POWDER WIZARD

Sam Scudder

Abstract

Avalanche forecasting, like weather forecasting, is a detail oriented and data intensive process. However, while weather forecasters have adopted high resolution computational models, avalanche forecasters continue to rely on data from a limited number of physical weather stations. Researchers have advocated for the use of machine learning to aid with avalanche forecasting, but previous hesitation to trust weather models in mountainous regions has deterred the development of new systems rooted in artificial intelligence. The Powder Wizard explores how modern neural network libraries can be used to generate detailed avalanche forecasts. This functionality will allow for more frequent and higher resolution forecasts as weather models become a trusted data source even in mountainous regions.

Introduction

Backcountry skiing is a rapidly growing winter pastime and the early closure of resorts caused by the COVID-19 pandemic generated even more interest in the sport towards the conclusion of the 2019/20 ski season in North America [1]. As the number of participants swells, it becomes increasingly important for backcountry recreationalists to make safe decisions for the sake of themselves and other members of the backcountry community.

Avalanche forecasts are a tool that backcountry travelers generally use to plan for trips into the backcountry. A forecast includes information about the weather, a general advisory for avalanche danger that day, and a more detailed list of additional hazards one should expect to encounter. Though a great resource, they have their pitfalls. Mountain ranges are are divided into smaller regions for the forecasts, but these regions can still encompass enormous areas of varying terrain. Many remote regions also lack forecasts of any sort.

Over two decades ago, both Swiss and American researchers proposed using neural networks to aid with avalanche forecasting [4, 5]. Even at that time, technology was slowly creeping its way into the world of snow science. Since the 1990s, The WSL Institute for Snow and Avalanche Research SLF has used the SNOWPACK computational model to understand different variables in the snowpack throughout the Swiss Alps and the resulting data is now part of their forecasting toolkit [2]. More recent work has been even more specific as researchers from the University of Calgary have used both the SNOWPACK model and its French equivalent, Crocus, to understand how surface hoar, the most notoriously dangerous feature in a snowpack, stabilizes over time [3].

At the moment, all of the aforementioned models are constrained by primarily using data from a finite number of physical weather stations. Because of a high density of weather stations throughout the European Alps, the Crocus and SNOWPACK models are relatively robust. But in

North American, there are fewer weather stations throughout larger mountain regions, meaning there are larger gaps in data.

Weather forecasting models are another data source that also provide the temperature, wind, and snow data that is used to formulate avalanche forecasts, but weather models still have not been used in the forecasting process because they historically have been inaccurate in mountain regions. As technology progresses, such models have become more trusted by the alpine climbing and backcountry skiing community. Eventually, these weather models may become accurate enough to become a primary data source for avalanche forecasts. If so, it will be perfect time to heed the call for computationally supported avalanche forecasting [6].

The Powder Wizard is a project that seeks to understand the process for training and deploying neural networks that can be used for generating complete avalanche forecasts. In summary the Powder Wizard strives to accomplish the following:

- Train a basic neural network for general avalanche advisories.
- Integrate a complete compilation of data into a training model to generate detailed forecasts.
- Deploy a system of neural networks that generate forecasts.

At the conclusion of this quarter, the Powder Wizard achieved two of its three goals. A complete overview of the project's progress will be given in the final section of this paper.

Related Works

The Open Avalanche Project [7] was a previous project that also explored the potential to leverage machine learning and weather models to create higher resolution avalanche forecasts. The project was terminated after machine learning models were trained to generate general avalanche

