a way that hopefully, the next prediction is more successful. One training cycle of the NN thus consists of the aforementioned feed forward step, followed by a backpropagation to calculate the gradients and then an update of the weights using these gradients in gradient descent [36, 40]. Eventually, one hopes to make predictions that match the target data as closely as possible.

NN are highly flexible in the sense that their behaviour is governed by many hyperparameters that may be tuned to optimise NN performance for a given task. The description and studying of these hyperparameters was the subject of our previous project [14] and we kindly refer the reader there for a more detailed description of NN and their hyperparameters.

### 2.3.2 Decision trees

Decision trees are a specific type of ML model that may be applied to both classification and regression. A decision tree consists of a set of nodes. At each node, a criterion based on a predictor and a threshold is evaluated and the data is then split accordingly [20]. A typical example of a decision tree is given in Figure 3. The criterion to be evaluated is formulated in such a way that the Information Gain (IG), given by the gini factor, is maximised [36]. The more nodes the tree possesses, the 'deeper' it is said to be. For regression, the nodes of the decision tree divide the parameter space into regions  $R_i$ . The tree makes predictions by using the inputs to evaluate in which region  $R_i$  the output value belongs. The predicted value that is returned is the mean value of the training values in this region [19]. Decision trees are very transparent models, as is easily seen in Figure 3. However, they are also prone to overfit,

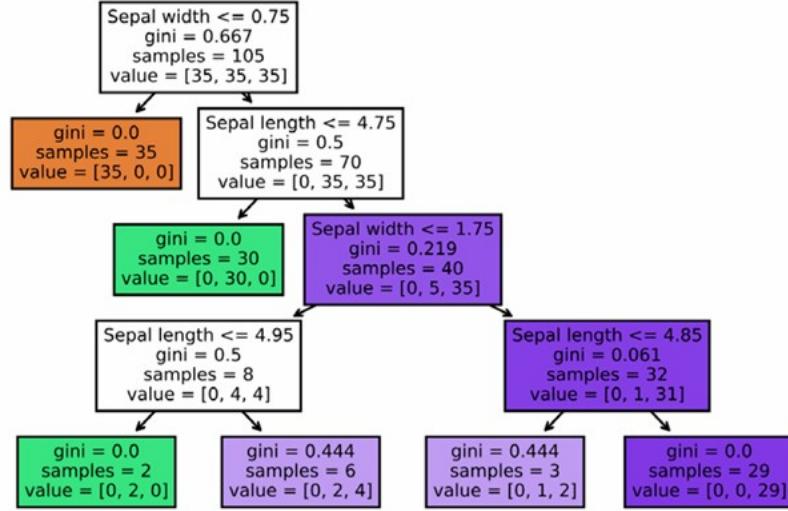


Figure 3: A typical example of a simple decision tree, taken from Raschka et. al. [36].

since the extreme division of the parameter space that results from a deep tree will lead to perfect fit on the training data, but poor generalisation on the test data [19, 36].

In order to leverage the strengths but minimise the weaknesses like overfit of decision trees, they are typically applied as ensemble method. An ensemble method is a meta-classifier made from aggregating individual classifiers such as decision trees. Thus, ensemble methods present greater generalisation performance than individual classifiers [36]. In an ensemble method,  $m$  different classifiers are trained. The trained model makes predictions by letting each classifier make a prediction from the input. This set of predictions is then averaged to find the final prediction as shown in Figure 4. This way, the overfit can be greatly reduced. Next, we will discuss two ensemble methods: Random Forest, which was used in this paper, and XGBoost, which was used by van der Meer et. al. [41] and which we will be comparing our results against.

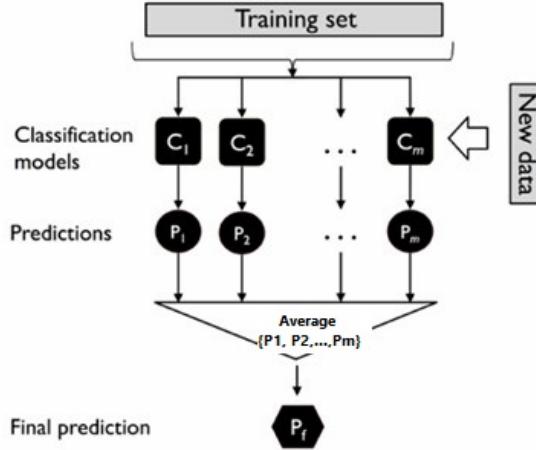


Figure 4: An illustration of how an ensemble method combines the predictions from its individual classifiers into one final prediction. Adapted from Raschka et. al. [36].

### 2.3.3 Random Forest

Random Forest is a bagging method [40, 16]. Bagging methods aim to reduce the high variance that is typically seen for decision trees by growing a set of trees  $C_1, C_2, \dots, C_m$  as illustrated in Figure 4. Each

tree is grown to a different bootstrap sample of the data, which means that it is a sample of the same size as the original data. The values in this bootstrap dataset have been drawn with replacement from the original data [17]. Since each tree is grown for a different bootstrap sample, they will all be slightly different. Thus, their combined predictions will show greater generalisation performance (thus, lower variance and less overfit) than a single decision tree [40, 16].

One substantial weakness of bagging, however, is that the different trees in the set  $C_1, C_2, \dots, C_m$  will typically still show a high degree of similarity. This results from the recursive binary splitting process: strong predictors will dominate the splitting process even across bootstrap samples. As a result, the splits of all trees will be similar, resulting in a fairly homogenous set of trees, also called correlated trees [40, 16, 20].

Random Forests make a small addition to the algorithm, which leads to a decorrelation of the trees. As the tree for a bootstrap sample is being grown, the predictor for the split will be chosen from a random subset of the predictors. It is this inherent randomness that gives Random Forest its name, and this feature gives strong predictors less of a chance to dominate across all trees. The result is a more heterogeneous set of trees, which leads to a great reduction in variance and overfit. Thus, Random Forest will typically perform better than a single tree or classic Bagging [40, 16, 20, 36].

### 2.3.4 XGBoost

Boosting methods also rely on building a set of classifiers, in this case decision trees. Unlike Bagging, which grows a new tree for each Bootstrap sample of data, Boosting methods sequentially grow a set of trees, where each tree is informed by the previous tree's mistakes. Each new tree is added to the model to create an aggregated classifier [36].

The exact manner in which a tree uses the previous tree information and how the trees are combined depends on the specific Boosting method. One popular example is Adaboost [36, 20], but in this section we will focus on Gradient Boosting, and XGBoost specifically.

