University of
South-Eastern Norway

www.usn.no

FMH606 Master's Thesis 2021
Industrial IT and Automation

Development of models for estimating snow depth and snow density using machine learning methods

Henrik Nikolai Vahl

Faculty of Technology, Natural sciences and Maritime Sciences
Campus Porsgrunn

University of South-Eastern Norway

www.usn.no

Course: FMH606 Master's Thesis, 2021

Title: Development of models for estimating snow depth and snow density using machine learning methods.

Number of pages: 52

Keywords: Machine Learning, Neural Networks, Capacitive Sensors

Student:	Henrik Nikolai Vahl
Supervisor:	Nils-Olav Skeie
	Håkon Viumdal
	Magnus Brastein
External partner:	Skagerak Energi AS / Beathe Furenes

University of South-Eastern Norway

www.usn.no

Summary:

Skagerak Energi AS uses the snow water equivalency (SWE) in models for predicting inflow of water to hydroelectric power plant reservoirs. Today, snow depth and density measurements are done manually and because of the remote locations this must be done at, this is a slow and expensive process. In this thesis, data from a proposed remote node system based on capacitance measurements of the snow designed in an earlier thesis at USN, is used to evaluate if capacitance measurements is a suitable replacement for the manual snow depth and density measurements. To evaluate this, data from the remote node is analyzed and various machine learning (ML) methods is used to create data driven models that estimate the snow depth and density. Based on the ML models created, it seems the capacitance based remote node can be used to estimate snow depth and density during winter, but in the spring when Skagerak Energi AS wants to measure the SWE, the snow contracts because of the warmer temperatures and loses contact with the capacitance sensors. To solve this challenge, a non-contact sensor could be considered as an alternative.

Preface

This thesis is written as the last part of my two year master program in Industrial IT and Automation at the University of South-Eastern Norway (USN), and builds on the results from a master project and master thesis from former students at USN in collaboration with Skagerak Energi AS.

Skagerak Energi AS as the external partner for this thesis has requested a solution for a cheaper and faster way of measuring SWE. In the earlier work at USN mentioned, a suggestion for what type of sensor to use has been proposed, and a prototype of a remote node has been built.

To evaluate how well this proposed solution is, I have analyzed data from sensors in MATLAB and created several ML models using different frameworks and methods. Originally, the plan was to use C# and ML.NET for the machine learning, but making neural networks is not available in ML.NET natively and the ML algorithms available gave inadequate results, so most of the machine learning has been done in python using Keras.

This thesis was worked on January-May 2021, and measurements were done from November to April 2021. In spring when the temperature starts to get warmer, it turns out that the proposed model can't be used because the snow contracts and looses contact with the capacitance sensors, so even though the models seem to work during winter, the remote nodes are not a solution to Skagerak Energi AS' problem since they need the SWE estimates in spring.

I want to thank the supervisors of this thesis, Nils-Olav Skeie, Håkon Viumdal, and Magnus Brastein for their guidance in the duration of this project, and especially Nils-Olav Skeie for collecting the data from the sensors and doing manual measurements of snow depth and density, and pictures of the prototypes for use in this thesis.

Porsgrunn, 18.05.21

Henrik Nikolai Vahl

Contents

Nomenclature

Abbreviations

Short form	Meaning
AI	Artificial Intelligence
API	Application Programming Interface
Csv	Comma Separated Values
EDGE	Enhanced Data rates for GSM Evolution
GPR	Gaussian Process Regression
GPRS	General Packet Radio Service
GUI	Graphical User Interface
LoRaWAN	Long Range Wide Area Network
ML	Machine Learning
MSc	Master of Science
NIPALS	Nonlinear Iterative Partial Least Squares
NN	Neural Network
ONNX	Open Neural Network Exchange
PCA	Principal Component Analysis
PC	Principal Component
PLS-R	Partial Least Squares Regression
SWE	Snow Water Equivalent
USB	Universal Serial Bus
USN	University of South-Eastern Norway

Symbols

Symbol	Description	Unit
h_s	Snow depth	cm
ρ	Snow density	g/cm^3
ρ_w	Water density	g/cm^3
$Mass_{Sample}$	Mass of snow sample	kg
$Volume_{Sample}$	Volume of snow sample	m^3
r	Radius of measurement pipe	m
h	Snow depth in measurement pipe	m

1 Introduction

This introduction chapter provides information about the motivations for this thesis as well as the work this thesis is a continuation of, the objectives this thesis sets out to solve, and an overview of the report structure.

1.1 Background

This master thesis is a continuation of a MSc group project [1] and MSc master thesis [2] from 2019 and 2020 respectively, where a sensor for measuring snow depth and snow density was designed.

The external partner for this project, Skagerak Energi AS, is a power producer in southern Norway that delivers large amounts of clean renewable energy from hydroelectric power plants [3]. In places like Norway with cold winters large amounts of water is stored as snow during winter, and as the snow melts in the spring, it finds its way into the reservoirs of hydroelectric power plants. The hydrologist group at Skagerak Energi AS develops models to predict the inflow of water for energy production at hydroelectric power plants. Snow melt will highly affect the inflow to the reservoirs and is an important input to these prediction models.

Presently, Skagerak Energi AS is making manual measurements of snow depth and snow density at the end of winter. This is an expensive and time-consuming process as these measurements often have to be done in remote locations not reachable by car.

USN has worked on finding a small, low cost, and energy efficient alternative through a MSc project and MSc master thesis already. In the 2019 group project, several solutions were considered and tested to find a good solution [1]. The proposed solution from the project was a system based on capacitance measurements. This proposed system was further developed in a 2020 master thesis. In the 2020 thesis, a prototype of the system was designed and built. Some measurements were done to test that the measurements worked and partial least squares regression (PLS-R) models for snow depth, snow density, and SWE were created [2].

1.2 Objectives

Skagerak Energi AS is looking for a measurement system that can be deployed remotely to measure snow depth and snow density and transmit information to their data center to be used for inflow prediction. A measurement system based on capacitance was developed in 2020 *[2]* and in this thesis, data from this system will be analyzed and used to create models based on machine learning to evaluate how well this system works. In the 2020 master thesis, PLS-R models were created, but the models were unsatisfactory. There was limited snow in 2020 to take measurements, so the data set used was quite small and this may have had an effect *[2]*.

In this thesis, new data will be used together with machine learning methods will be created to evaluate if the system proposed and designed earlier at USN is a suitable replacement for the manual measurements Skagerak Energi AS is currently doing.

The snow water equivalent (SWE) that is used in Skagerak Energi AS' prediction models, is calculated from the snow depth and snow density. Therefore, with models based on the new measurement system that can estimate both snow depth and snow density, the SWE can also be estimated using equation (1.1).

$$SWE = h_s \frac{\rho}{\rho_w} \tag{1.1}$$

Where h_s is the snow depth, ρ is the snow density, and ρ_w is the density of water, measured as 1 g/cm^3. SWE is measured in centimeters of water.

1.3 Report Structure

This report consists of a short overview of some of the methods that are used today, when measuring SWE, as well as a description of the system that the models in this project is based on. Then a theory chapter introduces different machine learning frameworks and concepts before data is analyzed and presented in chapter 4. Here the data that is used to train the machine learning models is analyzed, to see if there are any correlations that a machine learning model could take advantage of. Finally, the model creation and results are described in chapter 5, before we are discussing the results of the models and the viability of a continued development of the system.

2 System Description

The data used in this thesis to evaluate the usage of capacitance sensors for snow depth and snow density, comes from primarily two prototypes built the autumn of 2020, based on the prototype created in the spring of 2020 MSc thesis [2]. This chapter describes the prototypes that were used to collect the data, and used to create the machine learning models for snow depth and snow density. Furthermore, we give a short overview of other ways of measuring SWE used today.

2.1 Snow Water Equivalent Sensors

The system used in this thesis is based on capacitance measurements, but there are other ways to measure SWE available today. This section gives a short overview of some of these sensors.

2.1.1 Snow Pillow

Snow pillows are pillows filled with an antifreeze liquid that is placed on the ground before the first snow fall. As snow accumulates on the pillow, the pressure in the pillow increases, and the SWE can be calculated. Snow pillows are one of the cheaper options when it comes to monitoring SWE, but it comes with some challenges.

The ground where the pillow is placed must be prepared before installation as it has to be levelled and well drained. There is also a possibility that the pillow may puncture, for instance because of damages due to animals. Effects like bridging causes some of the weight of snow on top of the snow pillow to be transferred to the surroundings, and can make the measurements uncertain. When installed in remote locations, there is also the challenge of transporting several 100s liters of anti-freeze liquid to fill the pillow in difficult and remote terrain [4].

2.1.2 Snow Scale

Snow scales are large platforms with sensors to measure the weight of the snow on top of them. Like snow pillows, it is necessary to prepare the ground as the snow scale should be levelled and well drained. The groundwork for installing a snow scale is often more comprehensive than a snow pillow and may be a challenge in remote locations if an excavator is needed. They are more expensive than snow pillows, but since there is no anti-freeze liquid used, there is nothing that can leak and cause any environmental damage. Also, animals will not be able to cause any damage. Furthermore, the size of snow scales can be changed easier than a snow pillow, when a larger area needs to be covered. In addition, the latter also reduces the risk of bridging [4].

2.1.3 Gamma Attenuation Sensor

The CS725 gamma attenuation sensor by Campbell Scientific is designed to directly measure SWE in a snowpack. The basis of the technique used, is that the ground underneath the snow contains some natural radioactive elements. This implies that, there is a certain amount of electromagnetic energy that always radiates. The CS725 sensor measures the electromagnetic energy from the ground, and when there is water between the ground and the sensor, the gamma rays from the ground are attenuated. Based on the attenuation level, the SWE can be calculated [5].

The major advantage of this approach is that the sensor can be connected to a datalogging device, so when placed in a remote location, the data could be transmitted digitally, and reduce the need for visiting the site. The attenuation isn't affected by what type of snow it is, and hence, the sensor works for any snow or ice. When mounted at a height of 3 meters, the sensors measurements cover a quite large area; 50-100m^2. The sensor is dependent on enough potassium and thallium present in the ground, and performance can thus be site specific [4]. In addition it is more expensive than snow pillows and snow scales and has an upper limit of 600mm of snow, though this restriction is dependent on the ground radiation [4].

2.2 System description

There are three different prototypes used to collect data for this thesis. They are all based on the same sensors but vary slightly and are placed at different locations.

2.2.1 Prototypes

The prototypes measure snow depth and snow density based on capacitive soil measurement sensors. Five of these sensors are mounted on each prototype, at different heights to be able to measure the capacitance at different snow depths. Variations of snow density should change the output of the sensors, and if the variations are consistent enough, a model estimating snow density can be created using machine learning. To find the SWE, the snow depth is also required. We will develop a model estimating the snow depth, based on the same measurements, although this might prove challenging, at least outside of just being able to estimate the depth based on just the height of the highest moisture sensor covered with snow and setting the snow depth equal to the sensor height from the ground.

The three prototypes used are:

1. This prototype is located at Sjusjøen and was installed before the first snow fall winter of 2020. The location is well suited for measuring, as the temperatures are reliably sub-zero during the whole winter, making it a good place to get realistic density measurements, as snow accumulates naturally on the sensors. The location is a much more snow safe are than Porsgrunn where USN in located. The prototype is built using a light gray plastic pipe with a height of 2 meters and a diameter of 68 mm. The soil moisture sensors are mounted on the same side of the pipe at heights of 10, 30, 50, 80, and 110 cm. In addition to the soil moisture sensors, there is a temperature sensor

mounted at the top of the prototype. Finally, there are two pressure sensors. One located at the base measuring snow pressure, but possibly due to the membrane being too small the measurements do not vary much. In addition, the small size might face challenges with bridging. The second pressure sensor is inside the pipe and measures atmospheric pressure [6]. Figure 2.1 shows what the prototype looks like. The picture is from December 2020 and shows what it looks like when the first three moisture sensors are covered.

2. Prototype #2 is built differently than the two others in that instead of a pole, it is a plastic box with dimensions 20x15x7.5 cm. This prototype has been used in both Sjusjøen and Porsgrunn. This prototype has four capacitance sensors as well as a temperature sensor and an ultrasonic sensor which data is not considered in this thesis. The capacitance sensors on this prototype varies from the others. One of the sensors has a Veroboard as a capacitance plate with a bigger area, 16x6cm. One has two standard plates of 6x2cm. The remaining two has standard 6x2cm plates. The capacitance sensors are mounted on different the sides of the box [6].
3. Prototype #3 is similar to prototype #1, but the pipe is black, and the height is somewhat lower, 1.8 m and the diameter is 110 mm. It is located in Porsgrunn. The soil moisture sensors are mounted at heights of 23, 36, 64, 92, and 122 cm [6].

Figure 2.1: Prototype #1 in Sjusjøen December 2020

2.2.2 Data acquisition

The prototypes have an Arduino in it that is connected to all the sensors, taking measurements every minute, transmitting them to a Windows PC every minute via USB where the data is logged. The sensor values along with an ID for each prototype is stored in a .csv file every thirty minutes. The log file also contains manual measurements of snow depth and snow density. These manual measurements were done by Nils-Olav Skeie by inserting a pipe into the snow taking a sample and using equation (2.1). The snow depth is measured using the measuring tape-scale on the prototype.

$$Snow\ Density = \frac{Mass_{Sample}}{Volume_{Sample}} = 1000\frac{Mass_{Sample}\ [g]}{\pi r^2 h\ [cm^3]} \tag{2.1}$$

Where $Mass_{Sample}$ is the mass from the sample taken with the measurement pipe, r is the diameter of the measurement pipe, and h is the snow depth where the sample is taken.

