Towards deep learning solutions for classification of automated snow height measurements (CleanSnow v1.0.2)

Jan Svoboda[1], Marc Ruesch[1], David Liechti[1], Corinne Jones[2], Michele Volpi[2], Michael Zehnder[1,3], and Jürg Schweizer[1]

[1]WSL Institute for Snow and Avalanche Research SLF, Davos, Switzerland
[2]Swiss Data Science Center, ETH Zürich and EPFL, Zürich, Switzerland
[3]Institute of Integrative Biology, ETH Zürich, Zürich, Switzerland

Correspondence: Jan Svoboda (jan.svoboda@slf.ch)

Abstract. Snow height measurements are still the backbone of any snow cover monitoring, whether based on modeling or remote sensing. These ground-based measurements are often realized using ultrasonic or laser technologies. In challenging environments, such as high alpine regions, the quality of sensor measurements deteriorates quickly, especially in extreme weather conditions or ephemeral snow conditions. Moreover, the sensors by their nature measure the height of an underlying object and are therefore prone to return other information, such as the height of vegetation, in snow-free periods. Quality assessment and real-time classification of automated snow height measurements are therefore desirable to provide high-quality data for research and operational applications. To this end, we propose CleanSnow, a machine learning approach to the automated classification of snow height measurements into a snow cover class and a class corresponding to everything else, which takes into account both the temporal context and the dependencies between snow height and other sensor measurements. We created a new dataset of manually annotated snow height measurements, which allowed us to train our models in a supervised manner as well as quantitatively evaluate our results. Through a series of experiments and ablation studies to evaluate feature importance and compare several different models, we validated our design choices and demonstrated the importance of using temporal information together with information from auxiliary sensors. CleanSnow achieves a high accuracy of almost 98% and represents a new baseline for further research in the field. The presented approach to snow height classification finds its use in various tasks, ranging from snow modeling to climate science.

1 Introduction

Snow height measurements are key in many fields, such as water resources management, avalanche forecasting, climate science, and even tourism. A variety of complex models simulating and calculating snowpack properties therefore exist. For example, estimating snow water equivalent (SWE) (e.g. Jonas et al., 2009) in order to assess water resources. In addition, snow height is an important parameter for snow hydrological (e.g. Mott et al., 2023) and snow cover modeling (Lehning et al., 1999) used in operational avalanche forecasting (Morin et al., 2020; Pérez-Guillén et al., 2022; Herla et al., 2023). In climate science, snow cover is one of the key variables that strongly affect the global energy balance and the atmospheric circulation, due to its high albedo, high emissivity, and low thermal conductivity (e.g. Flanner et al., 2011). Snow height signals have also been used to determine vegetation growth and plant phenology (e.g. Jonas et al., 2008; Fontana et al., 2008; Vitasse et al., 2017) and to monitor climate change (e.g. Matiu et al., 2021). Finally, the snow cover directly influences tourism, transportation, and recreational activities (e.g. Willibald et al., 2021).

Snow height data are nowadays available, sometimes in almost real-time, from airborne or satellite remote sensing and ground-based automated weather stations (AWS). One of the sensors often mounted at meteorological stations in high alpine regions is an ultrasonic snow height sensor (Ryan et al., 2008). Due to the measurement method, snow height data come with a variety of errors that arise from the harsh mountain conditions the sensor is not originally designed to operate in. In addition, ultrasonic sensors only measure the distance to the underlying object, be it snow or anything else. It is therefore important to validate whether the information coming from the snow height sensor really corresponds to snow or not.

Arguably the most precise way of assessing the quality of snow height measurements is via visual inspection of the data by a human expert (Robinson, 1989), which is however not easily transferable and does not scale well (Fiebrich et al., 2010). A common practice in both manual and automated snow height quality assessment is to distinguish between snow and grass based on static climatological or minimum snow height thresholds. Random errors are typically detected using a maximum snow height threshold or snow height variance (Avanzi et al., 2014).

There are other sensors usually mounted at an AWS, whose temporal structure can provide information on whether the measured snow height relates to snow or not, as well as give some indications on the precision of snow height measurement. Fusion of temporal information from multiple sensors results in high-dimensional multivariate time-series signals, which increases the complexity of the problem. The first attempts to leverage other sensor information include the MeteoIO library developed by Bavay and Egger (2014) and the thresholding method of Tilg et al. (2015). Both algorithms are based on a series of thresholding rules that follow the physical properties of snow. In particular, with the presence of snow, snow surface temperature (TSS) is expected to be $\leq 0°\text{C}$. Ground temperature (TG) is expected to be constantly around $0°\text{C}$, as snow insulates the ground from atmospheric temperature variations (Domine, 2011). Reflected short-wave radiation (RSWR) is expected to be high since snow has a much higher albedo than soil or vegetation. When no snow is present, both TSS and TG typically show diurnal variations, in line with the air temperature (TA). However, it is rather difficult to capture correlations between different features in high dimensional space by defining thresholding rules. Moreover, thresholding approaches are known to be rather cumbersome to modify and do not generally transfer well to other station data.

Machine learning, instead, is an appropriate choice in such cases, and has already shown its power in other tasks concerning weather and climate data (e.g. Vaughan et al., 2022; Luković et al., 2022; Lam et al., 2023). Blandini et al. (2023) addressed the high dimensionality of the data with a random forest (RF) approach to snow height quality assessment, solving both snow height classification and anomaly detection at the same time. RF models (Breiman, 2001) are amongst the most popular choices of machine learning algorithms for tabular data (Grinsztajn et al., 2022). Multivariate time-series signals contain both temporal dependencies between different data points from the same sensor, as well as inter-sensor correlations between measurements from multiple different sensors. Apart from an attempt by Goehry et al. (2023), simple models such as random forests or multilayer perceptron (MLP) neural networks (Rosenblatt, 1958; Hornik et al., 1989; Cybenko, 1989) cannot explicitly account for the temporal nature of the data without engineering complex and artificial features, and are therefore a rather poor design choice. To correctly capture temporal patterns in the data, we instead choose to work with neural network models specifically designed to operate on time-series data, e.g., recurrent neural networks (RNNs) (McCulloch and Pitts, 1943; Kleene, 1951), long short-term memory (LSTM network) (Hochreiter and Schmidhuber, 1997), Temporal Convolutional Networks (TCNs) (Lea et al., 2016), TimesNet (Wu et al., 2023) or Transformers (Vaswani et al., 2017).

We developed CleanSnow, a machine learning model for the automated classification of snow height signals into a snow and a no-snow class. To approach this binary classification problem, we employed a Temporal Convolutional Network that explicitly accounts for the temporal relationships between different points in snow height time series data. To train our TCN, we created a new manually annotated snow height dataset composed of 20 measurement stations with around 20 years of data per station. This dataset also allows us to validate our design choices and evaluate the model in several different scenarios including challenging cases such as snow cover melt or plant growth periods.

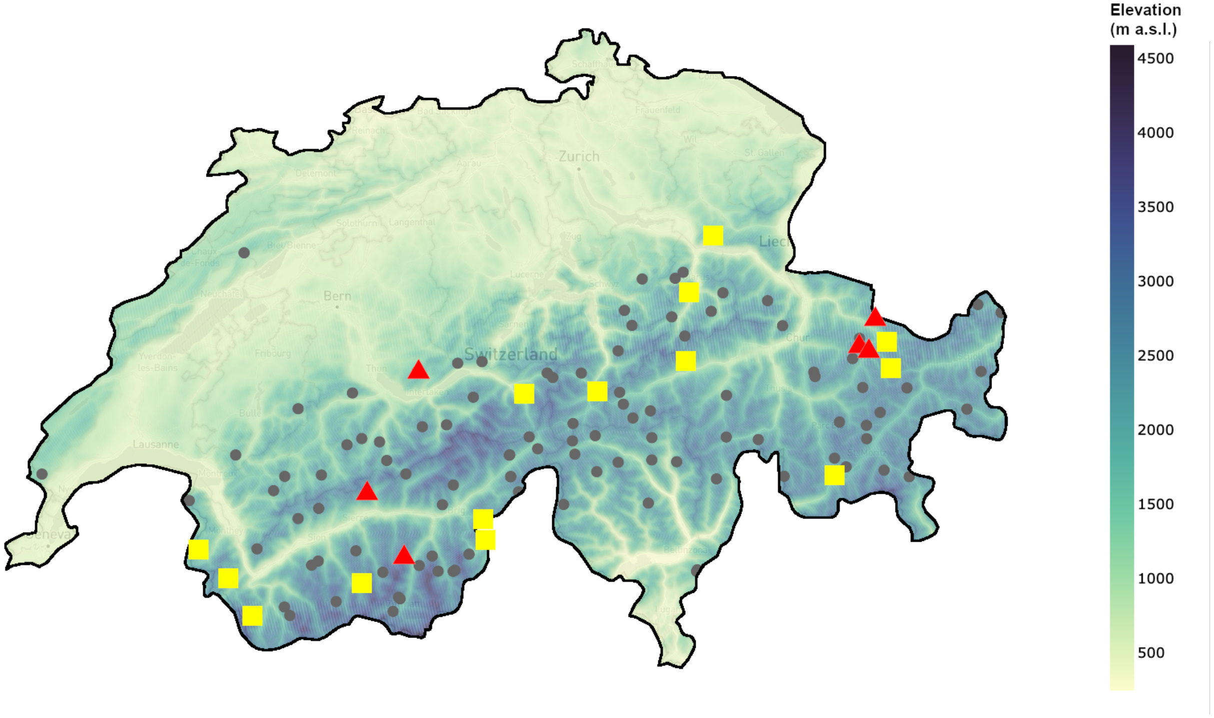


Figure 1. Map of IMIS stations in Switzerland. Stations marked as full gray circles were not part of the new annotated dataset. Yellow squares are the stations that were used for training (14 stations) and red triangles indicate stations used for testing (6 stations). Background colours indicate elevation in m a.s.l.

2 Data

We used snow height data from the Swiss Intercantonal Measurement and Information System (IMIS) (Lehning et al., 1999; Liechti and Schweizer, 2024), a network of 131 AWS (as of May 2024) focused on snow measurements that are distributed throughout the Swiss Alps and Jura (see Figure 1), mostly located above 2000 m a.s.l. The stations acquire data regularly in 30-minute intervals and provide meteorological data in addition to snow height. To analyze snow height (HS), we also leverage measurements such as air temperature (TA), snow surface temperature (TSS), wind speed (WV), relative humidity (RH), and reflected shortwave radiation (RSWR).

2.1 Data preparation

For model development and validation, we prepared a dataset with reliable ground truth information. Manually annotating snow height data is a tedious process, and doing so for the whole IMIS network is intractable. Therefore, we identified a subset of IMIS stations that we then manually annotated.

Annotating historical data is rather difficult, as there is no way of checking whether there really was snow at the station or not. This means that assessing the presence of snow with the help of information from other sensors, such as air temperature (TA), snow surface temperature (TSS), ground temperature (TG) and reflected short-wave solar radiation (RSWR), should be considered a best-effort approach.