Gradient boosting, as its name suggests, uses the gradient of the loss function to calculate the required updates to make the new tree perform better than the previous one. The initial tree is initialised as a tree with a single leaf node, thus, it returns a constant value given by the minimum of the loss function.

$$F_0(x) = \operatorname{argmin}(\gamma) \sum_{i=1}^n L(y_i, \gamma)$$

Next, we grow a set of trees of size  $M$ . For each tree  $m$  in the set, one computes the error (also called the pseudo-residual) that was made by the previous tree  $m - 1$  for each data point using the negative loss gradient:

$$r_{im} = - \left[ \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}$$

Note that the use of  $F(x) = F_{m-1}(x)$  so that the residuals are calculated based on the previous trees is the Gradient Boosting method of incorporating the previous model's mistakes in the updates to the new tree. The updates  $\gamma_{jm}$  represent the updates to the tree and are found by minimising the loss function:

$$\gamma_{jm} = \operatorname{argmin}(\gamma) \sum_{x_i \in R_{ij}} L(y_i, F_m(x_i) + \gamma)$$

This expression may be solved for the updates  $\gamma_m$ , which are then added onto the existing model by scaling with a learning rate  $\eta$ :

$$F_m(x) = F_{m-1}(x) + \eta \gamma_m$$

These steps are repeated until  $M$  trees have been grown in this manner. Each tree will incorporate an update based on the previous tree's mistakes [36, 20, 38]. Thus, an initially weakly performing model is continuously improved ('boosted') by adding small updates based on the gradient of the loss function. Adopting the learning rate  $\eta$  to scale the data ensures that the update steps remain small in order to prevent overfitting [36].

XGBoost, which stands for 'Extreme Gradient Boosting', is a specific application of Gradient Boosting. It uses several novel approximations in computing the tree splits, an algorithm for parallel tree learning, and a block structure [20, 9]. The XGBoost algorithm has significantly sped up boosting calculations and shows high predictive power [36]. XGBoost is available as a Python module and its association with good performance has led to its widespread use, among others by van der Meer et. al. [41].

## 2.4 Metrics for evaluation of model performance

The aim of the models in this study is to make predictions based on input data. In order to evaluate how well this prediction  $\hat{y}$  approximates the target values  $y$ , one uses a cost function. The chosen cost function depends on the task to which the algorithm is applied and what is deemed the most informative metric for this task.

In this study the used cost function is the Mean Absolute Error (MAE):

$$MAE = \frac{1}{n} \sum |y_i - \hat{y}_i| \quad (6)$$

which evaluates absolute difference between the prediction  $\hat{y}$  and the target  $y$  for each data point, which is then averaged to find a mean distance for the entire data set [4]. A low MAE means that there is little distance between the predicted points  $\hat{y}$  and the target  $y$ , thus indicating that the predicted values show a close match with the target values [4]. The MAE was the first of two metrics that were used to evaluate model performance, and it was also largely chosen for comparability with Van der Meer et. al. [41].

However, use of other metrics may provide additional information about model performance. Therefore we also computed the  $R^2$  score, which is given as

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y}_i)^2} \quad (7)$$

which is often called 'goodness of fit' [43] as it evaluates how well the model fits the target data using the residuals. The closer the  $R^2$  is to unity, the better the model is fitting to the data.

## 3 Methods

The following section will describe the data processing, preparation of training data and setup of the model architecture.

### 3.1 Study Area and Surface Mass Balance data

According to the Randolph Glacier Inventory (RGI), there are two subregions in Central Europe with glaciers: the Alps and Southeast Europe. The Alps have the highest number, with 4,034 glaciers, compared to Southeast Europe, which has only 45. For this study, we focused on the Swiss Alps, which contain 1,400 glaciers (Fig. 5). Among these, 176 glaciers have been well-documented by GLAMOS (Glacier Monitoring Switzerland). Observations using stakes across the Alps date back to 1939 and continue up to 2023, though they are not continuous. The elevation of the monitoring sites ranges from 2,000 meters above sea level (m.s.l.) to 3,500 m.s.l., with most sites located within the 2,500 to 3,000 m.s.l. range (Fig. 6).

The GLAMOS surface mass balance data is unique due to its extensive spatiotemporal coverage. Surface mass balance is measured with stakes that are relocated to their initial positions at specific intervals, recording the height above the surface. Alongside the surface mass balance data, site-specific density measurements are also provided. Measurements are typically taken twice a year: once during maximum snow depth, usually in April, and once at the end of the melt season, typically in September.

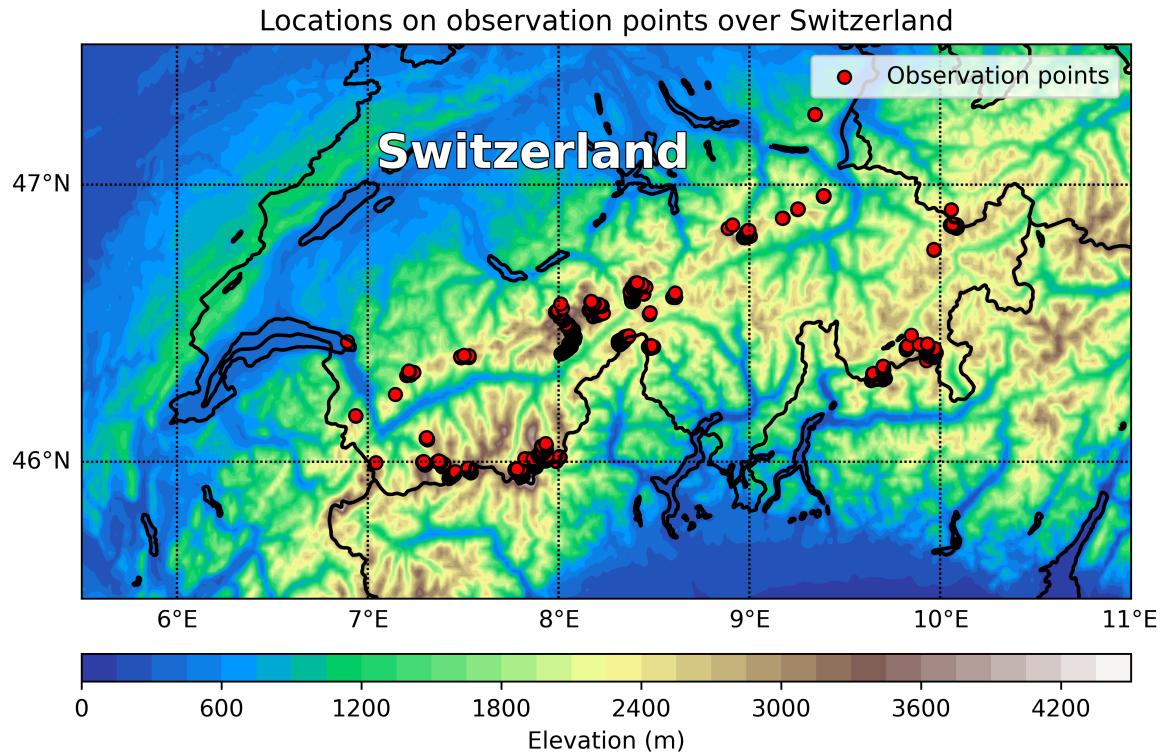


Figure 5: Study region and location of the stacked data used for training the model. The Surface Mass Balance data has been downloaded from Glacier Monitoring Switzerland (GLAMOS [11]).