The manual measurements are added to the .csv file along with a flag, marking whether the measurements are valid for supervised learning in a machine learning model. After a manual measurement is added, it is assumed that the snow depth and density is relatively consistent over the next five hours, ten samples, and automatic measurements in this period is marked for supervised learning. Figure 2.2 shows an example of a .csv file with the header showing the contents of the different fields.

```
TimeSample;NodeAdr;CapSens#1(Bot)(mV);CapSens#2(mV);CapSens#3(mV);CapSens#4(mV);CapSens#5(mV);
PressSens#1(mBar);PressSens#2(Atm:mBar);TempSens#1(degC);ManSnowHeight(cm);ManSnowDensity(kg/m3);
ManLogInfo;SupervisedLearningFlag;
12.02.2021 11.14;01;2729,5;2749,0;2724,6;2710,0;2710,0;997,2;969,7;-21,7;000;000;;OFF
12.02.2021 11.44;01;2735,6;2755,1;2725,8;2722,2;2719,7;996,8;973,4;-17,9;000;000;;OFF
12.02.2021 12.15;01;2739,3;2756,4;2729,5;2728,3;2727,1;999,6;973,4;-15,3;000;000;;OFF
12.02.2021 12.45;01;2739,3;2758,8;2728,3;2732,0;2734,4;998,4;974,0;-14,0;000;000;;OFF
12.02.2021 13.15;01;2738,1;2756,4;2729,5;2732,0;2729,5;998,0;972,2;-14,3;000;000;;OFF
12.02.2021 13.45;01;2739,3;2758,8;2729,5;2739,3;2739,3;999,2;974,6;-11,4;000;000;;OFF
12.02.2021 14.16;01;2749,0;2767,4;2736,9;2756,4;2753,9;1000,8;977,7;1,9;000;000;;OFF
12.02.2021 14.46;01;2745,3;2763,7;2734,4;2757,6;2753,9;1000,0;977,1;,4;81;233;tørr og lett snø;ON
12.02.2021 15.16;01;2741,7;2761,3;2734,4;2753,9;2751,5;999,6;974,6;-2,5;81;233;tørr og lett snø;ON
12.02.2021 15.46;01;2735,6;2756,4;2729,5;2739,3;2734,4;998,8;973,4;-10,1;81;233;tørr og lett snø;ON
```

Figure 2.2: Part of the contents of a data file from prototype #1

The .csv file contains information about the date and time of the measurement, along with the node ID, sensor measurements, the manual snow depth and snow density measurements, comments about the manual measurement, and the supervised learning flag.

2.2.3 Capacitive Soil Moisture Sensor

The capacitive soil moisture sensors are, as the name suggests, created to measure soil moisture. For its intended use, the sensor is inserted into soil, and since capacitance varies with the permittivity of the dielectric medium. And the permittivity of soil is dependent of, among other things, moisture, the output of the sensor will depend on the soil moisture [7]. For the nodes used in this thesis, the dielectric medium the soil moisture sensors are interacting with is air and snow instead of soil with the variations in permittivity when the sensors are covered in snow are large enough to measure changes in the density of the snow. Figure 2.3 shows the type of capacitive soil moisture sensors used in the nodes. The probe consists of a coplanar capacitance that is connected to the output of a 555 timer as part of a low pass filter. This creates a sawtooth waveform where the peak to peak depends on the dielectric medium of the probe [7]. A peak detector is then used to provide the output of the sensor.

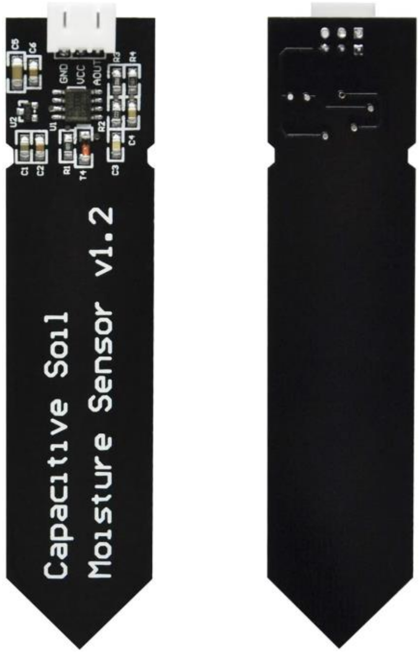


Figure 2.3: Capacitive soil moisture sensor [8]

2.2.4 Data communication

The measurements that are collected by the nodes must be transmitted to the data center to be used with models estimating the SWE and further to be used in the models for the inflow of water to the reservoir. Considering the remote locations these nodes will be placed in, some form of wireless communication protocol is preferred. There are primarily two things to take into consideration when it comes to choosing a protocol for this system, the first being range and the second power consumption. To get the data to the data center, the nodes would need internet access.

One option is to take advantage of the existing infrastructure of the mobile network. In Norway, the coverage is quite extensive, but whether this is an option or not depends on the coverage in at the node locations. It would be a good solution as no new infrastructure would be built, and access to the internet would be available straight from the node. The mobile network protocols GPRS, EDGE and newer ones have a high power consumption which could pose a challenge [9].

Using a satellite internet service might work, but these solutions are generally made for household use and are probably not energy effective.

Other options include long range protocols such as LoRaWAN that has a longer range than the mobile network protocols and is more energy efficient [10], but unless there happens to be a LoRaWAN network in the area, the infrastructure would have to be built. There is at least one open LoRaWAN network available in Norway [11], but without knowledge of where the remote nodes will be placed, it’s hard to tell if that specific network will be of use.

Assuming that there is GPRS or EDGE coverage at the node sites, one of those protocols would probably be best to avoid having to set up extra infrastructure and having increased costs with both building and maintaining it.

3 Theory

This chapter gives a short overview of some of the machine learning platforms and methods used to create data driven models with machine learning.

3.1 Machine Learning Frameworks

TensorFlow

TensorFlow is a free open source machine learning (ML) platform created by Google, that let you build, train, and deploy machine learning models [12]. The platform is supporting several languages through APIs, but the only language that is covered by the API stability promises is python. The other languages with official APIs are JavaScript, C++, Java, Go, and Swift. In addition, there is support for other languages like C#, Haskell, Julia, and MATLAB through the community [13].

Keras

Keras is a commonly used deep learning API, the high-level API, of TensorFlow. Keras comes with a lot of different layers, optimizers, and metrics that makes creating and training a ML model quickly possible. But if the built-in layers don't fit your network it's possible to create custom layers. There are also some data processing built in, like image, time-series, and text data [14].

The available layers allow you create a wide selection of neural networks and there are several core, convolution, recurrent, and pooling layers to choose from.

Keras is the most adopted deep learning solution and it is used at Netflix and Uber, as well as scientific organizations like CERN and NASA [15].

MATLAB

MathWorks MATLAB deep learning toolbox lets you build, train, and deploy machine learning models through MATLAB. There are apps that can be used to train models without having to code anything as the code is generated for you. It is also possible to write your own code. MATLAB is the only framework discussed here that isn't free. The licence for MATLAB is quite expensive [16] and few industry businesses use it, but MATLAB is used by a lot of educational institutions as well as by research institutions.

The most popular classification, regression, and clustering algorithms are available. And code for C or C++ can be generated for embedded and high-performance applications [17]. MATLAB also includes automated machine learning which automatically pre-processes the data, extracts features, and identifies the best performing model [18].

The MATLAB models can be exchanged with TensorFlow and TensorFlow models can be imported into MATLAB.

ML.NET

ML.NET is Microsoft's free machine learning framework for C# and F#. It supports both supervised and unsupervised learning of different machine learning algorithms has algorithms for different problems like binary classification, regression, anomaly detection, clustering, and matrix factorization. ML.NET supports transfer learning where you can import a pre trained neural network model created in TensorFlow or following the ONNX standard and use and train them but does not support creating neural networks by itself [19]. Microsoft does have another open source library called Microsoft Cognitive Toolkit, that supports creating neural networks in C#, but development of this library was stopped in 2019 [20].

scikit learn

scikit learn is a free, open source machine learning library for python. It provides several algorithms for supervised and unsupervised learning and for different problems. It also provides tools for every step in the machine learning process, all the way from pre-processing to a finished model [21]. Scikit learn is used by several companies, for instance Spotify, and the development is financially supported by among others, Microsoft, Fujitsu, and Columbia University [22].

Pytorch

Pythorch is a machine learning framework from Facebook that is open source and free. There are two languages available, python and C++. It is a deep learning library, so it can be used to create and train neural networks [23]. It supports creating and training neural networks, and contains several libraries with tools for handling things like, audio, text, and computer vision for use in neural networks [24].

ONNX

ONNX, or Open Neural Network Exchange, is an open standard for machine learning models from The Linux Foundation created to enable models to easily be transferred from one framework to another [25]. Indeed all the frameworks mentioned above support exporting and importing ONNX models. ML.NET for instance does not support creating neural networks natively, but a neural network model created with TensorFlow can be imported in the ONNX format and be used with ML.NET.

3.2 Machine Learning Methods

There are several machine learning methods available depending on the problem that needs to be solved. As shown above, there are several machine learning algorithms and neural networks available. When training these models there are primarily three different approaches used, supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, both the inputs and outputs of the training data is known. The training of the model is based on the error between the model output and the expected output based on the training data for a given input. Regression and classification problems are usually done using supervised learning. In unsupervised learning, only the input data is provided for training. This is often used to find patterns in data like clustering and anomaly detection problems [19]. Reinforcement learning, uses a reward/punishment system where the model interacts with an environment and gets rewarded, based on the action taken and the model tries to maximize the reward. One example of where this could be used, is in a Tic Tac Toe AI, where the reward depends on how optimal the next move chosen by the model is.

In this thesis, supervised learning is the training method that will be used. In the models, where manual measurements of snow depth and density will be used as expected outputs for the models during training, this approach works well Reinforcement learning requires some form of interaction with the environment and is thus not a viable method for creating models for snow depth and snow density.

Models created using machine learning are data driven models. The models are trained using one of several algorithms and methods based on data provided during the training. Data is provided as the input in the model, and the model is modified based on the output according to the algorithm used. To be able to produce a good model it is therefore necessary to have enough, and diverse enough, data during training, as well as using the correct training algorithm. For most algorithms, including neural networks, it is generally best to standardize the input data in some way to avoid larger values from having a disproportionate effect on the output. In some cases, other pre-processing is also done like principal component analysis or removing outliers from a data set.

Artificial Neural Network

Artificial neural networks, or just neural networks (NN), is a machine learning method based on the human brain. To give a short overview, they consist of several artificial neurons connected to each other in layers, that produces one or more outputs based on one or more inputs. The neurons can have several inputs and each input has a weight expressing the strength of the connection between the neurons [26]. Each input for the neuron is multiplied with its weight, and all are added together with a bias value. Usually this sum is used with an activation function, depending on what the network is developed for. A neural network must have inputs and an output layer, but between them there can be any number of hidden layers. Figure 3.1 shows an illustration of a neural network with two inputs, two hidden layers with four neurons, and one output neuron with the connections between them.

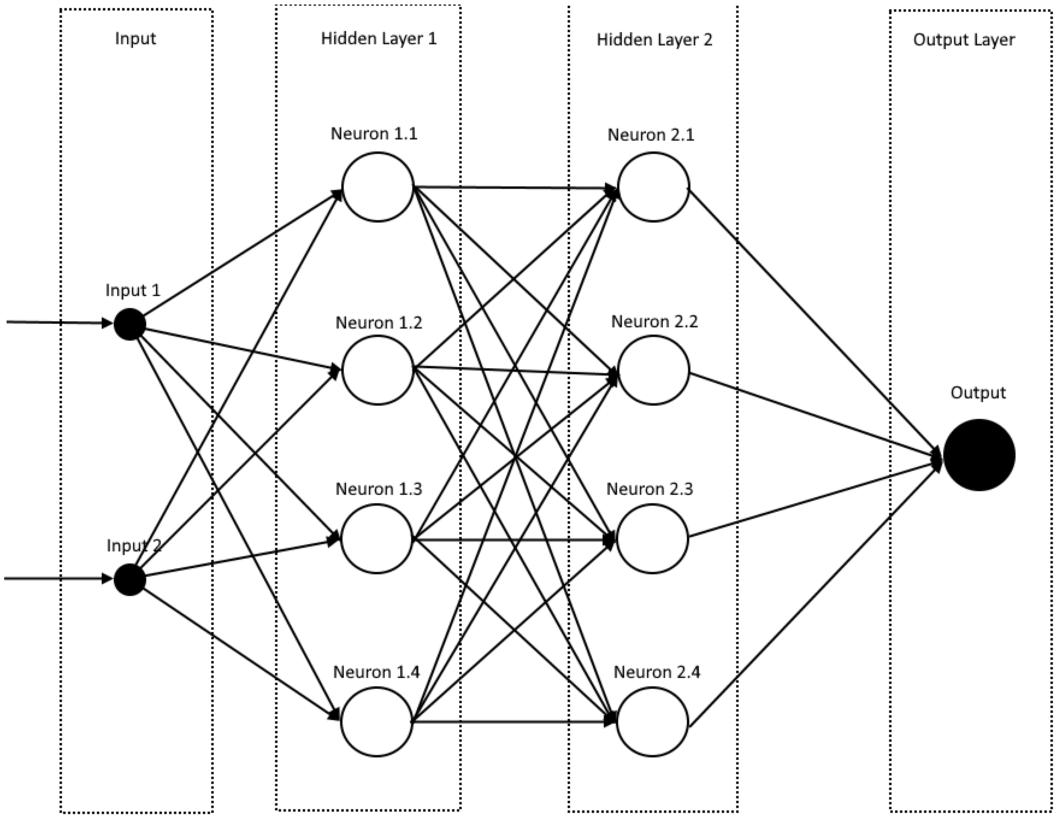


Figure 3.1: Illustration of a simple neural network structure

During the training of a neural network, the weights and biases of the network is changed. In the case of supervised learning, the change is based on the error between the output produced by the network, and the expected output given by the training data. There are different types of neural networks available, depending on the application at hand, like feedforward neural networks, as shown in figure 3.1, or recurrent neural networks where the output of a neuron is connected to its input.

Machine Learning Training Algorithms

There are several popular machine learning algorithms available in the different frameworks described earlier. Which algorithm to use depends on the problem the model needs to solve. Most of the frameworks provide and allow you to use mostly the same algorithms, though some have more than others. For instance, for regression problems ML.NET provides the following training algorithms [19]:

- -FastTreeRegressionTrainer
- -FastTreeTweedieTrainer
- -FastForestRegressionTrainer
- -GamRegressionTrainer

- -LbfgsPoissonTrainer
- -LightGbmRegressionTrainer
- -OlsRegressionTrainer
- -OnlineGradientDescentTrainer
- -SdcaRegressionTrainer

In addition, algorithms for other problems like binary classification, multiclass classification, ranking, clustering, and anomaly detection are available.

4 Data analysis

4.1 The Data Set

The three remote node prototypes have gathered data at two different locations, Sjusjøen and Porsgrunn. Prototype #1 is located at Sjusjøen, and is the prototype that received the most natural snow and the largest amount of good measurements. Because of this, the measurement data from prototype #1 is used for most of the data analysis and for the later modelling. Table 4.1 gives an overview of the measurements that are available from the different prototypes.