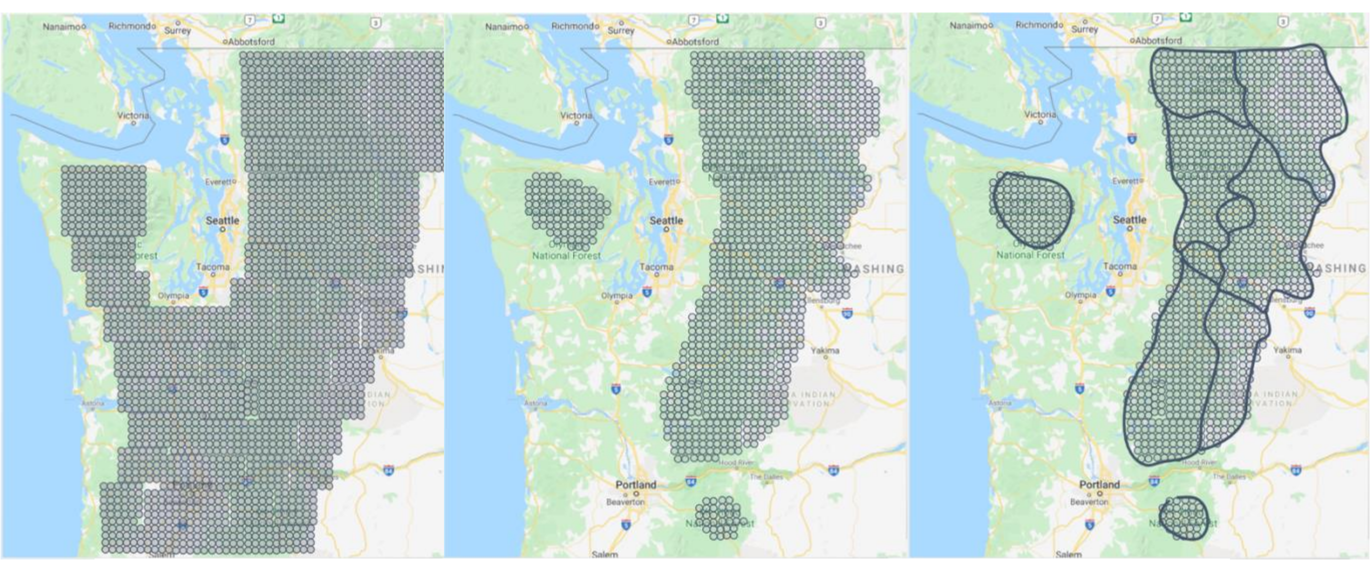


Figure 1 A visualization of the labeling process for NAM $12km_2$ datasets using regions covered by forecasts made by the NWAC.

advisories above, below, and near treeline throughout the Cascades. The project must be credited for laying out the groundwork for the Powder Wizard.

Because of the Open Avalanche Project, three winters worth of historical North American Mesoscale (NAM) forecasts have been cleaned and formatted to be easily uploaded into DataFrames using the Pandas library for Python. The NAM data and additional advisory labels were combined to make a training dataset. The Open Avalanche Project also cleaned and sorted one winter worth of historical avalanche forecasts from the Northwest Avalanche Center (NWAC). Because of this readily available data, the Powder Wizard was also initially trained to be used in the Pacific Northwest.

Apart from the functions used to import and separate the NAM data into labels and features, a majority of the programs written for the Open Avalanche Project are not used for the Powder Wizard because of different choices for the technology stacks. The Open Avalanche Projects trains its three models using the Scikit-learn machine learning library and deploys the system using Azure and C#. To make use of currently more popular libraries, the Powder Wizard will employ TensorFlow with the Keras library. The web application will be built using the Django web framework.

Technical Implementation

Data Preparation

The training dataset used for the Open Avalanche Project initially seemed to align with the historical NWAC forecasts. Each example from the training dataset included 295 columns of numeric data related to wind speed, air temperature, precipitation and snowmelt. The examples also included 45 columns of categorical data related to characteristics of the snowpack. Often times, the data was split into time increments or locations within the snowpack (i.e. snow accumulation over last 7 days, snow accumulation over last 3 days, freeze thaw likelihood 3 inches down in the snowpack, etc.). The origins of the data could be identified using the latitude and longitude and each example was assigned to 1 of 10 regions. It also had output labels for the avalanche danger above, near, and below treeline.

The NWAC provides forecasts for 10 regions with names similar to those used in the training dataset. Thus, the original training data as prepared by the Open Avalanche Project was used as the training dataset for the first part of this project. While adding the additional forecast details to the training dataset, it became clear that 10 regions used by the NWAC were not actually the same as the 10 regions identified in the original training dataset.