2.1.1 Snow/no-snow dataset

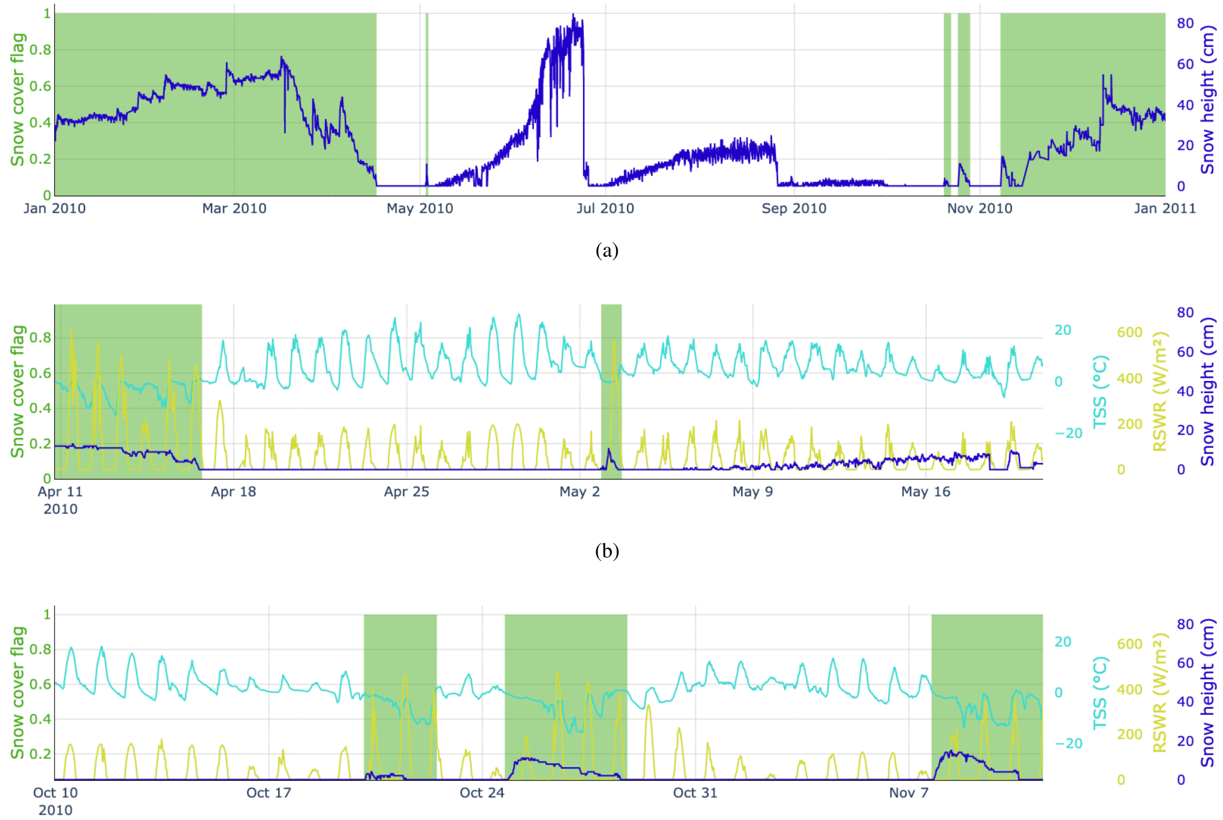
A subset of 20 stations (see Appendix A) which span different locations and elevations and vary in underlying surface (e.g., vegetation, bare ground, glacier, etc.) were selected and manually annotated with binary ground truth information regarding snow height data:

- Class 0 - No Snow - the surface is snow-free (e.g., vegetation, soil, rocks, etc.)
- Class 1 - Snow - the surface is covered by snow

The stations annotated with ground-truth information are depicted in yellow and red in Figure 1. An example of data annotation is shown in Figure 2, with two detailed views that emphasize the differences in behavior of TSS and RSWR in the presence and absence of snow cover. The selected stations mostly contain data between 2000 and 2023, at a 30-minute frequency, with a few exceptions for stations that were built later (BOR2, FLU2, LAG3, RNZ2 and SHE2; see Appendix A).

2.1.2 Evaluation subset

We left part of the annotated data out during model development, which we later used as an independent test set to evaluate the generalization ability (e.g. Section 5.2 of Goodfellow et al. (2016)) of our final approach on stations not seen at training time. We selected 6 stations (SLF2, WFJ2, KLO2, TRU2, STN2, SHE2) that contain challenging scenarios and are therefore suitable



(a)

(b)

(c)

Figure 2. Examples of manually annotated data for the calendar year 2010 at the station SLF2. (a) shows the snow cover flag in green (1 for snow cover, 0 otherwise) and snow height (in blue) for the whole year. (b) focuses on the end of winter season 2009/2010 illustrating the diurnal behavior of TSS and RSWR dependent on whether there is snow or not. (c) is the same as in (b) for the beginning of the winter season 2010/2011.

test cases. In particular, these stations are located at elevations where summer snowfalls occur, the snow season duration is very different, or where grass grows during the summer periods.

3 Methodology

To distinguish whether snow or other ground cover is under the sensor, other sensor measurements can be used. To this end, a combination of seven input variables can be selected, namely HS, TA, TSS, RSWR, VW, RH and solar altitude. We omitted TG, which was used during manual annotation, as it is not available at all IMIS stations and the sensor is also prone to defects. A detailed analysis regarding input variable selection is provided in Section 4.1.3.

Looking at a data point in the context of its temporal neighborhood helps in determining whether there is snow or not at a particular time step. In an operational setting, one would, however, like to be able to make a prediction for each incoming data point in real-time. This means we cannot access data points in the future, and the context for each data point has to be composed of itself and preceding data points (history). To reduce computational demands while still allowing for large enough context, we suggest working with window sizes of between 8 and 192 time steps, where 1 time step corresponds to 30 minutes. The effect of varying time window size on the results is summarized in Section 4.1.4.

To account for the multivariate temporal characteristics of our data, we opted to use Temporal Convolutional Networks, which have proven useful in many applications concerning time-series (Wan et al., 2019; Pelletier et al., 2019; He and Zhao, 2019; Hewage et al., 2020). Later, Section 4.2 provides a comparison of CleanSnow to other popular models, such as Random Forests, MLPs, a variation of an RNN called an LSTM, Transformers, and a recently released model for time-series processing called TimesNet, which yields state-of-the-art results on various standard benchmarks.

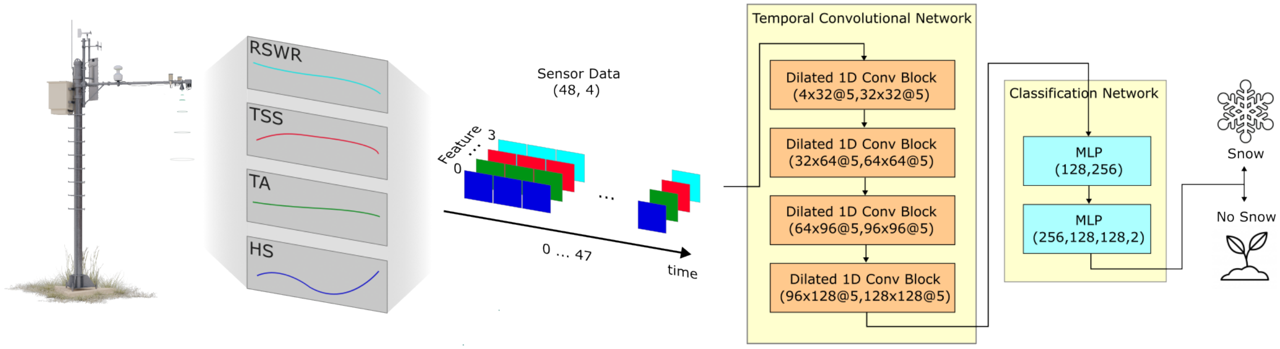


Figure 3. Flowchart of the proposed TCN-based architecture with a drawing of an IMIS station on the left. A time window of 4 input signals of length 48 (1 day) coming from an IMIS station is fed to the TCN, which causally aggregates information from all time steps into a 128-dimensional latent vector. This information is subsequently fed into the classification network, which applies a sequence of MLPs to classify the input signal into two classes - *Snow* or *No Snow*. Each dilated 1D conv block has filters described in the format (*input_features* × *output_features* @ *kernel_size*). The composition of each MLP is described as (*input_features*, *hidden_features_1*, ..., *output_features*).

3.1 Temporal Convolutional Network (TCN)

Based on well-known convolutional neural networks (CNNs) (Fukushima, 1988; Waibel et al., 1989; Weng et al., 1993; Lecun et al., 1998), TCNs are variations that consist of dilated, causal 1D convolutional layers that have the same input and output lengths. Dilation ensures that a specific entry in the output depends on all previous entries in the input, while causal convolution means that the i-th element of the output sequence may only depend on input elements that come before it (elements with indices $\{0, \ldots, i\}$).

As shown by Lea et al. (2016), with dilations and causal convolutions, TCNs can recover the behavior of RNNs while not suffering from typical drawbacks of RNNs, such as the vanishing gradient problem (Pascanu et al., 2013), and are therefore easier to train. The use of convolutions instead of a recurrent mechanism also potentially leads to further performance improvements due to the possibility of parallelization of the convolution operation.

We chose a 4-layer TCN architecture as shown in Figure 3, which has 4-dimensional time series with 48 time steps as the input. The number of layers and filter sizes were selected so that the output representation of the last point in the input time series is an aggregation of all previous time steps. In other words, the TCN produces an output representation of the last point in the input time series by aggregating information from the whole history available at the input. This representation is fed to an MLP classifier, which first produces a series representation and then uses this representation to produce output class probabilities.

3.2 Training

Snow height classification is a binary problem. Binary classification problems are typically optimized using the cross-entropy loss function (Good, 1952), which did not yield good results in our case. Many of the stations included in the dataset are located in places where snow is present for much of the year, resulting in considerable class imbalance in our data. Moreover, we would like our model to perform well on the challenging edge cases. Therefore, we chose to drive the optimization by the so-called focal loss (Lin et al., 2017), which allows the model to preferentially focus and train on examples that it has difficulty classifying correctly while down-weighting the simple cases throughout the training process.

The focal cross-entropy loss is defined as

$$\mathrm{FL} = -\sum_{i=0}^{N-1} \alpha_i (i - p_i)^{\gamma} \log_b(p_i), \tag{1}$$

where α_i is the so-called balancing factor for class i, further contributing to class balancing, γ is the focus parameter which controls the down-weighting of the easy examples, p_i is the probability of the sample belonging to the i-th class, $N = 2$ is the number of classes in the classification problem, and b is the logarithm base; typically $b = 10$.

We run training for a maximum of 300 epochs, feeding the model with a batch of 128 samples in each iteration. We allow for the possibility of early stopping if the validation loss has not improved for more than 50 epochs. The optimization process was governed by the AdamW (Loshchilov and Hutter, 2019) optimizer with an initial learning rate of 10^{-3}. The learning rate was subject to step decay with factor 0.1, three times, after 50, 100 and 150 epochs.

3.3 Dataset

In all experiments, we used the snow/no-snow dataset described in Section 2.1.1. This dataset was split into train and evaluation subsets (see Section 2.1.2). For model training, we further (randomly) divided the training subset into two parts using a 90/10 split: 90% used for training CleanSnow and 10% for validation. The validation set was used to monitor CleanSnow's performance during training and hyperparameter tuning, and enabled early stopping to prevent overfitting on the training data (Ying, 2019).

The whole training dataset contained approximately 7 million data samples, which would be rather impractical for experimentation, as it would yield extremely long training times and high compute demands, which might not always be available. To make our experiments more tractable, we selected roughly 30% of the data from every station in the training set using filtering by year (Table B1 shows which years were used from each station).

3.4 Hyperparameter tuning

We performed 5-fold cross validation with random training/validation splits in order to perform hyperparameter tuning using grid search for the following model architecture variables: *dropout* and *output_activation* for the TCN, *batch_norm* and *activation_function* for the MLP, *gamma* and *alpha* parameters of the focal loss as well as optimizer learning rate.

For all remaining experiments, we have fixed a random seed for the training/validation split in order to ensure easy and full reproducibility of our results. Random splitting inherently takes care of having samples from different stations and different time periods throughout the whole training subset.

We opt for a *batch size* of 128 samples as it is sufficiently large while still fitting into the GPU memory we had available. Due to limited computing resources, we do not optimize the remaining hyperparameters and we instead select them based on similar architectures available in other works and our experience with designing machine learning models.

4 Results

In this section, we summarize experiments performed to evaluate CleanSnow. With a series of ablation studies, we clarify various design choices and then compare our TCN, the model of choice, to other available options. We continue with a thorough evaluation of the TCN in different periods of the year, pointing out its strengths and weaknesses. Experiments are concluded with a case study that demonstrates the use of CleanSnow in vegetation science.