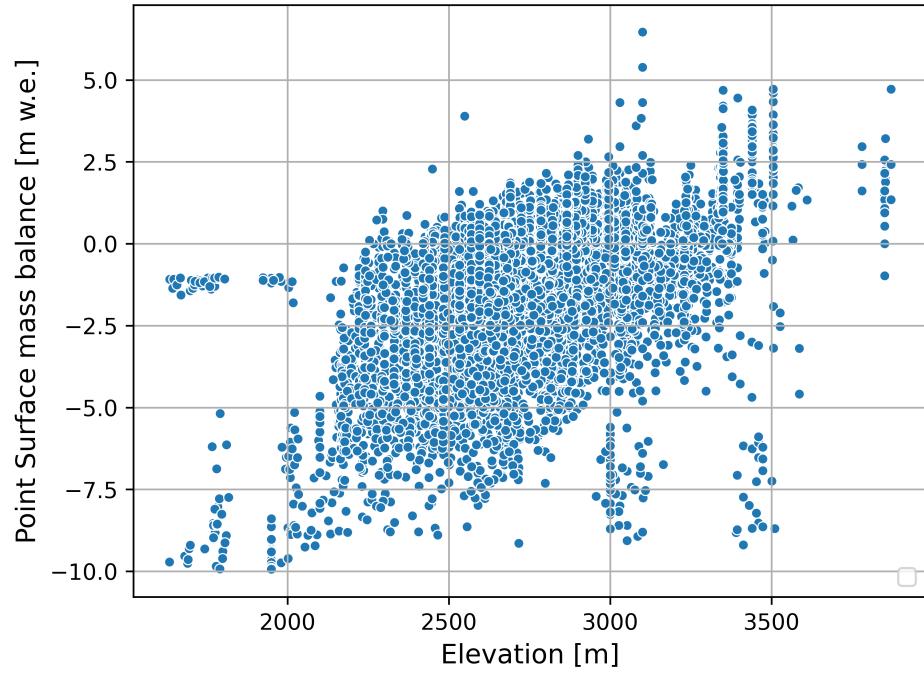


Figure 6: Variation of Surface Mass Balance w.r.t Elevation at which observation has been recorded.

However, due to weather and other constraints, the measurement dates can be inconsistent. To address this, all stake measurements have been standardized to align with the hydrological year, which runs from October 1st to September 30th of the following year. For more detailed overview of GLAMOS, one can refer to Geibel et. al. [11].

### 3.2 Training features

In Section 2, we discussed how surface mass balance is influenced by a range of factors, including meteorological elements such as air temperature and precipitation, as well as topographical features. Given that the energy balance of the glacier significantly impacts the melting rate of ice, additional features such as albedo, longwave radiation, and shortwave radiation have also been included in our analysis [3].

Table 1 provides a comprehensive summary of the various features used for model training. For each stake, data was extracted based on the nearest neighboring pixel relative to the stake's latitude and longitude coordinates. Since the spatial resolution of input features is very coarse, the elevation of each stake and the corresponding altitude height of the climate data were also incorporated as training features. A similar approach has also been adopted in the preprint by Van der Meer et. al. [41].

Table 1: The features that have been used to train the models, with associated symbols.

S.No.	Variable	Symbol	Source	Resolution
1	2 meter temperature	t2m	ERA5	0.25
2	Total Precipitation	tp	ERA5	0.25
3	Surface net long-wave (thermal) radiation	str	ERA5	0.25
4	Surface sensible heat flux	sshf	ERA5	0.25
5	Surface shortwave (solar) radiation	ssrd	ERA5	0.25
6	Forcast Albedo	fal	ERA5	0.25
7	Elevation of the stake	Point_elv	Insitu	-
8	Climate data elevation	alt_elv	ERA5	0.25

Given the wide variety of factors that influence surface mass balance, it was deemed interesting to this study to examine the impact of various input features on model training. Therefore, different combinations of input variables (Table 2) were prepared and split into 80% training and 20% testing datasets.

Table 2: Input variable combinations that were used to perform different trainings.

Number	Input Variables
1	't2m'
2	't2m', 'tp'
3	't2m', 'tp', 'str'
4	't2m', 'tp', 'str', 'sshf'
5	't2m', 'tp', 'str', 'sshf', 'ssrd'
6	't2m', 'tp', 'str', 'sshf', 'ssrd', 'fal'
7	't2m', 'tp', 'str', 'sshf', 'ssrd', 'fal', 'POINT_ELEVATION'
8	't2m', 'tp', 'str', 'sshf', 'ssrd', 'fal', 'POINT_ELEVATION', 'altitude_climate'

### 3.3 Model Architecture

The model architecture for the present study has been intentionally kept simple. The neural network was initially set up using the TensorFlow architecture [1]. However, this was very time-consuming in combination with the grid search, so scikit-learn's MLP Regressor module [33] was used as a more time-efficient alternative. XGBoost was implemented using the aforementioned XGBoost Python module [9]. The Random Forest, Linear Regression and Ridge Regression models were implemented using scikit-learn [33]. For all these models the hyperparameters were tuned using grid search, which was

implemented using `GridSearchCV` from scikit-learn [33]. The aforementioned model metrics  $R^2$  score and MAE were also computed using scikit-learn functions [33].

In their study, Van der Meer et. al. [41] also used a simple XGBoost model, which was optimized using grid search to perform best. However, in one of their cases, they predicted the monthly surface mass balance and aggregated the monthly predictions to obtain the annual surface mass balance to determine the loss. In contrast, in the present study, we convert the monthly input data to annual data prior to modeling, thereby enabling the model to directly predict the annual surface mass balance.

## 4 Results

This section presents the findings of our study, focusing on the performance of various machine learning (ML) and deep learning (DL) models with different combinations of input variables and algorithms. As outlined in 3.2, the impact of different combinations of input variables (Table 2) were prepared for input into the models. The different models were set up as described in Section 3.3, trained, and evaluated for performance on a testing set using the previously described  $R^2$  and MAE metrics.

### 4.1 Analyzing Input and Target Variables

Analyzing input features and their relationship with the target variable is essential for effective ML and DL model training. Including an excessive number of input variables can lead to model overfitting [36]. Figure 7 illustrates the correlation plot between various input variables and the target variable.

