Machine Learning for Snow Stratigraphy Classification

Julia Kaltenborn [*]
McGill University & Mila
julia.kaltenborn@mila.quebec

Viviane Clay
Osnabrück University
vkakerbeck@uni-osnabrueck.de

Amy R. Macfarlane
WSL Institute for Snow and Avalanche Research SLF
amy.macfarlane@slf.ch

Martin Schneebeli
WSL Institute for Snow and Avalanche Research SLF
schneebeli@slf.ch

Abstract

Snow-layer segmentation and classification is an essential diagnostic task for a wide variety of cryospheric science and climate research applications. To this end a Snow Micro Pen (SMP) can be used - a portable high-resolution snow penetrometer. However, the penetration-force measurements of the SMP must be labeled manually, which is a time-intensive task that requires training and becomes infeasible for large datasets. Here, we evaluate how well machine learning models can automatically segment and classify SMP profiles. Fourteen different models are trained on the MOSAiC SMP dataset, a unique and large SMP dataset of snow on Arctic sea-ice profiles. Depending on the user's task and needs, the long short-term memory neural network and the random forests are performing the best. The findings presented here facilitate and accelerate SMP data analysis and in consequence, help scientists to analyze the effects of climate change on the cryosphere more efficiently.

1 Introduction

Snow classification is an essential tool for polar science, cryospheric science, and climate change research. Snow layer segmentation and classification put forth knowledge about the atmospheric conditions a snowpack has experienced. This knowledge helps to discern fundamental snow and climate mechanisms in the Arctic and to analyze polar tipping points.

Traditionally, these snow stratigraphy measurements are made in snow pits - pits dug manually, vertically into snowpacks -, requiring trained operators and a substantial time commitment. The Snow Micro Pen (SMP), a portable high-resolution snow penetrometer [Johnson and Schneebeli, 1998], has been demonstrated as a capable tool for rapid snow grain classification and layer type segmentation. The resulting SMP profiles must be manually labeled, which requires time, training and becomes infeasible for large datasets. Machine learning (ML) algorithms could be used to automate this process. As a consequence this would 1) immensely accelerate the SMP analysis, 2) enable the analysis of large datasets and 3) make the training of interdisciplinary scientists in snow type categorization obsolete.

In previous work a nearest neighbor approach [Satyawali et al., 2009], random forests (RFs) [Havens et al., 2012] and support vector machines (SVMs) [King et al., 2020] were used to achieve an

[*]Work done at WSL Institute for Snow and Avalanche Research SLF and Osnabrück University

Tackling Climate Change with Machine Learning workshop at NeurIPS 2021

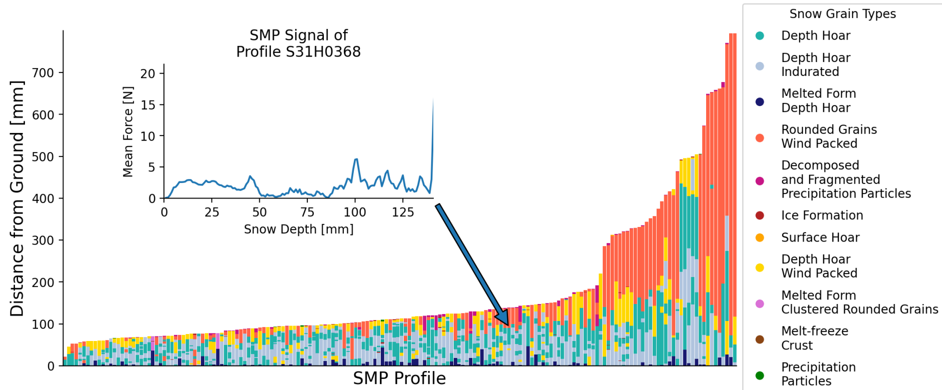

Figure 1: Overview of all used SMP profiles, where each bar is a profile (top of bar equals snow surface) and each color a grain type. The figure within shows a typical SMP signal. The x-axis of the SMP profile indicates how far the measurement tip has moved into the snowpack, and the y-axis is the measured penetration force.

automatic snow grain classification under certain conditions. In contrast to previous work, the models provided in this work, should 1) automate both classification and segmentation, 2) not include any additional measurements except from the SMP signals, 3) classify at least six grain types, 4) not include knowledge-based expert rules, 4) provide a high-resolution of 1 mm layers and 5) should be able to operate on large, unfiltered, real-world SMP datasets. If such a unified approach could actually be used in the field to analyze vast amounts of SMP profiles, it would be the first of its kind and a major enhancement for SMP signal interpretation.

