A climate forecasting model to assist Belgian wine-growers against bud freezing

Liège University, Faculty of Applied Sciences

A thesis presented in fulfillment of the requirements for the Master degree of "Civil Engineer in Data Science"

Adrien Leerschool

Supervisors :
Bertrand Conélusse, Sébastien Doutreloup

Academic year 2021-2022

Abstract

In Belgium, the global warming offers new opportunities for wine culture. In the past fifteen years, the area dedicated for vineyards has become eight times bigger, rising from 72 hectares in 2006 to 587 hectares in 2020 [14].
Nevertheless, the Belgian climate is not always indulgent with vineyards throughout the year. In the spring particularly, when the bud has already grown and has become more sensitive to frost, a few cold nights may arise and lead to the destruction of the grapes.

Hopefully, some methods exist to fight bud freezing. From one vineyard to another, different techniques will be used depending on, amongst other, the size of the plantation, the intensity of the frost or the financial means.
In 2017, half of the plantations had a size below one hectare [14]. For this kind of small area, the construction of expensive infrastructures is not viable and more adapted methods must be chosen. The most suitable solution, and often the only one conceivable in small vineyards, consists in lighting large candles all over the place to increase the temperature around the grapes.
Although this solution has the advantages to be easy and relatively cheap, it also has two drawbacks. The first one, common to other more expensive methods, is the temperature monitoring, which consist in determining when to take action. The second one, inherent to this particular method, is the need for a workforce available in a hurry to light all the candles in the middle of the night.

In our work, we propose to assist Belgian wine makers by means of Machine Learning (ML) techniques in the two aspects aforementioned. We achieved to accurately forecast the temperatures around the grapes up to 36 hours ahead with an error below 1° Celsius. Additionally, we also provide a live supervising model capable of detecting negative temperatures with an error below $0.5°C$. By doing so, we avoid the wine growers to unnecessarily wake up during night to monitor the temperatures and we also allow them to organize a workforce sufficiently early enough.

Acknowledgments

At the end of the academic year 2020-2021, it was the time for me to start thinking about a subject for my master thesis taking place the next year. Although the list of subjects for the data science master included some interesting projects, I was not confident about the topic to choose.

It was then that my father told me about a concrete issue they were having with his friends in their two vineyards. Back then, the doctor in climatology Sébastien Doutreloup, who was already working on that subject and needed someone to analyze the data, introduced me to the problematic. From that moment, I knew that I had a great opportunity to present an interesting, concrete and personal project that might help people around me.

In the first place, I thus wish to thank Sébastien who supported me with wise pieces of advice and provided all the necessary material for weather monitoring. It was the good time, the good place, he knew how to approach the problematic and therefore put me in a propitious environment to realize this master thesis.

Another main character that played a major role throughout the realization of this master thesis is my supervisor Bertrand Cornélusse. He initiated periodic meetings to maintain a constant working pace where he was able to give me technical advice and warned me when I fell behind while remaining encouraging. Together with Sébastien I could not have hoped for a better supervisor team.

I would also like to thank Sébastien Weykmans who works at `WalDigiFarm`, which is a company promoting the development of connected agriculture. He made available two `Sencrop` stations, which are connected devices specialized in agriculture and largely used in Wallonia. This might create opportunities to spread this work in the future.

Finally, I also wish to thank Jean-Marc Lewalle, a biologist by training who started the culture of the vines together with my father ten years ago. He helped me in the monitoring of the data but especially he explained me the functioning of vines with completeness and great interest, making each fact captivating.

Contents

Chapter 1

Introduction

1.1 Context

For a few decades now, Belgium is seeing an increasing number of vineyards emerging from all over its territory [14]. A constant increase of the temperatures seems to positively alter the climate for the wine industry [5]. As a matter of fact, in Uccle, a municipality in Brussels, the ten hottest years ever recorded since 1833 are in the beginning of the 21^{th} century [33]. Although no direct causal link has been explicitly found between those two events, the numbers speak for themselves.

In 2020, Belgium had 198 wine makers against 154 in 2019, this increase of 35% in only one year perfectly describes the fad in the vine growing activity. Even if Flanders has 326 hectares against 260 in Wallonia, the latter has produced 1 003 059 liters of the 1 853 034 made in the whole kingdom [14]. The strongest growth has been recorded in the province of Liège, from 9 wine makers in 2019 to 21 in 2020 (where only the professional growers have been referenced since the SPF Finances [1]does not account for amateurs), which is the region where the data has been collected [15].

Nevertheless, even if nowadays Belgian temperatures are more suitable than before for wine growing, some typical climate behaviors are destructive. Often occurring in the beginning of the spring, a well known freezing event, called *the spring frosts*, may appear [8]. According to Doutreloup & al., the frost represents a major constraint and threat for wines in Belgium [40].
Some time before this period, the buds start to disburse and their water content increase. As their development progress to become shoot, they progressively lose the capacity to withstand cold and therefore become very sensitive to negative temperatures [26].
Then, when a windless cold night arises, air stratifies and buds are trapped in the coolest layer, leading to the lost of a large proportion of the vineyard (sometimes all) by freezing. This kind of events therefore drastically impacts the harvest and subsequently the grower's income [8].

In addition, this shift of the temperature throughout the year does not only have a positive impact on the Belgian vines life cycle. The winters are milder and the buds germinate earlier [8]. The consequence of that is the hasty development of the vine which ultimately weakens their defenses,

[1]The Service Public Fédéral (SPF) Finances is the Belgian public entity that governs and supervises the finances of Belgium at a federal level

making them vulnerable against late frosts, which are still occurring at the same period.

Fortunately, means of action exist and are effective when used at the decisive moments. Without getting into the specifics, multiple techniques have been developed to protect vines against frost. By using a large horizontal wind turbine, one can break up the stratified temperature in order to avoid the cold from stagnating at the buds level [18]. Another can use the supercooling phenomenon, which consists in spraying the vine with water and catching the buds in an ice park, which prevents them to reach a temperature below 0° Celsius [36]. Finally, since the winemaker is fighting against frost, the more simple and straightforward solution consists in warming up the air surrounding the vines. The latter is done by lighting up large candles all over the vineyard.

The two first techniques mentioned above involve the construction of a costly infrastructure, which may not be adapted for the common Belgian vineyard. Indeed, in Belgium most of the wine comes from small producers who own small plots, 63% of them were below one hectare in 2017 [14]. Consequently, the construction of this kind of installation is not a worthwhile investment.
Conversely, the third method is well adapted since it requires no expensive equipment and can be deploy rather easily when needed. However, this simple solution has a drawback in comparison to others because it necessitates some manpower to light the candles in the middle of the night. This inconvenient implies people being called in a hurry when the monitored temperature drops below 0° Celsius. Moreover, temperatures within a single vineyard can highly fluctuate depending on the localization where the temperature is monitored, which leads to undetected frosts.

In this paper, we attempt to solve this problem by predicting the temperatures at the buds level sufficiently early enough to allow time to wine growers for organizing a workforce. Especially, we attempt to identify which portion of the vineyard specifically is subject to frost. Alternatively, we propose a live supervising fall-back solution to ensure error detection made in the first place.

With the rise of agrotechnologies, more and more farmers equip their crops with weather stations [45]. This allow them to monitor some useful information such as when to harvest or spread pesticide when special conditions are fulfilled for bacteria growth. This results in increasing the quality and quantity of the cultivated products, leading to higher profits for the farmers. By using such connected technologies, we hope to provide a highly reliable model which extrapolates the live temperatures to every parts of a vineyard, avoiding silent frosts that might have not been detected by the forecasting model.

In Belgium vineyards, one of the main challenges lies in the avoidance of bud freezing. In this field, few researches have been made and a lot of improvements need to be carried out to facilitate wine grower's work [9, 29, 41]. By combining these two approaches, we hope to provide to Belgian wine growers a realistic solution mixing far-sighted forecasts and a robust live supervising upon which they can confidently rely on.

1.2 Problem definition and objectives

In weather forecasting, as well as in many other fields, Machine Learning (ML) techniques have proven to be capable of competing with or outperforming existing dynamical models [19, 42]. These

relatively recent methods are constructed on the idea of making a computer program learns non-linear patterns between two related sets of data. By means of this kind of approach, we hope to make progress in this area and bring fresh ideas and new findings.

In this purpose, we created two distinct algorithms that will operate at two different temporal levels. This choice has been based on the willingness of creating a reliable ensemble capable of both forecasting and supervising temperatures within a vineyard.

The first algorithm consists in the live extrapolation of the data collected by weather stations. Depending on the topography and the weather conditions, large gaps can occur, in terms of temperature, between different parts of a vineyard. As well as horizontal discrepancies, significant vertical differences appear between station sensors (situated around one meter and half above the ground) and the actual feeling of the buds (around half a meter above the ground).

In order to create an adapted model for a given vineyard, the air movements and more specifically the air stratification within the plot, must be considered. Vineyards are often on sloping ground, preferably facing South, and have natural surrounding barriers such as a hedges or other cultures. These characteristics, inherent to each vineyard, lead to different air movements and will favor or not the stratification of the air.

As briefly mentioned earlier, when the wind is null, the air is not mixed and different heat layers might appear, with the coldest ones at the bottom. Since the buds are fifty centimeters above the ground, they are exposed to the coolest layer and may freeze although the station monitors a non-lethal temperature, one meter and half above the ground. On top of that, a bud at the top of a sloping vineyard, may feel a completely different temperature from an other situated at the bottom. These behaviors imply to not considering only one temperature for the whole vineyard but rather a coordinate-dependent temperature function.

The second algorithm will be charged of the forecasting. From the Regional Atmospheric Model (MAR), we extract regional forecasts 12, 24 and 36 hours ahead which are then downscaled to vineyards to make accurate local forecasts. The MAR has been developed in the end of the nineties by the University of Liège and gradually improved ever since. Although it was initially intended for the study of polar regions, it has been found to work well for Belgium as well. This model forecasting is divided into a grid where the side of each square pixel is 5 kilometers long.

Finally, by combining these two parts, we hope to be able to make reliable forecasts up to 36 hours ahead while also constantly reassess those predictions in live in order to avoid undetected freezing at all costs.

We developed multiple models with different parameters tuning in order to evaluate the most robust and reliable one. The latter are rated on their prediction accuracies as well as their confidences in the predicted temperatures.

1.3 Approach

The remainder of this work is structured as follows:

2. The **Background** chapter presents the available tools and the past researches made in this domain.

3. The **Methodology and data** chapter describes the procedures used to collect, store and manipulate the data. it synthesizes the hardware and software used as well as the actual content of the data.

4. In the **Models** part, we present our algorithms and the reasoning behind them.

5. The **Results and discussions** chapter details the performances and the accuracies of the models along with their interpretations.

6. **Conclusions and Future Work** finalizes the thesis and evokes paths that are worthwhile for future researches and improvements.

Chapter 2

Background

Not surprisingly, with the mushrooming of the Internet of Things (IoT), more and more sectors in agriculture equip their crops with sundry devices in order to collect data to improve their culture's yield. Indeed, in addition to the bigger profit, it is widely argued that the constant growing population and its increasing need of food entail the emergency of producing more than ever. In the meantime, the many problems incoming due to the globalization and the global warming such as the intense heat, the drought, the shift from fuel-based energy to green energy or the continuous need of new arable soils bring many challenges for the next decades [34].

Although the global warming made the wine culture more viable in Belgium, for other regions of the world, it has obviously become a very constraining problem. Since the beginning of the 20^{th} century, the global surface temperature was 1.09 [0.95 to 1.20] °C higher in 2011–2020 than 1850–1900 [37], which has a drastic impact on the biosphere upon which crops directly depend [30]. The generated disequilibrium translates into changes that directly destabilizes the wine growing industry. Clearly, the main aspect lies in the temperature modification, which entails a shift in premium winegrape regions and grape variety cultivation, but there are also other impacts less apparent such as the increase in insects and the diseases they spread, the spoilage of the wine quality and the lost of cultivable lands due to the rising sea level [32].

Concurrently, it has been observed that the literature of IoT in agriculture has significantly increased since 2010, a substantial majority of it comes from Asian scientists (especially from China) [45]. With the arrival of these new technologies, a door toward agriculture optimization based on climate forecasting has been open, leading to a clear trend of nowadays scientist interest. The sections below focus on a review of the studies of macro and micro weather prediction models and their ML-based techniques used along.

2.1 Machine learning techniques

Machine learning based techniques have this last decade proven their powerful ability to resolve complex problems where other conventional methods where unable to do so. The most famous breakthrough in this domain is the computer program called `AlphaGo` which was the first learning

algorithm capable of defeating 5-0 a Go [1]world champion [38]. This assessed the huge potential of such techniques, especially when looking at the enormous search space inherent to this Chinese board game.

The main idea of these methods is to learn the patterns linking some inputs to some outputs and then use this knowledge to estimate the results of entries never seen before. The learning process lies in the optimization of up to thousands of parameters by minimizing an error function representing the searching space. We assume here that the reader has some basic knowledge in mathematics and ML. Let us review few of them for the sake of clarity in the future sections.

2.1.1 Neural Network

A Neural Network, also called an Artificial Neural Network (ANN), is a computing system inspired by the biological neural networks that constitute animal brains. It attempts to simulate the connections, as we know them, between neurons and to make these connections more or less strong by attributing them a value.

In this artificial representation of a brain, each neuron is connected to other ones by a weighted link, a large weight convey a strong connection whereas a weight close to zero means no correlation between them. In order to organize the networks, neurons are arranged in vertical layers staked end to end (see figure 2.1) where the arrows represent the weighted connections between the neurons. The input layer constitutes the first layer (leftmost) and the output layer the last one (rightmost) and the layers situated in between are called *hidden* layers.

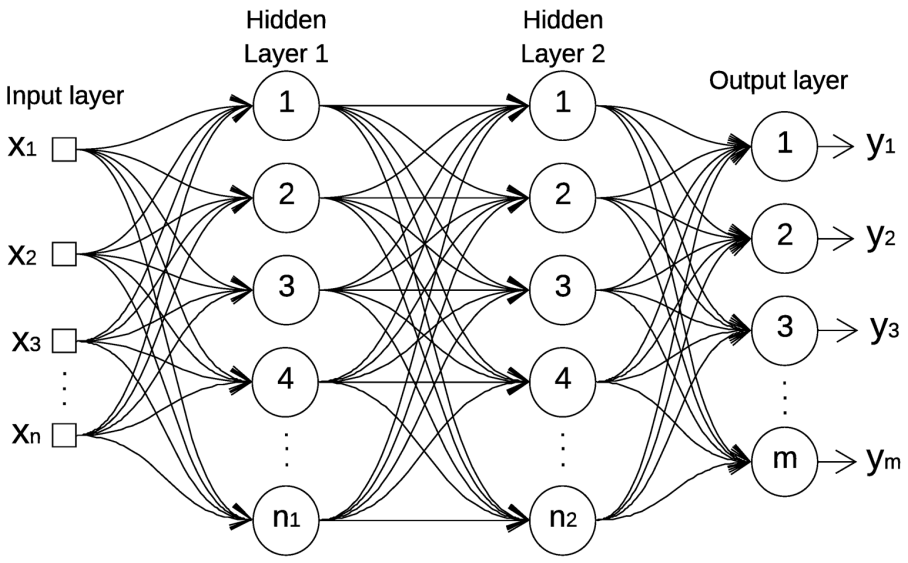

Figure 2.1: Schematic representation of an ANN [39]

[1]The game of Go is a famous 2,500 years old game invented in China where two players confront each other on a square board with 19 tiles side length. The goal is to surround more territory than the opponent.