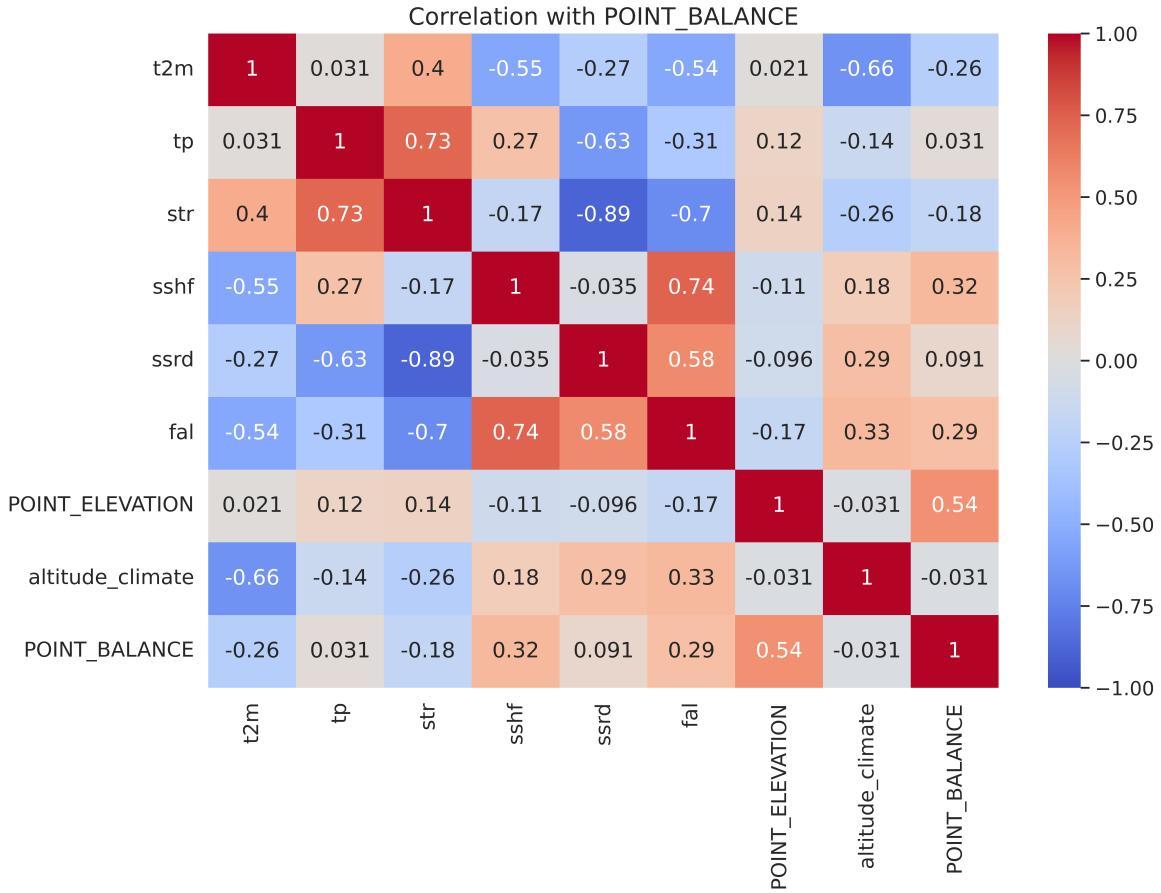


Figure 7: Correlation plot between different variables. Surface Mass Balance shows significant correlations with temperature at 2 meters (t2m), surface sensible heat flux (sshf), forecasted albedo (fal), and elevation of the stack. The variables also show strong inter-correlations.

As shown in Figure 7, the surface mass balance has a significant correlation with variables such as temperature at 2 meters (t2m), surface sensible heat flux (sshf), forecasted albedo (fal), and elevation of the stack. However, there is no significant correlation with total precipitation (tp), surface shortwave radiation (ssrd), and altitude from the ERA5 climate data. Additionally, many variables exhibit high inter-correlation among themselves. Therefore, even though input variables may not be directly correlated, they could still hold valuable information. This suggests that further evaluation is necessary to fully understand their contributions.

Table 3: Performance of the each model with different combination of input dataset. It is interesting to note that almost all model see a performance boost when elevation of the point in included in the training (7). Random Forest performs best for the sparser combinations (1-6), after which it is overtaken by XGB Regressor.

Input Variables	Model	Train $R^2$	Test $R^2$	Train MAE	Test MAE
1	LinearRegression	0.07	0.07	1608.31	1606.73
	Ridge	0.07	0.07	1608.31	1606.73
	RandomForestRegressor	<b>0.59</b>	<b>0.48</b>	<b>979.66</b>	<b>1103.18</b>
	XGBRegressor	0.28	0.27	1352.47	1368.96
	ANNRegressor	0.10	0.10	1571.03	1574.16
2	LinearRegression	0.07	0.08	1607.83	1603.01
	Ridge	0.07	0.08	1607.83	1603.01
	RandomForestRegressor	<b>0.60</b>	<b>0.50</b>	<b>972.45</b>	<b>1085.07</b>
	XGBRegressor	0.54	0.46	1066.52	1143.75
	ANNRegressor	0.21	0.20	1447.77	1461.82
3	LinearRegression	0.10	0.12	1565.16	1559.44
	Ridge	0.10	0.12	1565.16	1559.44
	RandomForestRegressor	<b>0.60</b>	<b>0.51</b>	<b>971.66</b>	<b>1079.27</b>
	XGBRegressor	0.57	0.50	1025.30	1102.34
	ANNRegressor	0.40	0.38	1229.44	1249.93
4	LinearRegression	0.13	0.14	1538.29	1533.76
	Ridge	0.13	0.14	1538.29	1533.76
	RandomForestRegressor	<b>0.60</b>	<b>0.52</b>	<b>970.84</b>	<b>1076.59</b>
	XGBRegressor	0.58	0.51	1008.58	1097.87
	ANNRegressor	0.46	0.43	1161.08	1200.41
5	LinearRegression	0.13	0.15	1530.62	1525.37
	Ridge	0.13	0.15	1530.62	1525.37
	RandomForestRegressor	<b>0.60</b>	<b>0.52</b>	<b>969.56</b>	<b>1068.59</b>
	XGBRegressor	0.59	0.51	994.18	1086.36
	ANNRegressor	0.42	0.40	1191.10	1215.30
6	LinearRegression	0.13	0.15	1531.21	1526.10
	Ridge	0.13	0.15	1531.22	1526.11
	RandomForestRegressor	<b>0.61</b>	<b>0.53</b>	<b>957.92</b>	<b>1051.63</b>
	XGBRegressor	0.59	0.52	980.49	1059.06
	ANNRegressor	0.46	0.44	1143.58	1168.28
7	LinearRegression	0.48	0.48	1132.27	1164.78
	Ridge	0.48	0.48	1132.27	1164.81
	RandomForestRegressor	<b>0.97</b>	0.80	<b>246.73</b>	<b>627.21</b>
	XGBRegressor	0.92	<b>0.81</b>	425.57	<b>612.76</b>
	ANNRegressor	0.81	0.73	624.34	732.39
8	LinearRegression	0.51	0.51	1096.80	1118.34
	Ridge	0.51	0.51	1096.80	1118.35
	RandomForestRegressor	<b>0.97</b>	0.83	<b>233.16</b>	588.26
	XGBRegressor	0.93	<b>0.84</b>	390.95	<b>561.95</b>
	ANNRegressor	0.83	0.78	638.26	708.57