In order to achieve such an automation, this paper compares different ML algorithms of the supervised and semi-supervised learning domain, which were trained to automatically segment and classify SMP profiles collected during the MOSAiC expedition [Shupe et al., 2020].

2 Methods

2.1 Data

The MOSAiC dataset is a unique and extensive dataset characterizing seasonal and spatial variation of snow on the central Arctic sea-ice [Macfarlane et al., 2021]. From a total of 3680 profiles, 164 collected between January and May 2020 were labeled by a snow expert (Figure 1). The manual labelling is achieved by assigning snow grain markers to a SMP signal based solely on the stratigraphy, frequency, strength, and gradient of the force signal. The properties of the different snow grain types are described in Fierz et al. [2009]. The labeled and unlabeled profiles were preprocessed and prepared for the models (Appendix D). The resulting dataset is not easy to classify due to its high complexity, extreme class imbalance and label uncertainty. The raw data becomes publicly available on 1st January 2023 on the open MCS or PANGEA archives [Macfarlane et al., 2021]. Before that the data is available upon request.

2.2 Models

The overall task for the models is to produce a segmented SMP signal with a grain-type assignment for each segment. To do so, the models used here first classify each data point of the SMP signal and subsequently summarize the classified points to distinct snow layers. We chose models from different learning domains - supervised and semi-supervised learning -, and task domains - independent classification and sequence labeling - to diversify the model search and provide a more detailed model overview. The code to train and evaluate the models is available online.[2] As a simple baseline the majority vote classifier is employed, predicting always the majority class.

[2] `https://github.com/liellnima/snowdragon`

As semi-supervised classifiers three different approaches are implemented. The first approach includes three cluster-then-predict models [Soni and Mathai, 2015, Trivedi et al., 2015] that all use a majority vote classifier within the clusters, but differ in the clustering algorithm: K-means clustering [Forgy, 1965, Lloyd, 1982], Gaussian mixture modeling, or Bayesian Gaussian mixture modeling. Furthermore, a self-trained classifier [Yarowsky, 1995] based on a balanced random forest is employed, and a graph-based algorithm propagating labels to unlabeled data points, called label propagation [Zhu and Ghahramani, 2002].[3]

Five different supervised classifiers are tested. This includes both 1) random forest (RF) [Ho, 1995, Breiman, 2001] and 2) balanced random forests [Chen et al., 2004], which are ensembles of diversified decision trees. 3) Support vector machines (SVM) [Cortes and Vapnik, 1995] construct hyperplanes in a high-dimensional space to separate classes from each other. 4) K-nearest neighbors (KNN) [Fix and Hodges Jr, 1952, Cover and Hart, 1967] is a local classification approach that compares samples to its nearest neighbouring data points during prediction. And lastly, 5) an easy ensemble classifier [Liu et al., 2008], an ensemble of balanced adaptive boosting classifiers.

For the supervised sequence-labelling models three different artificial neural network (ANN) architectures are employed: 1) Long short-term memories (LSTMs) [Hochreiter and Schmidhuber, 1997] are recurrent neural network (RNN) architectures that have different memory cells to forget, store and retrieve information necessary for classification decisions. 2) Bidirectional long short-term memories (BLSTMs) [Schuster and Paliwal, 1997] consist of two independent LSTMs processing time-series both forwards and backwards. 3) An encoder-decoder network [Cho et al., 2014], consisting of RNNs, where time-series information is encoded into a vector and decoded during classification. An attention mechanism was added to the encoder-decoder network to improve the model's ability to learn long-term dependencies [Bahdanau et al., 2014].[4]

2.3 Evaluation

The evaluation of the different models is based on balanced accuracy, absolute accuracy, weighted precision, F1 score, ROC AUC, log loss, fitting and scoring time. Qualitative evaluations such as the "smoothness" of the predicted SMP profiles are considered as well and can be found online. The results presented in the following section cannot be compared directly with results from Satyawali et al. [2009], Havens et al. [2012] and King et al. [2020], because they use different and fewer snow grain types and provide their algorithms with additional snow pit data or use manually pre-segmented profiles. For comparability reasons, the models used in previous work are also implemented in this work. For specifications about the models and experimental setup refer to Appendix C.