The figure 2.1 represents the most basic NN, which is called a Multi-Layers Perceptron (MLP). In this particular case, each cell (we assume here that cell and neuron represent the same thing) of a layer is connected to all cells of the previous layer as well as all cells of the next layer. The information is then propagated from the inputs $[x_1, x_2, ..., x_N]$ through the network by applying some non-linear transformations on them till reach the last layer, i.e. the output layer $[y_1, y_2, ..., y_N]$. It should be underscored that the weights accomplish the transformation (the arrows) while the neurons (the numbered circles) store the intermediate values. Therefore, the value of the weights are the very parameters of NNs. The information flows through layer by layer and each neuron's value is computed as follow

$$y_j = f\left(\sum_{i=1}^{N} w_{ij}x_i + b_j\right) \tag{2.1}$$

$$= f\left(w_{1j}x_1 + w_{2j}x_2 + ... + w_{Nj}x_N + b_j\right) \tag{2.2}$$

Where

- i is the size of the previous layer.
- j the size of the current neuron's layer.
- w_{ij} represent the weight between the neuron i of the previous layer and the current neuron j.
- x_i is the value stored in the i^{th} neuron of the previous layer.
- y_j is the value that the current cell will receive after computation.
- b_j is the offset of the current neuron j. It is the only value independent of all the inputs.
- f is what is called an activation function. It is often nonlinear, which is precisely the reason of the non-linearity of NNs.

Once the value of each neuron of the same layer has been computed, the algorithm goes to the next layer and does the exact same computations with the values of the previous layer's neurons as inputs, so on so forth till reach the output layer.
On figure below is a scheme representation of the computation for one cell.

The explanations above described how a NN is passing from an input sequence (which might not be a one-dimension array, depending on the problem tackle) to an output sequence. Nevertheless, each network is created and specifically fitted to a given problem and the processes behinds involve a whole series of calculations and optimizations.

Initially, the weights linking the neurons are randomly set and do not convey coherent information from the input to the output layer. They have to be adapted to apply the right modifications on the input so as to obtain the corresponding output. This procedure, called *training*, will progressively modify the value of the weights through a somewhat guided trial-and-error process by minimizing an error function. The error function, also called the *loss* function, determines how far a generated output is from the actual targeted data. As an example, a classical loss function is the Mean Squared Error (MSE), given by

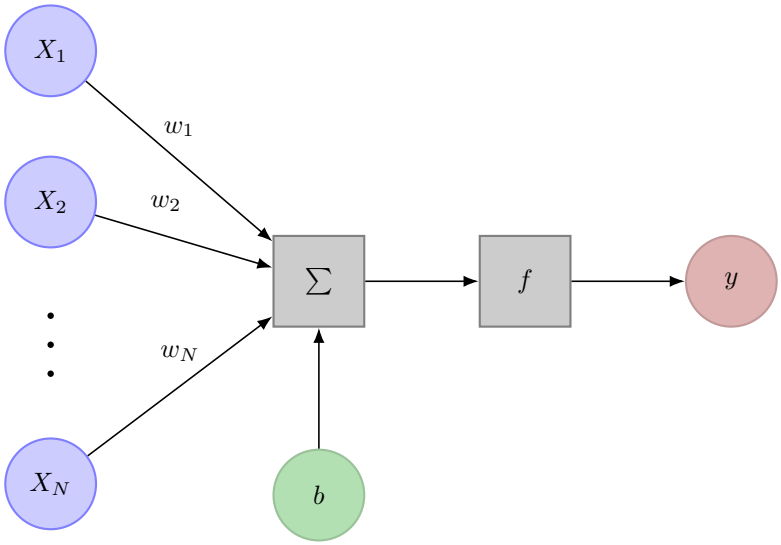

Figure 2.2: Graphical representation of a neuron insight a neural network

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{2.3}$$

where the $\hat{y}_i$'s are the values generated by the network and y_i's the goal values. Therefore, the MSE gives an idea of the *goodness* of the proposed output. Then, a backward pass is conducted in the opposite direction (i.e. from the output layer to the input layer) to slightly modified each weight iwith the purpose of diminishing the aforementioned error function value at the next forward pass with a new input.

In summary, from a list of paired vectors, i.e. inputs and their corresponding outputs, the algorithm extracts one pair at a time, feeds the NN with the input vector, compares the output computed by the network and the actual aimed output, evaluates the error and then modifies the weights accordingly with the intent of obtaining a smaller error with the next input-output pair.

Of course, there exist many different types of NNs whose net design are more or less suited for one problem to another. In the next section, the functioning of what is called a Recurrent Neural Network (RNN) is detailed. The specific structure of this kind of net allows the neurons to memorize the previously stored value, making them more suited for sequence data, which is the present case with temperature records.

2.1.2 Recurrent Neural Network

In some cases, the pattern that a Neural Network tries to understand is continuous, it follows a time series where each point directly depend on the previous one. This kind of pattern is observed in climatology, the features recorded obey physical laws that usually do not abruptly change over a short period of time. Willing to integrate this causal link between two following states, the classic

cells constituting a NN are replaced by some kind of memory cells, that will somehow reintroduce their previous state when computing the next one. When at least on layer of a NN is composed of this kind of neuron, the network is said to be *recurrent*. To graphically represent the different between the two type of neuron, on can see on figure 2.3 below the loop back arrow that take the previous output and reinstate it into the ongoing computation.

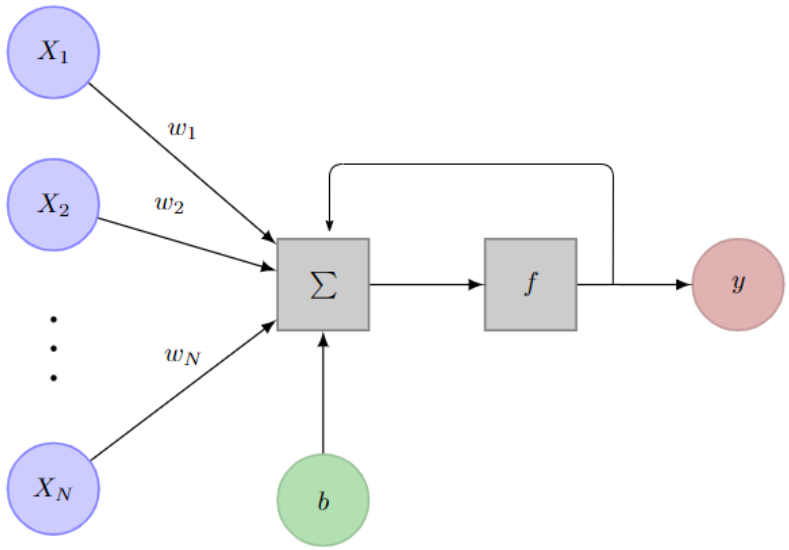

Figure 2.3: Graphical representation of recurrent neuron

Moreover, it exists different types of cells composing a RNN, some of them keeps in memory only the very previous value it encountered, others indirectly memorize the whole time series which passed through them into what is called a *cell state*. A Long Short-Term Memory (LSTM) cell has the latter characteristic in addition to some other features controlling the importance of the previous states that should be considered when processing the current input. For the sake of simplicity, the reader should retain that LSTM networks are a special type of RNNs where each cell is able to learn long-term dependencies.

2.1.3 Random Forest Regressor

A robust approach in classification and regression consists in using algorithm based on a tree structure. Basically, it is represented by a graph composed of nodes and leafs where nodes contain conditions to test and leafs contain values to return.
For example, on figure 2.4 below is drawn a representation of a possible regression tree made from a dataset containing four numeric variables ($[V_1, V_2, V_3, V_4]$). Each combination of these four variables has an output value, which is represented by the green triangles. When it comes to predict the output of a new input sequence, the regression tree starts by testing the condition of the root node, situated at the top. Depending if the condition is satisfied or not, the algorithm is going to go respectively to the left part or the right part and repeat the same operation on the corresponding

subtree. At the end, when the algorithm reach a terminal node, i.e. a leaf, the algorithm simply returns the value contained in it.

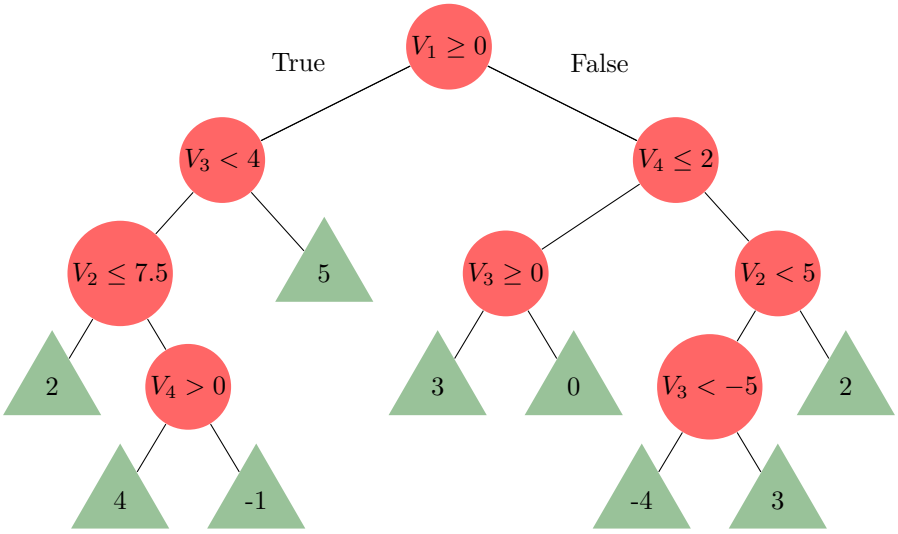

Figure 2.4: Example of a regression tree

Unfortunately, a single tree is not robust, it often leads to overfitting and the prediction accuracy is not great compared to other methods. Nevertheless, in order to keep the simplicity of this kind of algorithm and to simultaneously increase its robustness, one can combine the predictions of many trees and average the outputs to obtain more stable results [4]. As its name speaks for itself, a Random Forest Regressor (RFR) combines the predictions of multiple regression trees in order to make a robust and accurate one while preserving the simplicity of a tree-based algorithm. We used the `sklearn` library from `Python` that implements the RFR algorithm which follows the method proposed by L. Breiman for training the trees [7].

2.1.4 Normalizing Flows

Normalizing Flows (NFs) is defined as a sequence of bijective transformations stacked in series which create a reversible mapping between inputs and a known distribution, which is often the Gaussian distribution in practice. A bijective transformation is an operation that has a unique output for every input and vise-versa. Furthermore, the composition of bijective functions is also a bijective function, meaning that the transformation can be arbitrarily complex.

For an input sequence $\boldsymbol{x} \in \mathbb{R}^N$, a series of bijective transformations $f_\theta = f_1 \circ f_2 \circ ... \circ f_K$ is performed to obtain a latent random variable $\boldsymbol{z} \in \mathbb{R}^N$ with a known and tractable probability density function $p_{\mathcal{Z}} : \mathbb{R}^N \to \mathbb{R}^N$. By means of these transformations, the distribution of the variable of interest can be indirectly inferred through f_θ by

$$p_\theta(\boldsymbol{x}) = p_{\boldsymbol{z}}(f_\theta(\boldsymbol{x}))|\mathrm{D}\boldsymbol{J}_{f_\theta}(\boldsymbol{x})| \tag{2.4}$$

where $\mathrm{D}\boldsymbol{J}_{f_\theta}$ is the determinant of the Jacobian matrix of f_θ with respect to $\boldsymbol{x}$.

Training the model is done by injecting input-output pairs from which the model learns to derive a Gaussian distribution through the maximization of the log-likelihood $\sum_{i=1}^{D} \log p_\theta(\boldsymbol{x_i}, \boldsymbol{y_i})$ for a given dataset of length D. When doing so, the transformations are actually passing from a complex and irregular data distribution to a simpler regular distribution or in other words a *normal* distribution. This being done by successive transformations, it can be visualized as flows which end to end gave the name to this method.

$$\left.\begin{matrix}\boldsymbol{x}\\ \boldsymbol{y}\end{matrix}\right\} \quad \rightarrow \quad f_1 \circ f_2 \circ f_3 ... \circ f_K \quad \rightarrow \quad \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{2.5}$$

Where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the inputs and their corresponding outputs. The importance of the invertibility of the transformations comes into play when the model is then used to make predictions. Indeed, the idea behind the mapping is then to generate samples from the known distribution and to use the inverse transformations to create multiple scenarios explaining the given input (see figure 2.5). Ultimately, a distribution is drawn from these scenarios, which will be the prediction of the model, inherently integrating the uncertainty of the studied phenomenon.

$$\hat{\boldsymbol{y}} \quad \leftarrow \quad f_1^{-1} \circ f_2^{-1} \circ f_3^{-1} ... \circ f_N^{-1} \quad \leftarrow \quad \begin{cases}\boldsymbol{x}\\ \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})\end{cases} \tag{2.6}$$

Of course, the so called transformations can be of any nature as long as they remain reversible. Therefore, in the present case and as it will be explained further, one can train a neural network to learn the mapping between the inputs distribution and the normal distribution.

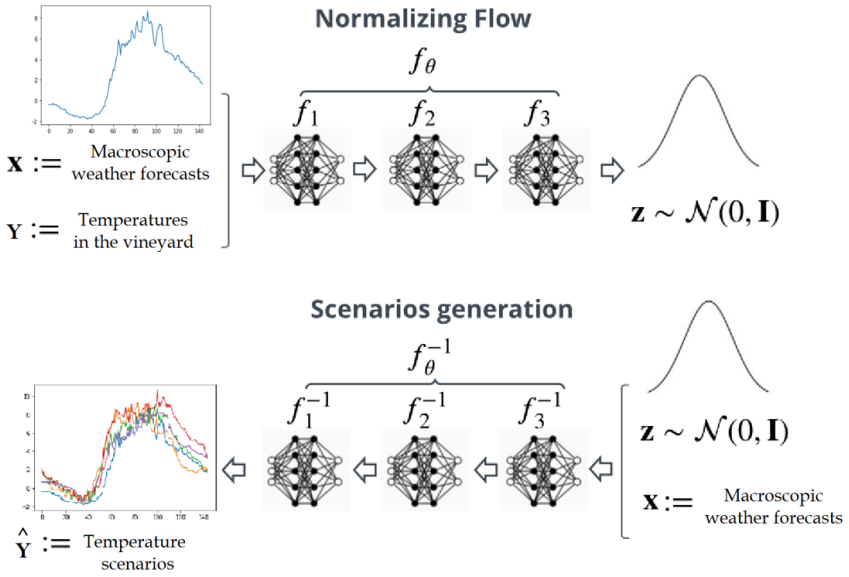

Figure 2.5: Schematic representation of a normalizing flow [13]

2.2 Review of the literature

This section presents a review of the literature discussing ambient air temperature forecasting. The global state of the ML in this field is first investigated, then a brief presentation of already existing similar works is discussed.

2.2.1 Machine Learning in weather forecasting

As mentioned in the introduction section, using algorithms based on ML techniques bring significant improvements in the accuracy of the predictions. Although it is harder to implement and more computationally costly to train, it allows to capture underlying patterns that simpler models do not. It is why their use has been widely spread and many of nowadays weather prediction models are based on such methods [42].

In comparison to this work, there are two important factors that need to be considered, the temporal distance of the prediction, how many hours, days or weeks ahead of the current time and the scale of the prediction, global, regional or location-specific. In the section hereafter is depicted the models presented in *A Review of Neural Networks for Air Temperature Forecasting* by Tran et al. (2021) [42]. The models chosen in this paper have been judged relevant for the discussed topic since they are dated from 2005 to 2020 and talk to very similar works.

In section 2.2.2 is discussed the macroscopic scale predictions with different time ahead values. Low scale weather predictions models are then presented in section 2.2.3 with different time intervals as well.

2.2.2 Regional weather forecasting

The substantial majority of weather forecasting papers found on the internet present models working on a relatively large scale. As a matter of fact, except in some field like agriculture, it is not needed to work with a meter precision when forecasting. Nevertheless, since these models try to predict the non-linear and chaotic pattern of the ambient air temperature, their review remain relevant and insightful.