## 4.2 Performance of Different Models

The performance metrics for different model-input combinations, including the impact of adding each new variable, are summarized in Table 3. It provides the performance metrics (Train  $R^2$ , Test  $R^2$ , Train MAE, Test MAE) for all models as given in Section 3.3 across all different input combinations as given in Table 2. In contrast to Van der Meer et. al. [41], where monthly annual mass balance is predicted and aggregated to annual surface mass balance, we have resampled the input data to annual, therefore predicting annual surface mass balance.

The data is also represented in Figures 8 and 9. These are plots of the obtained values across all

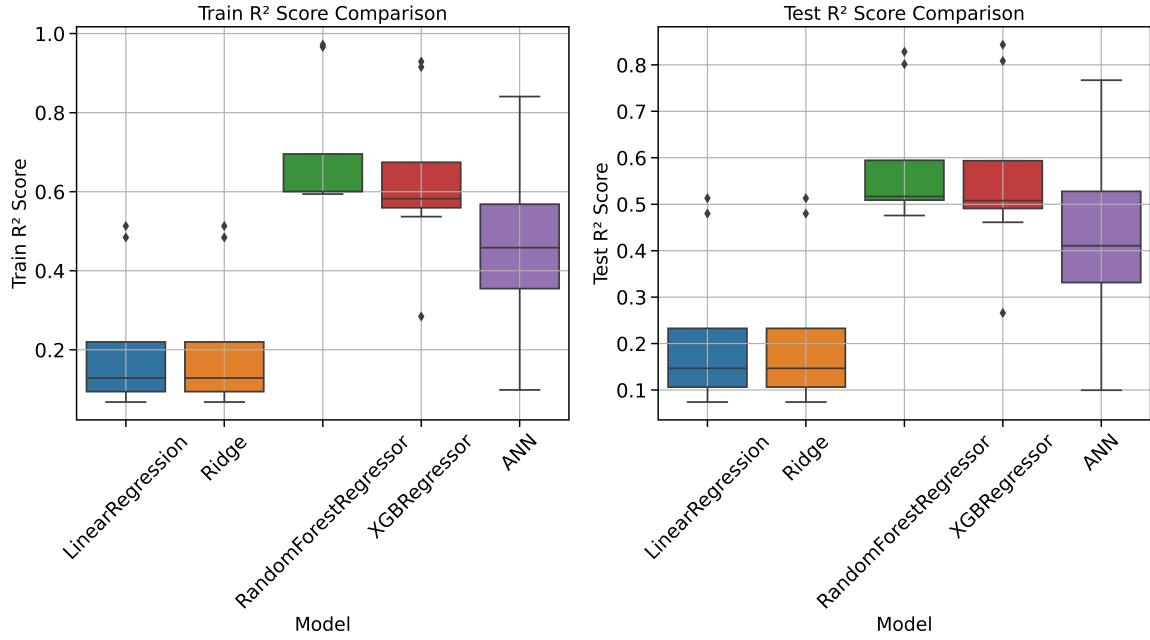


Figure 8: Comparison of each models with different input datasets combination. It is interesting to note that Random forest performs well across all data sets. The wide box and whiskers of ANN shows that it is influenced greatly by the amount of input variables.

input variable combinations for each model. The relatively small box and whiskers for the Random Forest method, as compared to other models, shows that it performs fairly across all data sets. The large box and whiskers for the Artificial Neural Network (the MLP regressor), on the other hand, shows its great variability as function of the input variables. This highlights that model performance is greatly influenced by the inputs that it is fed.

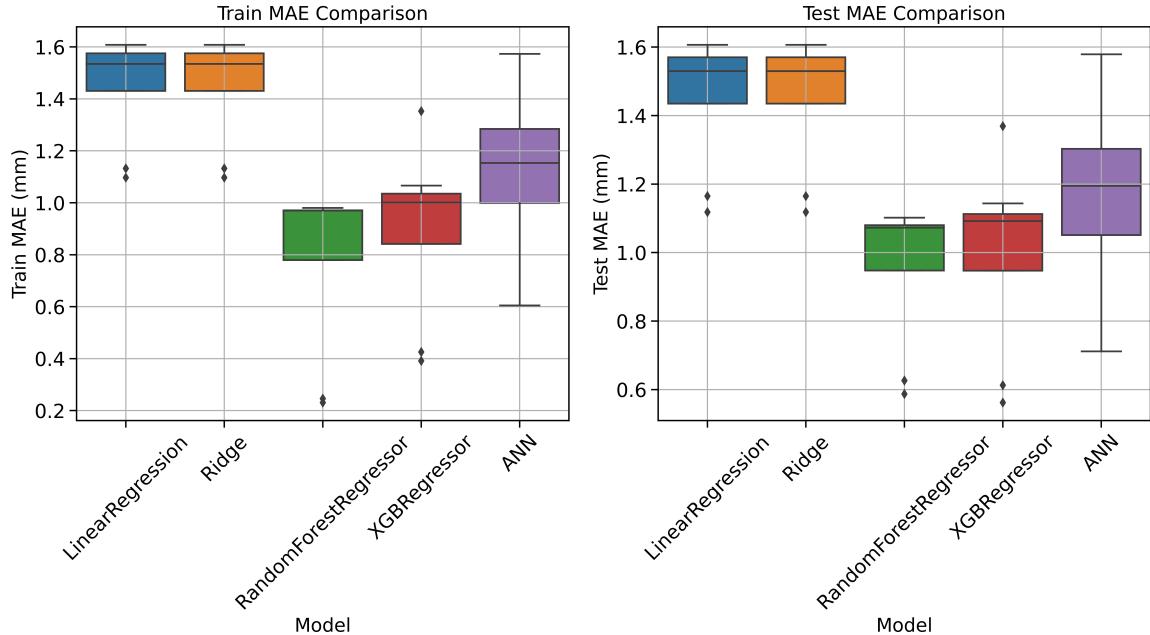


Figure 9: Comparison of Mean Absolute error for each model with different input combinations. It is interesting to note that Random forest performs well across all data sets. The wide box and whiskers of ANN shows that it is influenced greatly by the amount of input variables.