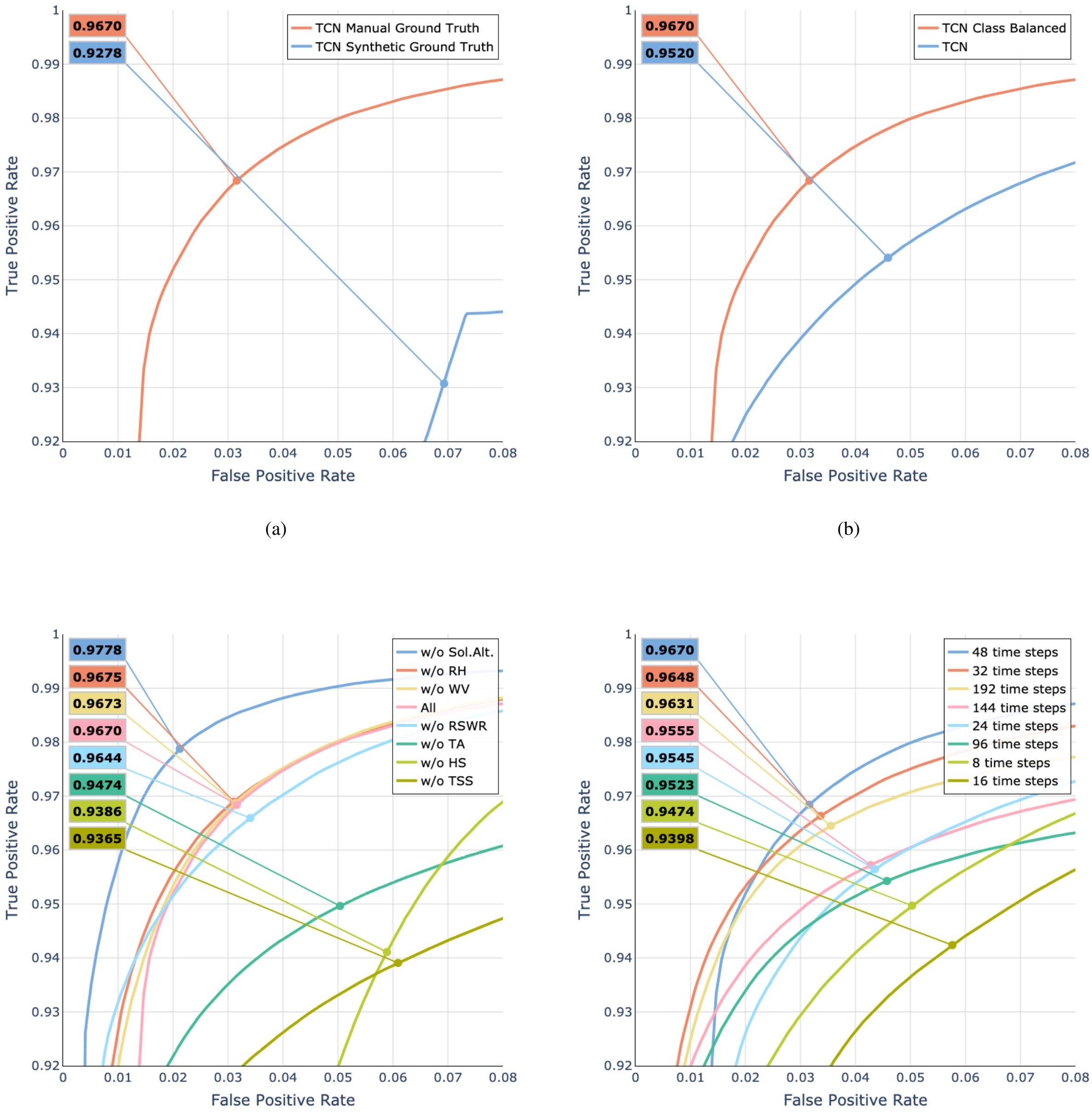
4.1 Experiments with CleanSnow setup

In the following sections, different experiments with the CleanSnow configuration and model comparisons are shown to explain our design choices and their contribution to obtaining the best results. All experiments were performed using a TCN with seven input features, namely HS, TSS, TA, RSWR, RH, WV and solar altitude (which encodes information about the date and time of the day).

Models were compared using the Receiver Operating Characteristic (ROC) curve (Egan, 1975), which is a plot showing the performance in terms of the true positive rate (TPR) and the false positive rate (FPR), and the F1-score.

4.1.1 Synthetic ground-truth experiments

To demonstrate the need for annotated data, we trained a model using synthetic ground truth based on empirical rules developed according to human expert knowledge (see Appendix C). We compared the model trained with the synthetic ground truth information to the model trained with the manually annotated data. The results in Figure 4(a) demonstrate the inability of



(a)

(b)

(c)

(d)

Figure 4. ROC curves for various ablation studies. Every plot additionally shows the macro-F1 score for the threshold where TPR = FPR (the point on each curve). (a) Importance of manually annotated ground truth data. (b) Effect of class balancing. (c) Importance of input features. (d) Influence of sequence length on model performance.

thresholding rules to generate reliable ground-truth information that could be leveraged for training. This resulted in the *TCN Synthetic Ground Truth* model not learning the correct relationships between different input variables, yielding an F1-score of 93% and therefore having a lower performance than *TCN Manual Ground Truth*, which was trained with our manually annotated dataset and achieved a F1-score of 97%.

4.1.2 Class balancing

Our training dataset included roughly twice as many snow-covered samples as snow-free samples. We applied class balancing by adjusting the class weights of the focal cross entropy loss and observed how that affected the performance of CleanSnow. We have assigned a weight of 1.0 to the class representing snow and a weight of 0.5 to the class representing bare ground, as there are approximately twice as many data samples from the snow-covered period. Figure 4(b) shows that class balancing improved the performance from an F1-score of 95.2% to 96.7% and was therefore a valid design choice in our pipeline.

4.1.3 Feature importance

We performed an ablation study training CleanSnow with a leave-one-out strategy for the input features to validate their importance for the model decision-making (Figure 4(c)).

The HS, TSS, TA and RSWR signals were found to be important (i.e. their removal resulted in a reduction in model performance with a decrease in the F1-score of up to 4%), in line with what was discussed above for manual data annotation. On the other hand, removing WV and RH from the input features only marginally improved model performance, suggesting that they have no positive effect. Hence neither feature provided any additional information useful for classification. However, for other tasks such as, e.g., snow height anomaly detection, WV might very well be an important signal carrying information about snow transport by wind and related phenomena. Interestingly, removing solar altitude, which encodes information about date and time, improved the performance of the model (increasing the F1-score by 1.5%). We attribute this to the fact that solar altitude information potentially makes the model decide based on the date and time of the year, which is undesirable. As much as date and time information are generally valid indicators of the season and therefore have a strong influence on the presence of snow, they might hamper decision-making, especially at the beginning and end of the snow season and in the case of summer snowfalls, whose occurrence varies from year to year.

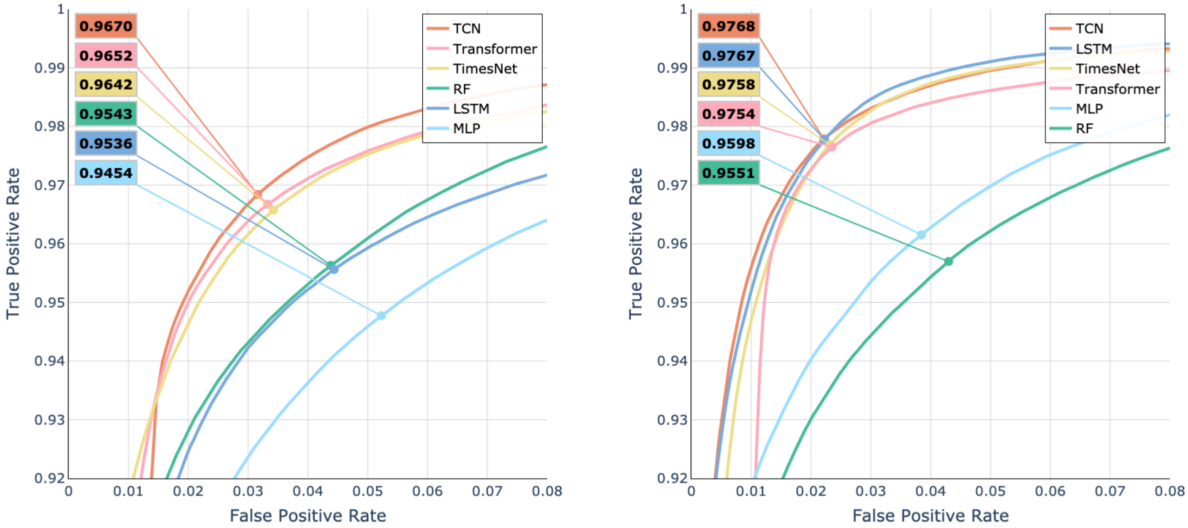
Therefore, we chose our final model to have four input features, namely HS, TSS, TA and RSWR.

4.1.4 Sequence length selection

One of the key parameters to choose is the length of the history the model can use to predict the current time step. Figure 4(d) shows the relationship between history length and model performance. The best results were obtained with a history length of 48 time steps (24 hours) achieving an F1-score of 97%; very similar results were obtained with a history of length 32 (18 hours) with an F1-score of 96%. A history length shorter than 24 time steps deteriorated the performance. Likewise for history lengths larger than 96 time steps. Accordingly, we selected the history length to be 48 time steps as a compromise between sufficient but not too much context for the model.

4.2 Model selection

To choose the right architecture for the task at hand, we experimented with several state-of-the-art machine learning models for single time-step and time-series processing, compared their performance, and finally selected the one that performed the best overall. Our model of choice was the TCN. A short description of the other models we evaluated is provided in Appendix D.



(a)

(b)

Figure 5. Model comparison shown as ROC curves for two different versions of the six models (LSTM, TCN, TimesNet, Transformer, MLP and RF): Model performance with (a) all seven input features - HS, TSS, TA, RSWR, RH, WV and solar altitude, and (b) with the four relevant input features - HS, TSS, TA and RSWR. Every plot additionally shows the macro F1-score for the threshold where TPR = FPR (the point on the curve).

To have a balanced model that does not favor one of the classes, we selected the decision threshold as the point where $TPR = FPR$. We evaluated each model for two scenarios: one with all seven input features and one with only the four relevant features.

Figure 5 shows the overall best performance of the TCN with an F1-score of 97.8%. Removing RH, WV and solar altitude, which were identified as irrelevant features resulted in a significant improvement of the LSTM model, equaling the performance of the TCN having an F1-score of 97.7%. Nevertheless, we opted for the TCN as it was on par with the LSTM, and the results in Figure 5(a) suggest that the TCN is more resilient to unimportant features in the input. In addition, the TCN is known to be easier to train compared to LSTMs. Interestingly, for RF the performance is less dependent on the selection of input features, suggesting its ability to deal with uninformative inputs.