An interesting article conducted in similar weather conditions, presented by three members of the Ulm University of Applied Sciences in Germany, have developed three distinct models and have analyzed their performances to predict local temperature up to 24 hours ahead [28]. The first two models are shallow nets, both with one hidden layer made of LSTM cells (see section 2.1.2), where the first one takes only temperature as input (univariate) and the second one takes additional features such as humidity and pressure besides temperature (multivariate). As they wisely notice it, the univariate version serves as comparison to assess the precision gained by the extra features. The last model is a deep convolutional LSTM network (see Kreuzer et al. (2020) [28] for additional information about the structure of the latter model and what is a convolution network).

They gathered their data from five stations dispatched in Germany with an hour timestep and took the Root Mean Square Error (RMSE) and the Mean Absolute Scaled Error (MASE) measures to assess the goodness of fit.

Model	Naive	SARIMA	LSTM	LSTM(multi)	convLSTM
MASE	1.00	1.05	1.18	0.99	0.93
RMSE	2.87	2.55	2.83	2.37	2.10

Table 2.1: Results by means of the RMSE and the MASE measures

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\hat{X}_t - X_t)^2} \qquad MASE = \frac{\sum_{t=1}^{T}|\hat{X}_t - X_t|}{\sum_{t=1}^{T}|X_{t-s} - X_t|} \tag{2.7}$$

Where $\hat{X}_t$ is a forecast and X_t the actual data. The MASE is a measure introduced by Rob J Hyndman (2005) [24] that compares a method's average absolute value to the average absolute value of the seasonal naive forecast [2]for a seasonal lag s.

They compared their three models to the naive forecast and the Seasonnal AutoRegressive Integrated Moving Average (SARIMA) model, which basically is a simple common method that computes the next prediction based on weighted sum of the previous observations plus some constants (in-depth explanations can be found in Box et al. (2016) [6]).

In their conclusion, they observed that in the first few hours of forecasting the SARIMA and univariate models perform better. However, after this short period, the multivariate models give better results, especially for the convolution LSTM model. Their results are gathered in table 2.1.

For the station situated in Ulm Germany, they obtained the following standard deviations for the different models at different time distances (see figure 2.6).

In Tran et al. (2020), they investigated the impact of the number of neurons and the depth of an ANN in forecasting maximum temperature time series. With an input of six-features length and a single output, they ranged from one hidden layers to five and to three neurons to 125 in a single layer, resulting in a number of parameters going from 49 to 1001. They assessed the performances of each network by means of the RMSE measure (see equation 2.7) across 55 stations located in South Korea. The data were collected on these same stations with a daily timestep. Globally, for all number of parameters, the RMSEs of their models sprawl around a value of three. Accordingly to what they concluded, they were no significant differences between the different models, which implies that simple ones perform as well as more complicated ones [43].

2.2.3 Location-specific weather forecasting

Luckily, two of the few available papers discussing low-scale weather forecasting tackle the issue of damages caused by frost in vineyards. The two papers have been wrote by the same members of Meiji University and present a model based on the Support Vector Machine (SVM) technique. They first study the causality of each feature in order to keep only the ones that matter and then develop a model to predict frost in a few hours ahead in local area.

The data were collected in Ikeda, a small town of the Hokkaido region situated in the North of Japan, known for is suitable weather for authentic European wine grapes cultivation [1]. With

[2]A naive forecast consists in reusing the actual value at time t for predicting the value at time $t+1$, without readjusting the factors.

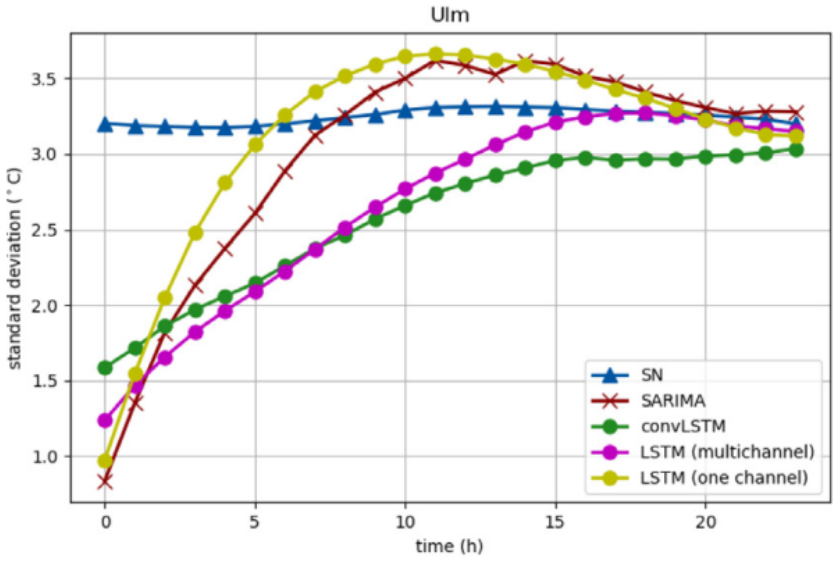

Figure 2.6: Standard deviations of the models presented by Kreuzer et al. (2020) [28]

a timestep of one minute, the sensor equipment has recorded a binary variable for the frost (F) (Yes/No) together with the air temperature (A), the wind speed (W), the relative humidity (H), the net radiation (R) and others.
In the first paper, they conducted a simple correlation analysis between each variable that we just mentioned (see table 2.2) in order to see the individual causal link between each of them and the freezing events.

	A	H	R	W	F
A		-0.438	0.646	0.457	-0.425
H			-0.473	-0.430	0.325
R				0.401	-0.283
W					-0.298

Table 2.2: Correlation analysis of variables from Ding et al. (2019) [11]

Then, they constructed miscellaneous models by using only different subsets of the variables and tried to predict one, two and three hours ahead. Not surprisingly, they concluded that temperature is the key factor, none of their models achieved to predict frost without the involvement of air temperature. They observed that relative humidity tends to be an early predictor on a relatively longer period (two to three hours) and they rejected the wind speed parameter as it appeared very noisy [11].

In the second paper, they focused their researches on vineyards specifically. They decided to use only the ambient air temperature (AT) and the vapor pressure (VP) correlated with humidity to carry out their experiments. Two of the important parameter of the SVM-based model is the

moving average window which controls the length of the previous observations to take into account and the method used to compute this moving average window. They present two methods in their paper, the simple method which considered the k previous observations equally and the exponential method which gives a higher weight to closer observations.

Obviously, they observed that there is a trade-off between the reliability of the prediction and the prediction time (the further in time the less accurate). They concluded that a too long or a too short moving average window has drawbacks as well and an appropriate size must be selected. Finally, they obtained slightly better results when using exponential moving average method in contrast to the simple method. Furthermore, they trained a model with data of April and May and achieved to predict frost in October, showing that the weather specifics of Spring are similar to the ones of Autumn.

Chapter 3

Methodology and data

In order to achieve the goals fixed in the introduction section, a series of steps had to be taken to collect, gather and process the data used for the models. In this section, the materials needed for monitoring weather factors and the Regional Atmospheric Model (MAR) are first presented. Then, in the second part, an exploration of the data as well as the description of its pre-processing is proposed. Finally, we will carry out an analysis of the overall data to leave to the reader a fairly good understanding of what it is made of.

3.1 Materials

In order to create the datasets needed for train and test the models, three different devices were used. Leaving the composition and the role of the data for the next section 3.2 below, the focus is here on the material used and its specificity. First, the MAR model is introduced, which has existed for a few decades now, then we describe the weather stations and the thermal chips that were installed in the vineyards.

3.1.1 The Regional Atmospheric Model

Evidently, local monitoring is not enough to make accurate forecast predictions, the use of a macroscopic model is needed. In this purpose, the MAR model is used in series with the chips to combine regional predictions with local climatic behaviors to make valid forecasting.

The "Modèle Atmosphérique Régional" (MAR) model is a weather forecasting program which has been initially developed to study the ice sheet in polar regions in both Greenland and Antarctica [2, 16, 27, 48]. Recently, the MAR model has proven to be also efficient in more temperate regions such as Belgium [12, 17, 40, 44]. It is based on the Global Forecast System (GFS), which has a $\sim$ 13 kilometers spatial resolution and can produces forecasts for up to 16 days in advance.

Basically, the MAR makes deterministic forecast predictions by simulating the physical interactions between atmosphere, soil and vegetation. Each three hours, its predictions are re-framed by the forecasts of the GFS in order to make the next predictions starting off from the new observations.

Thereafter, the MAR model downscales the global climate predictions to pixels of five-kilometers side length.

The computer programs involved in the realization of such a complex model are time consuming and hence cannot deliver actual live forecast. This means that at a given time t, the model takes all the information gathered up to this moment and starts the computations. Then, after approximately six hours of data processing, the model outputs 24 hours of forecasts starting at the horizon initially scheduled at time t. Thus, for the 12 hours forecast horizon case it schematically gives

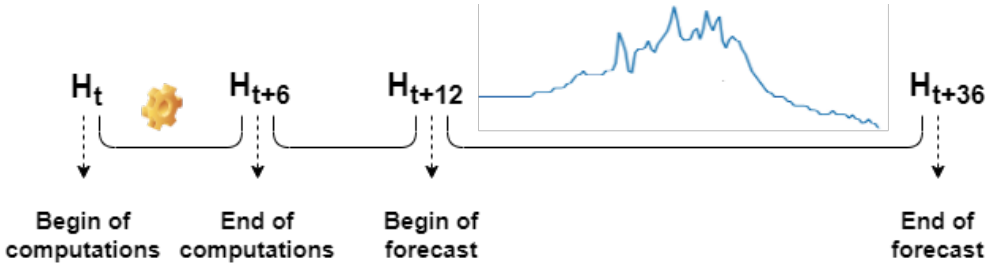

Figure 3.1: MAR forecasting representation for the 12 hours horizon

where $\boldsymbol{H_t}$ is the state of the forecast file at time t.
Consequently, each MAR output file is made of gradually augmenting forecast horizon sequences of 24 hours length, where the first row is 12, 24 or 36 hours ahead and the last row is respectively 36, 48 or 60 hours ahead.

Thanks to the Climatology and Topoclimatology Laboratory of the Liège University, which is in charge of this program, we have access to the latter predictions 12, 24 and 36 hours ahead.

3.1.2 The weather stations and the thermal chips

The stations used to monitor the weather parameters within the vineyards come from `Davis Instruments`, an American company which develops and manufactures systems, sensors and equipment for weather monitoring. The specific model used is the wireless model 6152FR with a passive shelter and air renewal in the sensor chamber. On table 3.1 can be found the specificity of the stations.

In contrast to the MAR and the weather stations, the chips only monitor temperature, their characteristics can be found on table 3.2.

Unfortunately, the chips do not have the same surrounding structure than the stations and suffer from biased recordings due to the sunbeams. Indeed, for example we can see on figure 3.2 that the curves representing the temperature recorded by the chips are highly irregular. Nevertheless, although this phenomenon might biases the model predictions during daytime, the focus in this thesis is on temperature forecasts during night, which is not subject to this issue.

In total, we used two weather stations (one in each vineyard) and 18 thermal chips (six in the Henri Simon vineyard and twelve in the village vineyard). On figure 3.3 & 3.4 can be found a schematic representation of both vineyards with the positions of the stations and the thermal chips.

Monitored feature	Resolution	Range	Accuracy
Temperature	$0.1°C$	$[-40°C; +60°C]$	$0.5°C$
Humidity	1%	$[0\%; 100\%]$	3%
Dew point	$1°C$	$[-76°C; +54°C]$	$1.5°C$
Barometric pressure	$0.1hPa$	$[880hPa; 1080hPa]$	$1.7hPa$
Wind speed	$0.5m/s$	$[1m/s; 54m/s]$	$1m/s$
Wind direction	$1°$	$[0°C; 360°C]$	$7°C$
Solar radiation	$1W/m^2$	$[0W/m^2; 1800W/m^2]$	5%
Rain	$0.25mm$	$[0mm; 999.9mm]$	4%

Table 3.1: Characteristics of the stations used in the two vineyards.

Monitored feature	Resolution	Range	Accuracy
Temperature	$0.06°C$	$[-40°C; +85°C]$	$0.5°C$

Table 3.2: Characteristics of the thermal chips used in the two vineyards.

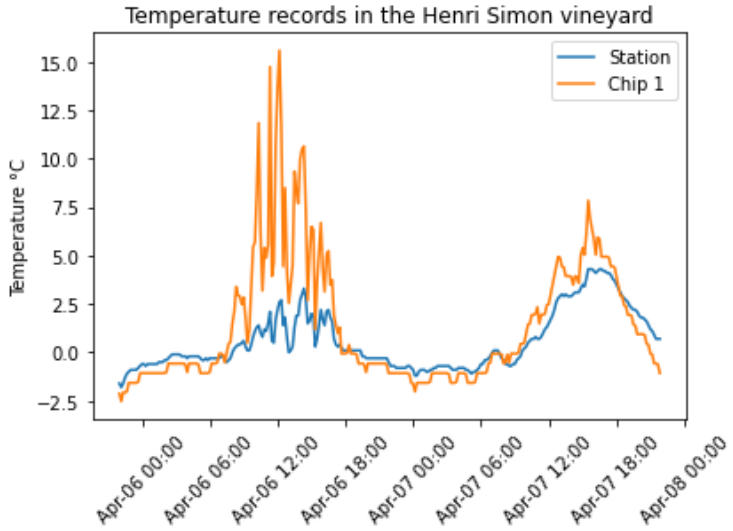

Figure 3.2: Example of chip biased records due to sunbeams

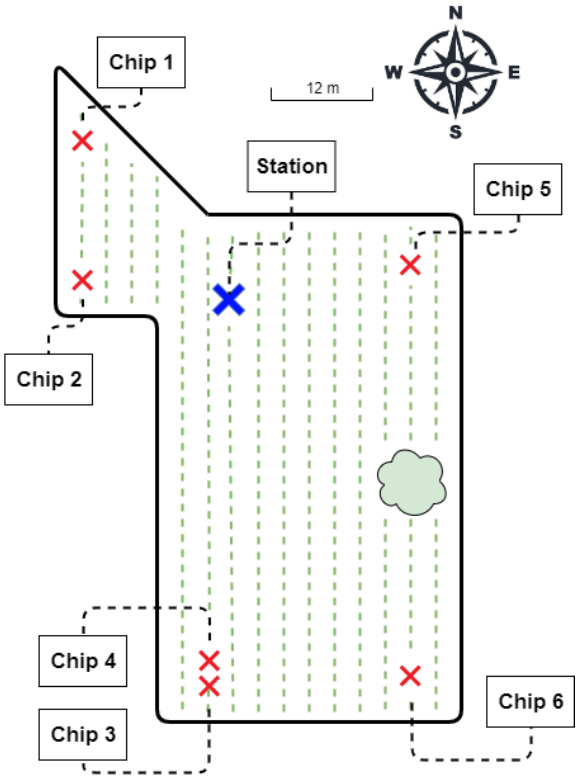

Figure 3.3: Schematic representation of the Henri Simon vineyard

3.2 Data

In keeping with the previous section, the three aforementioned types of devices hence generate three different datasets. Each of them has its own set of feature and covers completely or partially a full year time interval. Except the one 24 hours horizon dataset coming from the MAR, they contained more or less large slots of missing data due to technical issues. All the data were collected in or are related to two vineyards both situated in Lincé, a small village in the region of Liège (see figure 3.5). The first one, called the *Henri Simon*[1]vineyard is exposed to the South. The second one, called the *village* vineyard, is divided into two parts, one faces West and the second side faces South.