With the highest Train  $R^2$  and Test  $R^2$  and the lowest Train MAE and Test MAE values, it is evident from Table 3 that the Random Forest regressor performs very well across all input combinations. The addition of the 'POINT\_ELEVATION' variable (included from set 7) resulted in an overall improvement in model performance, demonstrating its influence on model training. Additionally, the XGB Regressor performed well, particularly when all variables were included, where it outperformed the Random Forest.

For the simplest model with only 't2m' as the input variable, the Random Forest regressor achieved a Test  $R^2$  of 0.48 and a Test MAE of 1103.18, outperforming Linear, Ridge, and MLP regressors. Adding extra variables ('t2m', 'tp', 'str', 'sshf', 'ssrd', 'fal', 'POINT\_ELEVATION', 'altitude ERA5\_climate') improved the performance of the Random Forest Regressor, which reached a Test  $R^2$  of 0.83 and a Test MAE of 588.26. For this combination of inputs, XGBoost performed slightly better with a Test  $R^2$  of 0.84 and a Test MAE of 561.95.

## 5 Discussion

Section 4 has evaluated the performance of ML and DL models using various input variables. This discussion will delve deeper into these findings, emphasizing key observations and their implications for predicting surface mass balance.

### 5.1 Importance of Input Features

The correlation plot depicted in Figure 7 discusses the relationships between input features and the target variable, emphasizing the significant correlations among variables such as temperature at 2 meters (t2m), surface sensible heat flux (sshf), forecasted albedo (fal), and elevation with surface mass balance. In contrast, certain variables from the ERA5 climate data—such as total precipitation (tp), surface shortwave radiation (ssrd), and altitude—did not show direct significant associations with the target variable.

It is important to note that the correlation plot in this context provides an average picture of these relationships, as the input variables were aggregated to match the temporal resolution of the target variable. Thus, the potential importance of factors with weak direct relationships should not be neglected. Their importance might become evident under specific conditions or within particular subsets of data, such as stakes experiencing higher precipitation levels versus those with significant temperature variations. This observed complexity aligns with studies conducted by [3] and [7], which emphasize the varying importance of different features and the contextual significance of seasonal snowfall patterns. Also, the correlation plot presented here amalgamates data from various stakes, suggesting that the relevance of variables may vary across different subsets.

### 5.2 Model Performance and Evaluation

The Random Forest regressor performs better than other models for the first sets of input combinations, as shown in Table 3. This model demonstrated strong generalization capabilities, consistently producing the highest Train  $R^2$  and Test  $R^2$  values while also achieving the lowest Train MAE and Test MAE values. The Random Forest regressor's ability to achieve a Test  $R^2$  of 0.48 and a Test MAE of 1103.18 using only the input variable 't2m' shows its usefulness even with limited input data. Though this performance is not very impressive in itself, the model's performance quickly improved when more variables were included, reaching a Test  $R^2$  of 0.83 and a Test MAE of 588.26 with all input variables ('t2m', 'tp', 'str', 'sshf', 'ssrd', 'fal', 'POINT\_ELEVATION', 'altitude ERA5\_climate'). Interestingly, XGB Regressor performance also consistently increased as more input variables were included. For combinations 7 and 8 it performed slightly better than Random Forest. Its best performance was for all input variables being included, resulting in a Test  $R^2$  of 0.84 and Test MAE of 561.95. It may also be interesting to note that the difference between training and testing scores seem less drastic for the XGB Regressor than for the Random Forest. This suggests that the XGB Regressor is less prone to overfit.

Overall, these findings show that gradient boosting techniques can effectively capture complex patterns in data to efficiently predict surface mass balance data. This is in line with findings by Van der Meer et. al. [41], who found a MAE of 417 mm. This is a better result than was achieved in this

study, though one should also consider that the data was processed differently as was at the end of Section 3.3.

In addition to agreeing with existing literature on the good performance of XGBoost, this study has also shown a nice complementary method to XGBoost in this context in the form of Random Forests. Especially in case of sparser input variables, our data suggests that Random Forest may perform better than XGBoost. Since the performance of both models is so close, it would be interesting to future research to consider how Random Forests may be implemented as a complement to or replacement of XGBoost (in case of sparse input features) in surface mass balance modelling.

As described before in Section 4.2, the performance of various model increased drastically with inclusion of elevation data for each stake into input variables. This was also evident from the correlation plot (fig 7), where strong correlation (0.57) was observed with surface mass balance. Figure 6 illustrates the relationship between elevation and surface mass balance, showing that lower elevations correspond to more negative surface mass balance, while higher elevations indicate a more positive balance. One possible explanation for this pattern is that point elevation is the only high-resolution data unique to each measurement stake, providing more valuable information to the model. In contrast, other datasets incorporated for training are of much coarser resolution, essentially providing the same input for different outputs. As noted by [3], ERA5 reanalysis data do not fully reflect point-scale data due to their coarse resolution. Similarly, a study by [31] emphasizes that coarser resolution fails to account for forces exerted by the Earth’s surface on atmospheric flow and thus cannot resolve the impact of complex terrain on local weather conditions.

### 5.3 Implications for Model Development and Future Research

With the emergence of Artificial Intelligence techniques, there followed an explosion of studies employing different algorithms for a wide variety of applications. Most of these studies conducted utilize deep learning methods such as neural networks to incorporate a non-linearity component in the models. The present study indicated that, even though the deep learning method is state of the art, traditional machine learning techniques are still valuable and should be utilized more. As demonstrated by Schwartz-Ziv & Armon [37] and Grinsztajn et. al. [12], tree-based methods perform well on medium-sized datasets (samples under 10,000). Given that glacier mass balance datasets are often medium-sized datasets with correlated input features, we recommend that studies planning to apply machine learning to model the Earth system also consider the possible usefulness ensemble-based methodologies.

This study’s key finding indicates that choosing the appropriate set of input variables is crucial to developing predictive models that work. A thorough examination of the input features and their interactions can help reduce overfitting, a common problem in the training of ML and DL models. In our study, we demonstrated the importance of point elevation. Though we tried 8 different input parameter combinations, this variable was included relatively late. In future, it might be interesting to compare XGBoost and Random Forest performance on sparser data sets that do include point elevation. In addition, an even more extensive research of input variable combinations could lead to interesting conclusions, since even though some variables lack strong direct correlations, they may nevertheless contain important information that needs more investigation. Future studies ought to look into the significance of input variables over different time periods. Research by [3] and [7] has looked at seasonal and monthly variations in the significance of features. Further understanding of how to maximize predictive accuracy can also be gained by looking into complicated feature selection methods and how they impact model performance.

## 6 Conclusion

The present study highlights the intricate relationship between various meteorological, topographical, and radiational factors in predicting surface mass balance over valley glaciers. Notably, point elevation emerged as an important high-resolution input variable, providing significant predictive power compared to coarser resolution ERA5 climate data. Our findings underscore the importance of thoughtful feature selection and high-resolution data in enhancing model accuracy, as indicated by previous studies. Through a comprehensive analysis of different statistical models—Linear Regression, Ridge Regression, Neural Networks, XGBoost, and Random Forest—we have demonstrated that the Random Forest model consistently outperformed other models across different sparse combinations of input variables.