3 Results

All models were able to outperform the majority vote baseline (Appendix A) in all metrics except fitting and scoring time. Among the semi-supervised models the self-trained classifier and the label-propagation performed best, while the cluster-then-predict models need further improvements, since they score as low as the baseline for the labels depth hoar wind-packed and precipitation particles (Figure 2). The label-propagation algorithm produces very fragmented predictions, which stands in contrast to the desired smooth, expert-like predictions. This makes the self-trainer the preferred model of the comparatively weak semi-supervised model group. Of all supervised classifiers, the RF, the balanced RF and the SVM have similar high scores, making it possible to choose among those three models. The balanced RF even achieves the highest balanced accuracy of all models (0.672), however, the RF does not overestimate the minority classes so strongly and produces more accurate overall predictions (0.726). The KNN model and the easy ensemble cannot keep up with the performances of the RFs and the SVM (Appendix A). The ANNs have the highest performance values for all metrics, except balanced accuracy, fitting and scoring time (Appendix A). The encoder-decoder achieves of all models the best absolute accuracy with 0.78. Its performance scores are slightly better than the LSTM's, but the LSTM is more than 8 times faster and much robuster to hyperparameter tuning. In contrast to the other model groups, the performances of all three ANNs depend strongly on the hyperparameter tuning. In summary, the LSTM and the encoder-decoder network show the highest performance values of all models, followed by the RFs, the SVM and the self-trained classifier.

[3] Refer to Bishop [2006] for detailled explanations about several of these algorithms.

[4] Refer to Jurafsky and Martin [2021] for detailled explanations.

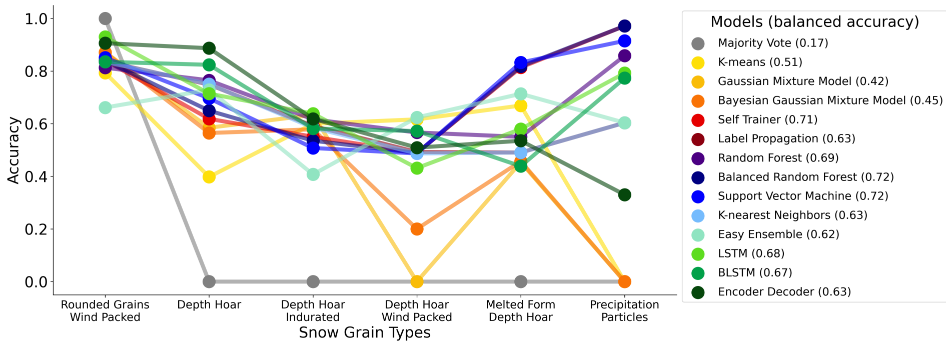

Figure 2: Class accuracy values of each model. The x-axis represents the snow grain types (descending with number of examples in the label dataset), the y-axis shows accuracy, and the colors the models (yellow-red: semi-supervised, purple-blue: supervised, green: ANNs). The label “rare” is omitted.

Label-wise, the majority class "rounded grains wind-packed" is classified very well by all models, depth hoar wind packed and indurated are difficult to predict for all models, depth hoar and its melted form are somewhat difficult to predict for most models, and the class “precipitation particles” yields accuracy performances spread along the complete range (Figure 2). The balanced RF and SVMs are especially good at predicting rare classes. The ANNs, in contrast, are, of all models, scoring highest for the three largest classes. There are a few models which score consistently high for different labels, such as the LSTM, BLSTM and the RF.

4 Discussion

There is no model that outperforms all the others, instead several models perform well under different circumstances. The underwhelming cluster-then-predict models could be improved substantially by using a different predictor than the majority vote within the clusters. However, by outperforming the majority vote classifier, they showed that clustering the data exhibits additional information. The ANNs show the most expert-like predictions and the best overall performances. The main reason for that might be that they interpret the data truly as time-series and can access (long-term) time dependencies which the other models cannot. For this reason, models interpreting the task as label-sequencing task might generally perform better than time-independent models.