Using the latitude and longitude to manually compare data points from the training data with the NWAC forecasts, it turned out many of examples used in the historical training data did not fall within the regions that are covered by the NWAC forecasts. Additionally, while all the data points that shared a single region within training dataset consistently shared output labels, the output labels did not seem to align with the corresponding daily forecasts from the NWAC.

It is possible that the Open Avalanche Project used semi-supervised machine learning techniques to label additional data. Without specific documentation showing how this was done, it seemed best to relabel the weather data in its entirety (Figure 1).

The labeling process began by first ensuring the dates for both the NWAC data and the NAM data overlapped. The available NAM dataset included seasonal data from November of

2013 to April of 2017, annually excluding the months between April and October. Meanwhile, the NWAC data was available for a single year: March of 2016 to March of 2017. Within the yearlong window, there were also random days for which either the avalanche data or the weather data was not recorded. In the end, complete sets of avalanche and weather data only existed for 146 days.

With regards to relabeling the data, the first step was to manually define bounds that defined the entirety of the region covered by the NWAC. After this, more specific bounds were created to split the NAM data into regions that aligned with the 10 regions defined by the NWAC. Using these regions, the output data was relabeled to match the NWAC forecasts and additional labels related to specific hazards that were also part of the historical avalanche forecasts were incorporated into the training data as originally planned.

After narrowing down the data, the quantity of training data did drop significantly. The original training dataset included 74,768 examples of NAM data over three winters, the final dataset consisted of 18,840 examples of NAM data combined with forecasts from the NWAC.

Portion of Forecast	Category	Labels
General Advisory	Below Treeline	Low Moderate Considerable High
	Near Treeline	
	Above Treeline	
Detailed Forecast	Likelihood	Very Unlikely Unlikely Likely Very Likey Certain
	Size	Small Medium Large Very Large Extremely Large
	Hazard	Wind Slab Loose Wet Persistent Slab Storm Slab Wet Slabs Cornices

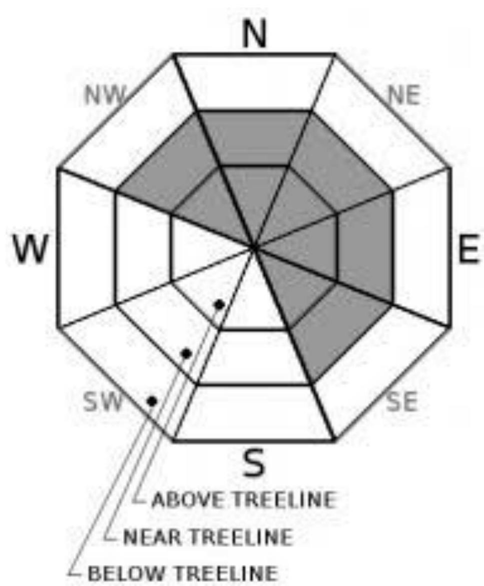


Figure 2 A list of the of items that go into a complete avalanche forecast and their possible values. The avalanche rose, shown on the right, is used to identify slop aspects and altitudes where avalanche events are likely to occur. There are 256 possible configurations of the rose.