Table 4.1: Prototype sensor and measurement periods

Prototype #	Sensors	Period
1	5x soil moisture sensors Temperature Atmospheric pressure	November 22-27, 2020 December 28-31, 2020 February 12-13, 2021 March 30 – April 2, 2021
2	4x capacitance sensors Temperature sensor	February 12-13, 2021 (Sjusjøen) February 15-18, 2021 (Porsgrunn)
3	5x soil moisture sensor Temperature sensor	February 18-28, 2021 March 1-27, 2021

There are some challenges with this data set that restricts the possibilities of creating a good machine learning model. As temperatures rise above the freezing point, the snow around the nodes recede, and contact between the soil moisture sensors and the snow is lost. This is presumably because of the snow crystals shrinking in size as the temperature goes up. Figure 4.1 shows how the snow has receded from the moisture sensors on prototype #1 in March 2021. In the figure all the snow around one of the soil moisture sensors is melted so there is no contact left. The same happened with prototype #3 in February and March. Prototype #3 also has a challenge because of the color of the pipe as the black plastic can warm up in the sun an accelerate melting. Because of this, the measurements from prototype #1 in November-February is the most reliable data to analyze and create a model from. Prototype #2 was built to test an ultrasonic sensor as well as testing capacitive sensors with larger capacitive plates. Unfortunately, in the measurements done to test the larger capacitive plates, something must have gone wrong because the output voltages vary too little.

Figure 4.1: Snow contracts and recedes from the moisture sensor when the temperature gets too high

The temperature at the location of prototype #1 was below zero from the end of November and throughout the measurements performed in February, except for two measurements done in this month. All measurements from November 2020 were without any snow, while the December and February measurements all have snow at varying depths and densities.

There is also some uncertainty in the accuracy of some of the density measurements. In December and February the assumption was that the density of the snow would be quite similar and stable over time, but when more frequent measurements were done in March and April the density varied quite a lot even during a period of a few hours. Possibly because of temperature changes. But the temperatures were also higher at the time, the variations may not be as frequent with sub-zero temperatures.

4.2 Data Analysis

The prototype #1 November-February data is the most promising data considering the challenges with snow melting close to the sensors when the temperature gets above 0℃. Figure 4.2 shows the variations of the soil moisture sensor outputs over time in this period. The sensors are ordered from the bottom up, so moisture sensor #1 is at the bottom and #5 is at the top.

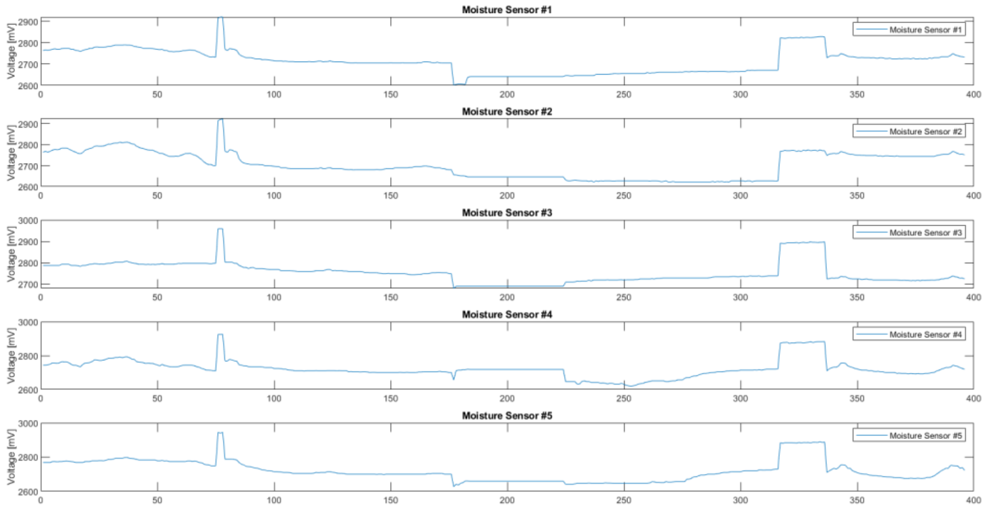


Figure 4.2: Variation of moisture sensor outputs over time

From the figure there are two sharp rises in output that happens with all five sensors at the same time at sample points 76 and 318. Looking at the other variables as well, this turns out to be some sort of error that also affects the two pressure sensors. In the same sample points as these spikes, the atmospheric pressure sensor shows a value of -433 mBar for instance. The cause of the error is unknown, but the temperature sensor seems to be unaffected. All the sample points that contain erroneous measurements are removed for the rest of the analysis.

For a machine learning model to be able to estimate the density and depth of the snow, some sort of correlation between them must be present. Figure 4.3 compares the output of the bottom soil moisture sensor to the other parameters measured, including the manual measurements.

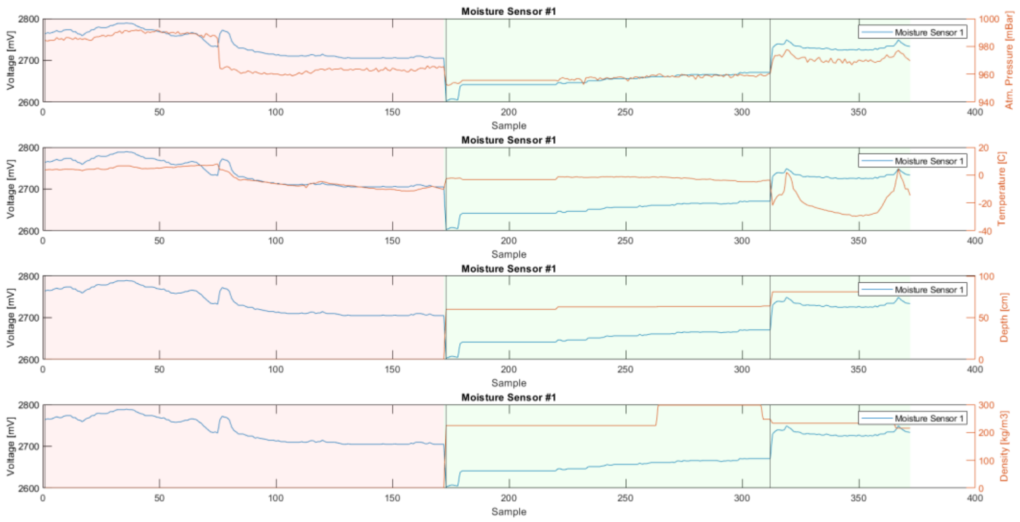


Figure 4.3: Comparison of moisture sensor #1 output over time with temperature, atmospheric pressure, snow density, and snow depth

The background in the figure is coloured red when the sensor is not covered by snow, and green when it is. The vertical lines indicate the start of a new period, so the first line is the start of the December samples and the second the February samples. In terms of numbers samples 1-172 are November, 173-312 is December, and 313-372 is February. The output varies by about 100 mV even when not covered and seems to be loosely correlated with both temperature and atmospheric pressure. Comparing this with figure 4.4 where sensor #5 is shown it look like the temperature is strongly correlated at least in February where sensor #5 that never gets covered in snow follows the change in temperature.

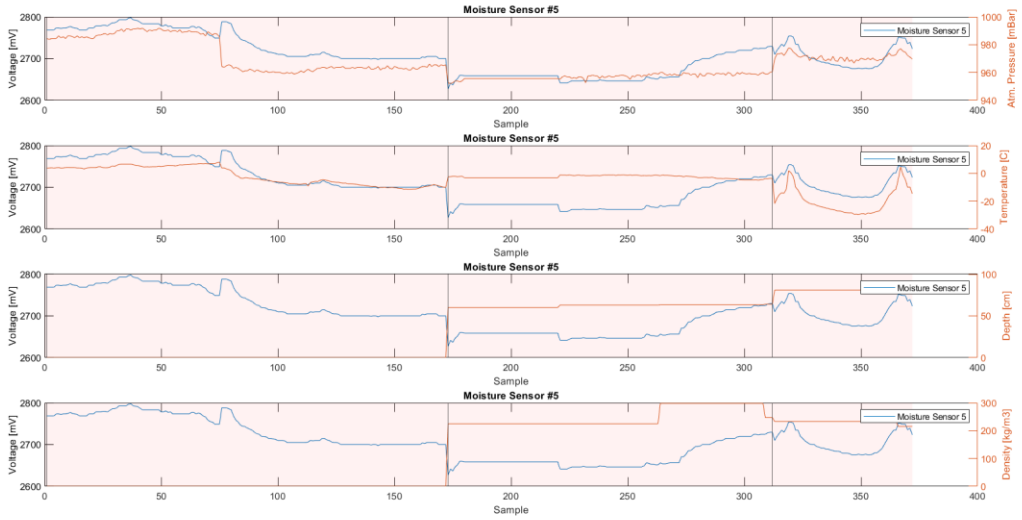


Figure 4.4: Comparison of moisture sensor #5 output over time with temperature, atmospheric pressure, snow density, and snow depth

Looking at the same data for sensors #2, #3, and #4, that can be seen in appendix D, the correlation between temperature and sensor output seems to be affecting the sensors higher up on the pole more, than the lower ones. Probably due to the temperature changes being lower further down in the snow.

Another observation from figure 4.3 and 4.4 is that there seems to be little correlation between snow depth, snow density, and sensor output.

4.2.1 Principal Component Analysis

Just plotting different features of the data against each other can give an indication of the correlation between the different features, but there are seven variables and by using principal component analysis (PCA) the correlations become more clear and easy to spot. PCA is a method for decomposing the data based on the variance in the data set by substituting the original variables with principal components [27]. In this thesis, the software 'The Unscrambler X' is used for the PCA, and the NIPALS algorithm to calculate the principal components. The moisture sensor variables all have values in the thousands, so if the raw data is used in the PCA, those are almost guaranteed to have the biggest variance. Usually when the variance varies a lot between variables, it is common practice to scale all variables, to prevent that a variable doesn't have the largest, or smallest variance simply because all the measurements are a factor of 100+ compared to the other variables. In this case the difference in variance between the different variables is quite large, from about 3000 to 80. Hence, the data will be scaled. The principal components are found by finding the direction of maximum variance in the data matrix, and that are orthogonal to each other. The first PC1 is along the largest variance, PC2 is orthogonal to PC1 and along the second largest variance, PC3 is orthogonal to PC1 and PC2 and along the third largest variance and so on [27].

By aligning the PCs in order of variance, the variance in the data set can be explained using fewer variables. Figure 4.5 shows the explained variance of the principal components. From the original seven variables, only four principal components are needed to explain 96% of the variance in the data. PC1 explains 72% and together with PC2, 83% of the variance is explained.

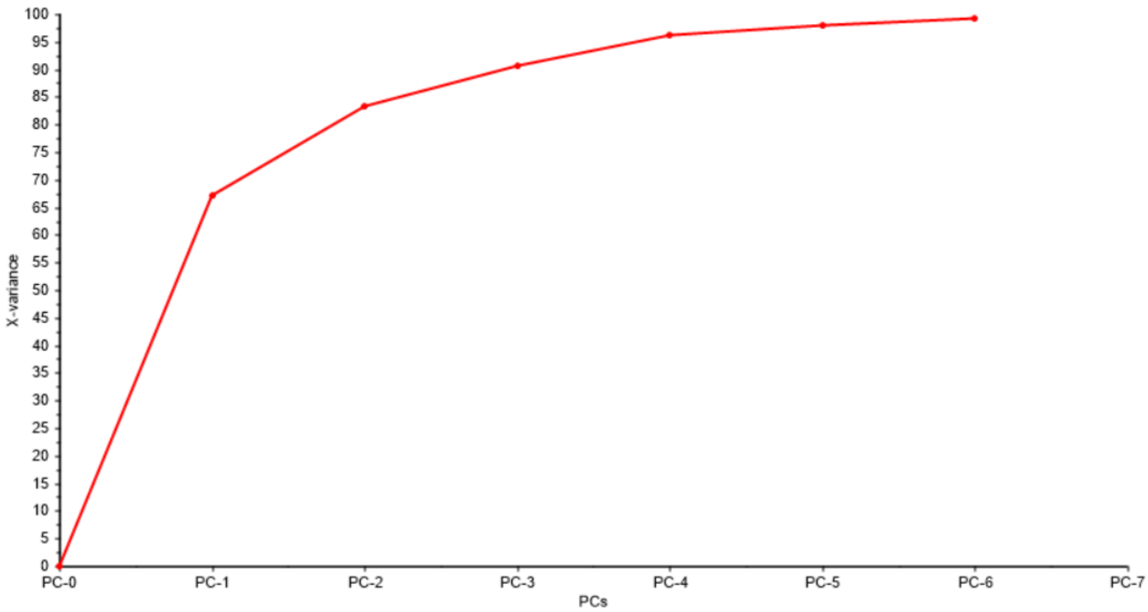


Figure 4.5: Explained variance of the principal components

In figures 4.6 and 4.7 the scores and loading plots, respectively, for PC1 and PC2 are shown. In these plots, PC1 is along the x-axis and PC2 along the y-axis. From the scorings-plot in figure 4.6 three or four groupings can be seen. The samples are coloured based on which month they are from, and for the most part, the groupings seem to be based on when the samples are from. The exceptions are some samples from February, that could be grouped with the one of the November groups. A closer look at the samples in question, shows that these are the samples that can be seen in figure 4.3 and 4.4 where the temperature spikes up at sample 319 and 367 so it makes sense that they would be similar to the November measurements based on the plots from figure 4.3 and 4.4. The two groupings from the November samples stems from the samples that were removed because of bad measurements.

The group in the lower left in figure 4.6 are the first 80 samples, all the samples before the faulty measurements.