For inclusion of more input variables (7 or 8) XGBoost was the best performer as is in line with previous literature, however, Random Forest remained a close second. In addition to agreeing with literature on the potential of XGBoost in Surface Mass Balance modelling, this study has highlighted Random Forests as possibly interesting complement to this method, especially for a sparser set of input variables. This work lays a solid foundation for future research aimed at refining predictive models for glacier mass balance, particularly by exploring seasonal and monthly variances in feature importance, and optimizing model architectures and training strategies.

## 7 List of codes

All codes used for this report are given in one compiled code file 'Project3\_compiled\_final.ipynb' which can be found in the GitHub repository [https://github.com/BrittHaanen15/Project3\\_FinalSubmission.git](https://github.com/BrittHaanen15/Project3_FinalSubmission.git). The repository also contains the input data, a map, the figures, and a csv file of the numerical results that formed the basis for Table 3.

What follows is a brief description of each part of the code with references to used libraries and sources. In the code file itself these sources are either named in the comments or, in the case of libraries, apparent from the syntax. The assistance of GitHub CoPilot is applicable to all parts of the code.

1. **Preamble:** this code imports the relevant packages NumPy [15], pandas [47], geopandas [29], scikit-learn [33], matplotlib [25], tqdm [10], seaborn [46], standard modules warnings, sys and joblib [42], xgboost [9], xarray [24] and json [34].
2. **Data import and processing:** this code imports the surface mass balance data and generates the associated plots for the Study Area section. It also performs the correlation analysis by plotting the correlation matrix.
3. **Setup of test function:** this code sets up the function that is used to test a particular model. It includes splitting of the data using `train_test_split`, setup of the grid search, hyperparameter tuning, and computation of the evaluation metrics using scikit-learn [33].
4. **Definition of models:** this code sets up the inputs for the models and the models themselves. Linear Regression, Ridge Regression, Random Forest, and MLP Regressor (ANN) are set up using scikit-learn [33]. The XGBoost model is defined using the xgboost module [9]
5. **Generation of results:** this code implements the test function to test all models across various inputs. The results are stored, exported numerically for tables and plotted as box plots using matplotlib [25].

## References

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [2] Peter Adams. Glossary of glacier mass balance and related terms, prepared by the working group on mass-balance terminology and methods of the international association of cryospheric sciences (iasc). *ARCTIC*, 64(4), December 2011.
- [3] Ritu Anilkumar, Rishikesh Bharti, Dibyajyoti Chutia, and Shiv Prasad Aggarwal. Modelling point mass balance for the glaciers of the central european alps using machine learning techniques. *The Cryosphere*, 17(7):2811–2828, 2023.
- [4] Anishnama. Understanding cost functions in machine learning: Types and applications. Web Article, March 2023. Medium, <https://medium.com/@anishnama20/understanding-cost-functions-in-machine-learning-types-and-applications-cd7d8cc4b47d>.
- [5] Douglas Benn and David J A Evans. *Glaciers and Glaciation, 2nd edition*. Routledge, February 2014.
- [6] Erik W. Blake. Glacier mass-balance measurements: A manual for field and office work, by g. Østrem and m. brugman. *ARCTIC*, 46(3), January 1993.
- [7] Jordi Bolibar, Antoine Rabatel, Isabelle Gouttevin, Clovis Galiez, Thomas Condom, and Eric Sauquet. Deep learning applied to glacier evolution modelling. *The Cryosphere*, 14(2):565–584, 2020.
- [8] Katherine Calvin, Dipak Dasgupta, Gerhard Krinner, Aditi Mukherji, Peter W. Thorne, Christopher Trisos, José Romero, Paulina Aldunce, Ko Barrett, Gabriel Blanco, William W.L. Cheung, Sarah Connors, Fatima Denton, Aïda Diongue-Niang, David Dodman, Matthias Garschagen, Oliver Geden, Bronwyn Hayward, Christopher Jones, Frank Jotzo, Thelma Krug, Rodel Lasco, Yune-Yi Lee, Valérie Masson-Delmotte, Malte Meinshausen, Katja Mintenbeck, Abdalah Mokssit, Friederike E.L. Otto, Minal Pathak, Anna Pirani, Elvira Poloczanska, Hans-Otto Pörtner, Aromar Revi, Debra C. Roberts, Joyashree Roy, Alex C. Ruane, Jim Skea, Priyadarshi R. Shukla, Raphael Slade, Aimée Slangen, Youba Sokona, Anna A. Sörensson, Melinda Tignor, Detlef van Vuuren, Yi-Ming Wei, Harald Winkler, Panmao Zhai, Zinta Zommers, Jean-Charles Hourcade, Francis X. Johnson, Shonali Pachauri, Nicholas P. Simpson, Chandni Singh, Adelle Thomas, Edmond Totin, Andrés Alegría, Kyle Armour, Birgit Bednar-Friedl, Kornelis Blok, Guéladio Cissé, Frank Dentener, Siri Eriksen, Erich Fischer, Gregory Garner, Céline Guivarc'h, Marjolijn Haasnoot, Gerrit Hansen, Mathias Hauser, Ed Hawkins, Tim Hermans, Robert Kopp, Noémie Leprince-Ringuet, Jared Lewis, Debora Ley, Chloé Ludden, Leila Niamir, Zebedee Nicholls, Shreya Some, Sophie Szopa, Blair Trewin, Kaj-Ivar van der Wijst, Gundula Winter, Maximilian Witting, Arlene Birt, and Meeyoung Ha. *IPCC, 2023: Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, H. Lee and J. Romero (eds.)]*. IPCC, Geneva, Switzerland. IPCC, July 2023.
- [9] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016.
- [10] Casper O. da Costa-Luis. ‘tqdm’: A fast, extensible progress meter for python and cli. *Journal of Open Source Software*, 4(37):1277, 2019.