Training Neural Networks

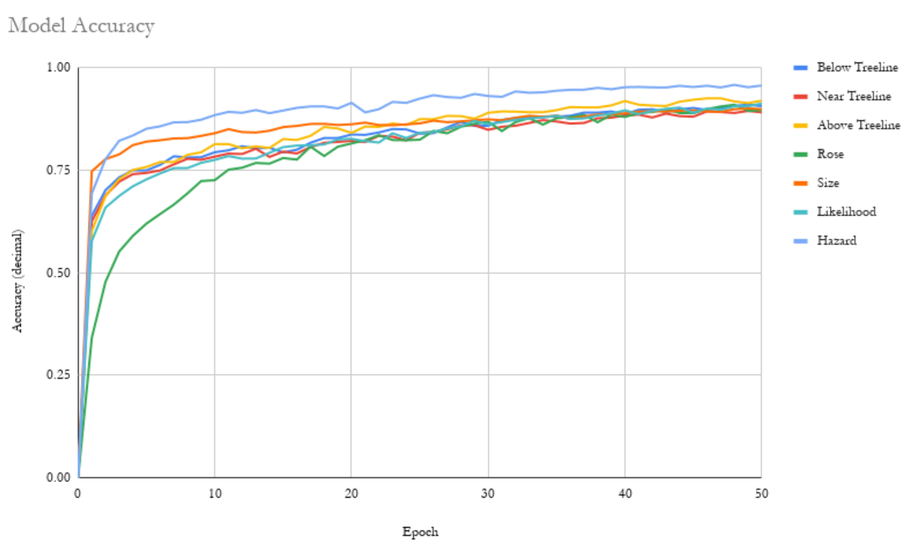
The Keras library for TensorFlow was used to format and train the Sequential Neural Networks (SNN) that power the Powder Wizard. This section will walk through the technical process of implementation including how data was organized to be used with Keras for TensorFlow.

The project began by attempting to train a single SNN to predict avalanche danger below treeline. Using the training data originally prepared by the Open Avalanche Project, the entire set of 240 input variables and 4 possible output labels were used to train the model. To do so, the weather data and labels were converted to tensors that could be used as inputs for a Keras training function.

Originally, two models were tested. The first was a simple 2-layer network with the 240 input neurons, correlating to the input variables, and 4 output neurons, correlating to the possible output labels. Once functional, the model attained a maximum accuracy of 25%. To experiment with other options, a second 2-layer neural network was trained using TensorFlow feature columns. The resulting model had a similar accuracy of 25%. The Powder Wizard seemed to be content with randomly selecting a forecast for each day.

After trying to apply the same data preparation and training technique to create SNNs for predicting avalanche danger above and near treeline, these neural networks achieved similar accuracies of approximately 25%. Researching possible reasons for the low accuracy, it seemed that over training, under training, and invalid training data could all have contributed to the suboptimal results.

After completely relabeling and incorporating additional detailed forecast information into the datasets, the pipeline for training the SNNs was reworked. Instead of converting the revised training data from Pandas DataFrames to TensorFlow tensors, the data was converted to NumPy arrays for input into the Keras fit function. Additionally, all of the categorical input data was cut from the training model, decreasing the number of input neurons from 340 to 295. The changes



Category	Final Accuracy (%)
Below Treeline	90.6
Near Treeline	98.2
Above Treeline	92.0
Hazard	95.7
Size	90.0
Likelihood	91.2
Rose	90.0

Figure 3 The left-hand image shows the incremental increase in accuracy over 50 epochs for the training data. The right-hand chart shows the final accuracies for each label at the end of the training.

resulted in a more accurate model. Having made multiple changes at once, it is unclear what ultimately changed the accuracy of the training function. However, it is a likely a misuse of tensors affected the original two models.

With a functional data input pipeline in place, models were trained for the additional output labels. For each label, the same 2-layer SNN was used and there were 295 input neurons. The only variation between the models was the number of output neurons. The models trained for the avalanche advisories below, near, and above treeline each have 4 output neurons. The models trained for the Likelihood and Size of avalanches have 5 output neurons. The model trained for the hazard type has 6 output neurons. The model trained for the avalanche rose has 28 output neurons. The final model has fewer neurons than possible configurations because less than 28 configurations were observed in the training process.

Each of the models achieved relatively high accuracy when trained using 50 epochs and additional epochs negligibly affected the accuracies (Figure 3).

Deployment

The complete Powder Wizard system consists of 7 SNNs that can be used to generate complete avalanche forecasts (Figure 4). To do so, the same input data is compiled and entered into each of the SNNs as an array of NumPy arrays. The SNNs each generate an array whose values indicate the likelihood of each of the possible label values. Taking the index of the maximum value in the array and comparing it with a corresponding list of labels allows the program to output strings that depict the label value.