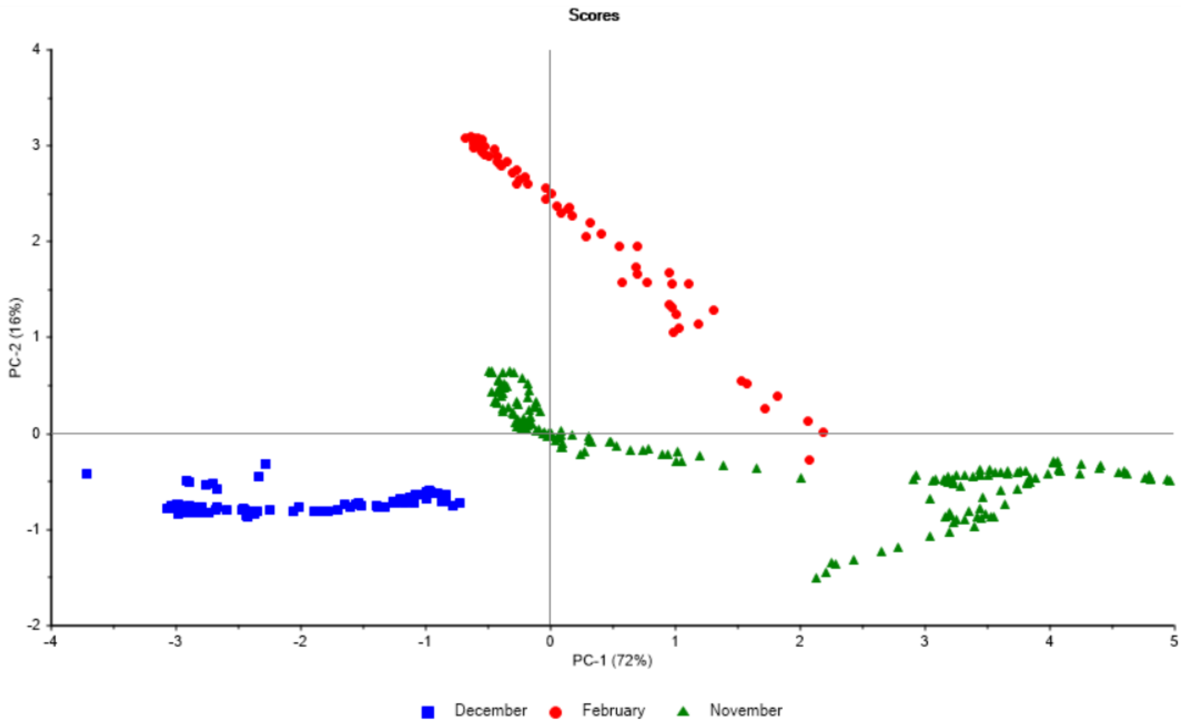


Figure 4.6: Score plot of principal component 1 and principal component 2

In the loading plot in figure 4.7, the correlation between the different variables can be seen. All the moisture sensors, S1, S2, S3, S4, and S5, S1 being the lowest sensor and S5 being the highest, are strongly correlated with the atmospheric pressure. PC1 is explained by these five variables, but S4 and S5 are most important in PC1. The temperature is contributing most of PC2, the other variables have quite low loadings there.

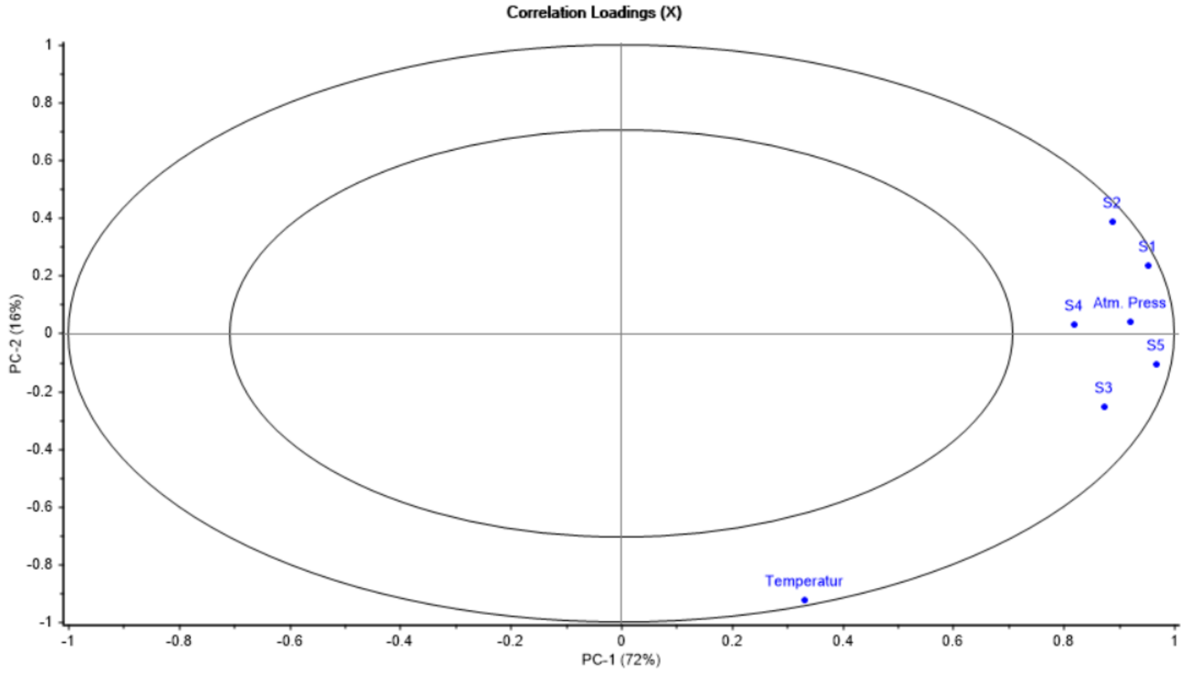


Figure 4.7: Loading plot of principal component 1 and principal component 2

Based on the PCA analysis of the November, December, and February data, the moisture sensors are correlated with the atmospheric pressure. By looking at only the December and February data, when there actually is snow present, the PCA looks very different. Figure 4.8 shows the loadings for PC1 and PC2. In this case, temperature is negatively correlated with most of the other variables. With this data, S1 and atmospheric pressure contributes most and are very strongly correlated. PC2 is mainly S3 and in figure 4.9 that shows PC1 and PC3, S4 is the main contributor to PC3.

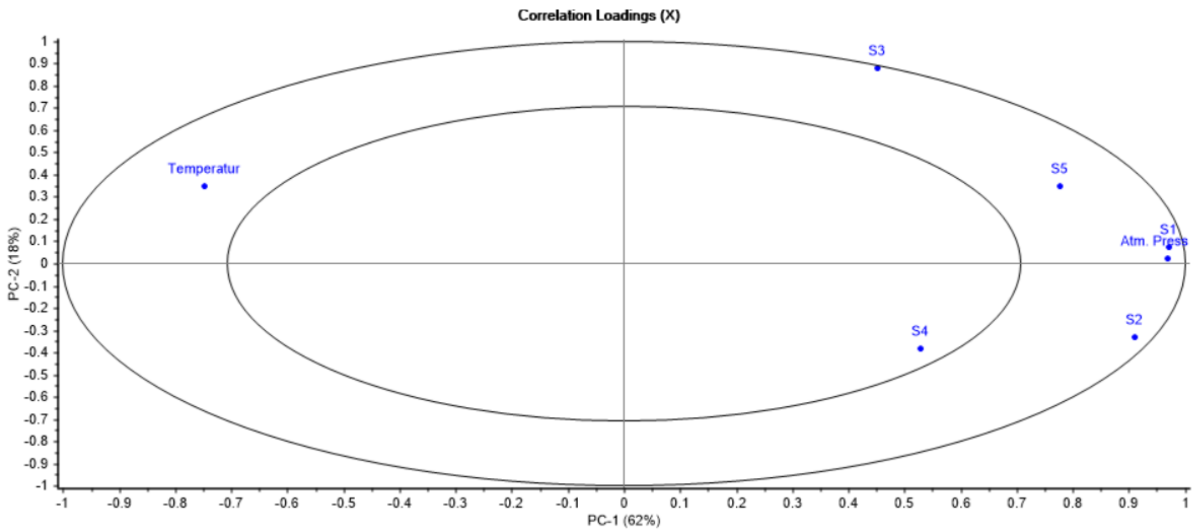


Figure 4.8: Loading plot for principal component 1 and principal component 2 using December and February data

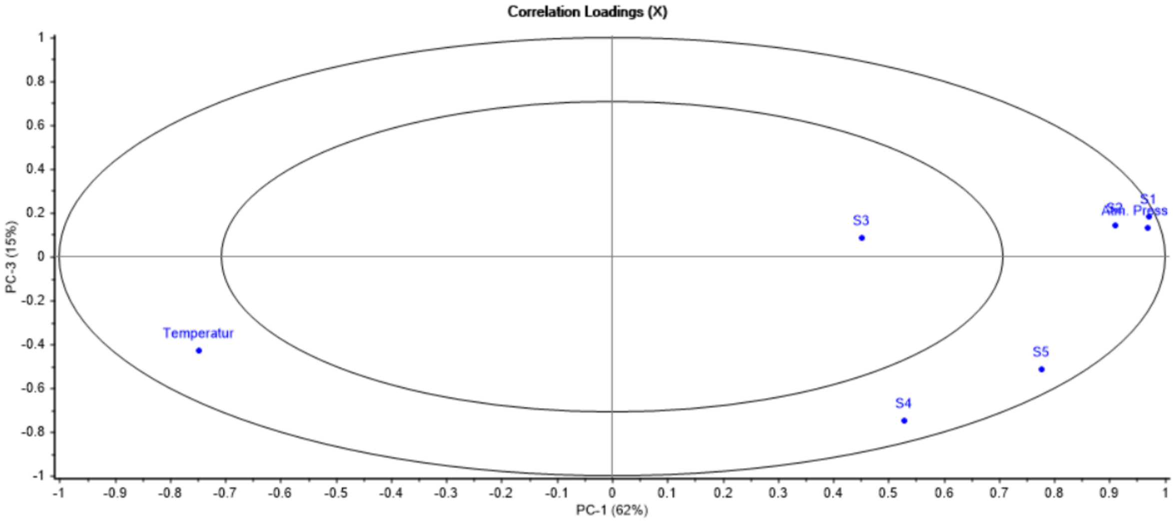


Figure 4.9: Loading plot for principal component 1 and principal component 3 using December and February data

Hence, it seems that when it's colder, or when there is snow covering the sensors, temperature is more correlated with the moisture sensors. Texas Instruments is one of the companies that produce the 555 timers, among others the TLC555 that seems to be used in the soil moisture sensors. According to a response from a Texas Instruments engineer, the TLC555 circuits can be affected by temperature [28].

4.3 Rain

The snow sensor is designed to be deployed all year round, and though the measurement of snow is its main purpose, being able to detect rain would also be of interest. A dataset made with prototype #3 in Porsgrunn, accompanied with measurements of precipitation and humidity from November 2020 is available to see if that is possible. The measurements are available for the period 22.11.20-27.11.20. The precipitation and humidity measurements were done at a higher sampling rate than the capacitance measurements. To get an even amount of sampling points, the data from prototype #3 was padded using linear interpolation to fill in the missing samples. Some samples were removed due to the same issues as with prototype #1, where some measurements were much higher than the rest, and the atmospheric pressure was negative. There was little rain in the timeframe where the measurements were done, and in figure 4.10 the soil moisture sensors values are plotted against the precipitation.

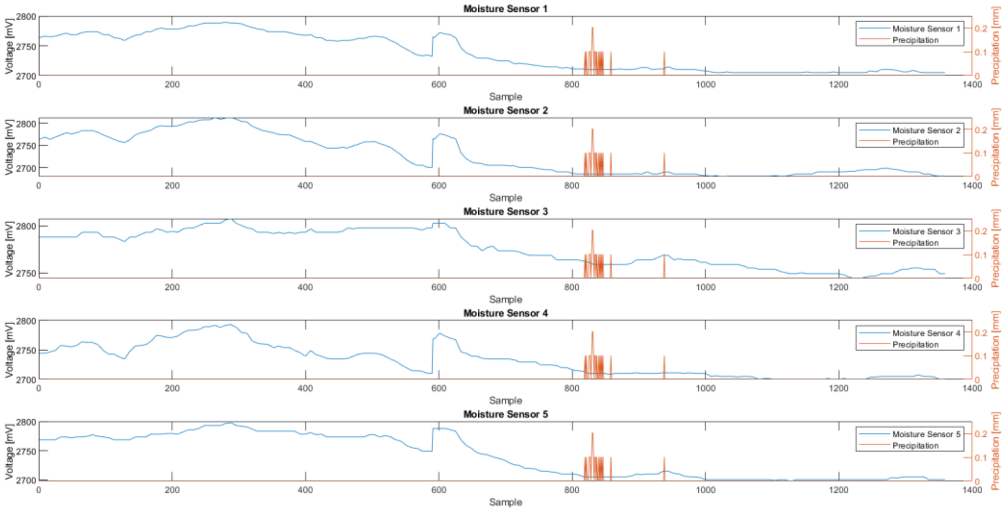


Figure 4.10: Moisture sensor outputs compared to precipitation

As seen in figure 4.10, there seems to be no significant change in the capacitance measurements in the period where there is precipitation measured. There were few samples with precipitation and in those that had, there were small amounts of it. In figure 4.11 the moisture sensors are plotted against the temperature. It shows that there is a much higher correlation between the temperature and moisture sensor output than precipitation.

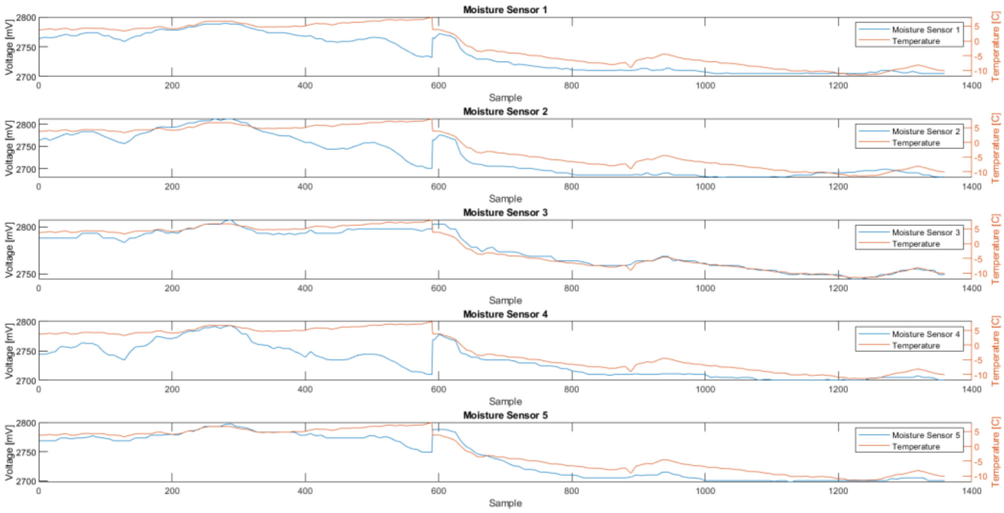


Figure 4.11: Moisture sensor outputs compared to temperature

The relationship between humidity and the moisture sensor outputs can be seen in figure 4.11. Between samples 150 and 250 it looks like there might be some inverse correlation, but later between sample 1000 and 1200 where the humidity falls again, the moisture sensors seem unaffected.