The first dataset was created from the **forecasts** made by the MAR model for the region including the two vineyards. It is worth noting that these are therefore outputs of the deterministic model discussed in section 3.1.1, not actual records of the real weather. Hereafter are the list of the features recorded and their meaning.

1. **Temperature :** given in degree Celsius [°C].

[1]Henri Simon is a famous writer in Wallonia during the 18^{th} and 19^{th} century who lived in Lincé. We gave his name to the vineyard because of a statue of himself standing nearby.

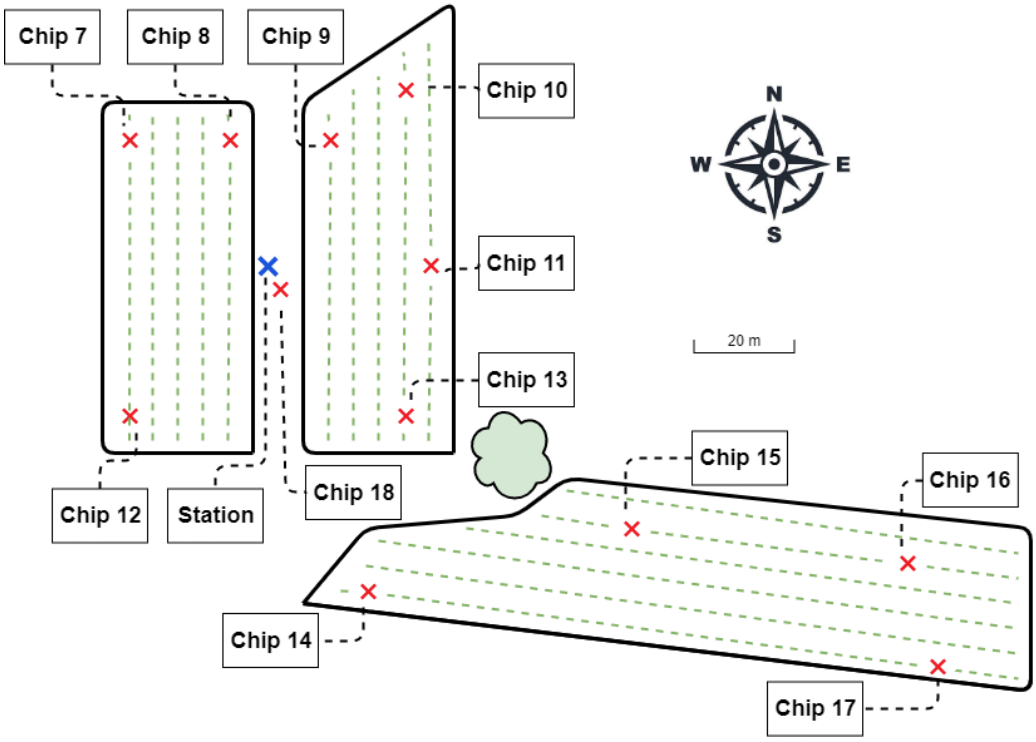

Figure 3.4: Schematic representation of the village vineyards.

2. **Relative humidity :** expressed as a percentage (given by a value comprised in $[0; 100]$), it is defined as the ratio of the partial pressure of water vapor in the air to the equilibrium vapor pressure of water at a given temperature. In other words, it indicates the content of water vapor in the air with respect to its maximal capacity in given conditions.

3. **Dew point :** given in degree Celsius [°C], is the temperature to which air must be cooled to become saturated with water vapor.

4. **Pressure :** given in hectopascal [hPa].

5. **Wind speed :** given in meters per second [m/s].

6. **Wind direction :** given in degrees [°], where the 0° and 360° degree is North.

7. **Solar radiation :** given in Watts per square meter $[W/m^2]$, measures the intensity of the sun's radiation reaching a horizontal surface.

8. **Rain :** given in millimeters [mm].

9. **Wind gust :** given in meters per second [m/s]

Figure 3.5: Map of Belgium with the location of the vineyards from which the data comes from

10. **Low cloud coverage :** expressed as a percentage (given by a value comprised in $[0; 1]$), it represents the fraction of the sky covered by clouds which spans up to an altitude of 1800 meters.

11. **Medium cloud coverage :** expressed as a percentage (given by a value comprised in $[0; 1]$), it represents the fraction of the sky covered by clouds situated between 1800 and 6300 meters of altitude.

12. **High cloud coverage :** expressed as a percentage (given by a value comprised in $[0; 1]$), it represents the fraction of the sky covered by clouds lying above an altitude of 6300 meters.

The data coming from the MAR predictions is divided in three parts, the 12 hours, 24 hours and 36 hours forecast horizons.

The second dataset, which is composed of two parts, comes from the weather stations installed in the two vineyards, one in each. Contrary to the MAR model, they record the real weather conditions in the vineyards at approximately 1.5 meters above the ground. The recorded features are the same as the eight first ones of the MAR model just listed above.

Finally, chips record the temperature at approximately 0.5 meter above the ground (for the most of them) and have been scattered within the two vineyards, six in the Henri Simon vineyard and twelve in the village vineyard. On table 3.3 below is a summary of the features collected by each devices.

MAR		Weather stations		Chips	
Variable	Unit	Variable	Unit	Variable	Unit
Temperature	$°C$	Temperature	$°C$	Temperature	$°C$
Humidity	%	Humidity	%		
Dew point	$°C$	Dew point	$°C$		
Pressure	hPa	Pressure	hPa		
Wind speed	m/s	Wind speed	m/s		
Wind direction	°	Wind direction	°		
Solar radiation	W/m^2	Solar radiation	W/m^2		
Rain	mm	Rain	mm		
Wind gust	m/s				
Low cloud coverage	%				
Medium cloud coverage	%				
High cloud coverage	%				

Table 3.3: Summary of the features recorded by the MAR, the weather stations and the thermal chips.

As briefly explained in the background section, the models need first to go through a training step before making any prediction. Subsequently, the model need to go through a *validation* and *testing* stage to assess its accuracy. The reason is that after training it is essential to be able to evaluate the goodness of fit of the model to decide if it has properly captured the mechanisms of the assigned task or not. If the model does not reach satisfactory results, i.e. if it does not perform well on the data associated with the validation process, it goes back to the training step. Eventually, the model shows satisfying outcomes and is ultimately tested to estimate its accuracy on future predictions (see figure 3.6).

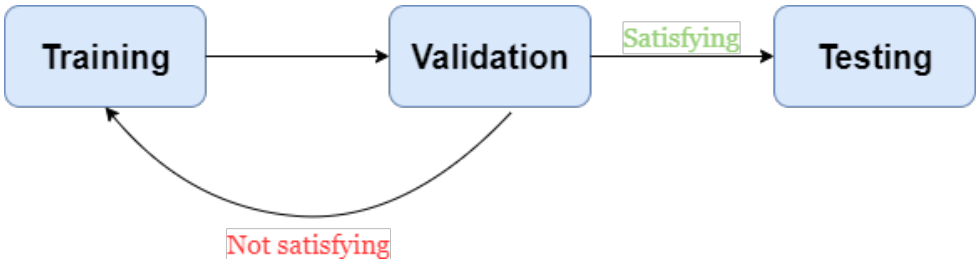

Figure 3.6: Stages for the creation of a model.

The difference between the validation and the testing stage lies on the principle that the data used to evaluate the final model needs to be completely independent from the data used in the training process. Since the validation somehow influences the model parametrization by deciding if it should keep training or not, it is good practice to split the two operations.
As implicitly mentioned, all these stages go along with their own distinct dataset to not bias the testing and validation step. As a matter of fact, models explicitly or implicitly memorized the data used for training, if the latter is reused for the validation and/or testing process, the results will henceforth not reflect the actual accuracy. Therefore, the datasets are all divided in three parts, each allocated to one of the stages described above.

3.2.1 Data transformation

The data exploited for the development of the models come from different sources and have therefore divergent standards. Globally, they contrast in the units, which is easy to standardize, and the frequency at which each measure is made. In the latter issue, a timestep of ten minutes has been chosen and set at fixed times. So each variable has been re-scaled and shifted to obtain a dataset where each row index is a timestamp with rounded time. In other words, a complete recording day starts at $00:00$ and finishes at $23:50$ with a measure every ten minutes.

Obviously, time is an important factor when it comes to make forecast predictions, a night in the middle of winter is going to be much colder than a sunny afternoon during summer. From this assumption, we need to efficiently integrate this information into the data in order to make it relevant for the models. As a matter of fact, time follows a cyclical logic which does not appear under a classical notation. When passing from December (12^{th} month of the year) to January (1^{st} month of the year), it goes numerically from twelve to one, which may be misleading for a learning model. The hours in a day and the days in a month raise the same issue.

To address this problem, the months, the days and the hours of a given record have each been decomposed into two cyclical variables : their sine and their cosine. By doing so, time representation is unambiguous and perfectly integrates its periodic pattern. Both values are necessary since during a sine or cosine oscillation it goes two times through each value. The explained transformation can be summarized as follow :

$$M \quad \begin{array}{ll} \rightarrow & M_{cos} = \cos\left(\frac{2\pi}{12} \times M\right) \\ \rightarrow & M_{sin} = \sin\left(\frac{2\pi}{12} \times M\right) \end{array} \tag{3.1}$$

$$D \quad \begin{array}{ll} \rightarrow & D_{cos} = \cos\left(\frac{2\pi}{31} \times D\right) \\ \rightarrow & D_{sin} = \sin\left(\frac{2\pi}{31} \times D\right) \end{array} \tag{3.2}$$

$$H \quad \begin{array}{ll} \rightarrow & H_{cos} = \cos\left(\frac{2\pi}{24} \times H\right) \\ \rightarrow & H_{sin} = \sin\left(\frac{2\pi}{24} \times H\right) \end{array} \tag{3.3}$$

Where M, D and H respectively represent the months, the days and the hours and the X_{cos} and X_{sin} indices represent the cosine and the sine of the value to which it is attached.

Similarly, the wind direction, initially given in degrees where the $0°$ and $360°$ degree is North, was modified in order to integrate the cyclical logic of the direction. Again, numerically speaking,

passing from 359° to 1° results in a relatively large difference, although it is quasi negligible in terms of direction. The change is

$$W_d \quad \begin{array}{l} \rightarrow \\ \rightarrow \end{array} \quad \begin{array}{l} W_{cos} = \cos(W_d) \\ W_{sin} = \sin(W_d) \end{array} \tag{3.4}$$

Where W_d, W_{cos} and W_{sin} are respectively the wind direction (in degrees), the cosine and the sine of the wind direction.

On figure 3.7 below is shown the representation for the hours in a day and the representation for the wind direction by their sine and cosine waves.

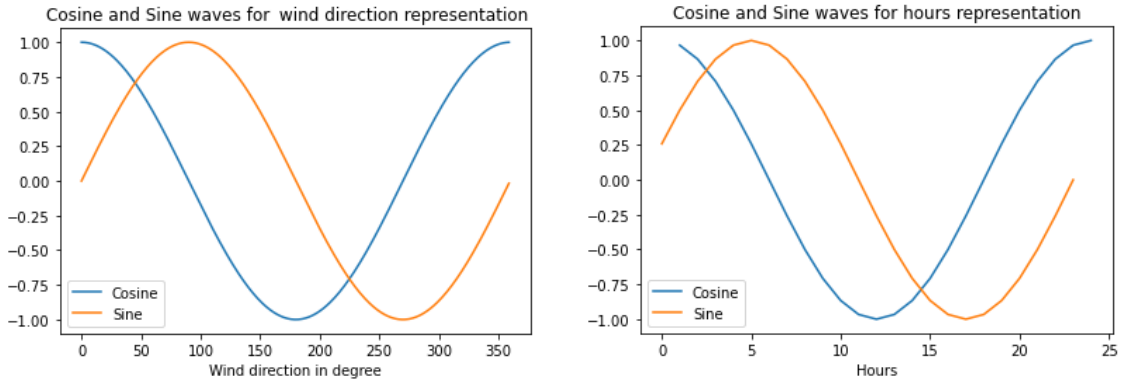

Figure 3.7: Cosine and sine waves representations

In addition, the weather stations integrate in their software the daylight saving time, forcing a shift of one hour from the end of March to the end of October every year of data.

Finally, the data is composed of a variety of weather factors that each comes with a different scale. Indeed, the sines and cosines of the time variable and the wind direction are ranging between -1 and 1 whereas the pressure is expressed in thousands. Before using the data, it is good practice to scale the data in order to have a more stable training process [3]. Therefore, all the data used for training the different models were previously scaled.

3.2.2 Data analysis

In this section, a meaningful analysis is conducted in order to have a fairly good understanding of what the data is made of. Both relationships between the elements within a same dataset and elements coming from different datasets are exhibited with expressive plots. A brief summary of the missing data across the different sources of data is also presented.

Let us first analyze the MAR dataset, which is the one that covers the less temporal space. As a matter of fact, its homonym model keeps in memory up to the past six months forecasts, knowing that the first data collection has been made in early April, it leaves the datasets filled with only a nine months year cover. Hopefully, the fact that there is forecasts with a 12, 24 and 36 hours

	12h forecast	24h forecast	36h forecast
Date begin	03/10/2021	14/01/2022	14/01/2022
Date end	11/07/2022	12/07/2022	12/07/2022
Total number of records	40 607	25 919	25 919
Number of missing/corrupted records	869	0	724
Number of common records with the chips in the Henri Simon vineyard	188 240	122 378	118 904
Number of common records with the chips in the village vineyard	366 258	227 294	220 927

Table 3.4: Summary of the MAR dataset

horizon enlarges the amount of available data from the MAR forecasts.

The table 3.4 gather all the information about the dataset content, where a record stands for a complete measure of all the factors at a punctual moment (with a measure every ten minutes). In addition, since this dataset will be combined with the thermal chip records for training and testing the models responsible for the forecasting, the total number of input-output pairs for each vineyard is laid out in the two last rows.

Where the common records are the sum of the common records between the MAR dataset and each individual chips within the corresponding vineyard.

In order to visualize the underlying relationships between the different weather factors, on figure 3.8 below are displayed the correlation plots between the features of the MAR dataset for the three different forecast horizons. Not surprisingly, except for the wind speed parameter that slightly differs from the 12 hours horizons to the two other ones, no notable differences exist between those plots. The temperatures recorded by the chip n°1 have been added to quantify the correlation with the different factors. As expected, both temperatures are highly correlated, with a value close to 0.92 for every forecast horizons, as well as the humidity and the dew point.

The data relative to the stations is composed of two datasets corresponding to the two stations. They have been installed in the first half of February 2021 and therefore record data ever since. Nevertheless, some time slots are missing due to their small storage capacity, up to only two weeks, leaving empty slots among the records when nobody was there to free their memory. In the same optic as with the MAR data, a summary of the two station datasets can be found on table 3.5.

The correlation plots of the stations records were displayed as well (see figure 3.9). Certainly, the same relationships between the factors can be observed with again no notable difference between the plots.