4.3 Performance analysis per station

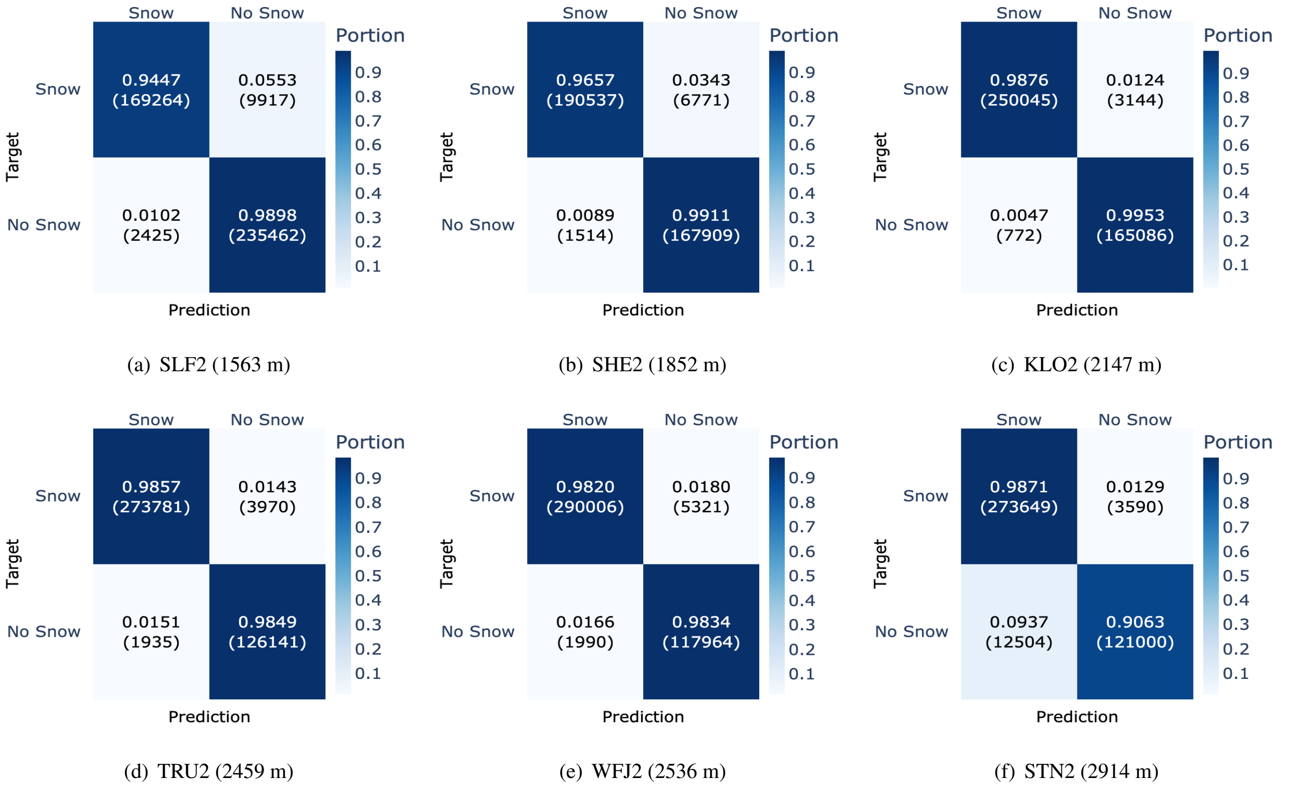
To better understand the generalization capabilities of the model, we evaluated its performance for each test station separately. The results in terms of confusion matrices are presented in Figure 6 and suggest good generalization capability of the model for most stations, except SLF2 and STN2. The stations SLF2 (1563 m) and STN2 (2914 m) were considerably outside the elevation range that was available during training. Moreover, these two stations are rather special cases compared to most of the other stations and can be considered out-of-distribution samples. The station SLF2 is located on a meadow in the village of Davos, which seems to have a positive effect on the classification into the class *no snow*, as it was the only station with an F1-score for class *no snow* higher than for class *snow*. The station STN2, instead, stands on a glacier, which results in very different ground properties compared to any other station in the dataset. This is reflected by a lower F1-score for the class *no snow*, especially as STN2 reached an F1-score of only 94.5% (which is 2% less than any other station in the test set). In addition, from Figures 6 and 7 one can further conclude that the model generally classifies the presence of snow slightly better than the absence of snow.

It is also important to understand whether CleanSnow generalizes to stations at different locations with different elevations. The Results presented in Figure 7 suggest that the model performance was very stable for stations at elevations between abouty 2100 and 2700 m a.s.l., while it decreased for stations located either below or above this range. This corresponds to the fact that 80% of stations in our training set were in this range and only two stations were below 2000 m and one station was at 2800 m.

The seemingly good performance of the model should however be analyzed in detail. There are periods for which it is rather easy to correctly classify snow as *snow* and snow-free ground as *no snow*, and other times of the year when the problem becomes much harder. This is discussed in detail later in Section 4.4.

4.4 Performance for different times of the year

Classification of snow height measurements into snow and snow-free ground can be both a simple and rather challenging task depending on the location and time of the year. We provide a per-month performance analysis in Figure 8, which shows that the model mostly had trouble predicting snow-free ground in winter months. This is because very little training data for that class were available during December, January, February and March. Furthermore, we had no snow-free samples in the test set for February and March. In summer instead, the results suggest CleanSnow was able to detect most of the summer snowfalls (with approximately 20% performance drop compared to full winter) while retaining very good performance on predicting snow-free ground. At the end of winter, in May and June, the model performance was also very good, suggesting that CleanSnow can be used to accurately predict the snow disappearance date (as a longer snow-free period after a long period with constant snow cover).



(a) SLF2 (1563 m)

(b) SHE2 (1852 m)

(c) KLO2 (2147 m)

(d) TRU2 (2459 m)

(e) WFJ2 (2536 m)

(f) STN2 (2914 m)

Figure 6. Performance evaluation of each test station separately, shown in terms of confusion matrices ordered by elevation: SFL2, SHE2, KLO2, TRU2, WFJ2, STN2. Each confusion matrix has targets as rows and predictions as columns.

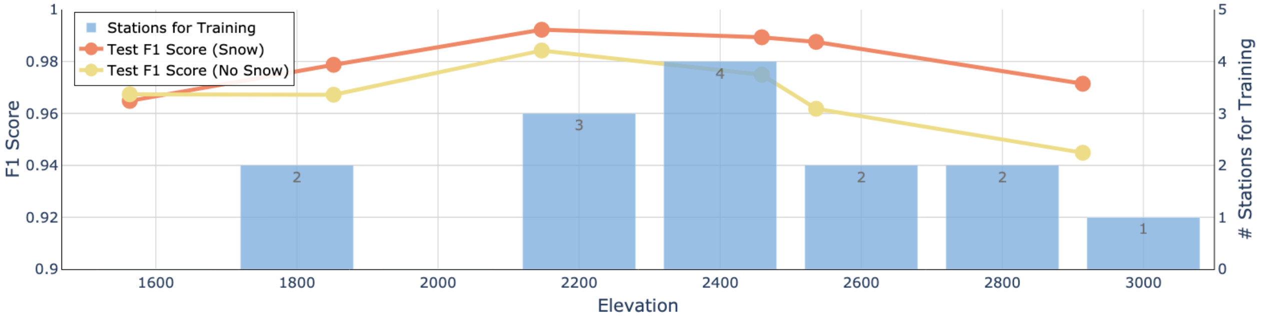


Figure 7. Model performance for the six stations in the test subset as a function of elevation. The F1-score is shown separately for the classification of snow (red line) and no snow (green line). The blue columns indicate the elevation distribution in the training subset (14 stations).

In addition, we analyzed the model performance for each season. To this end, we split the test dataset into four different seasonal clusters:

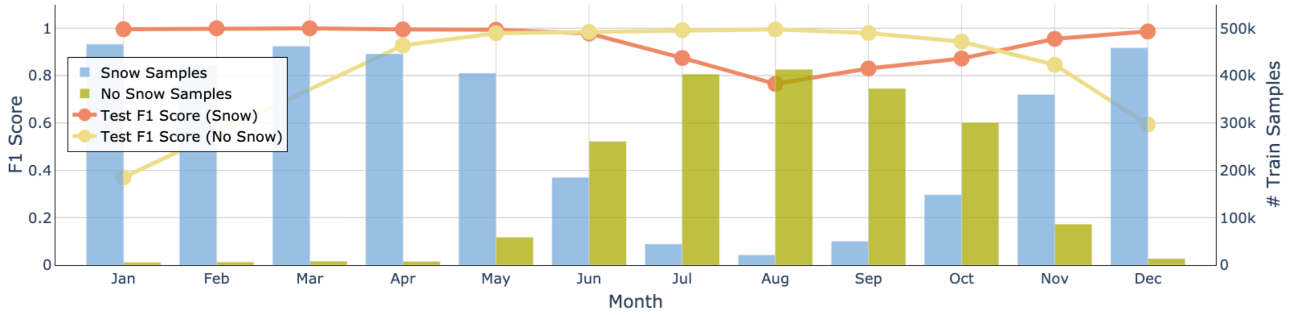


Figure 8. Performance of the model for each month of the year. The F1 score is shown separately for the classification of snow (red line) and no snow (yellow line). The blue columns indicate the distribution of snow samples, while the yellow columns indicate the distribution of the no-snow samples.

- **Winter season** was defined as the period with mostly continuous snow cover (December, January, February, March, and April)
- **Summer season** was the part of the year typically without snow (July, August and September)
- **End of winter season** defined the snowmelt period resulting in snow-free ground (May, June and July)
- **Start of winter season** included the months when it starts snowing more often and at some point a continuous snow cover forms on the ground (September, October and November)

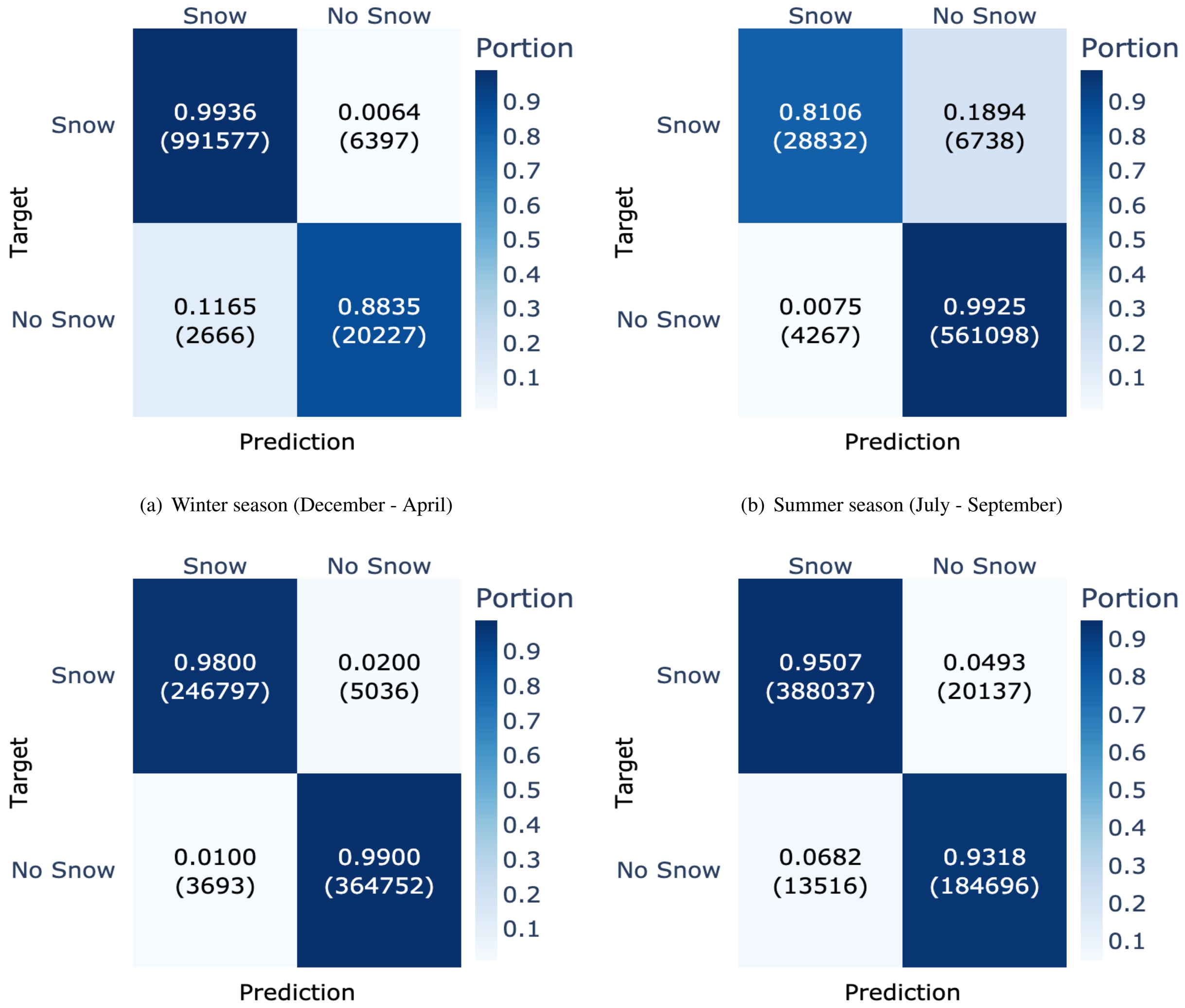
In the following sections we describe the model performance for each of the four seasonal clusters in detail and point out some season-specific challenges.

4.4.1 Winter season

For snow classification, the middle of winter is presumably the easiest time of the year. Besides some low-elevation stations and some exceptional seasons with a very late onset of winter or very early snowmelt, the task should be rather trivial, as the snow cover is continuous in time. Figure 9(a) demonstrates that the model confidently classified snow (TPR = 99.36%) in contrast to snow-free ground (TPR $= 88.35\%$).