The code provided alongside this paper enables SMP users from different fields for the first time to automatically segment and classify snow pack profiles. A practitioners guide on how to choose a suitable model is provided in Appendix B. Based on the findings described here, a SMP classification and segmentation tool could be developed and integrated into the existing SMP analysis package snowmicropyn.[5] Such a tool would make knowledge about snowpacks easier and faster accessible for all scientists and would thus facilitate climate change research. An immediate impact of this project is that the connected analysis of the MOSAiC SMP dataset makes the retrieval of essential information about the state of the Arctic cryosphere possible. This way, it enables climate and cryospheric scientists to understand the effect of climate change on our planet’s cryosphere and especially the Arctic better.

The most important limitation of the models is that the classification might never be able to reach accuracy close to 100% because of the grain types’ nature and the subjectivity of the manual labelling process. Due to snow metamorphism, where snow grains are transforming into others, snow classes are not discrete, but more of a continuous nature. Transforming, overlapping and internal subclasses of the grain labels impede the classification and make it difficult for experts to find clear labels and consequently the training labels are afflicted with high uncertainty.

[5] `https://snowmicropyn.readthedocs.io/en/latest/`

5 Conclusion

This work shows that an automatic classification and segmentation of SMP profiles is possible, even when real-world, unfiltered SMP profiles are used. Depending on the classification needs, and the temporal and computational resources at hand, the (balanced) RF and the LSTM are especially well suited for automatic SMP analysis. In future studies, we would like to test the generalization capabilities of the models by analyzing their predictions for MOSAiC SMP profiles from a different season (e.g. melting season). The MOSAiC dataset provides humanity for the first time in history with a large amount of detailed data about the Arctic's condition. The ML-driven approach used here to analyze SMP profiles will be one of many methods to make the knowledge behind the data accessible - knowledge that is essential to understanding and mitigating climate change impacts.

Acknowledgements

Data used in this manuscript was produced as part of the international Multidisciplinary drifting Observatory for the Study of the Arctic Climate (MOSAiC) with the tag MOSAiC20192020. The data was collected during the Polarstern expedition AWI PS122 00.

References

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `http://tensorflow.org/`. Software available from tensorflow.org.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

C. Chen, A. Liaw, and L. Breiman. Using random forest to learn imbalanced data. 110:1–12, 2004.

K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

F. Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1): 21–27, 1967. doi: 10.1109/TIT.1967.1053964.

CyberZHG. Keras self-attention. `https://github.com/CyberZHG/keras-self-attention`, 2020.

C. Fierz, R. L. Armstrong, Y. Durand, P. Etchevers, E. Greene, D. M. McClung, K. Nishimura, P. K. Satyawali, and S. A. Sokratov. The international classification for seasonal snow on the ground. 2009.

R. A. Fisher. Statistical methods for research workers. In *Breakthroughs in statistics*. Springer, 1992.

E. Fix and J. L. Hodges Jr. Discriminatory analysis-nonparametric discrimination: Small sample performance. Technical report, CALIFORNIA UNIV BERKELEY, 1952.

E. W. Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, 21:768–769, 1965.

S. Havens, H.-P. Marshall, C. Pielmeier, and K. Elder. Automatic grain type classification of snow micro penetrometer signals with random forests. *IEEE transactions on geoscience and remote sensing*, 51(6): 3328–3335, 2012.

T. K. Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

J. B. Johnson and M. Schneebeli. Snow strength penetrometer, Nov. 3 1998. US Patent 5,831,161.

D. Jurafsky and J. H. Martin. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition, 2021. URL `https://web.stanford.edu/~jurafsky/slp3/`. In progress. 3rd ed. draft. Can be found at `https://web.stanford.edu/~jurafsky/slp3/`.

J. King, S. Howell, M. Brady, P. Toose, C. Derksen, C. Haas, and J. Beckers. Local-scale variability of snow density on arctic sea ice. *The Cryosphere*, 14(12):4323–4339, 2020.