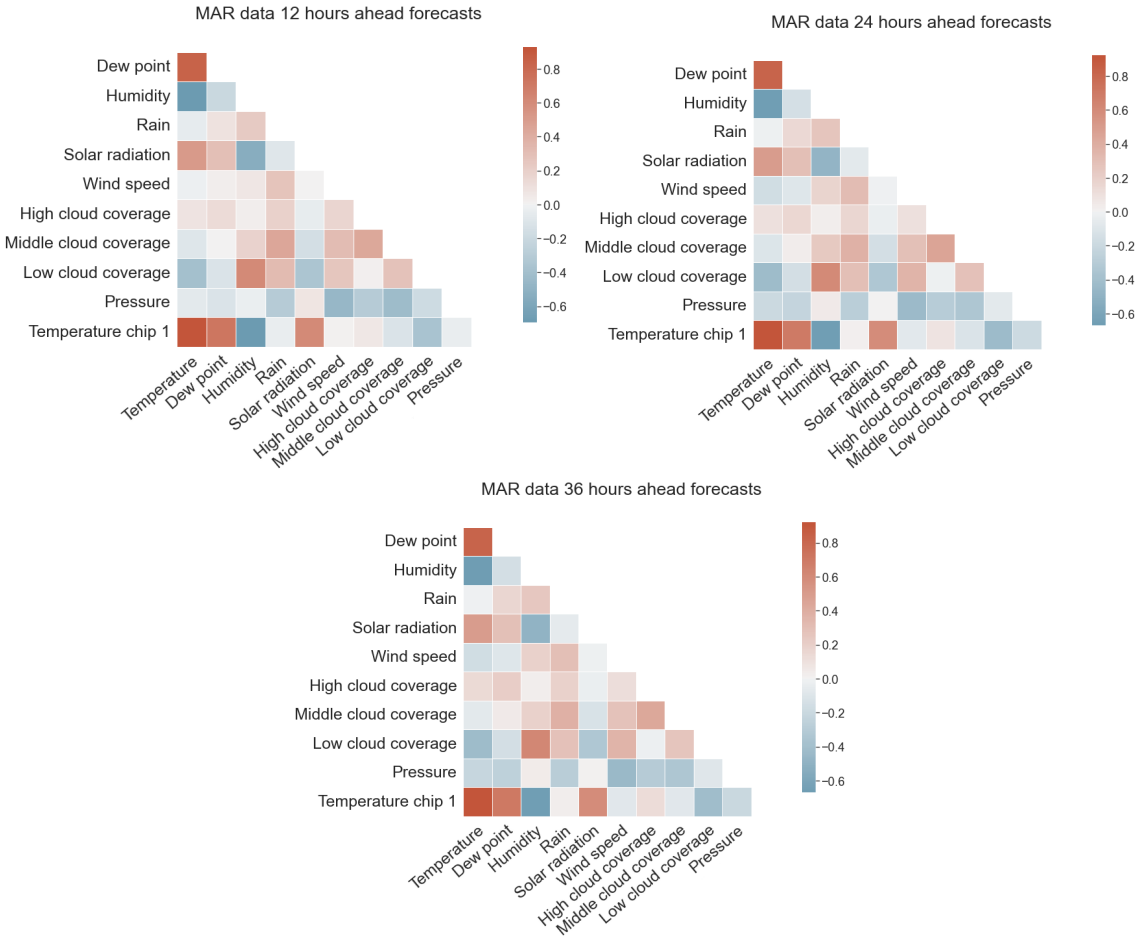

Figure 3.8: Correlation plots of the MAR data for the three forecast horizons

An interesting point to mention is the relationship between the temperature, the relative humidity and the dew point. Indeed, one can notice that on the five correlation plots (see figure 3.8 & 3.9) it seems that the correlation values between those three elements are invariant. As a matter of fact, a relatively good approximation of the dew point temperature for relative humidity above 50% is given by

$$T_d = T - \frac{100 - RH}{5} \qquad \text{or} \qquad RH = 100 - 5(T - T_d) \tag{3.5}$$

where T_d is the dew point temperature, T the temperature and RH the relative humidity. It is clear that the dew point directly depend on the temperature and that the relative humidity is inversely proportional to it. In addition, the equation on the right corroborates the definition given in section 3.2, when $T = T_d$ the air is saturated and the relative humidity is at 100%.

	Henri Simon vineyard station	village vineyard station
Date begin	09/02/2021	09/02/2021
Date end	24/06/2022	24/06/2022
Total number of records	71 978	71 985
Number of missing/corrupted records	16 293	16 345
Number of common records with the chips	294 724	566 579

Table 3.5: Summary of the station datasets

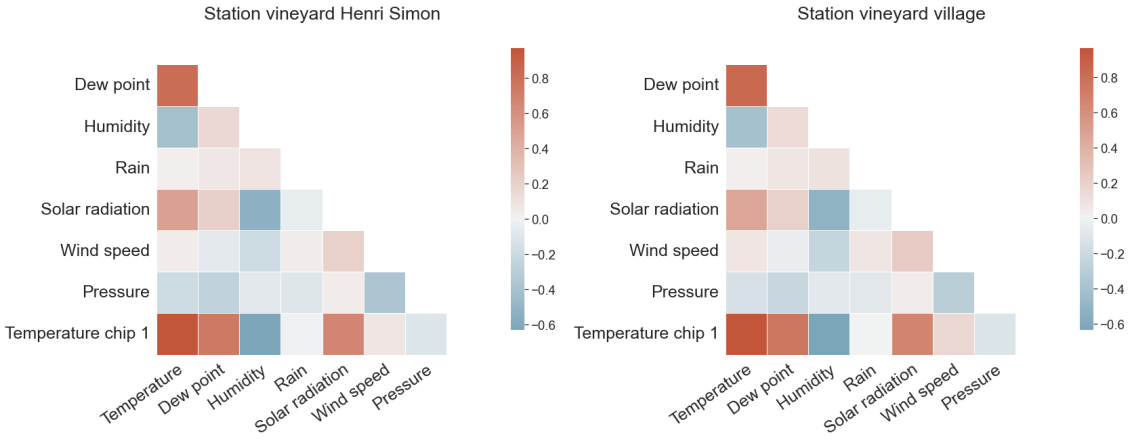

Figure 3.9: Correlation plots of the variables of the station situated in the Henri Simon vineyard (left) and in the village vineyard (right)

Finally, the last set of data is composed of all the chip records, where each one has its own dataset with its amount of missing row. Their small size comes with a low memory and battery capacity, which implies to empty and recharge them approximately every month and half. That is, they have to be regularly removed from the vineyards and therefore no longer monitor the temperature during a few days. Moreover, some of them appear to have serious time shifts in their records and others even completely stopped working.
However, although those conditions involved a lot of missing values, the amount of data remained widely sufficient to train and test the different models.

On figure 3.10 below are the scatter plots between the temperatures recorded by the village station and the temperatures recorded by the chip n°7, 13, 16 & 18. In these plots, the temperature records have only been kept after sunset and before sunrise in order to avoid biased values due to the sunbeams impacting the chips. The chips 7 and 13 are both approximately 50 meters away from the station, in opposite directions (see figure 3.4). Nevertheless, although the distance between the two devices is relatively small, large discrepancies are observed on the scatter plots.
On the other hands, the chips 18, which is place at $1.5m$ above the ground just next to the temperature sensor of the station, assesses the effectiveness of the chips (its points cloud perfectly lies on the $y = x$ line, see bottom left plot on figure 3.10).

As it is relayed by the scatter plots of the chips 7 & 13, one can notice that the area surrounding the former chip appears to be subject to colder temperatures than the latter. From a topographical point of view, this behavior can be explain by the fact that the chip $n°7$ lies lower in altitude than the $n°13$ ($242m$ vs. $248m$) and is exposed to a phenomenon of stagnating cold air. When the wind is null and the air is not stirred up, the air starts to stratify with the coolest parts lying in the bottom. On slopping grounds, the air simply goes down the hill and does not aggregate. But if some natural barriers such as hurdles or in this case a ground conductive to the formation of a cold-air pool, the cold air is trapped and therefore threatens the vines.

On the two plots on the upper right of the figure 3.10, some particular points have been highlighted in red. These points represent one of the main problematic of this work, they express the existing discrepancies between the measures made by a global temperature measure situated at $1.5m$ above the ground (a.k.a. the station weather sensor) and the different parts of the vineyard at the buds level (a.k.a. the thermal chips). Indeed, they have been pointed out because there is at least a difference of temperature superior or equal to $2°C$ between the station and the chip sensors with a temperature below zero for the chip sensor. Some extreme points even reach nearly $-5°C$ (lethal for sprouted buds) according to the chip whereas the station records a temperature of $\sim 2°$ (safe temperature). In turn, the green dots represent the inverse issue, less threatening but troublesome still since they denote false freezing events. Nevertheless, in the latter case, their sparsity suggests that they are punctual events probably caused by external factors.

Now that the data composition and the relationships binding the different datasets as well as the recorded factors and the different available tools have been presented, the structure of the models that will process all this information are introduced in the following chapter.

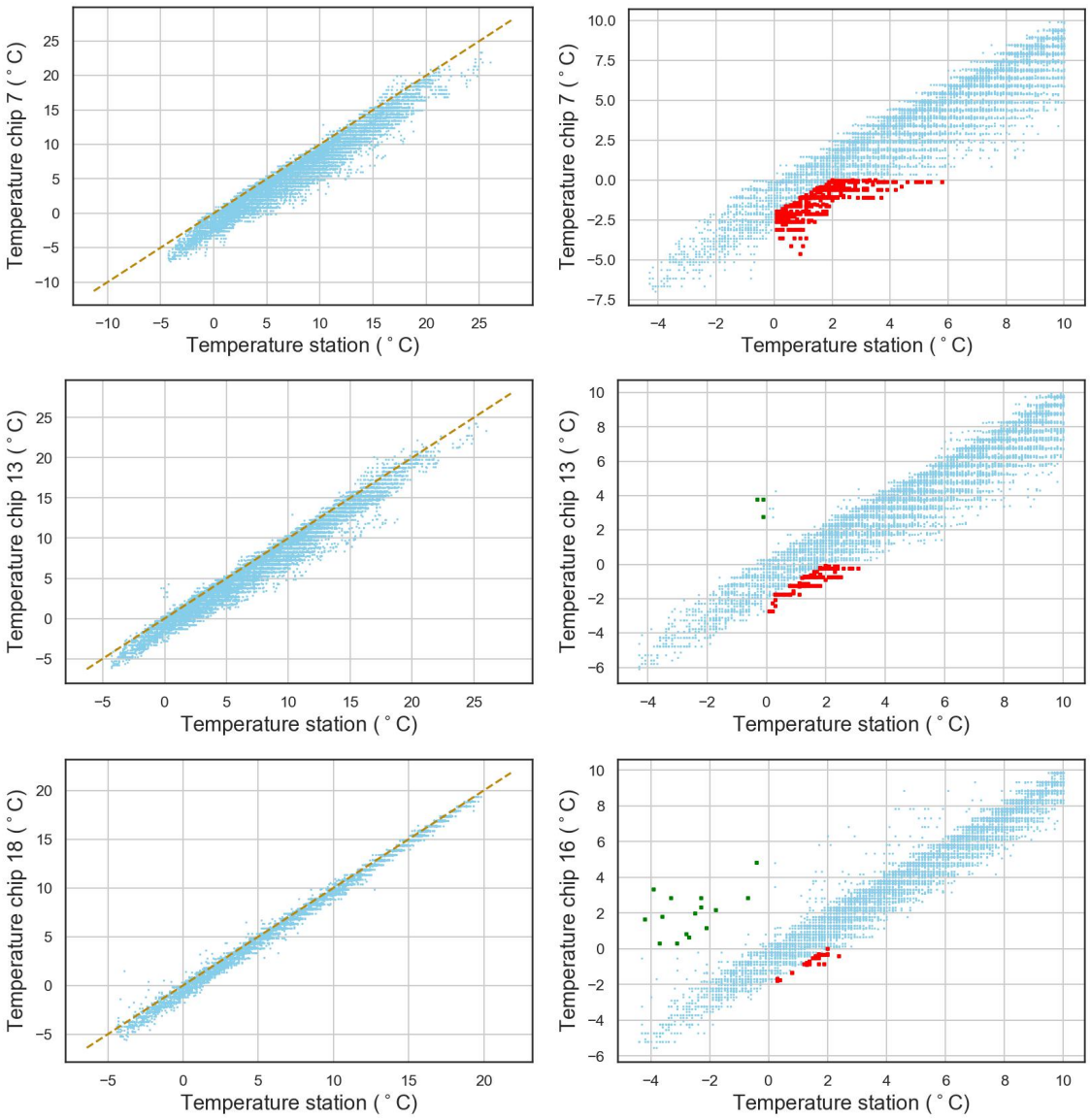

Figure 3.10: Scatter plot between the temperatures recorded by the station situated in the vineyard village and the chip 7, 13, 16 and 18.

Chapter 4

Models

In this chapter is presented the architecture of the models as well as the steps taken for their parametrization. The following will be divided into two main sections, the first one will expose the forecasting algorithms, also referred to as the *MAR-to-chip* algorithms, while the second one will address the live supervising models, also referred to as the *station-to-chip* models. But foremost, a brief summary on how the confidence of the models are assessed is introduced.

As a reminder, the forecasting models take as input the deterministic predictions of the MAR and downscale them through probabilistic processes to obtain location-specific forecasts. For their part, the supervising models take in the live recordings of the weather stations situated in the vineyards together with a 3D-coordinate vector and estimate the temperature at the given position.

4.1 Uncertainty

In air temperature forecasting, due to its chaotic nature, uncertainty is a major factor when assessing the liability of a prediction. As a matter of fact, on figure 4.1 below one can observe that the temperature within the month of April 2021 for a given hour and position can greatly vary.

As one can see, for a given hour, the temperature can vary sometimes more than $10°C$. Consequently, forecasting processes are inevitably subject to a certain variance that at some point can no longer be further reduced.

In probabilistic models, a prediction is significantly more informative when it is coupled together with what is called a *prediction interval* that expresses the confidence of the model. Indeed, in very sensitive domains such as healthcare, it is very important to assess the certainty of a model, if the latter is not confident, one might not take its prognosis for granted and avoid dramatic consequences. Inversely, when the model affirms a good assurance, one can follow its prognosis and provides adequate solutions.

In the present case, when it comes to forecast temperatures in sensitive periods, wine growers would like to know if the chance of freezing is quasi null or if they will have to wake up eventually. Thus, the forecast prediction interval will be expressed by lower and upper bounds encompassing the forecast curve. Hence, at a given timestep, a model will therefore provide three values, a lower

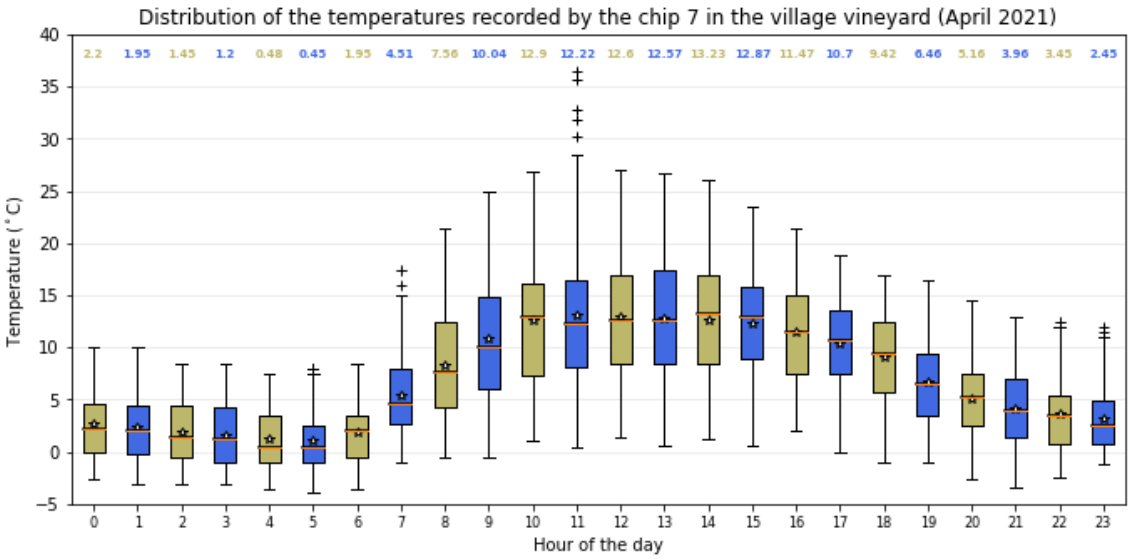

Figure 4.1: Temperature distributions on a hourly basis of the chip 7 in the village vineyard during the month of April 2021

bound and a upper bound expressing that there is 95% chance that the predicted temperature will actually fall between this interval, according to the model, and a center value which represents the actual prediction.