4.4.2 Summer season

In contrast to full winter, the classification of snow in the summer is more challenging. Besides snow-free ground, there were many stations where vegetation grew (approximately 20% of the data in the test set). This resulted in non-zero snow height sensor measurements, which do not correspond to snow. Exceptions were stations at high elevations (e.g., on a glacier) and winters when the snow did not melt until the beginning of summer.



(a) Winter season (December - April)

(b) Summer season (July - September)

(c) End of winter season (May - July)

(d) Start of winter season (October - December)

Figure 9. Confusion matrices for each of the four seasonal clusters. Each confusion matrix has targets as rows and predictions as columns.

The snow height signal for snow-free ground typically oscillates with high frequency and either stays around zero or grows in the presence of vegetation under the sensor. The surface temperature and air temperature will most of the time oscillate high above 0°C, showing a diurnal cycle. During overcast periods or in the presence of precipitation, TA and TSS will show the same value. Due to the lower albedo of snow-free ground, smaller amounts of RSWR are measured. Based on the above assumptions, summer snowfalls can be detected when TA equals TSS, which is followed by larger values of RSWR with a simultaneous decrease in TSS. If there is vegetation growing under the station, the HS signal counter-intuitively decreases as the plants get pressed down by the snow. In the case of snow-free ground under the sensor, the HS signal will increase as expected during a snowfall.

Figure 9(b) demonstrates that the model accurately detected snow-free ground with 99% accuracy. The effect of summer vegetation is shown in Figure 10(a). On the other hand, detecting a snowfall in the summer proved to be difficult, and even more so when vegetation was present. In this very difficult setting CleanSnow achieved a performance of 81%. A partial detection of a summer snowfall is shown in Figure 10(c). CleanSnow succeeded in detecting the main event but failed to correctly classify a few hours both at the start and the end of the summer snowfall.

4.4.3 Start and end of winter season

The transition periods between winter and summer and vice versa are key periods for the detection of the first snow and its disappearance, which are both dates of interest in climate science. These two seasonal clusters contain both data with rather continuous snow cover and with bare ground or vegetation growth. Such data are therefore a perfect test case for CleanSnow.

In our experiments, the end-of-winter season was the easier case to classify, achieving a very competitive performance of 98% for snow and 99% for snow-free ground (Figure 9(c) and Figure 10). We attribute this high accuracy to the fact that the transition from snow-covered to snow-free ground was often rather smooth, and once the snowpack had melted, there were not many periods with snow persisting on the ground. The beginning of summer was typically represented by high air temperatures, which caused TSS to oscillate with the daily cycle indicating snow-free ground; simultaneously RSWR noticeably decreased once the snow had completely melted.

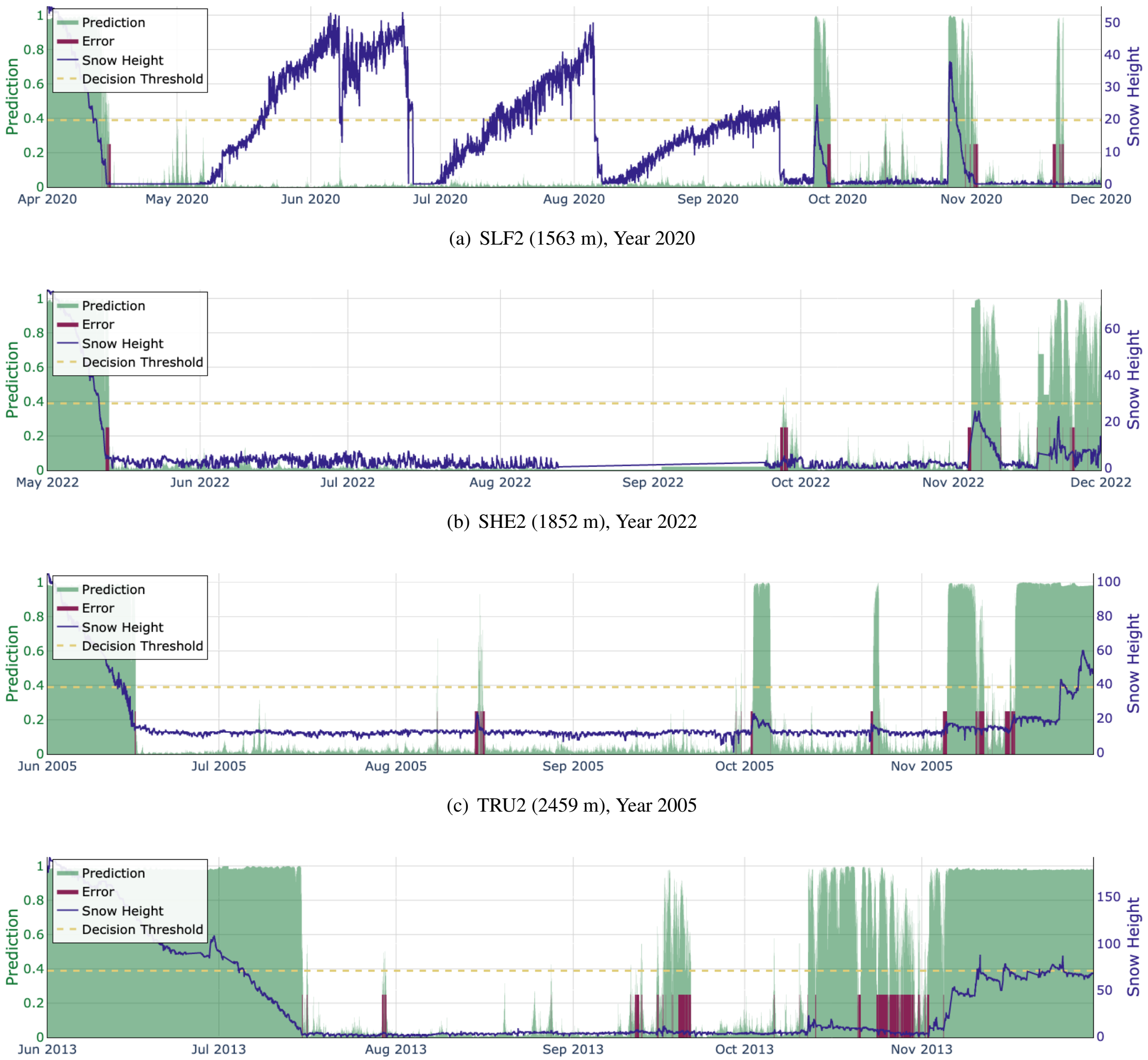
On the other hand, classification during the start-of-winter season was more challenging: the model achieved an accuracy of 95% for snow and 93% for snow-free ground (Figure 9(d) and Figure 10). There were multiple snowfalls at the beginning of the season after which the snow melted again completely. In addition, in late autumn and the beginning of winter, temperatures occasionally dropped and the ground froze overnight. This resulted in TSS being constantly less than or equal to $0°$C even without snow, which might force the model to focus more on RSWR and HS during decision-making, potentially decreasing its decision power.

4.5 Comparison to manual observations

A perfect test case are stations with concurrent manual observations, i.e., measurements manually performed by human observers. Such measurements were available for the two stations WFJ2 and SLF2 located in the region of Davos.

Since the manual measurements were done only once per day, we resampled our predictions from 30-minute intervals into 24-hour intervals. We averaged probability scores over the 24 hours (48 automatic measurements) to obtain the per-day probability score.

The performance comparison on annotated automatic measurements versus manual observations in Figure 11 confirms that we had produced high-quality annotations for the historical data. Some days with snow were erroneously annotated as snow-free ground. This can be related both to short snowfalls which disappear in daily aggregation and also to the fact that manual



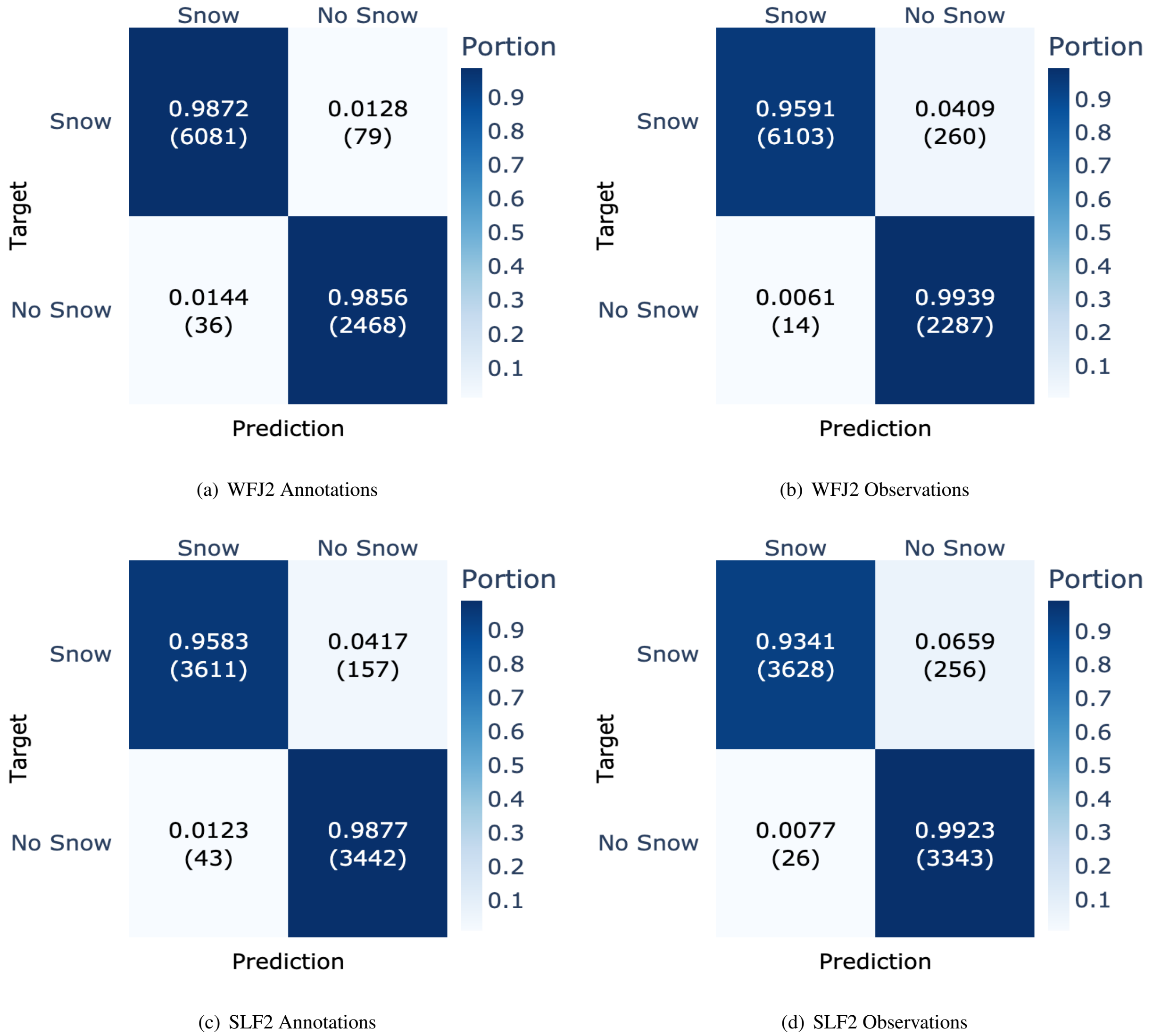
(a) SLF2 (1563 m), Year 2020

(b) SHE2 (1852 m), Year 2022

(c) TRU2 (2459 m), Year 2005

(d) STN2 (2914 m), Year 2013

Figure 10. Examples of classification results by CleanSnow. The snow height signal is depicted in blue. The model predictions in terms of probability (0 - 1) are shown in green. The dashed horizontal line denotes the decision threshold selected to balance the model performance on predictions for both classes. The red-shaded areas mark regions with classification errors (i.e. samples being assigned to the wrong class). (a) shows a correct classification of summer vegetation growth (the non-zero blue curve is classified with probability lower than the decision threshold, therefore being assigned to class *no-snow*). (b) is an example of early October snowfall that has been classified partially correctly. (c) demonstrates the model's capability to detect summer snowfalls as well as scattered snowfalls at the beginning of winter. (d) is evidence that the model does not always perform well, here making mistakes at the beginning of the next winter season.