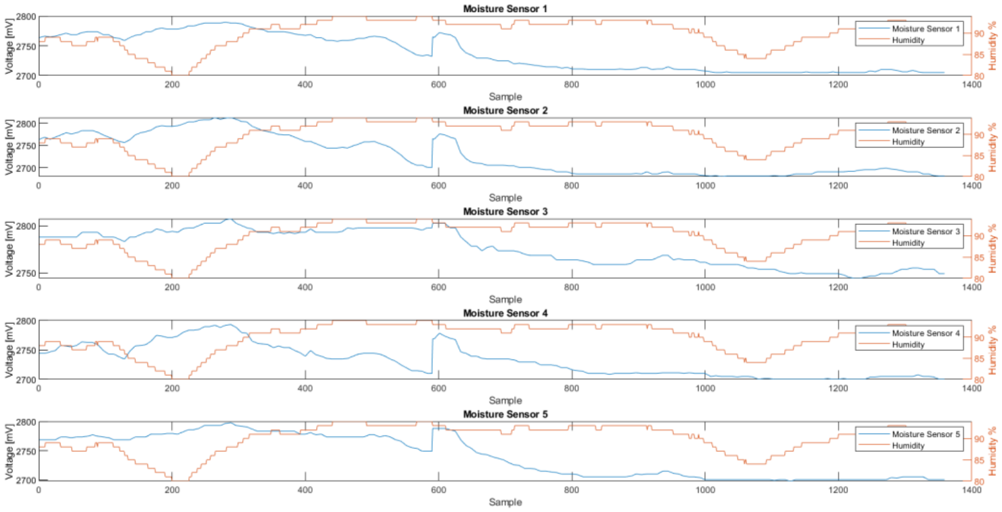


Figure 4.12: Moisture sensor outputs compared to humidity

Even though the amount of data samples which registered some amount of precipitation is limited, the lack of correlation with the moisture sensors outputs suggest that detecting rain with this type of remote node is unfeasible.

5 Machine Learning Models

In this chapter machine learning models are built and tested using different methods and algorithms. Since SWE is calculated based on the snow depth and snow density models for both snow depth and density are built. Three different machine learning platforms are used, each building and testing models independently of each other. Estimating the snow depth and snow density is a regression problem and different regression algorithms are tested to find the best model possible, as well as models built from neural networks.

The data marked with the supervised learning flag is used for training and validating the models, and the rest is used to test the accuracy of the models. For all the model's measurements from prototype #1 is used. There are seven inputs, the five soil moisture sensors, atmospheric pressure, and temperature.

5.1 MATLAB

MATLAB has many regression algorithms and the possibility to build shallow neural networks. It has a regression learner application that trains models with different algorithms. Which implies that you don't have to undertake each algorithm individually. The application also performs normalization for you, as does the shallow NN regression application used later. The best model created in MATLAB was created using this application with 5-fold cross-validation, and an automatic PCA that removes dimensions after 95% of the variance is explained. In this case the two first principal components were used. The best training algorithm for both models ended up being Gaussian Process Regression. Figure 5.1 and 5.2 show the density and depth, respectively, estimated by the models against the measured density and depth. The test data is as explained earlier the measurements without the supervised learning flag, so they are not entirely correct. The measured plot only shows the last manual measurement, implying that when the measured values change you would expect to see some change in the predicted plot before the jump takes place.

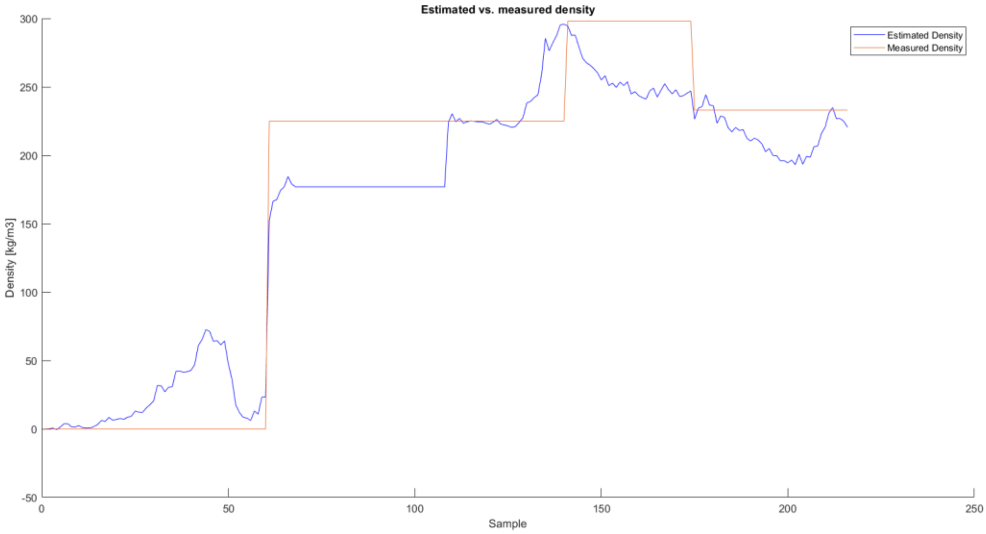


Figure 5.1: Snow density model estimations vs. measured density

The density model seems to perform reasonably well when snow is present at the sensors. As seen at the start of figure 5.1, where there is no snow, the model estimates that there is a density. The peak of the incorrect density, at about sample 45, corresponds to the lowest temperature period in November.

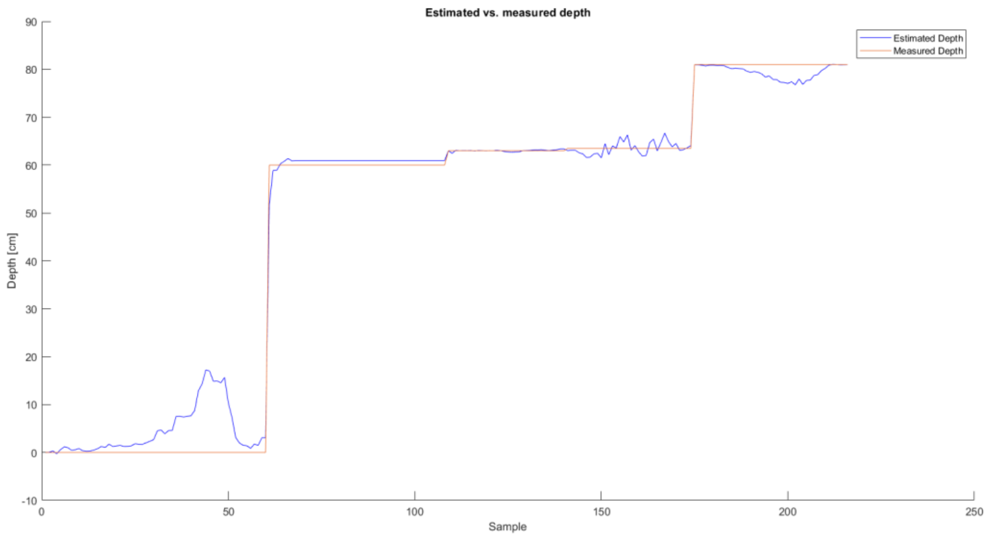


Figure 5.2: Snow depth estimated by model vs. measured depth

The depth model seen in figure 5.2 seems to have a problem estimating the snow depth in November, where the measured depth is zero just like the density model did. The rest of the model follows the measured depth quite closely. The jump at sample point 175 looks like it follows the measured value too closely but, this is where the data changes from December to February, implying that a smooth transition at that point would not be correct.

MATLAB has the capability of creating deep and shallow neural network through applications as well as machine learning algorithms. Figure 5.3 shows the density estimates from a shallow neural network built in MATLAB.

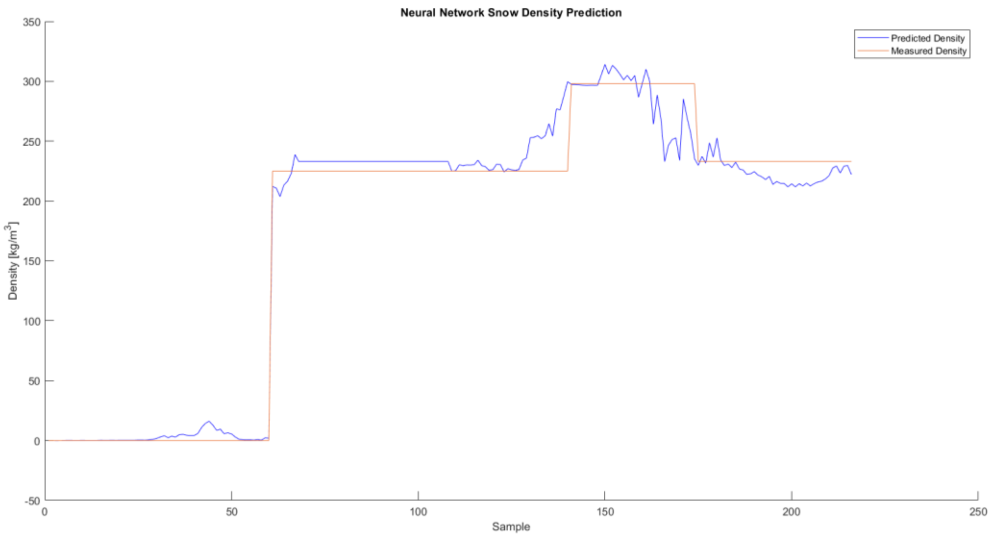


Figure 5.3: Snow density estimates vs. measured density

The neural network has one hidden layer, with 50 neurons and the training algorithm used is Bayesian Regularization. Figure 5.3 shows that the NN model handles the lack of snow better than the GPR model however, it still estimates snow when there is none. For the rest of the estimates the NN model seems to follow the measured density slightly better although not to a large extent.

Figure 5.4 shows the NN based snow depth model. This model also consists of one hidden layer with 50 neurons, and the best results were found using the Bayesian Regularization training algorithm. Like all models so far, November is a challenge. The GPR model follows the measured depth slightly worse that the NN model, except at the start, where the NN model both has a larger error in the period when there is no snow, and a slightly higher value at the start of December.

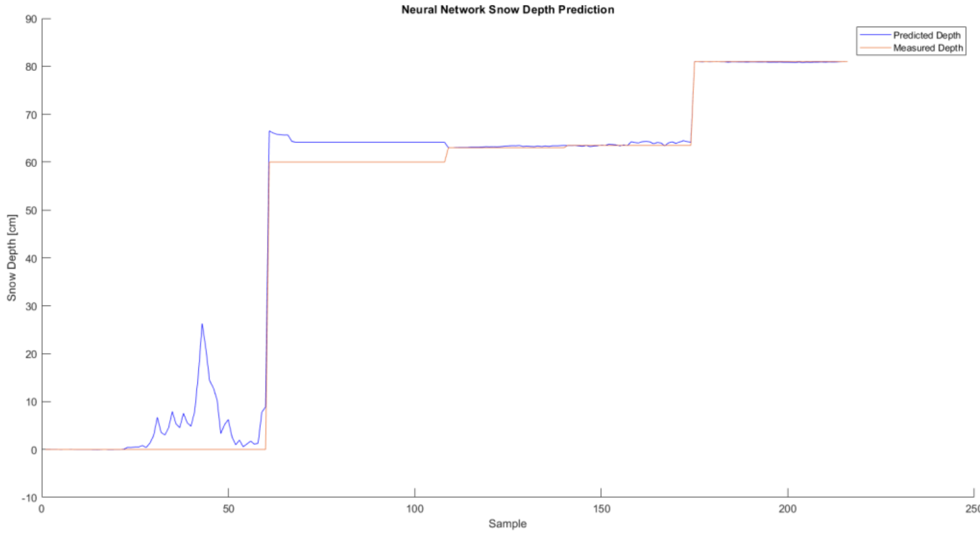


Figure 5.4: Snow depth estimated by model vs. measured depth

5.2 ML.NET

ML.NET has a handful of regression algorithms built in. It has an automatic trainer like MATLABs applications but, it is harder to test all the algorithms since only the algorithm with the best validation results are offered at the end of the process. In this case a tree algorithm might perform quite well, since the number of unique values in the output data is limited, but ideally the model should be able to estimate values other than what it was trained with.

To train and test the models, a Windows forms application was used. The primary purpose of the form was to plot, the estimates and measurements in a graph. None of the built-in regression algorithms worked as well as the NN or GPR based models from MATLAB. The number of algorithms to choose from is also more limited. The best results came using a Fast Forrest algorithm. Figure 5.5 shows the results of the density model.

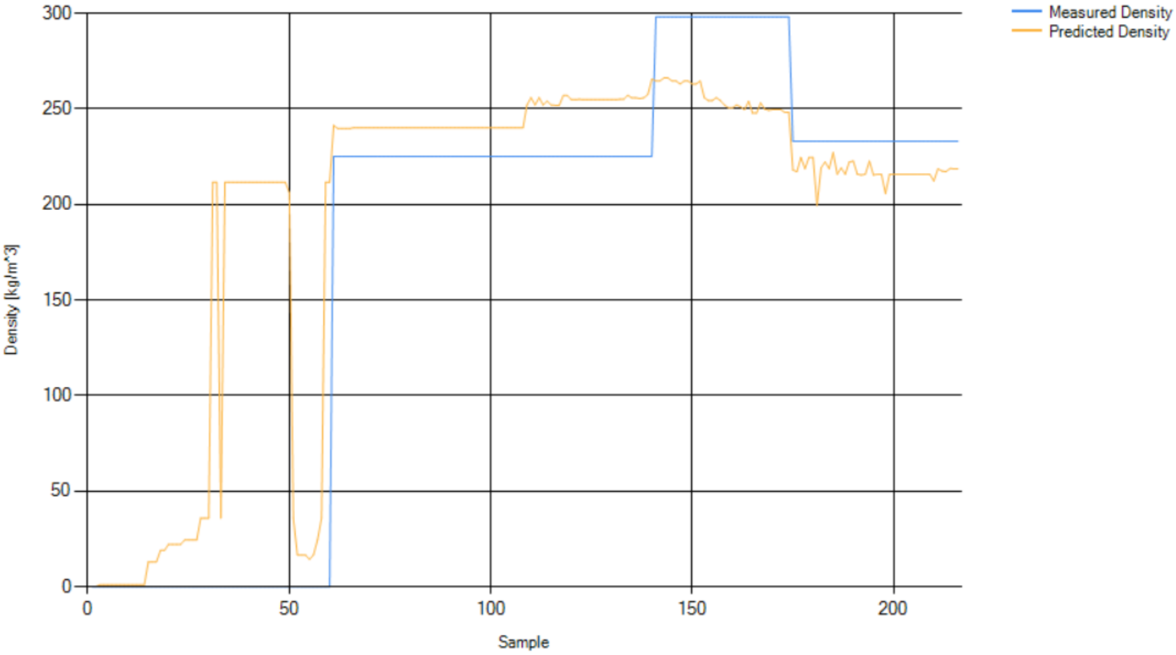


Figure 5.5: Snow density estimated by model vs. measured snow density

This model struggles more than the others at the start when there is no snow, and looks like it estimates less accurately than the previous models, especially at the higher densities.