With the RFRs, it is very easy to express the confidence of a prediction since one forecast is made up from the average of a hundred individual predictions. Hence, the interval is estimated by computing the desired quartiles amongst all the generated forecasts. Similarly, the NFs model is what is called a *generative* model, i.e. it generates many scenarios and averages all the values to obtain a forecast distribution. Afterward, from the scenarios, it only remains to also compute the desired quartiles to obtain the bounds.

Unfortunately, with neural networks there is no such averaging-based approach and a solution needs to be established to circumvent this issue. In this domain, no standard method exists and it is up to everyone to adopt his/her own solution to the problematic. One can train the nets on different subsets of the data to obtain a batch of models with their share of forecasts, which again can be used to compute the quartiles. An other can give slightly differing inputs to mimic the uncertainty of an observed phenomenon and see the impact on the output to infer the confidence. Lastly, one can use a Bayesian Neural Network (BNN) which is a classic NN but with integrated noise in the computation from one neuron to another. Basically, instead of having weights made of punctual values, they are each expressed by a normal Gaussian distribution with a certain variance. This means that at each forward pass through the network, the weights will have different values hence generating different outputs. This latter approach has been chosen over the other two because it does not necessitate to train and test many models nor apply modifications on the input data.

4.2 Forecasting models

In order to optimally reproduce the underlying mechanisms governing the relationship between macroscopic weather events and the actual local temperature perceived near grapevines, four different types of algorithm were investigated. The first two rely on Neural Networks (NNs), the third one on tree-based methods and finally the last one on Normalizing Flows (NFs).

Since the beginning of 21^{th} century, NN-based models have been used extensively in air temperature forecasting, but only few studies used learning methods such as RNN and Long Short-Term Memory (LSTM), although they are highly promising [42]. In this perspective, the first two models are a simple Multi-Layers Perceptron (MLP) (which will be used as benchmark) and a RNN with LSTM cells. On the other hand, generic models based on a tree structure have also proven to be efficient in air temperature forecasting [31, 49]. Amongst the wide range of different tree-based algorithms, the Random Forest Regressor (RFR) was judged the most suitable for its computational efficiency and its inherent capability of disregarding corrupted data [25]. Finally, if one look at the temperature along a complete day, it can be seen as an irregular but still consistent distribution of a random variable. Keeping that in mind, it seems reasonable to attempt to predict the temperatures by understanding the key-generation of such distribution. For this task, recent techniques based on Normalizing Flows have proven to be efficient and therefore might be interesting to be investigated [13, 35].

The MLP model is made of three hidden layers with 30 neurons each surrounded by an input layer of size 23 and a single neuron as output layer. The RNN has the exact same architecture than the MLP except an additional hidden layer of size 30 as well. As mentioned in the above section 4.1, in order to introduce noise into the models to mimic the uncertainty of such prediction, BNN were used. Therefore, for both nets the weights linking the input layer and the first hidden layer as well as the last hidden layer and the output layer are both made of Bayesian weights. Their structures can be visualized on figure 4.4 below.

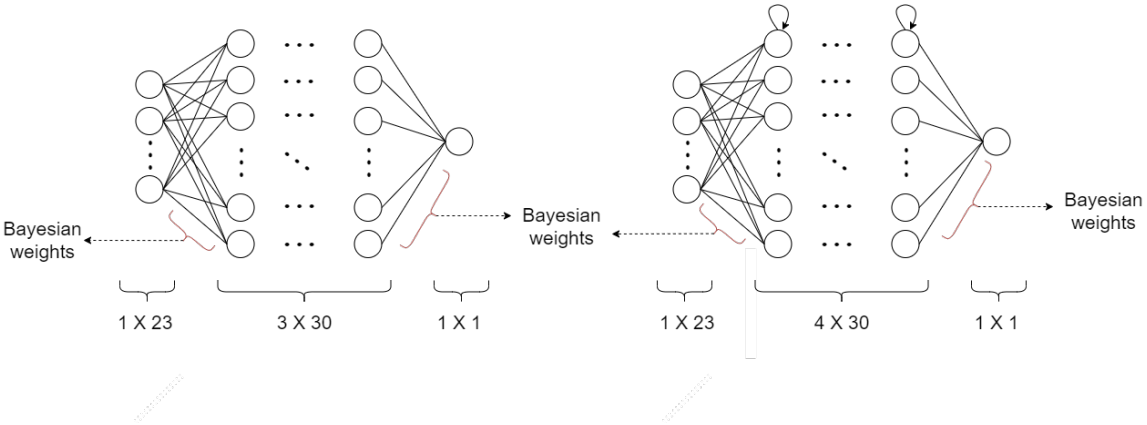

Figure 4.2: Representation of the MLP (left) and LSTM (right) networks used for downscaling the MAR forecasts to the vineyards

In order to determine the number of layers and the number of neurons per layer, a series of em-

pirical tests were conducted to eventually highlight a better performing structure. Unfortunately, any configuration stood out from the crowd and demonstrated substantially better results (see table 4.1 for the MLP model and table 4.2 for the LSTM model). To perform those results, the variances of the Bayesian parts of the networks were set to zero to only consider the structure.

		Layers					
		1	2	3	4	5	6
Neurons	10	1.673	1.732	1.753	1.690	1.725	1.702
	20	1.729	1.726	1.687	1.686	1.708	1.797
	30	1.746	1.687	1.666	1.751	1.861	1.764
	40	1.792	1.707	1.721	1.678	1.768	1.856
	50	1.695	1.714	1.785	1.725	1.722	1.770
	60	1.719	1.791	1.735	1.781	1.783	1.823
	70	1.748	1.782	1.750	1.822	1.831	1.790

Table 4.1: MAE of the MLP forecasting model for different sizes and numbers of layer.

		Layers					
		1	2	3	4	5	6
Neurons	10	1.683	1.769	1.627	1.731	1.774	1.685
	20	1.700	1.747	1.729	1.684	1.794	1.704
	30	1.715	1.734	1.785	1.627	1.708	1.622
	40	1.712	1.748	1.715	1.827	1.797	1.777
	50	1.802	1.701	1.708	1.765	1.653	1.755
	60	1.763	1.738	1.777	1.805	1.747	1.721
	70	1.842	1.739	1.777	1.779	1.725	1.769

Table 4.2: MAE of the LSTM forecasting model for different sizes and numbers of layer.

Obviously, no particular pattern seems to appear when looking across the different tested arrangements. Hence, a partially arbitrarily choice was made by selecting a configuration lying in the middle of the table to avoid too light nor too heavy structure but still keeping a rather satisfying overall score. Those scores have been highlighted in red in both tables 4.1 & 4.2.

The RFR was used with its classic parametrization which consist in one hundred estimators, i.e. the 100 best trees were kept for making forecast, and no maximum depth limit. In every model a sample of one hundred predictions is used to compute the mean and the quartiles, therefore the choice for the number of estimator trees was straightforward. On the other hand, constraining the maximum depth of trees did not substantially impacted the performances. Indeed, the maximum depth reached without any limit is generally around 45. On figure 4.3 below is plot the MAE of the RFR depending on the maximum depth. Obviously, once above 15, the variation between the different depths is negligible and it was decided to not enforce any limit.

The chosen architecture for the NFs model integrally comes from the implementation given in the paper written by Dumas & al. [13]. They developed multiple scenario-based probabilistic fore-

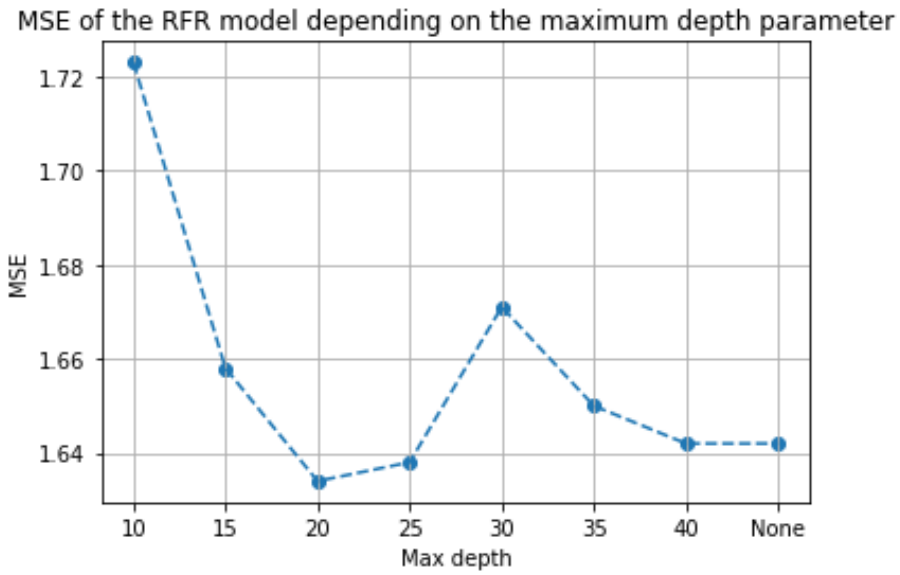

Figure 4.3: MAE of the RFR depending on the maximum depth parameter

casting models to help decision-makers to manage electricity systems based on renewable energy sources. Their models, especially the NFs model, was given inputs such as wind speed or solar radiation, which are exactly the same type of weather factors used in this thesis.

As explained in the background section, the bijective transformations which pass from a complex and irregular distribution to a classical and tractable Gaussian distribution can be made of NNs. In their paper, Dumas & al. decided to use an integration-based transformer by using the class of Unconstrained Monotonic Neural Networks (UMNN) proposed by Wehenkel and Louppe [46]. Without entering into details, under some assumptions this type of net was proven to be very suitable for approximating universal density of continuous random variable. Again, to determine the number of layers and their sizes, an empirical analysis was conducted and revealed no particular pattern (see table 4.3). A UMNN of four layers of size four hundreds each was selected to effectuate the bijective transformations.

		Layers			
		2	3	4	5
Neurons	200	1.911	1.849	1.904	1.931
	300	1.897	1.929	1.909	1.844
	400	1.854	1.908	1.805	1.863
	500	1.809	1.861	1.859	1.867

Table 4.3: MAE of the NFs model for different size and number of layers of the UMNN transformer

Furthermore, the output layer size of the UMNN corresponding to the dimension of the Gaussian latent distribution, which is the "random seed" for the sample generation, has been set to 50.

4.3 Supervising models

In their structures, the models remain globally the same and their hyper-parameters were selected with the same empirical methods.

The MLP model is made of four hidden layers with 60 neurons each surrounded by an input layer of size 20 and a single neuron as output layer. The RNN has the exact same architecture than the MLP except that the hidden layers are made of 50 neurons instead of 60. Again, in order to introduce noise into the models to mimic the uncertainty of such prediction, BNN was used. Similarly to the forecasting NN models, the weights linking the input layer and the first hidden layer as well as the last hidden layer and the output layer are both made of Bayesian weights. Their structures can be visualized on figure 4.4 below.

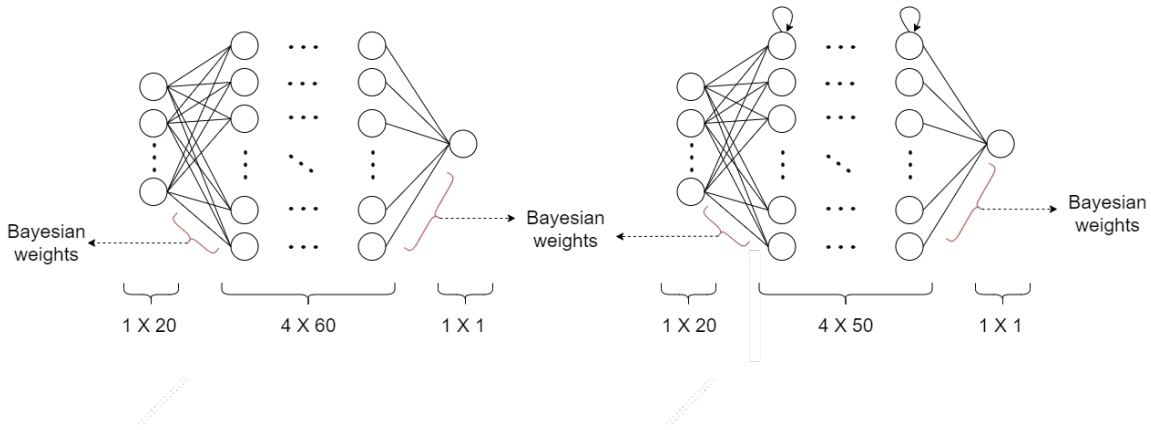

Figure 4.4: Representation of the MLP (left) and LSTM (right) networks used for supervising the vineyards

The results of the empirical analysis are presented on table 4.4 & 4.5 where the chosen sizes and number of layers are highlighted in red.

With regard to the RFR and the NFs models, empirical analysis did not shown any best performing structure neither. Therefore the same constitution and hyper-parameters were maintained.

		Layers							
		1	2	3	4	5	6	8	10
Neurons	10	0.758	0.966	0.849	1.211	0.718	0.745	0.741	0.741
	20	1.638	0.864	0.721	1.737	0.692	0.703	0.694	0.713
	30	0.874	0.978	0.738	1.003	0.687	0.687	0.724	0.669
	40	1.024	0.829	0.800	0.677	0.680	0.698	0.660	0.714
	50	0.710	1.054	0.825	0.697	0.668	0.685	0.686	0.680
	60	0.713	0.672	0.667	0.651	0.679	0.669	0.676	0.693
	70	0.678	0.656	0.663	0.664	0.686	0.677	0.679	0.629
	80	0.679	0.688	0.645	0.655	0.664	0.665	0.664	0.700
	100	0.665	0.661	0.667	0.670	0.675	0.671	0.666	0.670

Table 4.4: MAE of the MLP supervising model for different size and number of layer.

		Layers							
		1	2	3	4	5	6	8	10
Neurons	10	0.736	0.767	0.751	0.734	0.703	0.689	0.710	0.674
	20	0.783	0.718	0.711	0.757	0.663	0.653	0.643	0.640
	30	0.678	0.837	0.746	0.685	0.634	0.634	0.633	0.684
	40	0.809	0.674	0.679	0.646	0.621	0.621	0.626	0.646
	50	0.701	0.718	0.750	0.614	0.624	0.619	0.622	0.621
	60	0.595	0.620	0.625	0.624	0.626	0.628	0.625	0.638
	70	0.616	0.604	0.602	0.597	0.615	0.640	0.620	0.649
	80	0.605	0.602	0.621	0.599	0.623	0.614	0.608	0.638
	100	0.614	0.613	0.606	0.629	0.640	0.621	0.607	0.601

Table 4.5: MAE of the LSTM supervising model for different size and number of layer.

Chapter 5

Results and discussions

The following chapter exposes the results of the different models for forecasting and supervising. A first section is dedicated to the steps taken behind the creation of the training, validation and testing datasets. Then a section is dedicated to expose the established evaluation metrics that will accurately report meaningful insights on the results. Finally, the performances of the different models are laid out.

5.1 Training, validation and testing sets

In section 3.2 we exposed how the data were collected and the different modifications made upon without specifically saying what is actually given to the models as input. This choice is based on the willingness to test different sets of inputs in order to quantify the gained or lost accuracy by using a given variable or not. Of course all possible combinations can not be tested but a subset of it might gives interesting results.