G. Lemaître, F. Nogueira, and C. K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, 18(1):559–563, 2017.

X.-Y. Liu, J. Wu, and Z.-H. Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2008.

S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

H. Löwe and A. Van Herwijnen. A poisson shot noise model for micro-penetration of snow. *Cold Regions Science and Technology*, 70:62–70, 2012.

A. R. Macfarlane, S. Arndt, R. Dadic, H.-R. Hannula, M. Jaggi, N. Kolabutin, D. Krampe, M. Oggier, R. Pirazzini, I. Raphael, J. Regnery, E. Shimanshuck, D. N. Wagner, and M. Schneebeli. Snowmicropen raw data (sn_smp_31, sn_smp_43 and sn_smp_49) during mosaic expedition, 2021. In Review.

K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

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.

P. Satyawali, M. Schneebeli, C. Pielmeier, T. Stucki, and A. Singh. Preliminary characterization of alpine snow using snowmicropen. *Cold Regions Science and Technology*, 55(3):311–320, 2009.

M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

M. D. Shupe, M. Rex, K. Dethloff, E. Damm, A. Fong, R. Gradinger, C. Heuzé, B. Loose, A. Makarov, W. Maslowski, et al. Arctic report card 2020: The mosaic expedition: A year drifting with the arctic sea ice. 2020.

R. Soni and K. J. Mathai. Improved twitter sentiment prediction through cluster-then-predict model. *arXiv preprint arXiv:1509.02437*, 2015.

M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133, 1974.

S. Trivedi, Z. A. Pardos, and N. T. Heffernan. The utility of clustering in prediction tasks. *arXiv preprint arXiv:1509.06163*, 2015.

L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995.

X. J. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. 2002.

A Appendix

Model	Absolute Accuracy	Balanced Accuracy	Precision	F1 Score	ROC AUC	Log Loss	Fitting Time	Scoring Time
Majority Vote	0.390	0.143	0.152	0.219	nan	nan	**0.001**	$< \mathbf{10^{-3}}$
K-means	0.620	0.440	0.609	0.612	nan	nan	384.7	0.010
GMM	0.649	0.363	0.586	0.612	nan	nan	151.3	<u>*0.008*</u>
BGMM	0.646	0.382	0.626	0.625	nan	nan	224.8	0.009
Self trainer	0.692	<u>0.670</u>	<u>0.736</u>	0.708	0.918	0.840	19.3	0.292
Label propagation	<u>0.714</u>	0.538	0.717	<u>0.712</u>	0.916	1.499	<u>10.5</u>	3.352
RF	<u>0.726</u>	0.596	0.731	<u>0.726</u>	0.927	0.704	72.2	0.965
Balanced RF	0.696	<u>**0.672**</u>	<u>0.741</u>	0.712	0.919	0.836	9.9	0.579
SVM	0.705	0.656	0.731	0.710	<u>0.934</u>	<u>0.668</u>	18.6	7.451
KNN	0.712	0.536	0.714	0.710	0.891	3.584	<u>*0.006*</u>	1.837
Easy Ensemble	0.616	0.591	0.700	0.639	0.878	1.656	46.1	42.494
LSTM	*0.754*	<u>0.584</u>	*0.751*	*0.746*	<u>**0.944**</u>	<u>**0.633**</u>	<u>349.1</u>	<u>2.299</u>
BLSTM	0.736	0.575	0.742	0.734	0.927	0.793	974.9	3.410
Encoder Decoder	<u>**0.780**</u>	0.541	<u>**0.780**</u>	<u>**0.774**</u>	*0.943*	*0.642*	2911.2	5.755

Table 1: Results of each model on the testing dataset with the best parameters found during hyperparameter tuning. The best score in each category is bold and the second best italic. The best scores of each model category are underlined. The first category is the baseline, the second the semi-supervised models, the third the independent supervised classifiers and the last the ANNs. Log-Loss and ROC AUC score are not compared if not all models within a group have them.

B Appendix

B.1 Practitioners Guide

The following criteria should be considered when choosing a model for an automatic SMP analysis:

A) Time and resources for hyperparameter tuning. The LSTM and the encoder-decoder network are recommended when plenty of tuning time is available. Especially, the encoder-decoder network performs badly if not tuned well. The SVM and the balanced RF need little tuning time, whereas the RF is the go-to-model in case (almost) no tuning time can be provided.