- [11] Lea Geibel, Matthias Huss, Claudia Kurzböck, Elias Hodel, Andreas Bauder, and Daniel Farinotti. Rescue and homogenization of 140 years of glacier mass balance data in switzerland. *Earth System Science Data*, 14(7):3293–3312, 2022.
- [12] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35:507–520, 2022.
- [13] Britt Haanen and Pragay Shourya Moudgil. Project 1: Modeling digital elevation model using machine learning. Course project submission, October 2024.
- [14] Britt Haanen and Pragay Shourya Moudgil. Project 2: Using gradient descent and neural network for regression and classification. Course project submission, November 2024.
- [15] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
- [16] Trevor Hastie. Trevor hastie - gradient boosting machine learning. Recording of lecture at <https://www.youtube.com/watch?v=wPqtzj5VZus>, December 2014.
- [17] M Hjorth-Jensen. Applied data analysis and machine learning 5: Resampling methods. Lecture notes, August 2024.
- [18] M Hjorth-Jensen. Week 35: From ordinary linear regression to ridge and lasso regression. Lecture notes, August 2024.
- [19] M Hjorth-Jensen. Week 46: Decision trees, ensemble methods and random forests. Lecture/instruction, November 2024.
- [20] M Hjorth-Jensen. Week 47: From decision trees to ensemble methods, random forests and boosting methods. Lecture notes, November 2024.
- [21] M Hjorth-Jensen. 3. linear regression. Lecture notes, n.d.
- [22] Regine Hock. Temperature index melt modelling in mountain areas. *Journal of hydrology*, 282(1-4):104–115, 2003.
- [23] Regine Hock, Andrew Bliss, Ben Marzeion, Rianne H. Giesen, Yukiko Hirabayashi, Matthias Huss, Valentina Radic, and Aimée B. A. Slanger. Glaciernip – a model intercomparison of global-scale glacier mass-balance models and projections. *Journal of Glaciology*, 65(251):453–467, May 2019.
- [24] S. Hoyer and J. Hamman. xarray: N-D labeled arrays and datasets in Python. *Journal of Open Research Software*, 5(1), 2017.
- [25] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.
- [26] Matthias Huss and Andreas Bauder. 20th-century climate change inferred from four long-term point observations of seasonal mass balance. *Annals of Glaciology*, 50(50):207–214, 2009.
- [27] Matthias Huss, Daniel Farinotti, Andreas Bauder, and Martin Funk. Modelling runoff from highly glacierized alpine drainage basins in a changing climate. *Hydrological Processes*, 22(19):3888–3902, August 2008.
- [28] Matthias Huss and Regine Hock. A new model for global glacier change and sea-level rise. *Frontiers in Earth Science*, 3, September 2015.

- [29] Kelsey Jordahl, Joris Van den Bossche, Martin Fleischmann, Jacob Wasserman, James McBride, Jeffrey Gerard, Jeff Tratner, Matthew Perry, Adrian Garcia Badaracco, Carson Farmer, Geir Arne Hjelle, Alan D. Snow, Micah Cochran, Sean Gillies, Lucas Culbertson, Matt Bartos, Nick Eubank, maxalbert, Aleksey Bilogur, Sergio Rey, Christopher Ren, Dani Arribas-Bel, Leah Wasser, Levi John Wolf, Martin Journois, Joshua Wilson, Adam Greenhall, Chris Holdgraf, Filipe, and François Leblanc. *geopandas/geopandas*: v0.8.1, July 2020.
- [30] M. Kuhn, E. Dreiseitl, S. Hofinger, G. Markl, N. Span, and G. Kaser. Measurements and models of the mass balance of hintereisferner. *Geografiska Annaler, Series A: Physical Geography*, 81(4):659–670, December 1999.
- [31] Changgui Lin, Deliang Chen, Kun Yang, and Tinghai Ou. Impact of model resolution on simulating the water vapor transport through the central himalayas: implication for models' wet bias over the tibetan plateau. *Climate dynamics*, 51:3195–3207, 2018.
- [32] Ben Marzeion, J. Graham Cogley, Kristin Richter, and David Parkes. Attribution of global glacier mass loss to anthropogenic and natural causes. *Science*, 345(6199):919–921, August 2014.
- [33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [34] Felipe Pezoa, Juan L Reutter, Fernando Suarez, Martín Ugarte, and Domagoj Vrgoč. Foundations of json schema. In *Proceedings of the 25th International Conference on World Wide Web*, pages 263–273. International World Wide Web Conferences Steering Committee, 2016.
- [35] David R. Rounce, Regine Hock, Fabien Maussion, Romain Hugonnet, William Kochtitzky, Matthias Huss, Etienne Berthier, Douglas Brinkerhoff, Loris Compagno, Luke Copland, Daniel Farinotti, Brian Menounos, and Robert W. McNabb. Global glacier change in the 21st century: Every increase in temperature matters. *Science*, 379(6627):78–83, January 2023.
- [36] V Mirjalili S Raschka, Y Liu. *Machine Learning with PyTorch and Scikit-Learn*. Packt Publishing Ltd, 1 edition, 2022.
- [37] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.
- [38] J Starmer. Gradient boost part 2 (of 4): Regression details. Video, April 2019. <https://www.youtube.com/watch?v=2xudPOBz-vs>.
- [39] Daniel Steiner, Andreas Walter, and Heinz Jürg Zumbühl. The application of a non-linear back-propagation neural network to study the mass balance of grosse aletschgletscher, switzerland. *Journal of Glaciology*, 51(173):313–323, 2005.
- [40] J Friedman T Hastie, R Tibshirani. *The elements of Statistical Learning*. Springer, 2 edition, 2017.
- [41] Marijn van der Meer, Harry Zekollari, Matthias Huss, Jordi Bolibar, Kamilla Hauknes Sjursen, and Daniel Farinotti. A minimal machine learning glacier mass balance model. September 2024. Preprint.
- [42] Guido Van Rossum. *The Python Library Reference, release 3.12.4*. Python Software Foundation, 2024.
- [43] D Verma. R2 score: Linear regression. Web Article, May 2023. Medium, <https://medium.com/@deependra.verma00/r2-score-linear-regression-e095a1188e87>.
- [44] C. Vincent, A. Fischer, C. Mayer, A. Bauder, S. P. Galos, M. Funk, E. Thibert, D. Six, L. Braun, and M. Huss. Common climatic signal from glaciers in the european alps over the last 50 years. *Geophysical Research Letters*, 44(3):1376–1383, February 2017.

- [45] C. Vincent, G. Kappenberger, F. Valla, A. Bauder, M. Funk, and E. Le Meur. Ice ablation as evidence of climate change in the alps over the 20th century. *Journal of Geophysical Research: Atmospheres*, 109(D10), May 2004.
- [46] Michael L. Waskom. Seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.
- [47] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.
- [48] M. Zemp, M. Huss, E. Thibert, N. Eckert, R. McNabb, J. Huber, M. Barandun, H. Machguth, S. U. Nussbaumer, I. Gärtner-Roer, L. Thomson, F. Paul, F. Maussion, S. Kutuzov, and J. G. Cogley. Global glacier mass changes and their contributions to sea-level rise from 1961 to 2016. *Nature*, 568(7752):382–386, April 2019.
- [49] Zemp, Michael, Paul, F, Hoelzle, M, and Haeberli, W. Glacier fluctuations in the european alps, 1850–2000: an overview and spatio-temporal analysis of available data. In *Darkening Peaks: Glacier Retreat, Science, and Society*. University of California Press, 2008.