As briefly mentioned in the background section 2.1 on ML techniques, to obtain unbiased results it is necessary to not reuse data already processed by the model. Indeed, especially in the RFR model, if the parameters were toyed on the very data that is used later for testing, the accuracy measures will inevitably be skewed. Additionally, even if the forecasting and supervising models do not receive the same input vector (i.e. respectively a MAR forecast and an actual live recording from a station) and do not have the same purpose, both types aim at the same outputs. Consequently, the data among the two testing datasets needs to have sufficient common records to allow a comparison between them.

On table 3.3 in the section 3.2 has been laid out the different factors that each device is forecasting/recording. Since the deployed models are well suited for relatively large input sequences, no feature was taken out in a first place. Including the transformed data and other additional parameters, a total of 23 input features are given to the forecasting models and 20 to the supervising models (see table 5.1).

The x, y and z Cartesian coordinates represent the position of the chips within a vineyard where the position (0,0,0) lies at the ground level beneath the station. In comparison with the z coordinate

	MAR-to-chip model input features		Station-to-chip input features	
	Variable	Unit	Variable	Unit
1.	Hour cosine	/	Hour cosine	/
2.	Hour sine	/	Hour sine	/
3.	Day cosine	/	Day cosine	/
4.	Day sine	/	Day sine	/
5.	Month cosine	/	Month cosine	/
6.	Month sine	/	Month sine	/
7.	x coordinate	m	x coordinate	m
8.	y coordinate	m	y coordinate	m
9.	z coordinate	m	z coordinate	m
10.	height	m	height	m
11.	Temperature	$°C$	Temperature	$°C$
12.	Humidity	%	Humidity	%
13.	Dew point	$°C$	Dew point	$°C$
14.	Pressure	hPa	Pressure	hPa
15.	Wind speed	m/s	Wind speed	m/s
16.	Wind cosine	/	Wind cosine	/
17.	Wind sine	/	Wind sine	/
18.	Solar radiation	W/m^2	Solar radiation	W/m^2
19.	Rain	mm	Rain	mm
20.	Wind gust	m/s	Slope	/
21.	Low cloud coverage	%		
22.	Medium cloud coverage	%		
23.	High cloud coverage	%		

Table 5.1: Summary of the features recorded by the MAR, the weather stations and the thermal chips.

which is relative to the station altitude, the height represents the vertical distance between a chip and the ground (16 are situated 0.5 meter above the ground and two at 1.5 meters). The slope variable indicates to the model if the chip currently processed is situated on the same slope as the station or not.

The division of the original datasets to obtain the training, validation and testing sets was done as follow. For the sake of simplicity, the size of the datasets will be expressed in days, where one day comprises 144 records in a row (six records per hour times 24 hours). As briefly mentioned in the data section 3.2, for one record of the MAR or the stations, many chips records are available, e.g. with one station record of the Henri Simon vineyard and six chip records made at the same time, one can create six input-output pairs. Although those are different from each other, their inputs are the same and their outputs only slightly differ. Therefore, to obtain fully even-handed results, it was decided to randomly allocate 22 days for the validation set and 25 days for the testing set where none of those time slots will appear in either training set. As just explained, since there is multiple chip records for each day, there are more than 22 and 25 days in the testing and validation set, depending on the number of functioning chips during those periods. Additionally, the ultimate goal of this thesis is to prevent bud freezing during the cold nights of Mars, April and May. Henceforth, three extra nights have been specifically added to the testing set to evaluate the models on this targeted events.
Since the data comes from two different vineyards, all the datasets are therefore made of two parts. The summary of the sizes of each set depending on their utility and the vineyard from which they have been collected can be found on table 5.2.

	MAR-to-chip datasets		Station-to-chip datasets	
	Henri Simon vineyard	Village vineyard	Henri Simon vineyard	Village vineyard
Training set	1031	1944	1763	3277
Validation set	127	254	83	174
Testing set	142	271	131	229

Table 5.2: The dataset breakdown expressed in days

5.2 Evaluation metrics

An important aspect that must be discussed before starting to test the models is the metrics through which they are evaluated. Across the next sections, the performances of the different models will be assessed and compared by means of the classic Mean Absolute Error (MAE) and a lesser-known method called the Continuous Ranked Probability Score (CRPS). The latter can be seen as a kind of extension of the MAE to probabilistic forecasts. While the MAE computes the difference between each point of a forecast and the actual observations, the CRPS computes the difference between the cumulative distribution of all the prediction scenarios and the cumulative distribution of the observations. Actually, when using CRPS with a deterministic model, it turns out to be the MAE. Gneiting and Raftery [20] proposed the following formulation

$$\mathrm{CRPS}(F, x) = E_F|X - x| - \frac{1}{2}E_F|X - X'| \tag{5.1}$$

where X and X' are independent copies of a random variable with distribution function F and finite moment, and E_F is the expectation according to the probabilistic distribution F. In the beginning of the 21^{th} century, the CRPS has attracted a renewed interest in the atmospheric sciences

community simultaneously for its simple implementation and its expressiveness [10, 21, 22, 23, 47]. It is negatively oriented, which means that the closer to zero the better.

As reminded in the uncertainty section 4.1, in probabilistic models, estimating the confidence of a prediction is a primary concern. Therefore, the accuracy of this confidence must be evaluate as well. Gneiting and Raftery [20] also proposed a formula to assess the correctness of such interval called the Interval Score (IS). Let us consider the classic central case $(1-\alpha)\times 100\%$ prediction interval. the lower and upper bounds are respectively given by $l=\frac{\alpha}{2}$ and $u=1-\frac{\alpha}{2}$ and for a given observation x, the IS is given by

$$S_\alpha(l,u;x)=(u-l)+\frac{2}{\alpha}(l-x)\mathbb{1}\{x<l\}+\frac{2}{\alpha}(x-u)\mathbb{1}\{x>u\} \tag{5.2}$$

This scoring rule has an intuitive appeal and the forecaster is rewarded for narrow prediction intervals together with a penalty for observations that miss the interval. A value close to the covering percentage indicates a very good interval prediction with quasi no observations lying outside the intervals whereas a high value indicates misleading prediction intervals.

Keeping in mind that the spring frosts arise during night, a good insight would be to evaluate the errors only occurring overnight. Henceforth, each testing set will have a global evaluation as well as a night-specific one, which takes into account only the predictions past the sunset and before the sunrise.

5.3 Results

Over the two following sections, forecasting as well as supervising models will be evaluated based on the metrics described in the above section. They are compared by means of error and score values but also with graphical visualization of their respective predictions. As explained in the section 3.1.1 talking about the MAR, the latter produces three different forecast horizons. For the sake of clarity, when a time horizon is mentioned it implicitly means there is a 24 hours difference between the first and last predicted time slot (e.g. for the 12 hours horizon, it actually means the forecast goes from 12 hours to 36 hours ahead of the current time, see figure 3.1).

5.3.1 Forecasting models

On table 5.4 below are gathered the results of the forecasting models for both vineyards with the 12, 24 and 36 hours horizons. The best results for each forecast horizon and vineyard have been highlighted with a color attached to the model performing the best (see table 5.3).

MLP	LSTM	RFR	NFs

Table 5.3: Color attribution for the models

At first glance, the yellow and red colors are the most present in the table, indicating that the LSTM and RFR models perform better in general than the other two. Although it was first intended to serve as benchmark, the MLP model yet appears to be the most accurate in two

categories, especially for night forecasts with the 24 hours horizon where it outperforms the LSTM and RFR models by more than half a degree precision. A surprising result is the performances of the NFs model in the 36 hours horizon for night forecasting where the two best models do not manage to go below $2°C$ whereas it reaches a MAE of 1.425°.

	Henri Simon vineyard			Village vineyard			
	MAE	CRPS	Night MAE	MAE	CRPS	Night MAE	
MLP	1.720	1.330	0.722	1.309	1.03	0.510	12h
LSTM	1.628	1.326	0.738	1.189	0.902	0.491	
RFR	1.781	1.325	0.79	1.396	1.034	0.686	
NFs	1.905	1.406	1.458	1.582	1.187	1.298	
MLP	2.452	1.866	2.028	1.887	1.468	1.238	24h
LSTM	1.916	1.560	1.423	1.910	1.632	1.752	
RFR	1.870	1.404	1.567	1.857	1.439	1.860	
NFs	2.408	1.927	1.568	2.252	1.763	1.802	
MLP	2.027	1.510	1.944	1.788	1.381	2.155	36h
LSTM	1.721	1.376	2.010	1.722	1.431	1.650	
RFR	1.863	1.398	2.280	1.744	1.300	1.578	
NFs	2.357	1.957	1.425	1.961	1.481	2.241	

Table 5.4: Errors of the forecasting models with a 12, 24 and 36 hours horizons over three metrics and two vineyards (temperatures expressed in $°C$)

Having the MAE and CRPS to express the errors and compare the models is a good first approach but strictly positive metrics are certainly not enough. Surely, two types of error might occur and their respective consequences have opposite effects. The error is simply computed by subtracting the observation to the prediction, which implies negative and positive errors. In the first case (negative error), the model predicts lower temperatures than the actual ones, which might lead to false alarm. In the second case (positive error), much more concerning, the forecast under estimates the temperatures which ultimately leads to undetected freezing events. To investigate this issue, the error distributions and their corresponding Kernel Density Estimation (KDE) (the blue curve that incorporates the histograms) are exposed on figure 5.1.

As expected, the farthest the forecast horizon the more spread out the error distribution. The two histograms on the top of figure 5.1 are relatively close to a centered normal distribution and have both a majority of errors lying in the interval $[-1; 1]$. As forecasts go further in time, one can notice that the variance increases and the errors lie on a larger range. Ultimately, the NFs and RFR models which have a night MAE of respectively 1.425 $°C$ and 1.578 $°C$ appear to be scatter and hence make very uncertain predictions. Moreover, the 24 hours forecast horizon errors for both

vineyards have their means shifted on the right, which indicates a tendency to over estimate the temperatures, eventually leading to silent frosts in critical cases.

As explained in the uncertainty section 4.1, when providing a prediction, it is important to also address the confidence of the model for that prediction. The latter is evaluated according to equation 5.2 and the results for the different models are displayed on table 5.5.

	Henri Simon vineyard			Village vineyard		
	75%	85%	95%	75%	85%	95%
MLP ($\sigma = 0.5$)	9.447	11.939	22.413	7.853	10.254	20.131
LSTM ($\sigma = 0.5$)	10.285	14.113	29.878	6.536	7.991	13.894
RFR	4.850	5.616	8.521	3.883	4.569	7.138
NFs	5.518	6.614	10.985	5.563	7.065	13.140

Table 5.5: IS of the four forecasting models for three centered intervals (75%, 85% and 95%)

The first thing to notice is the high values for the two NN-based methods, which indicate a poor interval score for this category of model. Nevertheless, the LSTM model in the village vineyard seems to perform better than in the Henri Simon vineyard. The best confidence estimator is undoubtedly the RFR model, which outperforms every other model for all intervals considered.

Globally, the first three models obtained satisfying results on the 12 hours horizon, especially for the LSTM and RFR models that appear to be the two best performers. Indeed, the LSTM model has a night MAE below $0.5°C$ and a standard deviation of $0.64°C$ which is a reasonable accuracy for a prediction up to 36 hours ahead. However, the prediction intervals of the latter remain not quite conclusive. In turn, the RFR model combines a rather good precision ($\sim 0.7°C$) while having a relatively good prediction interval score. Nevertheless, the standard deviation of overnight errors is high ($\sim 0.9°C$) which may not be accurate enough in sensitive periods.

On the other hand, the NFs model appear to be the worst model with error patterns acting strangely in comparison of the three others. As it will be shown further, the latter model does not seem to be able to generate distributions representing well the temperatures within a day.

Unfortunately, beyond 36 hours any model seems to make accurate prediction with a narrow standard deviation. The best model in terms of night MAE is the MLP but a look at its error density (see middle right plot on figure 5.1) reveals a too much spread out distribution. Indeed, the standard deviation of the night errors is $\sim 1.2°C$ which is too high to rely on. Evidently, the forecasts of the MAR are also less accurate, which is reflected through the increasing errors alongside with the forecast horizon.

5.3.2 Supervising models

As the supervision is done in live and with a device situated within the monitored vineyard, the errors will be therefore much smaller than with the forecasters. As a matter of fact, one table 5.6

one can see that the errors and the negatively oriented scores are smaller than for the forecasting ones.
As expected from the previous section, the NFs model does not capture well the air temperature properties with live data neither. Indeed the errors and scores are nearly at least twice as bad as all other models.

The supervising results even more emphasize the better performances of the LSTM and RFR model. As a matter of fact, there is only the colors attributed to those appearing in table 5.6. It seems that the RFR model obtain better results when considering a full day supervision whereas the LSTM stands out during night supervision.

	Henri Simon vineyard			Village vineyard		
	MAE	CRPS	Night MAE	MAE	CRPS	Night MAE
MLP	0.844	0.664	0.435	0.536	0.418	0.356
LSTM	0.718	0.589	0.344	0.452	0.387	0.230
RFR	0.734	0.516	0.352	0.515	0.350	0.374
NFs	1.652	1.225	1.137	1.167	0.843	0.806

Table 5.6: Error of the supervising models over three metrics and two vineyards (temperatures expressed in $°C$)

Surprisingly, all models and evaluation metrics considered, both forecasting and supervising error results of the village vineyard are much better than the Henri Simon vineyard. Although the former is bigger and hence the chips are more spread out to cover the whole area (see figure 3.4), the models reach significant better results. A reason likely to explain this increasing accuracy might be the greater amount of available training data (1763 days for the Henri Simon against 3277 for the village vineyard, see table 5.2). Furthermore, it is divided into two sides which implies different sun exposure together with higher distances between the station and the chips (up to 100 meter for the farthest). That means that for bigger cultivation area, a single weather station could be sufficient to supervise the whole vineyard.

Similarly, the uncertainty estimation when supervising is also an important factor. Depending where in the germination process the buds are, wine growers might want to trigger an alarm more or less quickly. On table 5.7 are laid out the different ISs for 75, 85 and 95% centered intervals. Accordingly with the forecasting models, the RFR model also surpasses all other models for every interval range.

Unfortunately, the same relationship between the MAE and the IS also implies to choose a correct balance between the accuracy and the uncertainty for the NN-based models (see appendix A).

As briefly mentioned in the set breakdowns section 5.1, an interest in exploring different input subsets in order to estimate unnecessary or less important variables was shown. Indeed, when

	Henri Simon vineyard			Village vineyard		
	75%	85%	95%	75%	85%	95%
MLP ($\sigma = 0.5$)	4.816	6.147	11.996	3.350	4.205	7.735
LSTM ($\sigma = 0.5$)	5.028	6.899	14.593	3.652	5.134	11.486
RFR	1.890	2.027	2.445	1.377	1.430	1.615
NFs	4.589	5.569	9.597	2.843	3.136	4.234

Table 5.7: IS of the four supervising models for three centered intervals (75%, 85% and 95%)

equipping a crop with a weather station, many models are available with each a different set of captors and therefore with different prices. Consequently, investigating the impact of some factors might prioritize the purchase of some captors in front of others or even waive some if unneeded.

From table 5.1, the 10 first factors and the slope are never removed since they do not bring additional cost and express the time and static information such as the position of the chips. Additionally, from the review of the literature and the correlation plots of the station features (see figure 3.9), the temperature and the humidity as well as the resulting dew point are considered as essential indicators and are not removed neither. In summary, there are 14 fixed variables and the 6 remaining will be or not in the different tested subsets. On the list below are expressed the subsets where the variables in green are included and the ones in red are not.

1^{st} **subset** : [Pressure, Wind speed, Wind cosine, Wind sine, Solar radiation, Rain]

2^{nd} **subset** : [Pressure, Wind speed, Wind cosine, Wind sine, Solar radiation, Rain]

3^{rd} **subset** : [Pressure, Wind speed, Wind cosine, Wind sine, Solar radiation, Rain]

The subsets have been tested on the village vineyard data, the results can be seen on table 5.8.