(a) WFJ2 Annotations

(b) WFJ2 Observations

(c) SLF2 Annotations

(d) SLF2 Observations

Figure 11. Confusion matrices for comparison of model performance evaluated against our annotations (left) and against human observer measurements (right). Results for station WFJ2 are in (a) and (b), followed by results for SLF2 in (c) and (d).

observations were performed around 08:00 CET in the morning, while our data were daily averaged values. Such misalignment might produce additional disagreements between manual observations and our annotations.

The results also show that CleanSnow achieved a very good performance when evaluated against daily manual observations. The differences in performance between the two ground-truth sources (approximately 2% in TPR and 1.5% in TNR) were attributed to the inconsistencies between the manual annotations of automatic measurements and manual observations.

4.6 Comparison to other approaches

To further demonstrate the added value of our machine learning approach, we compared it to other state-of-the-art methods such as filtering used in the physics-based snow cover model SNOWPACK (Lehning et al., 1999). In particular, we considered the snow water equivalent (SWE) provided by SNOWPACK since the HS signal is filtered to calculate SWE. Therefore, SWE should be a good indicator of whether the HS signal relates to snow or not. If the HS signal does not represent snow, one would expect SWE to be 0. In addition, we also compared CleanSnow to thresholding-based filters implemented in the MeteoIO library, which were mainly designed to filter vegetation growth measurements in summer.

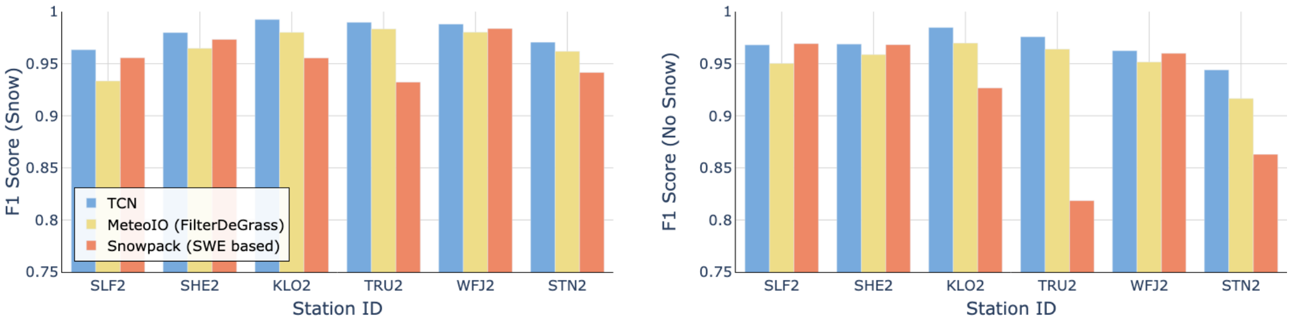


Figure 12. Comparison to other approaches shown as performance (F1-score) per station for CleanSnow (blue), the filter based on the SWE from SNOWPACK (red) and the thresholding filter from MeteoIO (yellow).

Figure 12 shows the comparison of the snow height classification by our TCN model to classification based on SWE calculated by SNOWPACK and the MeteoIO filter. The results suggest that the machine learning approach is superior in most cases. This might be attributed to the fact that both SNOWPACK and MeteoIO use thresholding-based rules based on TSS and TG to filter HS similarly to the approach described by Tilg et al. (2015). The optimal threshold values vary across different stations, which requires per-station calibration of the thresholds. Moreover, TG-based filtering is problematic since, as already mentioned, the TG sensor is prone to failures and the signal is therefore often missing at some stations.

4.7 Case study: Vegetation growth

Besides obvious applications in snow science, a reliable separation of snowfall from plant growth also has benefits for biological research. Removing HS measurements classified as snow allows for the extraction of a clean vegetation signal and the pinning down of reoccurring events in the life cycle of alpine vegetation – referred to as vegetation phenology. Since snow and plant heights have been recorded for a very long time, it is possible to relate the timing of green-up (i.e. the start of vegetation growth) or other phenological phases to snow climate parameters, and study phenological shifts over time – an excellent indicator of climate change (e.g. Inouye, 2022). We extracted 25 years of vegetation growth data from HS measurement data at TUJ2 (Culmatsch, 2262 m a.s.l.), an IMIS station characterized by tall plant growth. Within the 20 years of data, the algorithm flagged all snow days during the vegetation period which were then removed. Snow disappearance and snowmelt dates were defined as the first, respectively the last, day of the continuous winter snow cover. We fitted a logistic growth curve (Kong et al., 2022) to the clean plant growth measurements and defined the start of growth by a 10% threshold of maximum plant height (Figure 13). Vegetation green-up was directly linked to the timing of snowmelt, consistent with other studies (Jerome et al., 2021; Jonas et al., 2008), while late snowfall events shifted the start of growth towards later calendar days. Linear regression analysis revealed an earlier occurrence of green-up over the study period coinciding with an increase in spring temperatures measured at the station. Despite insignificant changes in snowmelt timing, the shorter lag between snowmelt and initiation of plant growth suggests a warming-driven advancement in phenology at the study site. This case study highlights the importance of long-term monitoring and automated machine learning approaches in understanding climate-induced phenological shifts, with implications for ecosystem dynamics in remote alpine regions.

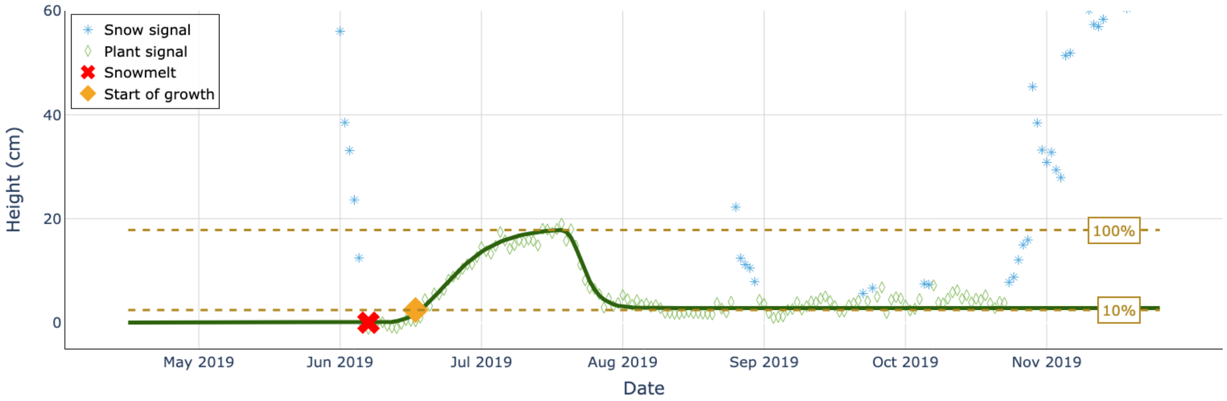


Figure 13. An example of a logistic growth curve (in dark green) fitted to height measurements data from TUJ2, in the vegetation season of the year 2019. Snow height data corresponding to snow are shown with blue stars, while plant signal is shown with green diamonds. The red cross marks the snowmelt date, while the orange diamond marks the start of plant growth.

5 Discussion

We proposed a deep learning-based approach to snow height signal classification, which is a crucial step in automating the snow height signal quality-checking process. In addition to selecting an appropriate model, we provided some good practices to develop machine learning models for automated snow height classification.

5.1 Best practices for snow height classification using machine learning

In our analysis, we aimed to establish good practices for further development of machine learning methods for snow height classification and quality assessment. We showed that learning from synthetic ground-truth data generated using thresholding rules proposed in the past did not work well, as the predefined thresholds did not generalize to all stations without modifications. This emphasizes the need for well-annotated data for training. Next, we pointed out the importance of addressing the class imbalance problem to achieve the best possible performance. Furthermore, we demonstrated the superiority of sequence-based models (TCN, LSTM, TimesNet and Transformer) over single time-step-based models (RF and MLP), which confirms the need for temporal context to achieve a high classification performance. We acknowledge the existence of techniques that allow one to feed RF and MLP models with sequences of data, e.g., lagged features (i.e., adding data from previous time steps as extra input features). Nevertheless, we argue that such techniques do not treat sequential data as a causal sequence, which is conceptually non-ideal and might potentially lead to the resulting model becoming less explainable in how it treats temporal information. Another important aspect to consider is the sequence length. We performed an analysis of the performance for the length of the time window (i.e., the size of the temporal context), which revealed that the ideal length was around 48 time steps, as shorter and longer time windows resulted in a deterioration of the model performance. Subsequently, we showed that it was important to evaluate the model performance during the critical times of the year (the start and the end of the winter season) to reveal their true performance.

5.2 Deep learning models for snow height classification

We studied the suitability of state-of-the-art deep learning models for the snow height classification task. Several cutting-edge deep learning architectures have been evaluated against each other, resulting in the superiority of a TCN over the other compared methods. CleanSnow reached an accuracy of 97.7% on the independent test set when we used a decision threshold that balanced the model performance on predictions for both classes - *snow* and *no-snow*. Hence, the results indicate that the approach generalizes well to unseen stations that are within the distribution of the training set. A detailed performance evaluation for each station in the test set showed that the model performed very well except for the data of the stations SLF2 and STN2, which are two particular cases that were not well represented in the training data. The station SLF2 is located low in a valley and STN2 is on a glacier. In addition to being out-of-distribution, such special environments, compared to those of most other stations in the dataset, might cause slightly different behavior of the auxiliary variables used during HS analysis and result in a performance decrease.

5.3 Generalization

The generalization ability of CleanSnow to elevations that are within the range included in the training set is good. These elevations represent the Alps, which is the region of interest for us. Generalization to out-of-distribution samples (stations located at elevations that are not well represented in the training data) is rather poor. Out-of-distribution generalization, however, remains an open problem in the machine learning community. One possibility for improving out-of-distribution generalization is to explicitly express some known behavior (e.g. physical constraints, etc.) in a neural network. Such models are known as Physics Informed Neural Networks (PINN) (Raissi et al., 2019) and can be implemented either by adding a regularization term to the loss function or by incorporating the constraints directly into the model architecture. In both cases, such constraints help the model to correctly extrapolate to situations that were not represented in the training data.

5.4 Limitations

One of the known limitations of CleanSnow is the fact that it operates on raw data, meaning the inputs may contain both anomalies (e.g. spikes) and missing values. Even though CleanSnow seems to be resilient to anomalies, it would be good practice to perform anomaly detection and filtering before running the proposed snow height classification models. We argue that filtering obvious spikes in the snow height signal is a rather trivial procedure and can be solved by employing statistical methods such as Hampel filtering (Pearson, 1999) or an exponential moving average filter (Kendall and Stuart, 1966). However. other more subtle variations are very challenging to detect by both the human eye and automated methods.

CleanSnow can only be applied in cases where the full history needed to make a prediction is available. At the moment, in the case of missing samples in the 48-time step context, the samples were discarded without being run through the model. Dealing with missing data is far more complicated than filtering anomalies. A simple solution for periods of up to several time steps would be linear interpolation. However, as the size of the interpolated interval increases, this fails to produce an accurate reconstruction of the missing data. To impute larger periods of missing data, methods that take into consideration both spatial and temporal context should be employed. This is, however, out of the scope of this work, and we therefore leave it as a possible future research direction.

6 Conclusions

Automated snow height measurements are key input data for many modeling approaches in climate sciences, snow hydrology, and avalanche forecasting. Erroneous snow height measurement deteriorate the performance of these models. We demonstrated how to mitigate the aforementioned issues by the use of deep-learning methods for automated snow height classification. Our contributions can be summarized as three-fold. First, we created a novel machine learning approach to snow height signal classification that operates directly on time-series data. Second, we provided an in-depth comparison of several machine learning models applied to snow height classification. Third, we introduced a new benchmark dataset with annotated snow height data, which sets a baseline and can be used for further research in the field. The proposed approach achieved a high accuracy of 97.7% and generalized well to previously unseen stations. CleanSnow can be implemented as a component of an arbitrary snow height quality assessment pipeline without the need for any special hardware.

Code availability. The exact version of the software used to produce the results in this manuscript is available at https://doi.org/10.5281/zenodo.14587841, while current and future versions of it can be found at https://gitlabext.wsl.ch/jan.svoboda/snow-height-classification.