The best performing algorithm for training the depth model, ended up being the Generalized Additive Models algorithm. The depth estimations from this model is shown in figure 5.6. Like every other model so far, the model is unable to detect a snow depth of 0 cm.

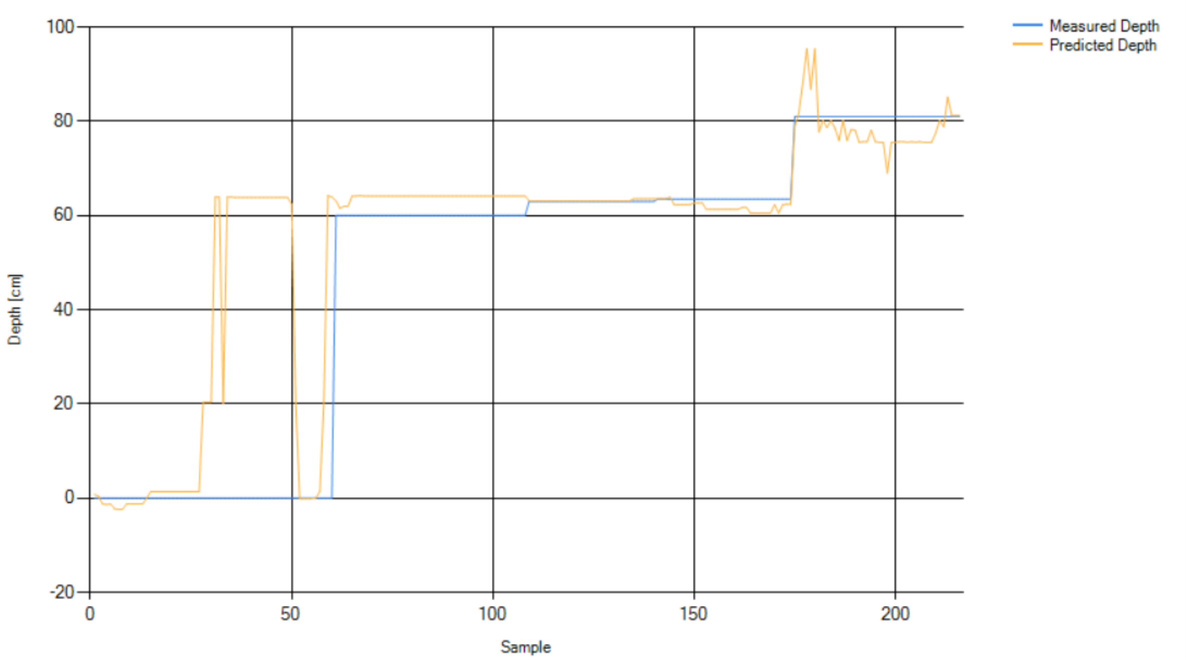


Figure 5.6: Snow depth model estimations vs. measured depth

5.3 Keras

Keras is a python-based API for TensorFlow, and is used for deep learning neural networks.

In this section neural network models are built, trained, and tested for both snow depth and snow density. Keras provides far more possibilities than the more shallow neural network in MATLAB, although there is no application with a GUI like MATLAB has, for neither deep, nor shallow neural networks.

Both MATLAB and ML.NET normalizes the data for you, but Keras does not. Hence, before the data can be used in the neural network the data is normalized. One NN was created for each model and different combinations of hidden layers, neurons in those layers, and optimizers were tested. Finding the best parameters was acheived by trial and error.

Density model

The results for the best density model can be seen in figure 5.7. The shape of the neural network is explained in table 5.1. The adam optimizer was used during training, and 20% of the training data was used for validating the model during training, optimizing the mean squared error. When using the built in Keras training and validating split, the fraction of the data that will be used for validating is set between 0.0 and 1.0. The split is chosen to use the last part of the data set for validating. To ensure that there was enough variation in the validation set, the order of the samples in the training data set is randomized. 1000 epochs were used for training the model.

Table 5.1: Snow density neural network structure

Layer	Neurons	Activation Function
Input Layer	7	
Hidden Layer 1	150	Relu
Hidden Layer 2	125	Tanh
Hidden Layer 3	100	Relu
Hidden Layer 4	75	Relu
Hidden Layer 5	50	Relu
Hidden Layer 6	25	Relu
Output Layer	1	Relu

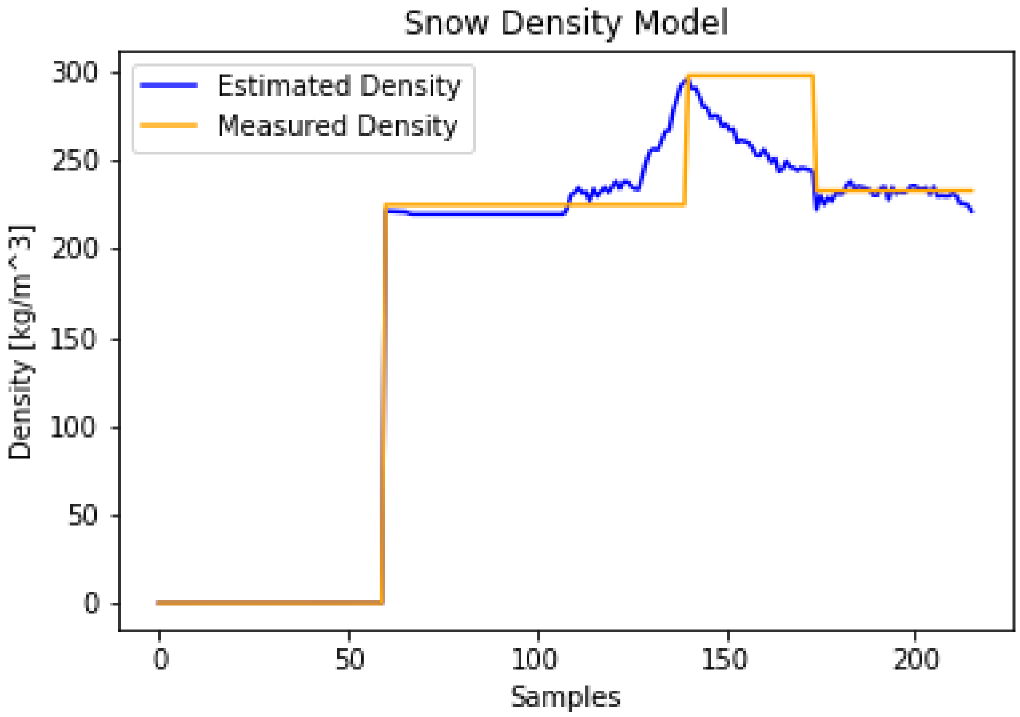


Figure 5.7: Snow density estimated by model vs. measured density

As seen in figure 5.7, this is the first model that handles the period with no snow quite well. There is a sample or two at the end where the model estimates a bit of snow, but it putperforms all the other models trained so far. For December and February, it is on par with the models created in MATLAB.

Depth model

The preparation of the data for the snow depth model is similar to the density model. Several combinations of layers, activation functions, neurons, and optimizers were tested attempt to train the best model possible. Table 5.2 describes the neural network that achieved the best estimations based on the testing data.

Table 5.2: Snow depth neural network structure

Layer	Neurons	Activation Function
Input Layer	7	
Hidden Layer 1	200	Tanh
Hidden Layer 2	125	Tanh
Hidden Layer 3	80	Relu
Hidden Layer 4	100	Sigmoid
Hidden Layer 5	75	Tanh
Hidden Layer 6	75	Sigmoid
Hidden Layer 7	50	Relu
Hidden Layer 8	50	Relu
Output Layer	1	Relu

Like the rest of the depth models there are some issues when it comes to the November samples, but it still provides a better result than the other depth models. Figure 5.8 shows the model estimations in comparison to the measured snow depth.

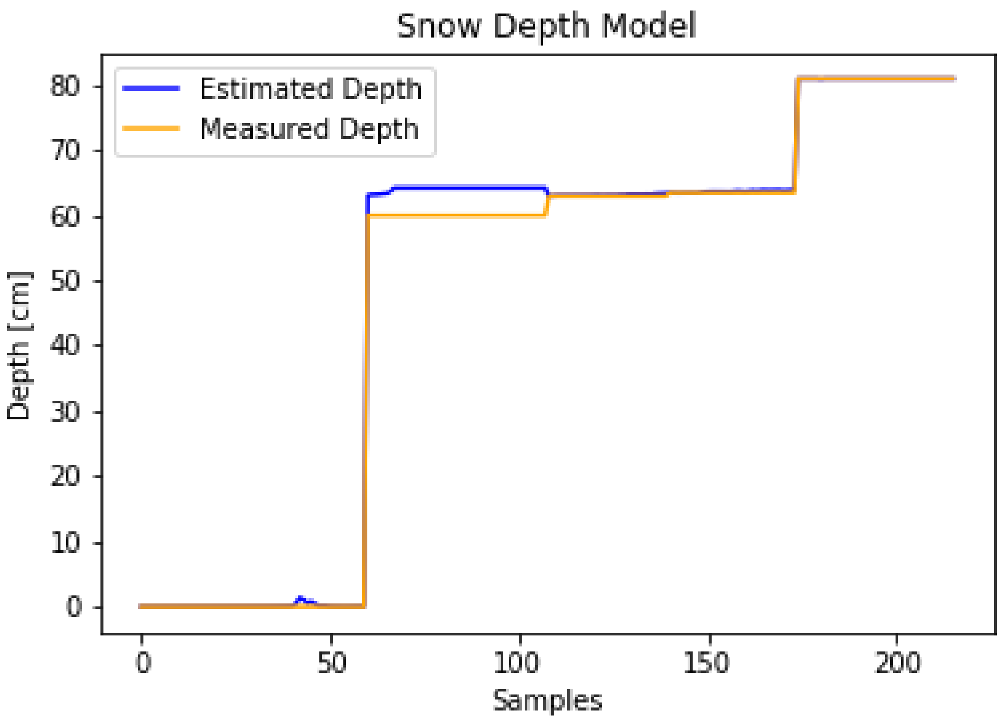


Figure 5.8: Snow depth model estimations vs. measured depth

SWE model

Most of the focus in this thesis is on the snow depth and snow density, because the SWE that is of interest, is easily calculated from the depth and density. The best models that has been created in this thesis were found using Keras. In figure 5.9 those two models are used to estimate snow depth and density, and from these predictions the calculated SWE is compared to the measured SWE. The combined model works quite well applied on the test data. It doesn't follow the measured SWE exactly everywhere, but this is because of the density measurements, where the model is expected to change value gradually.

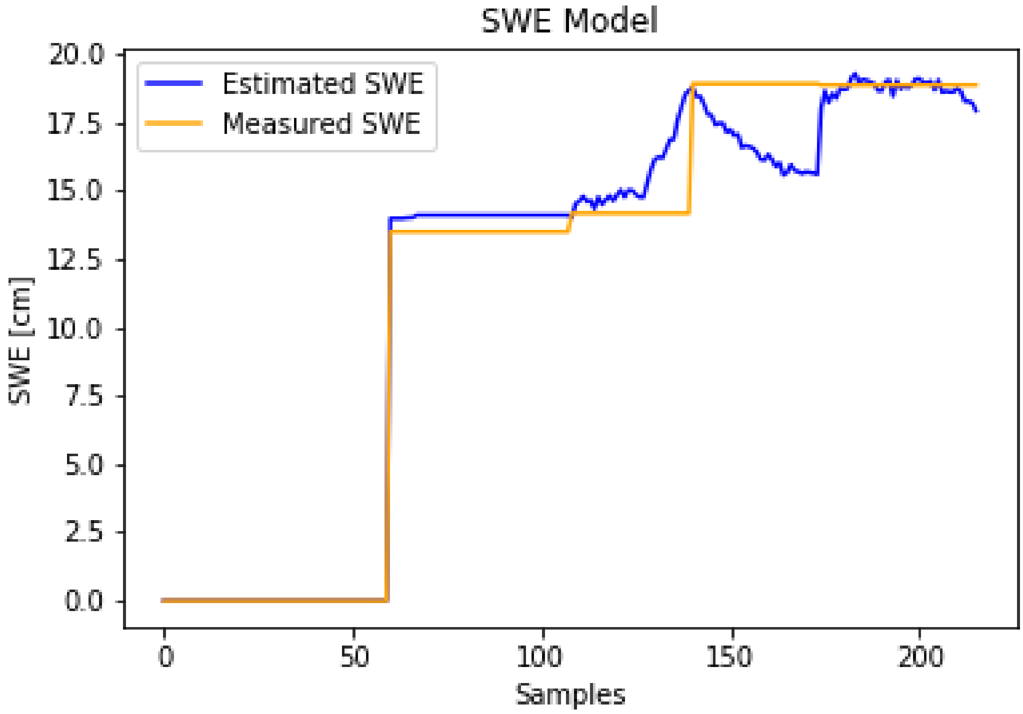


Figure 5.9: SWE estimated by calculating estimated depth and density vs. SWE from measured depth and density

6 Conclusion

The main objective of this thesis has been to evaluate whether the capacitance-based sensor approach to measure SWE is a feasible and good solution. Based on a set of models created and trained, it seems possible that this approach could be used to estimate both the snow depth and the snow density needed to calculate SWE. Unfortunately, due to there being little snow in Porsgrunn in the testing period and the black plastic heating in the sun on prototype #3 making the measurements done with that prototype a bit uncertain. This restricted the possibilities for training the ML models to prototype#1, that was placed in a snow safe location. There are some uncertainties if the manual measurements of the densities are correct in December, and there is a limited number of them and this led to the machine learning models being trained on somewhat limited data. A larger data set, typically possessing a larger variety in snow depth and snow density, would probably yield better models.

The results show that all the models have a challenge when there is no snow present. This isn’t necessarily a problem, since the whole purpose installing these sensors is to measure when there *is* snow, not *whether* there is snow. Though obviously, being able to detect snow would be preferrable. During winter when there is snow present, and the temperature is below zero degrees centigrade, the estimates from the models perform well and the estimates are pretty close to the observed values.

Unfortunately, as discovered in March and April, the snow retracts from the moisture sensors when the temperature increases. The planned use for this system is to help predict the inflow of water to reservoirs. This is mainly done during the spring when the temperatures are starting to rise, so even if the models work reasonably well during the winter, it is not very likely that they will work for their intended purpose.