B) Need for a simple to handle, off-the-shelf algorithm. Among the high-performing models, the RFs and the SVM are the easiest to handle off-the-shelf algorithm. The self-supervised algorithms and especially the ANNs require a somewhat deeper understanding of the models and the ability to implement those.

C) Importance of minority classes. When deciding on a model, the underlying task must be examined as well: In case of avalanche prediction it is essential to predict a buried layer of "surface hoar" - a very rare class, which needs to be detected no matter at which costs. In such a case of "minority class prediction" the balanced RF or the SVM should be considered. The ANNs and the RF, in contrast, are more suitable to achieve an overall good classification.

D) Availability of unlabeled data that is from the same distribution as the labeled data. In case a lot of unlabeled data from the same distribution and time is available, the self-trained classifier can be considered. The weak learner of the self-trained classifier can be chosen according to the criteria listed above. Since in this work we only had a small subset of unlabeled data stemming from the same distribution like the labeled data, further evaluations on the self-trained classifier and label-propagation remain open.

C Appendix

C.1 Model Setup

Python 3.6 was used throughout the project. All used packages can be found on `https://github.com/liellnima/snowdragon` in the requirements text file. PCA, t-SNE, k-means clustering, Gaussian mixture models, Bayesian Gaussian mixture models, RFs, SVMs and the KNN algorithm were used as made available through scikit-learn from Pedregosa et al. [2011]. [6] The easy ensemble for imbalanced datasets and a balanced variant of the RF are imported from imbalanced-learn from Lemaître et al. [2017]. [7] All ANN architectures were created with the help of TensorFlow [Abadi et al., 2015] and Keras [Chollet et al., 2015].[8] [9] The attention model within the encoder-decoder network was used as provided in the keras-attention-mechanism package by CyberZHG [2020].

C.2 Experimental Setup

A typical training, validation, testing and tuning framework was employed. Circa 80% of the labeled dataset is used as training and validation data, while circa 20% is used for testing. Validation is realized as a 5-fold cross-validation [Stone, 1974] during hyperparameter tuning. Moderate hyperparameter tuning was applied and all tuning results can be found in the github repository. [10]

The experiments were run on two different machines. A set of experiments that is compared to each other was always run on the same machine. The machine on which single experiments and the complete evaluation were conducted is a 64-bit system with an Ubuntu 18.04 (Bionic Beaver) operating system. The machine has 16 GB RAM and an Intel® Core™ i7-6700HQ CPU @ 2.60GHz × 8 (and the GPU was not used). Hyperparameter tuning, training and validation were run on an Azure virtual machine of the Dsv3-series, namely on a Standard_D4s_v3 [11] machine with Ubuntu 18.04 (Bionic Beaver) as an operating system, 16 GB RAM and 4 vCPUs. Wherever possible and throughout the snowdragon repository, the number 42 was used for random states and random seeds. The results are replicable when using 42 as random state and seed.

D Appendix

D.1 Data Preprocessing

During data preprocessing the data was cleaned and prepared for the models. Among other steps, measurement side effects were cleansed out, the force signal of each SMP profile was binned into 1 mm sections, and sliding windows were applied to add time-dependent variables from a Poisson shot noise model by Löwe and Van Herwijnen [2012]. The overall dataset was normalized, some snow grain classes were merged and profiles from the melting season were removed. Subsequently, the structure of the dataset was analyzed with feature selection (ANOVA [Fisher, 1992], and RF Feature Importance) and dimension reduction techniques (Principal Component Analysis [Pearson, 1901, Hotelling, 1933], and t-distributed Stochastic Neighbor Embedding [Van der Maaten and Hinton, 2008]) to understand the interaction between the different snow grain types better.

[6] `https://scikit-learn.org/stable/`

[7] `https://imbalanced-learn.org/stable/`

[8] `https://www.tensorflow.org/`

[9] `https://keras.io/`

[10] `https://github.com/liellnima/snowdragon`

[11] `https://docs.microsoft.com/en-us/azure/virtual-machines/dv3-dsv3-series`