Data availability. The manually annotated dataset that we used to both train and evaluate CleanSnow is publicly available for research under CC BY-NC[1] license at https://doi.org/10.5281/zenodo.13324736

Appendix A: List of stations in the snow/no-snow dataset

This section provides the list of IMIS stations used in our snow/no-snow dataset (see Section 2.1.1) together with their metadata. Table A1 shows the stations ordered by increasing elevation. The column *Subset* indicates whether a station was used for training or testing.

Station ID	Latitude [°N]	Longitude [°E]	Elevation [m]	Available since	Subset
SLF2	46.8127	9.8482	1563	November 1997	**test**
AMD2	47.1708	9.1468	1610	October 1997	train
GLA2	46.9966	9.0375	1632	November 2000	train
SHE2	46.7488	7.8124	1852	October 2001	**test**
ILI2	46.1913	6.8277	2022	March 2000	train
GUT2	46.6793	8.2896	2115	November 1999	train
KLO2	46.9091	9.8738	2147	November 1996	**test**
TUM2	46.7810	9.0214	2191	October 2002	train
FNH2	46.1007	6.9641	2252	September 1997	train
KLO3	46.8412	9.9316	2299	November 1996	train
LAG3	46.4245	9.6977	2300	November 2009	train
FLU2	46.7527	9.9464	2394	October 2003	train
RNZ2	46.6855	8.6267	2400	December 2008	train
TRU2	46.3709	7.5855	2459	November 1996	**test**
BOR2	46.2905	8.1093	2517	September 2001	train
WFJ2	46.8296	9.8092	2536	January 1996	**test**
ARO3	46.0874	7.5620	2602	September 1996	train
SPN2	46.2294	8.1176	2620	November 1996	train
FOU2	45.9717	7.0672	2800	October 1999	train
STN2	46.1678	7.7505	2914	October 1998	**test**

Table A1. List of stations that are part of the snow/no-snow dataset, together with their auxiliary information, ordered by elevation. The column Subset denotes whether station belongs to the train or test set.

[1] https://creativecommons.org/licenses/by-nc/4.0/

Appendix B: Subsampling of the training data

To run experiments in a reasonable time and make sure they were computationally tractable, we sub-sampled the training dataset to reduce the amount of training samples. In Table B1 we list which years were selected for each station for the training set.

Station ID	Selected years
AMD2	1998, 2001, 2004, 2007, 2010, 2013, 2016, 2019, 2022
GLA2	2001, 2004, 2007, 2010, 2013, 2016, 2019, 2022
ILI2	2002, 2005, 2008, 2011, 2014, 2017, 2020, 2023
GUT2	2000, 2003, 2006, 2009, 2012, 2015, 2018, 2021
TUM2	2004, 2007, 2010, 2013, 2016, 2019, 2022
FNH2	2000, 2003, 2006, 2009, 2012, 2015, 2018, 2021
KLO3	1999, 2002, 2005, 2008, 2011, 2014, 2017, 2020, 2023
LAG3	2011, 2014, 2017, 2020, 2023
FLU2	2005, 2008, 2011, 2014, 2017, 2020, 2023
RNZ2	2010, 2013, 2016, 2019, 2022
BOR2	2002, 2005, 2008, 2011, 2014, 2017, 2020, 2023
ARO3	1998, 2000, 2003, 2006, 2009, 2012, 2015, 2018, 2021
SPN2	1999, 2002, 2005, 2008, 2011, 2014, 2017, 2020, 2023
FOU2	2001, 2004, 2007, 2010, 2013, 2016, 2019, 2022

Table B1. List of years for each station that were selected as part of the sub-sampled training dataset.

Appendix C: Synthetic ground-truth generation

We generated synthetic ground-truth data by applying thresholding rules inspired by works of Bavay and Egger (2014); Tilg et al. (2015) to the HS measurements. In order for a sample to correspond to snow cover, the following condition had to be met:

$$\left(\left(\frac{1}{N}\sum_{n=0}^{N-1}\mathrm{TSS}_n\right) \leq 0.0\right) \wedge \left(\left(\frac{1}{N}\sum_{n=0}^{N-1}\mathrm{RSWR}_n\right) \geq 300.0\right), \tag{C1}$$

where N is the length of the time window.

Appendix D: Machine learning models

For completeness, we provide a short description of every machine learning model that was used in our performance comparison.

D1 Random Forest (RF)

Implemented in many data science libraries and easy to use, Random Forests (RFs) are a popular choice of machine learning algorithm that can provide satisfactory predictions in both classification and regression tasks. In practice, RF is an ensemble approach, which produces a final prediction as a combination of outputs of many decision trees. It often works well on tabular data, but there are no mechanisms that would allow for a more principled representation of temporal, spatial or graph structures.

In our experiments we used the RF classifier implementation from the Scikit-Learn library (Pedregosa et al., 2011), setting the number of decision trees to 1000 and maximum depth of each tree to 50. We left the other parameters at their default settings and trained the RFs using the Gini criterion (Gini, 1936).

D2 Multilayer Perceptron (MLP)

Being one of the first neural network models that can learn non-linear functions, MLPs have shown their power in natural language processing (NLP) and serve as a foundational component for many other neural network models nowadays. Finding their applications in both regression and classification tasks, MLPs can serve as an alternative to the RFs presented above. Putting them in comparison with RFs, MLPs can be generally more difficult to train for a given task and often exhibit lower performance, especially with tabular data. This is due to their nature of learning smooth (sometimes overly smooth) solutions, thereby causing them to not perform well on problems with a non-smooth decision boundary. Grinsztajn et al. (2022) argue this is due to the gradient descent approach to MLP optimization. They also show that MLPs are more affected by, e.g., uninformative features compared to RFs.

We designed an MLP composed of an input layer with 7 input dimensions and 32 output features, followed by 3 hidden layers with 64, 128 and 256 output features, respectively. Each hidden layer had batch normalization (Ioffe and Szegedy, 2015) and Rectified Linear Unit (ReLU) activation functions (Fukushima, 1969; Nair and Hinton, 2010) appended to it. The MLP was concluded with an output layer which takes a 256-feature representation and produces the final class probability score.

D3 Long short-term memory (LSTM)

Belonging to the family of recurrent neural networks (RNNs), the original models developed for time series processing, GRU (Cho et al., 2014) and long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) are variations that allow the model to better capture long-term dependencies compared to RNNs, which tend to forget inputs that came much earlier in the history. We chose to use an LSTM in our experiments, as it is one of the gold standards in deep learning for time-series processing.

The LSTM model we used in our experiments took an input with 7 dimensions and was composed of 3 recurrent layers with hidden dimensions of 64, 128 and 256, followed by an output MLP classifier that produced the final probability scores.

D4 TimesNet

Recently released and setting the new state-of-the-art performance on many standard benchmarks, TimesNet (Wu et al., 2023) has become one of the models of choice for time series processing in general. Its main characteristic is the transformation of a 1-dimensional time series signal into a 2-dimensional one, which allows it to capture complex temporal variations in the signal. The conversion of a time series into a 2-dimensional signal is based on detecting signal periods using amplitude information from a Fast Fourier Transform (FFT) and ordering the signal chunks into a 2-dimensional array. Applying 2-dimensional convolutions to this array allows it to capture both inter- and intra-period variations in the signal.

In our experiments we used a modification where the definition of signal periods is fixed and not determined by the FFT. We used 5 periods to split the signal, namely 48, 32, 24, 16 and 8. The model was then composed of 3 layers with each layer having 2 blocks and 128 hidden features.

D5 Transformer

Since they were published in 2017, transformers have revolutionized many areas of deep learning, achieving new state-of-the-art results mostly in natural language processing and computer vision. Transformers are models based on an attention mechanism (Vaswani et al., 2017) that were originally proposed for sequence-to-sequence tasks.

Here we employed a modification of the traditional transformer. In particular, we took the classical transformer encoder in order to produce a latent representation for the input sequence, where each point is conditioned on the past context. The encoder was composed of 2 layers with hidden dimensions of 128 and 4 attention heads. Both the input positional encoding and encoder have a dropout of 0.1 applied. The latent representation produced by the transformer encoder was average pooled and passed to an MLP readout network, which produced the classification probability scores.

Author contributions. JSc, MR and DL initiated the study and together with MV prepared the research idea and main goals. DL, MR and JSv prepared the data and carried out the manual data annotation. JSv and CJ analyzed the data, developed the models, prepared the experiments, analyzed the results and drafted the original manuscript. MZ contributed the vegetation experiment. All co-authors provided critical reviews and contributed to the final paper. JSc acquired the funding to support the study.

Competing interests. The contact author has declared that neither they nor their co-authors have any competing interests.

Acknowledgements. The authors gratefully acknowledge funding from Swiss Data Science Center grant C21-15L.

References

Avanzi, F., De Michele, C., Ghezzi, A., Jommi, C., and Pepe, M.: A processing-modeling routine to use SNOTEL hourly data in snowpack dynamic models, Adv. Water Resour., 73, 16–29, https://doi.org/10.1016/j.advwatres.2014.06.011, 2014.

Bavay, M. and Egger, T.: MeteoIO 2.4.2: a preprocessing library for meteorological data, Geoscientific Model Development, 7, 3135–3151, https://doi.org/10.5194/gmd-7-3135-2014, 2014.

Blandini, G., Avanzi, F., Gabellani, S., Ponziani, D., Stevenin, H., Ratto, S., Ferraris, L., and Viglione, A.: A random forest approach to quality-checking automatic snow-depth sensor measurements, The Cryosphere, 17, 5317–5333, https://doi.org/10.5194/tc-17-5317-2023, 2023.

Breiman, L.: Random Forests, Machine Learning, 45, 5–32, https://doi.org/10.1023/A:1010950718922, 2001.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y.: On the Properties of Neural Machine Translation: Encoder-Decoder Approaches, in: Proceedings of the Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST@EMNLP), pp. 103–111, Association for Computational Linguistics, https://doi.org/10.48550/arXiv.1409.1259, 2014.

Cybenko, G. V.: Approximation by superpositions of a sigmoidal function, Mathematics of Control, Signals and Systems, 2, 303–314, https://doi.org/10.1007/BF02551274, 1989.

Domine, F.: Physical Properties of Snow, in: Encyclopedia of Snow, Ice and Glaciers, edited by Singh, V. P., Singh, P., and Haritashya, U. K., pp. 859–863, U. K., Springer Netherlands, Dordrecht, https://doi.org/10.1007/978-90-481-2642-2_422, 2011.

Egan, J. P.: Signal detection theory and ROC analysis, Series in Cognition and Perception, Academic Press, New York, NY, ISBN 9780122328503, 1975.

Fiebrich, C., Morgan, C., McCombs, A., Hall, P., and Mcpherson, R.: Quality Assurance Procedures for Mesoscale Meteorological Data, Journal of Atmospheric and Oceanic Technology, 27, 1565–1582, https://doi.org/10.1175/2010JTECHA1433.1, 2010.

Flanner, M., Shell, K., Barlage, M., Perovich, D., and Tschudi, M.: Radiative forcing and albedo feedback from the Northern Hemisphere cryosphere between 1979 and 2008, Nature Geoscience, 4, 151–155, https://doi.org/10.1038/NGEO1062, 2011.

Fontana, F., Rixen, C., Jonas, T., Aberegg, G., and Wunderle, S.: Alpine Grassland Phenology as Seen in AVHRR, VEGETATION, and MODIS NDVI Time Series - a Comparison with In Situ Measurements, Sensors, 8, 2833–2853, https://doi.org/10.3390/s8042833, 2008.

Fukushima, K.: Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements, IEEE Transactions on Systems Science and Cybernetics, 5, 322–333, https://doi.org/10.1109/TSSC.1969.300225, 1969.

Fukushima, K.: Neocognitron: A hierarchical neural network capable of visual pattern recognition, Neural Networks, 1, 119–130, https://doi.org/10.1016/0893-6080(88)90014-7, 1988.

Gini, C.: On the Measure of Concentration with Special Reference to Income and Statistics, Colorade College Publication, 208, 73–79, 1936.

Goehry, B., Yan, H., Goude, Y., Massart, P., and Poggi, J.-M.: Random Forests for Time Series, REVSTAT-Statistical Journal, 21, 283–302, https://doi.org/10.57805/revstat.v21i2.400, 2023.