What we have seen from the test seems to be that a major issue is that retracting snow will likely be a challenge for any sensor system that depends on physical contact with the snow to make measurements unless one is able to prevent the snow from melting close to the sensors.

7 References

[1] J. M. Bjerke, M. N. M. Abril, A. Jaganjac and N. Pouladi, “Measurement of snow density,” Porsgrunn, 2019.

[2] M. N. M. Abril, “Development of a remote measurement node for snow density,” Porsgrunn, 2020.

[3] Skagerak Kraft, “About us,” Skagerak Kraft, 2021. [Online]. Available: https://www.skagerakkraft.no/about-us/category2464.html. [Accessed May 2021].

[4] H. B. Stranden, B. L. Ree og K. M. Møen, «Recommendations for automatic measurements of snow water equivalent in NVE,» Norwegian Water Resources and Energy Directorate, Oslo, 2015.

[5] Campbell Scientific Inc., “CS725 instruction manual,” August 2012. [Online]. Available: https://s.campbellsci.com/documents/us/manuals/cs725.pdf. [Accessed May 2021].

[6] N.-O. Skeie, “Description of snow capacitance and density measurement prototypes,” 2021.

[7] P. Placidi, L. Gasperini, A. Grassi, M. Cecconi og A. Scorzoni, «Characterization of Low-Cost Capacitive SoilMoisture Sensors for IoT Networks,» Perugia, 2020.

[8] GikFun, “Amazon capacitive soil moisture sensor,” [Online]. Available: https://www.amazon.com/Gikfun-Capacitive-Corrosion-Resistant-Detection/dp/B07H3P1NRM. [Accessed May 2021].

[9] A. Ambanloc, «What Communication Protocol to Use and When,» 2020. [Internett]. Available: https://neuronicworks.com/blog/wireless-communication-protocols/. [Funnet May 2021].

[10] LoRa Alliance, «What is LoRaWAN?,» November 2015. [Internett]. Available: https://lora-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf. [Funnet May 2021].

[11] The Things Network, «landing Page,» 2021. [Internett]. Available: https://www.thethingsnetwork.org/. [Funnet May 2021].

[12] Google Inc., “TensorFlow website,” [Online]. Available: https://www.tensorflow.org/. [Accessed May 2021].

[13] Google Inc., “TensorFlow API Documentation,” April 2021. [Online]. Available: https://www.tensorflow.org/api_docs. [Accessed May 2021].

[14] Keras, «About Keras,» 2021. [Internett]. Available: https://keras.io/about/. [Funnet May 2021].

[15] Keras, «Why choose Keras?,» 2021. [Internett]. Available: https://keras.io/why_keras/. [Funnet May 2021].

[16] The MathWorks inc., «Pricing and licensing,» 2021. [Internett]. Available: https://se.mathworks.com/pricing-licensing.html. [Funnet May 2021].

[17] The MathWorks Inc., «Machine Learning,» 2021. [Internett]. Available: https://se.mathworks.com/solutions/machine-learning.html. [Funnet May 2021].

[18] The MathWorks Inc., «AutoML,» 2021. [Internett]. Available: https://se.mathworks.com/discovery/automl.html. [Funnet May 2021].

[19] J. Capellman, Hands-On Machine Learning with ML.NET, Packt Publishing Ltd., 2020.

[20] CNTK Team, «CNTK GitHub readme,» April 2019. [Internett]. Available: https://github.com/microsoft/CNTK/blob/master/README.md. [Funnet May 2021].

[21] scikit learn, "scikit-learn index," [Online]. Available: https://scikit-learn.org/stable/index.html. [Accessed May 2021].

[22] scikit-learn, «About Us,» [Internett]. Available: https://scikit-learn.org/stable/about.html. [Funnet May 2021].

[23] FaceBook Inc., "Features," [Online]. Available: https://pytorch.org/features/. [Accessed May 2021].

[24] faceBook Inc., "PyTorch documentation," [Online]. Available: https://pytorch.org/docs/stable/index.html. [Accessed May 2021].

[25] The Linux Foundation, «ONNX Home Page,» [Internett]. Available: https://onnx.ai/index.html. [Funnet May 2021].

[26] N. Siddique and H. Adeli, Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing, Wiley, 2013.

[27] K. H. Esbesen, Multivariate Data Analysis - In Practice. An introduction to Multivariate Data Analysis and Experimental Design, Camo Software AS, 2002.

[28] T. Kuehl, "Texas Instruments - Clock & timing forum," April 2014. [Online]. Available: https://e2e.ti.com/support/clock-timing-group/clock-and-timing/f/clock-timing-forum/333853/tlc555---frequency-stability-with-temperature-change . [Accessed May 2021].

Appendices

Appendix A Thesis description

Appendix B Description of snow capacitance and density measurement prototypes

Appendix C Code

Appendix D Sensor measurement comparison plots

Appendix A – Thesis Description

University of
South-Eastern Norway

Faculty of Technology, Natural Sciences and Maritime Sciences, Campus Porsgrunn

FMH606 Master's Thesis

Title: Development of models for estimating snow height and snow density using machine learning methods.

USN supervisor: Nils-Olav Skeie and co-supervisors Håkon Viumdal and Ole Magnus Brastein

External partner: Skagerak Energi AS / Beathe Furenes

Task background:
Part of the work at the Hydrologist group at Skagerak Energi AS is to develop and maintain models to predict the amount of inflow for energy production in hydropower systems. The input to these models are based on among others the weather forecast to predict this inflow.
The springtime in northern countries can be a challenge for such systems as the melting of the snow will highly affect this prediction of inflow. Today manual measurements of the snow height and density are performed during the end of the wintertime to estimate the inflow. These measurements must often be taken at impassable locations so the manual operation takes time and is expensive.
An MSc project group autumn 2019 at USN evaluated different measurement principles, and an MSc master thesis spring 2020 made some preliminary conclusions based on insufficient snow measurements with a selected measurement principle. This project will be a continuation of the results from previous work at USN.
Skagerak Energi AS wants a robust autonomous measurement system for snow density which can be deployed remotely with a minimal footprint, and transmit the desired information to the data centre at Skagerak Energi AS. The focus in this thesis is data pre-processing and data-driven modelling, not the data communication nor system design.

Task description:
The suggested task description is:

- Make a literature survey of such systems available today.
- Evaluate the usage of a mechanistic model using the measurement from the node for estimating the snow density and snow height.
- Give an overview of machine learning platforms like TensorFlow and Keras, and support in programming languages like MATLAB, Python, R, Julia and C#.
- Evaluate the supervised and reinforcement learning methods for this projects and indicate any requirements for such learning methods.
- Describe the structure of making a data driven model based on machine learning algorithms and relate the structure to available methods in the ML.NET library for C#.
- Two prototypes are available, one with only capacitance sensor devices, and one with both capacitance, temperature and pressure sensor devices, and a set of samples will be available for developing a model. Make an analysis of the data sets with a discussion of the pre-processing of these data sets.
- Develop models for estimating the snow height, snow density and the snow water equivalent based on the data sets.
- Discuss the data communication challenges and propose a protocol between the measurement system and the data centre when the data centre will use the data

driven models developed in this project, including an option for supervised learning, for estimating the snow height, snow density and snow water equivalent (SWE),

- The air humidity affects the capacitance sensor devices. Evaluate if the change in air humidity can be used to indicate raining?
- Optional: make a paper for the SIMS conference in autumn 2021 (www.scansims.org).

Student category: IIA (EET, EPE, IIA or PT students)

The task is suitable for online students (not present at the campus): Yes

Practical arrangements:
Measurements from one prototype of the remote sensor node in the Lillehammer area will be available. Another prototype will also be available for the project at campus if more measurements are wanted.

Supervision:
As a general rule, the student is entitled to 15-20 hours of supervision. This includes necessary time for the supervisor to prepare for supervision meetings (reading material to be discussed, etc).

Signatures:

Supervisor (date and signature): 4-FEB-21 / Nils-Olav Skeie ....

Student (write clearly in all capitalized letters): Henrik Nikolai Vahl

Student (date and signature): 4-FEB-21 / Henrik Vahl ....

Appendix B - Description of snow capacitance and density measurement prototypes

1 SnowSensorNodes

Case Description of snow capacitance and density measurement prototypes
Author Nils-Olav Skeie (NOS)
Date 22-FEB-21 - Version 0

1.1 Introduction

In cooperation with Skagerak Energy AS in 2018, a project was started for estimating the water contents in snow. Skagerak Energy AS need this information for estimating the spring flood. The project started up with a MSc project in Autumn 2019, and continued with a MSc Thesis in spring 2020 and 2021. The focus in the MSc project was a system overview and a proposal for a system architecture for a remote snow sensor node. The main requirements are a small foot print and using a battery as the power source. The MSc thesis in 2020 was based on a prototype with five sensors measuring the capacitance in the snow, but due to the Covid-19 situation could not the prototype be used on the cabin on Sjusjøen. No snow in the Grenland area results in one measurement series on Viddaseter. Two new prototypes were build the autumn 2020 and one of these prototypes is located at Klufimovegen 190 at Sjusjøen.

All the prototypes are based on the humidity sensor for Arduino projects measuring the humidity in flower pots. The sensor device is shown in Figure 1.

Figure 1: The capacitanse sensor, measure about 10x2 cm.

This sensor device is located at different heights on the snow sensor to measure the capacitance at different heights. The hypothesis is that the water content in snow, the density of the snow, can be estimated using the capacitance of the snow. By using the capacitance at different heights, the density at different heights can be estimated, and the snow height can be estimated based on the sensors at different heights. According to the MSc project will it be a challenge to measure the snow height, therefor the first approach was to estimate the snow height based on the capacitance measurements.

The snow water content, or snow water equivalent (SWE), is estimated based on the following equation:

$$SWE = h\ \frac{\rho}{\rho_w}$$

where h is the snow height, ρ is the density of the snow, and ρ_w is the density of water $(1\,\mathrm{g/cm^3})$. Note that the unit for SWE is cm. The density of the snow is in the range of 100 to 400 $(\mathrm{kg/m^3})$ and can be calculated using a pipe with diameter d, as:

$$SnowDensity = \frac{Mass_{sample}}{Vol_{sample}} = 1000\ \frac{Mass_{sample}\ [\mathrm{g}]}{\pi r^2 h\ [\mathrm{cm^3}]}$$

where the mass is measured, and h is the snow height where the sample is taken.

The prototypes are based on an Arduino system collecting the values from the sensors, one measurement every minute, and transmitting the sensor values on the USB connection to a Windows computer for logging. An overview of the logging process is shown in Figure 2.

1

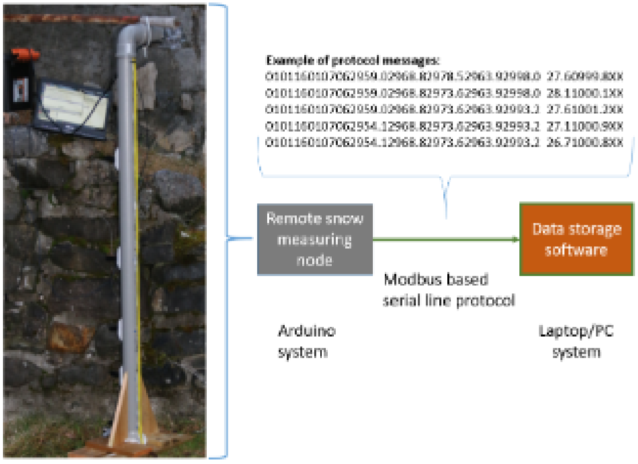


Figure 2: The system structure for logging the sensor values. The remote snow measuring node contains a set of sensors, read by an Arduino system located at the node, and transmitting the values on a Modbus RTU based protocol to the logging device.

The protocols consists of the location ID (prototype ID) and the sensor values from the specific node. The logging software on the Windows system must extract the values, filter the sensor values and log the values on a csv file. The csv file will also contain the manual snow height and snow density. These manual values must be added by a human and updated from time to time. Due to this time frame will these manual values be marked a valid only for a specific period of time by the supervise flag in the csv file.

1.2 Prototypes / Locations

There exists three prototypes:

1. Location #1: A prototype build using a plastic pipe with a height of 2 m and a diameter of 68 mm. The prototype is located at the cabin on Sjusjøen (Kluftmovegen 190, 2364 Næroset), installed outside at the end of November 2020, before the snow fall. Five capacitance sensors at different height in the plastic pipe, a temperature sensor (TMP−36) located at the top of the pipe, and two pressure sensors. One absolute pressure sensor at the bottom of the sensor to measure the snow pressure (the membrane is too small?) and an absolute pressure sensors inside the pipe to measure the atmospheric pressure. The sensor node is shown in Figure 2. The five capacitance sensors are located at the following heights: 10, 30, 50, 80, 110 [cm]. A measuring tape is glued to the pipe to measure the snow height. The Arduino system is located at the top of the pipe, together with a 12V PSU and the temperature sensor, and has been testing in temperatures down to $-26\,^{\circ}$C. The node is powered by 220V, with an integrated 12VDC PSU. The PSU is added due to the pressure sensors. The data is transmitted on the USB port, one message every minute.

2. Location #2: A prototype build in a small plastic box with the dimensions 20, 15 and 7.5 cm containing four capacitance sensors, one temperature sensor (TMP − 36) and an ultra sonic sensor. The measurement range for the ultra sonic sensor is 2 − 450 [cm]. The capacitance sensors have been modified so the first sensor has a vero board as the capacitance plate with an area of $16x6$ cm and the second capacitance sensor has two standard plates of $6x2$ cm and the rest of the capacitance sensors have the standard plate of $6x2$ cm. These sensors are mounted on different sides of the plastic box and the show height must be simulated by filling each side with snow. The ultra sonic sensor is mounted at the last side of the box, without any capacitance sensor. The sensor node is powered using the USB

2

```
TimeSample;NodeAdr;CapSens#1(Bot)(mV);CapSens#2(mV);CapSens#3(mV);CapSens#4(mV);CapSens#5(mV);PressSens#1(mBar);PressSens#2(Atm:mBar);TempSens#1
(degC);ManSnowHeight(cm);ManSnowDensity(kg/m3);ManLogInfo;SupervisedLearningFlag
28/12/2020 12:16;01;2602.5;2656.1;2680.7;2656.1;2627.0;947.9;952.6;-2.6;000;000;test with 60 cm;OFF
28/12/2020 12:46;01;2605.0;2656.1;2690.4;2710.0;2641.6;948.1;952.0;-2.4;000;000;test with 60 cm, lett snøvar;OFF
28/12/2020 13:16;01;2607.4;2652.6;2690.4;2714.8;2636.7;949.9;952.6;-2.1;000;000;test with 60 cm, lett snøvar;OFF
28/12/2020 13:47;01;2607.4;2651.4;2690.4;2714.8;2646.5;949.9;953.9;-2.1;000;000;test with 60 cm, lett snøvar;OFF
28/12/2020 14:17;01;2606.2;2651.4;2690.4;2717.3;2651.4;949.5;953.9;-2.4;000;000;test with 60 cm, lett snøvar;OFF
28/12/2020 14:47;01;2605.0;2650.2;2690.4;2717.3;2659.9;949.1;953.2;-2.2;000;000;test with 60 cm, lett snøvar;OFF
28/12/2020 15:18;01;2637.9;2646.5;2690.4;2719.7;2659.9;949.1;953.9;-2.1;000;000;test with 60 cm, lett snøvar;OFF
28/12/2020 15:48;01;2641.6;2646.5;2690.4;2718.5;2658.7;949.9;955.7;-3.2;000;000;test with 60 cm, lett snøvar;OFF
```

Figure 3: The csv file format for the logger application, with the header on the first line.

port. The data is transmitted on the USB port, one message every minute. This is a portable device used at different locations, both at the cabin and at home.