After the models are trained, each is exported as an H5 file. The files can be imported for use in other Python files. At this point, the models are yet to be redeployed in a web application.

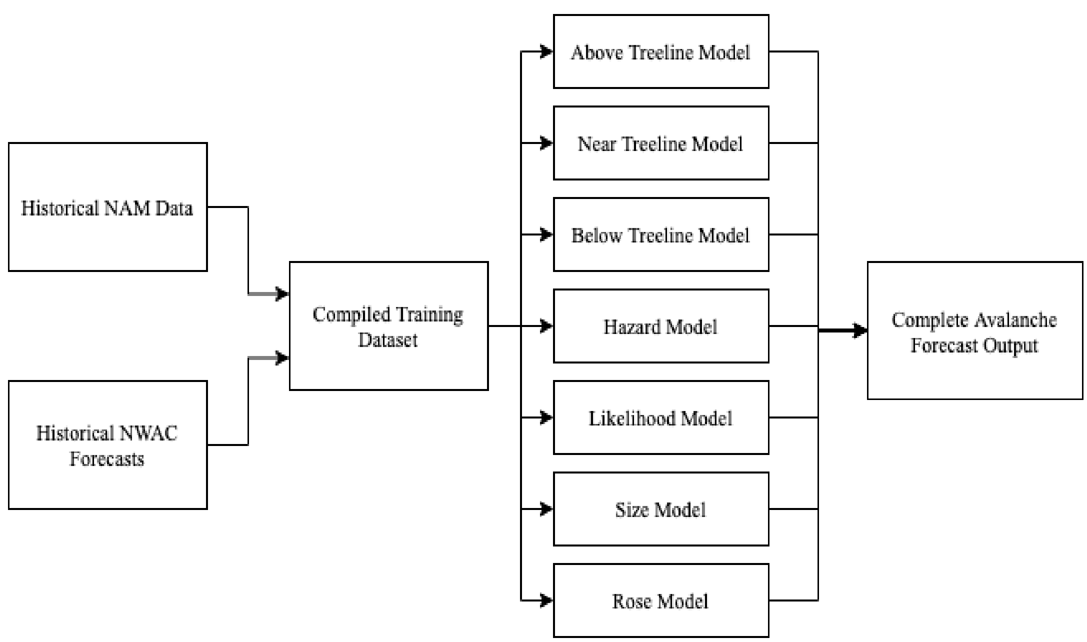


Figure 4 A diagram of the parts that comprise the Power Wizard system.

Milestones/Project Refection:

The original project plan had three intended deliverables with smaller tasks included as well. They were the following:

- Deliverable 1: A basic interactive map.
  - Combine weather data from SNOTEL and from NAM
  - Develop a basic interactive map
- Deliverable 2: A map containing a higher resolution of regions that could be used for unique forecasting.
  - Cluster data from weather forecasts to generate labels
  - Develop Sequential Forecasts
- Deliverable 3: Avalanche prediction application.
  - Use basic machine learning to train a model to generate forecasts
  - Apply model to other regions.

The original ideas now seem slightly unclear and overly ambitious, however they provided inspiration for the project as a whole. After working with the historical weather data and being drawn to the machine learning pipeline, I readjusted my project milestones to focus on the machine learning that would power the Powder Wizard. At the midway point in the quarter, I had adjusted my milestones to include baseline goals for attaining a minimal viable product and additional optional tasks. The following list is a more concise description of the deliverables I submitted at the midway point of the quarter:

- Deliverable 1: A semi-functional classification model.
  - Baseline:
    - Train a TensorFlow model.
  - Above and Beyond:
    - Achieve approximately 75% accuracy with all training models.
    - Reimplement using Tensor.js
- Deliverable 2: A revised dataset that includes more data form the NWAC forecasts.
  - Baseline:
    - Sort data and provide additional labels for the training data using detailed information from historical NWAC forecasts.
  - Above and Beyond:
    - Determine what NAM data is and isn't necessary for the training model to cut down the potential memory needed to deploy a more complete system.
    - Create more user-friendly input data that could be used to explain the how the entire system works.
- Deliverable 3: Model deployment.
  - Baseline:
    - Deploy model locally.
  - Above and Beyond
    - Design a user-friendly interface.
    - Include an interactive map in the application.