Good, I. J.: Rational Decisions, Journal of the Royal Statistical Society. Series B (Methodological), 14, 107–114, https://doi.org/10.1111/j.2517-6161.1952.tb00104.x, 1952.

Goodfellow, I., Bengio, Y., and Courville, A.: Deep Learning, MIT Press, http://www.deeplearningbook.org, 2016.

Grinsztajn, L., Oyallon, E., and Varoquaux, G.: Why do tree-based models still outperform deep learning on typical tabular data?, in: Advances in Neural Information Processing Systems, https://doi.org/10.48550/arXiv.2207.08815, 2022.

He, Y. and Zhao, J.: Temporal convolutional networks for anomaly detection in time series, in: Journal of Physics: Conference Series, vol. 1213, p. 042050, IOP Publishing, https://doi.org/10.1088/1742-6596/1213/4/042050, 2019.

Herla, F., Haegeli, P., Horton, S., and Mair, P.: A Large-scale Validation of Snowpack Simulations in Support of Avalanche Forecasting Focusing on Critical Layers, EGUsphere, 2023, 1–38, https://doi.org/10.5194/egusphere-2023-420, 2023.

Hewage, P., Behera, A., Trovati, M., Pereira, E., Ghahremani, M., Palmieri, F., and Liu, Y.: Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station, Soft Computing, 24, 16 453–16 482, https://doi.org/10.1007/s00500-020-04954-0, 2020.

Hochreiter, S. and Schmidhuber, J.: Long Short-term Memory, Neural Computation, 9, 1735–80, https://doi.org/10.1162/neco.1997.9.8.1735, 1997.

Hornik, K., Stinchcombe, M., and White, H.: Multilayer feedforward networks are universal approximators, Neural Networks, 2, 359–366, https://doi.org/10.1016/0893-6080(89)90020-8, 1989.

Inouye, D. W.: Climate change and phenology, WIREs Climate Change, 13, e764, https://doi.org/10.1002/wcc.764, 2022.

Ioffe, S. and Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, in: Proceedings of the 32nd International Conference on Machine Learning, vol. 37, pp. 448–456, https://doi.org/10.48550/arXiv.1502.03167, 2015.

Jerome, D., Petry, W., Mooney, K., and Iler, A.: Snowmelt timing acts independently and in conjunction with temperature accumulation to drive subalpine plant phenology, Global Change Biology, 27, 5054–5069, https://doi.org/10.1111/gcb.15803, 2021.

Jonas, T., Rixen, C., Sturm, M., and Stoeckli, V.: How alpine plant growth is linked to snow cover and climate variability, J. Geophys. Res.-Biogeosci., 113, G03 013, https://doi.org/10.1029/2007JG000680, 2008.

Jonas, T., Marty, C., and Magnusson, J.: Estimating the snow water equivalent from snow depth measurements in the Swiss Alps, Journal of Hydrology, 378, 161–167, https://doi.org/10.1016/j.jhydrol.2009.09.021, 2009.

Kendall, M. G. and Stuart, A.: The Advanced Theory of Statistics. Volume 3: Design and Analysis, and Time-Series, vol. 3 of *Griffin's statistical monographs and courses*, Charles Griffin & Company, London, ISBN 9780852642689, 1966.

Kleene, S. C.: Representation of Events in Nerve Nets and Finite Automata, RAND Corporation, Santa Monica, CA, https://doi.org/10.1515/9781400882618-002, 1951.

Kong, D., McVicar, T., Mingzhong, X., Zhang, Y., Peña-Arancibia, J., Filippa, G., Xie, Y., and Xihui, G.: phenofit: An R package for extracting vegetation phenology from time series remote sensing, Methods in Ecology and Evolution, https://doi.org/10.1111/2041-210X.13870, 2022.

Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., Ravuri, S., Ewalds, T., Eaton-Rosen, Z., Hu, W., Merose, A., Hoyer, S., Holland, G., Vinyals, O., Stott, J., Pritzel, A., Mohamed, S., and Battaglia, P.: Learning skillful medium-range global weather forecasting, Science, 382, 1416–1421, https://doi.org/10.1126/science.adi2336, 2023.

Lea, C., Vidal, R., Reiter, A., and Hager, G. D.: Temporal Convolutional Networks: A Unified Approach to Action Segmentation, CoRR, https://doi.org/10.48550/arXiv.1608.08242, 2016.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P.: Gradient-based learning applied to document recognition, Proceedings of the IEEE, 86, 2278–2324, https://doi.org/10.1109/5.726791, 1998.

Lehning, M., Bartelt, P., Brown, B., Russi, T., Stöckli, U., and Zimmerli, M.: SNOWPACK model calculations for avalanche warning based upon a network of weather and snow stations, Cold Regions Science and Technology, 30, 145–157, https://doi.org/10.1016/S0165-232X(99)00022-1, 1999.

Liechti, D. and Schweizer, J.: The Swiss network of automated snow and weather stations for avalanche forecasting – success factors to its robustness and longevity, in: Proceedings of the International Snow Science Workshop, ISSW International Snow Science Workshop, pp. 1174–1179, https://www.dora.lib4ri.ch/wsl/islandora/object/wsl:37841, 2024.

Lin, T., Goyal, P., Girshick, R. B., He, K., and Dollár, P.: Focal Loss for Dense Object Detection, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 2999–3007, IEEE Computer Society, https://doi.org/10.48550/arXiv.1708.02002, 2017.

Loshchilov, I. and Hutter, F.: Decoupled Weight Decay Regularization, in: Proceedings of the International Conference on Learning Representations, https://doi.org/10.48550/arXiv.1711.05101, 2019.

Luković, M., Zweifel, R., Thiry, G., Zhang, C., and Schubert, M.: Reconstructing radial stem size changes of trees with machine learning, Journal of the Royal Society Interface, 19, 20220349, https://doi.org/10.1098/rsif.2022.0349, 2022.

Matiu, M., Crespi, A., Bertoldi, G., Carmagnola, C. M., Marty, C., Morin, S., Schöner, W., Cat Berro, D., Chiogna, G., De Gregorio, L., Kotlarski, S., Majone, B., Resch, G., Terzago, S., Valt, M., Beozzo, W., Cianfarra, P., Gouttevin, I., Marcolini, G., Notarnicola, C., Petitta, M., Scherrer, S. C., Strasser, U., Winkler, M., Zebisch, M., Cicogna, A., Cremonini, R., Debernardi, A., Faletto, M., Gaddo, M., Giovannini, L., Mercalli, L., Soubeyroux, J.-M., Sušnik, A., Trenti, A., Urbani, S., and Weilguni, V.: Observed snow depth trends in the European Alps: 1971 to 2019, The Cryosphere, 15, 1343–1382, https://doi.org/10.5194/tc-15-1343-2021, 2021.

McCulloch, W. S. and Pitts, W.: A Logical Calculus of the Ideas Immanent in Nervous Activity, The Bulletin of Mathematical Biophysics, 5, 115–133, https://doi.org/10.1007/bf02478259, 1943.

Morin, S., Horton, S., Techel, F., Bavay, M., Coléou, C., Fierz, C., Gobiet, A., Hagenmuller, P., Lafaysse, M., Ližar, M., Mitterer, C., Monti, F., Müller, K., Olefs, M., Snook, J. S., van Herwijnen, A., and Vionnet, V.: Application of physical snowpack models in support of operational avalanche hazard forecasting: A status report on current implementations and prospects for the future, Cold Regions Science and Technology, 170, 102 910, https://doi.org/10.1016/j.coldregions.2019.102910, 2020.

Mott, R., Winstral, A., Cluzet, B., Helbig, N., Magnusson, J., Mazzotti, G., Quéno, L., Schirmer, M., Webster, C., and Jonas, T.: Operational snow-hydrological modeling for Switzerland, Frontiers in Earth Science, 11, https://doi.org/10.3389/feart.2023.1228158, 2023.

Nair, V. and Hinton, G.: Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair, vol. 27, pp. 807–814, https://doi.org/10.5555/3104322.3104425, 2010.

Pascanu, R., Mikolov, T., and Bengio, Y.: On the difficulty of training recurrent neural networks, in: Proceedings of the International Conference on Machine Learning, vol. 28 of *JMLR Workshop and Conference Proceedings*, pp. 1310–1318, https://doi.org/10.48550/arXiv.1211.5063, 2013.

Pearson, R. K.: Data cleaning for dynamic modeling and control, 1999 European Control Conference (ECC), pp. 2584–2589, https://doi.org/10.23919/ECC.1999.7099714, 1999.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E.: Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 12, 2825–2830, http://jmlr.org/papers/v12/pedregosa11a.html, 2011.

Pelletier, C., Webb, G. I., and Petitjean, F.: Temporal convolutional neural network for the classification of satellite image time series, Remote Sensing, 11, 523, https://doi.org/10.3390/rs11050523, 2019.

Pérez-Guillén, C., Techel, F., Hendrick, M., Volpi, M., van Herwijnen, A., Olevski, T., Obozinski, G., Pérez-Cruz, F., and Schweizer, J.: Data-driven automated predictions of the avalanche danger level for dry-snow conditions in Switzerland, Natural Hazards and Earth System Sciences, 22, 2031–2056, https://doi.org/10.5194/nhess-22-2031-2022, 2022.

Raissi, M., Perdikaris, P., and Karniadakis, G.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics, 378, 686–707, ISSN 0021-9991, https://doi.org/https://doi.org/10.1016/j.jcp.2018.10.045, https://www.sciencedirect.com/science/article/pii/S0021999118307125, 2019.

Robinson, D.: Evaluation of the collection, archiving and publication of daily snow data in the United States, Physical Geography, 10, 120–130, https://doi.org/10.1080/02723646.1989.10642372, 1989.

Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain., Psychological Review, 65 6, 386–408, https://doi.org/10.1037/h0042519, 1958.

Ryan, W. A., Doesken, N. J., and Fassnacht, S. R.: Evaluation of Ultrasonic Snow Depth Sensors for U.S. Snow Measurements, Journal of Atmospheric and Oceanic Technology, 25, 667 – 684, https://doi.org/10.1175/2007JTECHA947.1, 2008.

Tilg, A.-M., Ch., M., and G., K.: An automatic algorithm for validating snow depth measurements of IMIS stations (Abstract), 13th Swiss Geoscience Meeting, Basel, Switzerland, 20-21 November 2015, 339, 2015.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I.: Attention is All you Need, in: Advances in Neural Information Processing Systems, pp. 5998–6008, https://doi.org/10.48550/arXiv.1706.03762, 2017.

Vaughan, A., Tebbutt, W., Hosking, J. S., and Turner, R. E.: Convolutional conditional neural processes for local climate downscaling, Geoscientific Model Development, 15, 251–268, https://doi.org/10.5194/gmd-15-251-2022, 2022.

Vitasse, Y., Rebetez, M., Filippa, G., Cremonese, E., Klein, G., and Rixen, C.: 'Hearing' alpine plants growing after snowmelt: ultrasonic snow sensors provide long-term series of alpine plant phenology, International Journal of Biometeorology, 61, 349–361, https://doi.org/10.1007/s00484-016-1216-x, 2017.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K.: Phoneme recognition using time-delay neural networks, IEEE Transactions on Acoustics, Speech and Signal Processing, 37, 328 – 339, https://doi.org/10.1109/29.21701, 1989.

Wan, R., Mei, S., Wang, J., Liu, M., and Yang, F.: Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting, Electronics, 8, 876, https://doi.org/10.3390/electronics8080876, 2019.

Weng, J., Ahuja, N., and Huang, T.: Learning recognition and segmentation of 3-D objects from 2-D images, in: 1993 (4th) International Conference on Computer Vision, pp. 121–128, https://doi.org/10.1109/ICCV.1993.378228, 1993.

Willibald, F., Kotlarski, S., Ebner, P. P., Bavay, M., Marty, C., Trentini, F. V., Ludwig, R., and Grêt-Regamey, A.: Vulnerability of ski tourism towards internal climate variability and climate change in the Swiss Alps, Science of The Total Environment, 784, 147 054, https://doi.org/10.1016/j.scitotenv.2021.147054, 2021.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M.: TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis, https://doi.org/10.48550/arXiv.2210.02186, 2023.

Ying, X.: An Overview of Overfitting and its Solutions, Journal of Physics: Conference Series, 1168, 022 022, https://doi.org/10.1088/1742-6596/1168/2/022022, 2019.