3. Location #3: A prototype build at USN based on several plastic pipes, a height of about 1.8 m and a diameter of 110 mm. The prototype is located at Stathelle or Porsgrunn (USN), and covered by snow after the snow fall. The prototype consists of five capacitances at the following heights: 23, 36, 64, 92, 122 [cm] .A temperature sensor (TMP – 36) has been added to the system. The sensor node is powered using the USB port. The data is transmitted on the USB port, one message every minute. Black plastic pipes are used giving a challenge with the sun as the snow close to pipes is melting faster than the surroundings snow.

1.3 Csv file format

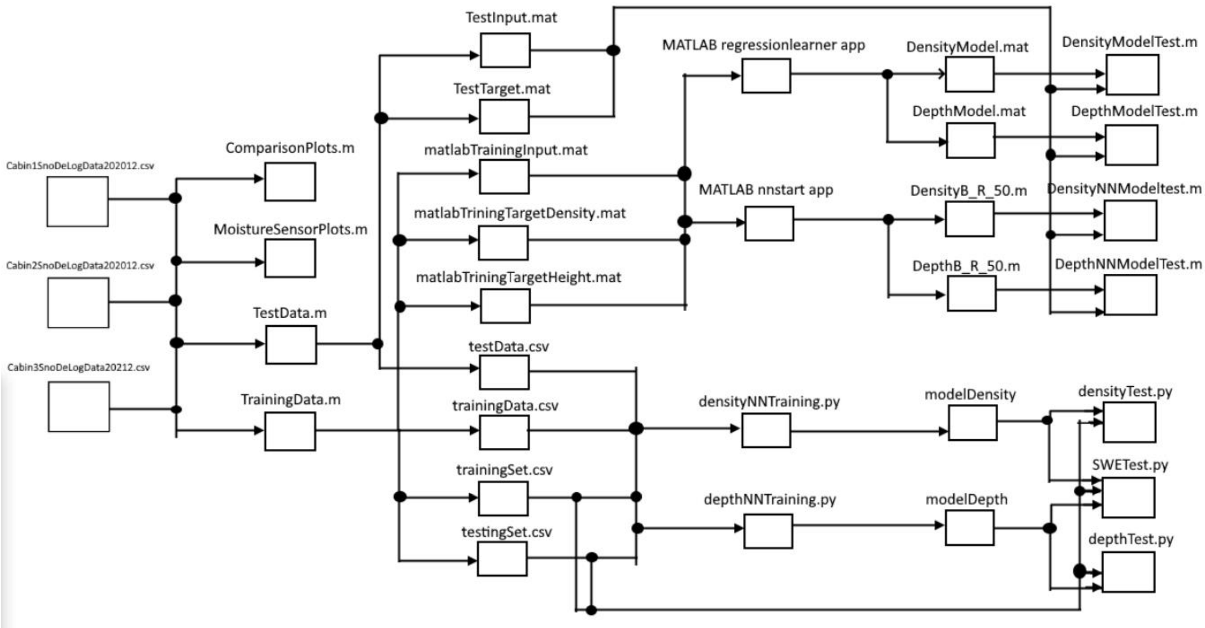
The csv file fields are separated by the semi colon and containing the following fields:

1	Date and time	DD/MM/YYYY HH:MM
2	Location	xx (01 - 99)
3	Sensor devices	Floating value, number depend on location
4	Manual snow height	integer, in cm
5	Manual snow density	integer, in kg/ m^3
6	Comments	string, lately with the snow density information
7	Supervised learning flag	ON/OFF flag, turn off after x samples

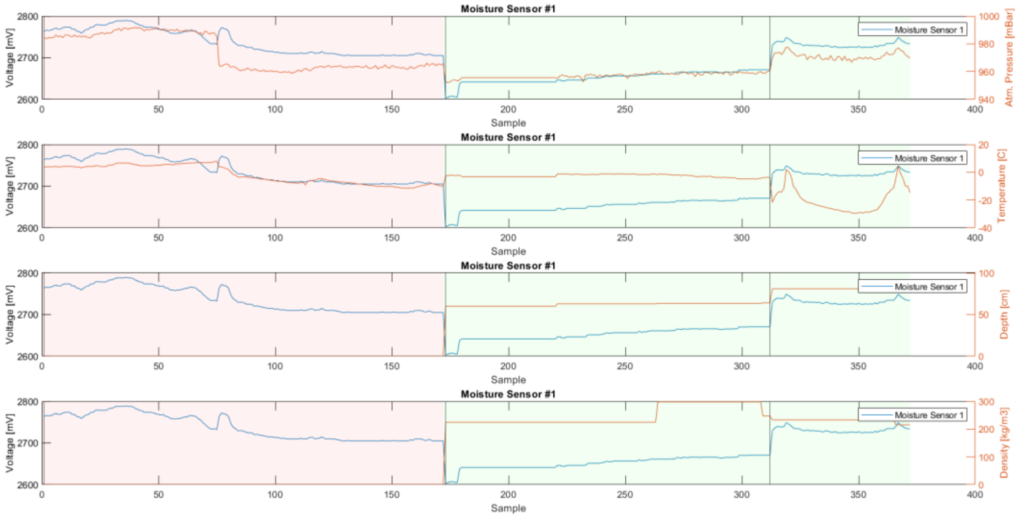
The manual inputs are part of configuration, and the file can be without these fields. An example of a csv file is shown in Figure 3.

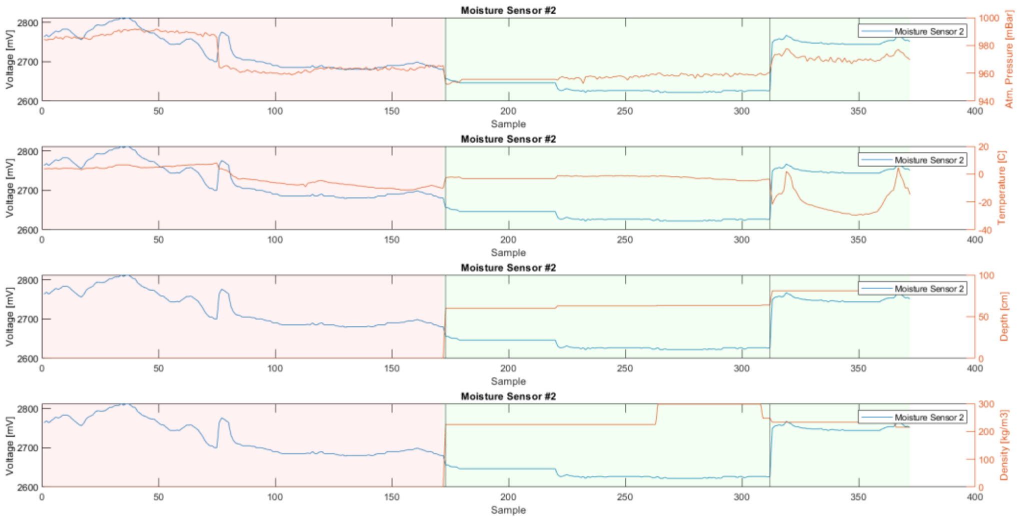
Appendix C – Code

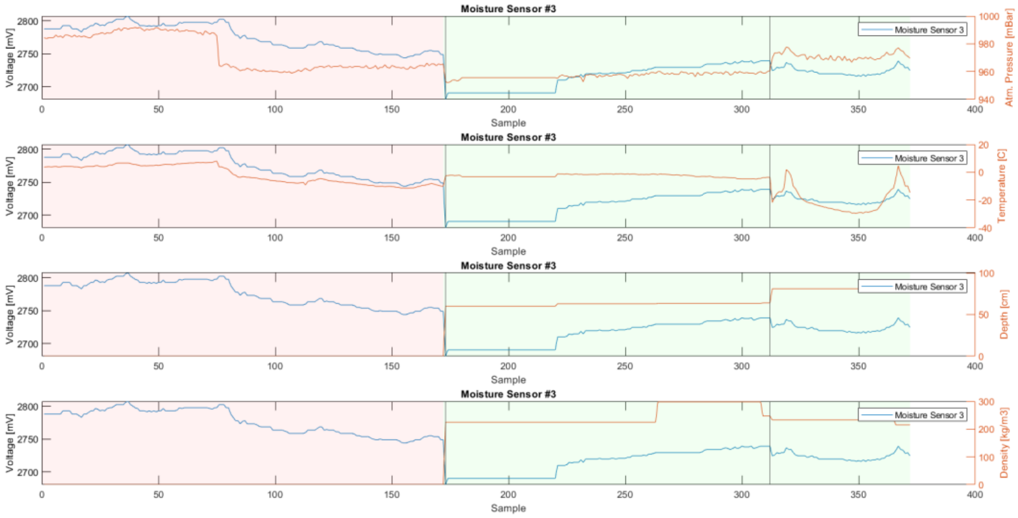
The figure below shows a flow-chart of different scripts and the files they produce and use, used to create the plots seen in this thesis. For the ML.NET the Visual Studio project is located in the MLRegression folder. The scripts can be found at https://github.com/Henrik-V/ThesisScripts

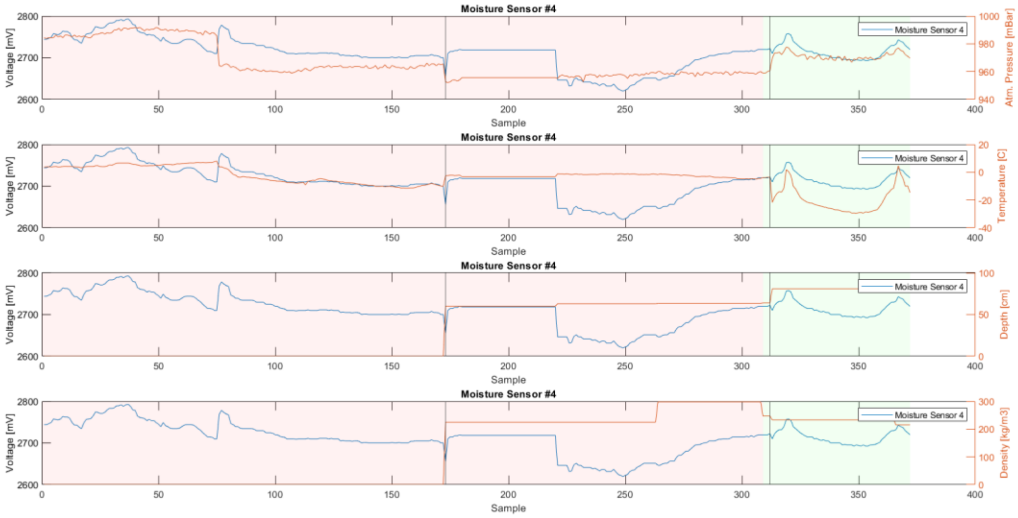


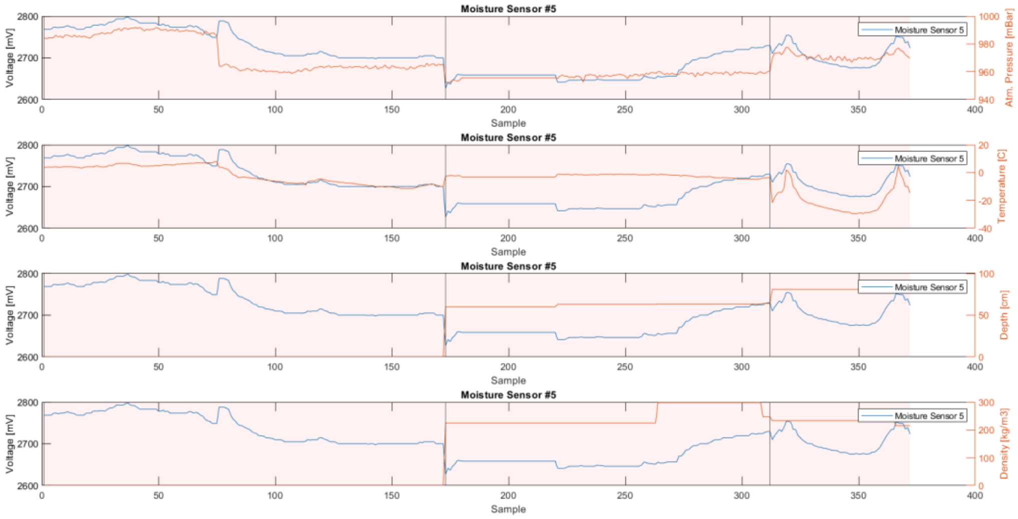
Appendix D - Sensor measurement comparison plots






Moisture Sensor #3
Moisture Sensor #3
Moisture Sensor #3
Moisture Sensor #3
Voltage [mV]
Sample
Moisture Sensor 3
Atm. Pressure [mBar]
Temperature [C]
Depth [cm]
Density [kg/m3]


Moisture Sensor #4
Moisture Sensor #4
Moisture Sensor #4
Moisture Sensor #4
Voltage [mV]
Sample
Moisture Sensor 4
Atm. Pressure [mBar]
Temperature [C]
Depth [cm]
Density [kg/m3]


Moisture Sensor #5
Moisture Sensor #5
Moisture Sensor #5
Moisture Sensor #5
Voltage [mV]
Sample
Moisture Sensor 5
Atm. Pressure [mBar]
Temperature [C]
Depth [cm]
Density [kg/m3]