I am disappointed that I was unable to deploy a functional application locally, but I do feel that I made significant steps towards future deployment. This being said, I did encounter unanticipated challenges while working on the project. I originally chose to work with data prepared by the Open Avalanche Project because I hoped it would allow me to skip the data processing required to launch most new projects. Thus, when I found the data to be less reliable than I anticipated, I thought it would be better to allocate time towards relabeling the data so the system would generate results I felt more comfortable with. This seemed like an ethically sound decision.

In the end, though I only accomplished 2 of my baseline milestones (Adding the training labels and training a TensorFlow model), I did accomplish one of the supplemental milestones (Achieve 75% accuracy). I also allocated a significant amount of time towards a task that was not used as a milestone (filtering and relabeling all of the training data). Thus, substituting the unfinished baseline milestone with the accomplished extra milestone and factoring in the additional task of data preparation, I feel I deserve a comfortable B for the effort I put towards this project.

Conclusion

Avalanche forecasts are one of the primary resources used by skiers to understand how they can stay safe in the backcountry, but they have remained the same for decades. Not only could artificial intelligence be used to bolster the current forecasting process, but models trained using data from higher resolution sources could allow for safer exploration of both local and remote backcountry destinations. Especially with the rapid growth of backcountry skiing in the past few years, now is the time for change.

Thus far, the work completed for the Powder Wizard includes successfully combining historical NWAC and NAM data and training several SNNs that can be used to generate complete avalanche forecasts. Immediate next steps include completing the online deployment of the Powder Wizard and a incorporating a system for scraping NAM data to be used for real time forecasting.

The Powder Wizard has also exposed some of the challenges associated with training neural networks using alternative data sources. The process of relabeling data for this project included generalizing the avalanche forecasts created using a very finite number of weather stations, inherently contradicting the benefits of using higher resolution weather data. Thus, if possible, it would be interesting to investigate how clustering or other semi-supervised learning techniques could be used to create better labels for the training data. It would also be interesting to see how models trained using NAM data only from regions directly surrounding weather stations would perform.

Personally, working on the Powder Wizard has been a humbling experience, but I am excited to have built up the momentum that will allow me to continue with this project. I now look forward to learning more about Python frameworks like Django and working with TensorFlow has inspired me to dive deeper into properly understanding neural networks.

I would like to thank Professor Ryan Kastner for a great quarter and for allowing me to pursue this project for his class.

References

1. Powder. "How COVID-19 Will Change the Future of Backcountry Skiing." *POWDER Magazine*, Powder Magazine, 29 May 2020, www.powder.com/stories/how-covid-19-will-change-the-future-of-backcountry-skiing/.

2. Jürg Schweizer, Sascha Bellaire, Charles Fierz, Michael Lehning, Christine Pielmeier, Evaluating and improving the stability predictions of the snow cover model SNOWPACK, Cold Regions Science and Technology, Volume 46, Issue 1, 2006,

3. Simon Horton, Bruce Jamieson, Modelling hazardous surface hoar layers across western Canada with a coupled weather and snow cover model, Cold Regions Science and Technology, Volume 128, 2016,

4. J. Stephens, E. Adams, X. Huo, J. Dent, J. Hicks, D. McCarty, Use of neural networks in avalanche hazard forecasting, Proceeding of the 1994 Snow Science Workshop

5. M. Schweizer, M. Fohn, J. Schweizer, Integrating Neural Networks and Rule Based Systems to build an Avalanche Forecasting System, International Association of Science and Technology for Development, 1994

6. James A. Floyer, Karl Klassen, Simon Horton, Pascal Haegeli, Looking into the '20s: Computer-Assisted Avalanche Forecasting in Canada, International Snow Science Workshop 2016 Proceedings

7. https://openavalancheproject.org/