	1^{st} subset			2^{nd} subset			3^{rd} subset		
	MAE	CRPS	Night MAE	MAE	CRPS	Night MAE	MAE	CRPS	Night MAE
MLP	0.638	0.502	0.338	0.761	0.592	0.317	0.792	0.616	0.253
LSTM	0.576	0.474	0.235	0.676	0.580	0.286	0.752	0.606	0.368
RFR	0.581	0.400	0.301	0.814	0.560	0.461	0.877	0.610	0.474
NFs	1.093	0.793	0.893	1.140	0.821	0.818	1.288	0.918	0.879

Table 5.8: Error of the supervising models in the village vineyard over three metrics (temperatures expressed in $°C$) for different subsets of the initial inputs.

As expected, as variables are successively removed from the inputs, the errors and scores increase alongside, at least during the day time. Indeed, one can observe in the CRPS columns that the

score increases as we removed input factors (especially when removing the solar radiations). However, during night supervision, the MAE seems to be less impacted by the factor removal, the latter even enhances the performances of the MLP model. Therefore, those results suppose that buying a weather station with only a thermometer and a hygrometer (needed to monitor relative humidity) is sufficient to reach a good precision. the same analysis was conducted for the forecasting input variables, see Appendix B.

Overall, the two NN-based models as well as the tree-based model shown satisfying results whereas the NFs model lag behind. Again the LSTM model obtained the best results with a night error of $0.23°C$ and a standard deviation of $0.27°C$. Still, the prediction intervals of the latter are not optimal and the RFR model offers a good compromise.

5.4 Test on critical nights

Although the beginning of the year 2022 was not threatening for vines in terms of frost, three nights in a row have reached negative temperatures in the sensitive period. Indeed, the buds did not already germinated when they occurred, hence they were ready to withstand those temperatures. The nights of the 1, 2 and 3 April reached respectively -1.5, -4.2 and -2.5 °C around six in the morning, they will be therefore used to observe the model predictions on those events.

In the following pages are displayed, side by side, the forecasts of each model with the 12 hours horizon (on the left) with its respective supervising model predictions (on the right) for those three critical nights (see figure 5.2, 5.3, 5.4 and 5.5).

Although never far away from the actual data (for the two first nights), the MLP forecasting model fails at correctly forecasting the lowest temperatures of the different nights (see figure 5.2). Especially on April 3 where it predicts temperatures around $1°C$ at the lowest while it actually drops till $-2.5°C$. Nevertheless, the supervising model realizes much better results and even under estimate the temperatures the first night.

In general for the forecasting part, the LSTM model performs equally to the MLP model, except that it better anticipates the drop in temperatures of the third night (see figure 5.3). On the supervision part, it performs just as well as the MLP but its prediction intervals ill-represent the true uncertainty, although it has the same variance as the MLP in its Bayesian layers.

So far, the RFR is the best performer in the forecasting accuracy together with its prediction intervals. Besides the third night where all models seem to be misled, the RFR forecast follow very closely the actual data and, as indicated by the ISs in table 5.5, its prediction intervals incorporate the substantial majority of the observation points (see figure 5.4). On the supervision side, the RFR appears to be also the most detailed model where its prediction intervals are simultaneously narrow and relatively correct, assessing a great reliability.

Finally, as relayed by the results in table 5.4 and 5.6, the NFs models appear to be the less accurate both for forecasting and supervising. the output distributions are all very similar as if the models learned one pattern that is adapted to every individual input sequence. Those results suppose that the model is not able to reproduce such complex curve and that the hypothesis of

considering the temperatures within a day as a distribution is not well-founded.

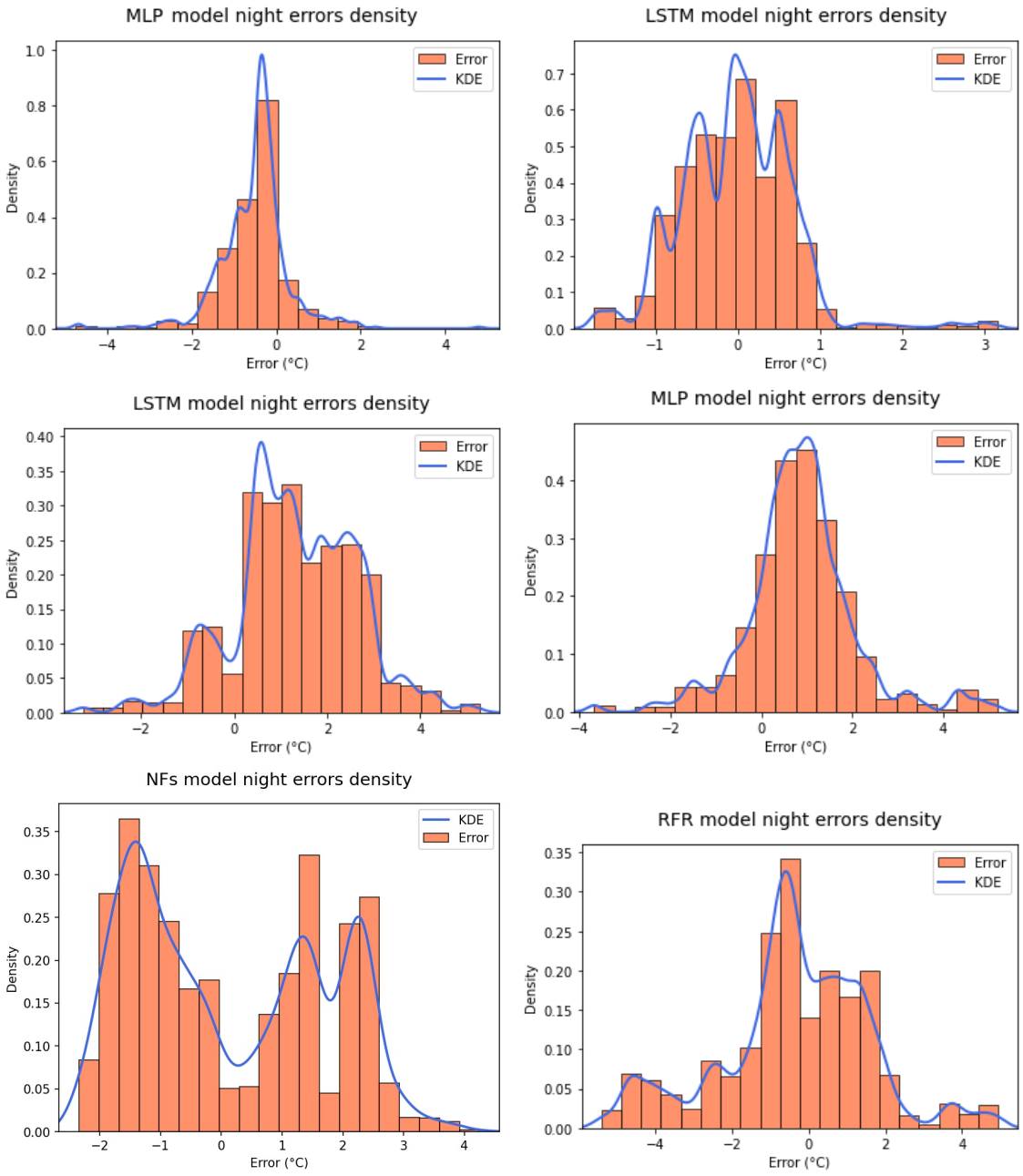

Figure 5.1: Error distributions of the best model forecasts during night. Following the same disposition as in table 5.4, the first, second and third rows are the best forecast average night error distribution for respectively the 12, 24 and 36 hours forecast horizons. The error distributions of the Henri Simon vineyard are on the left and the ones of the village vineyard on the right.

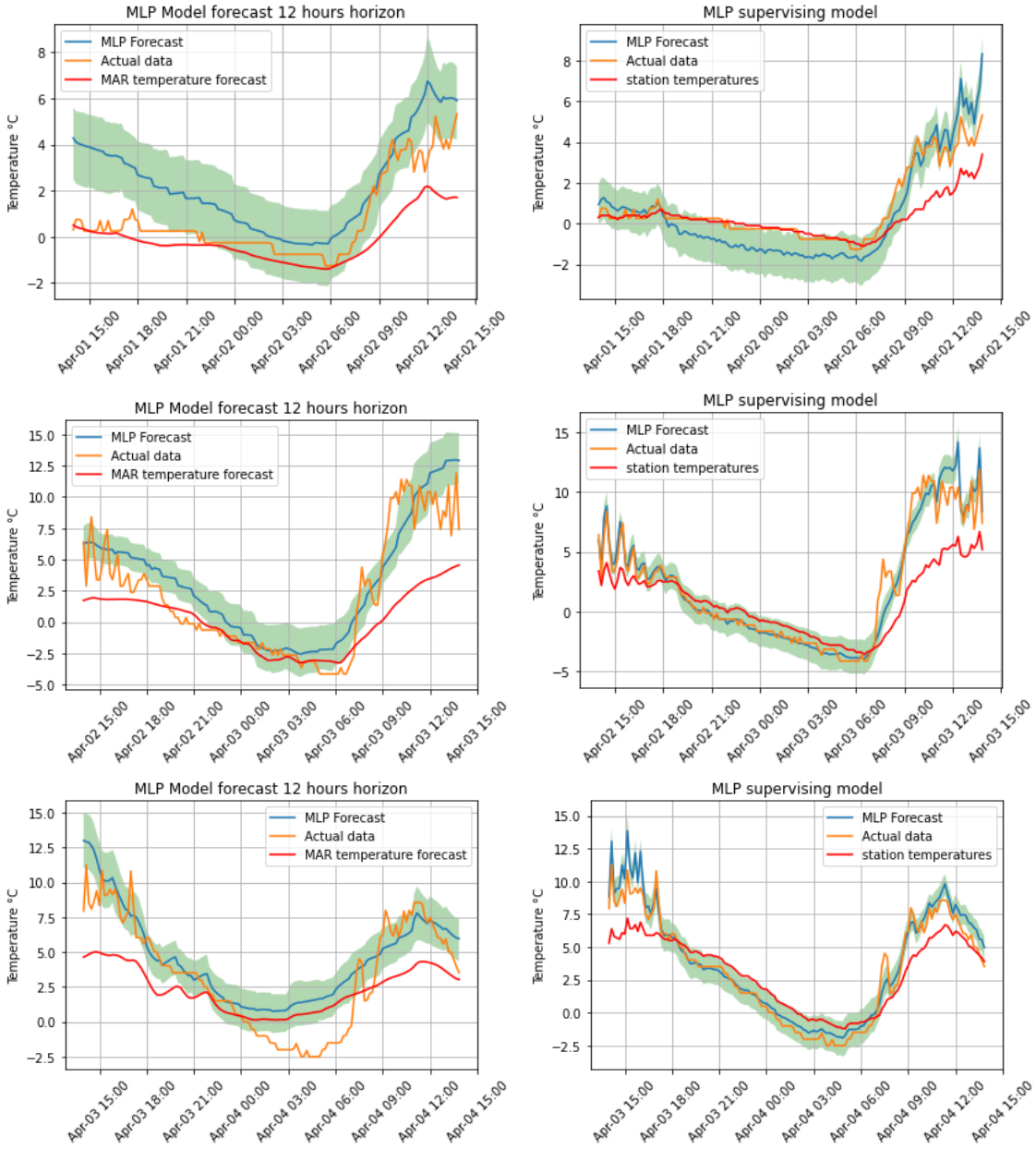

Figure 5.2: Forecasts and supervisions for the nights of April 1, 2 and 3, 2022, by the MLP models with a 95% prediction interval. For the first, second and third night, the actual data comes from the recording of respectively the chip $n°3$, 1 and 2.

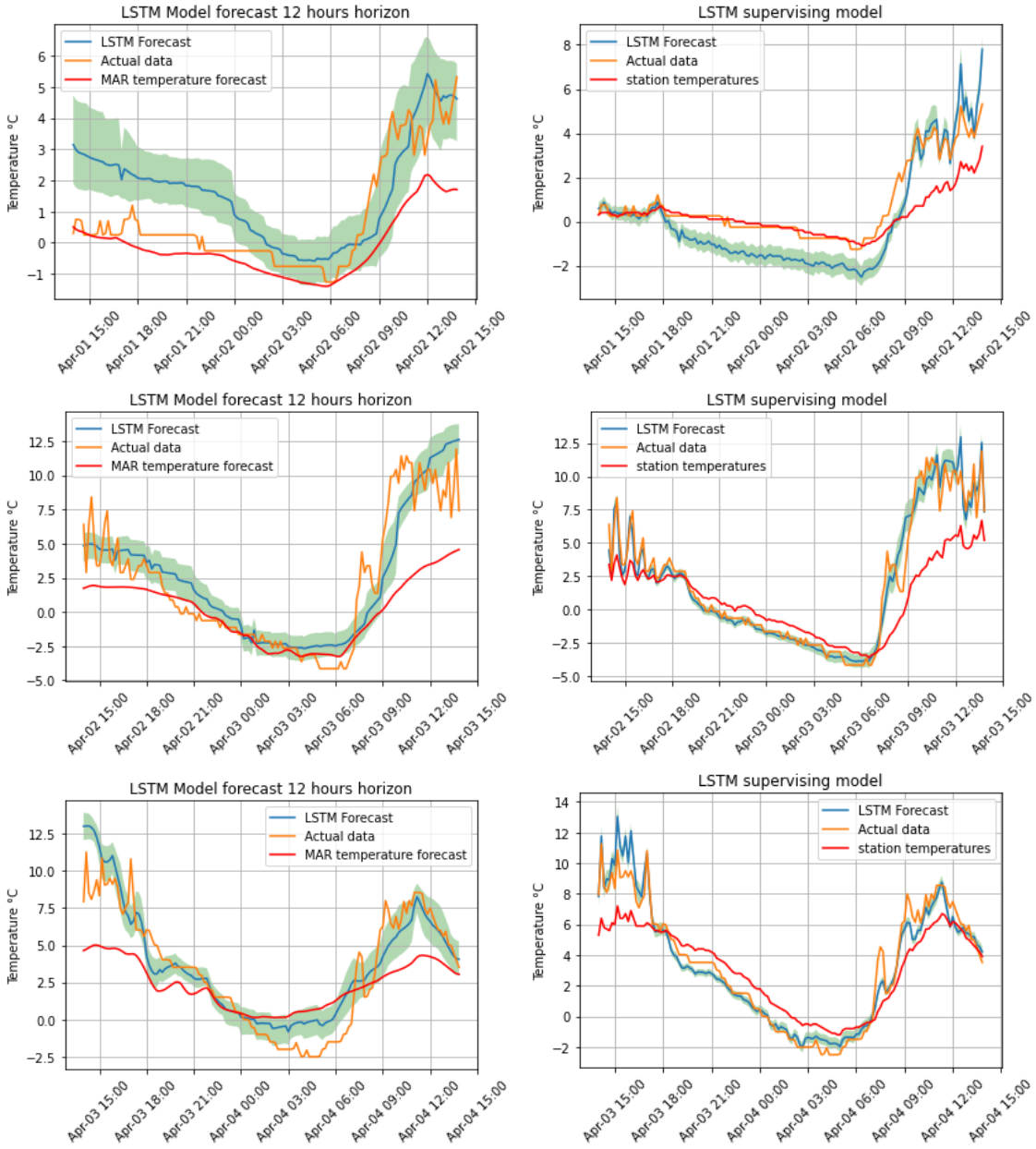

Figure 5.3: Forecasts and supervisions for the nights of April 1, 2 and 3, 2022, by the LSTM models with a 95% prediction interval. For the first, second and third night, the actual data comes from the recording of respectively the chip $n°3$, 1 and 2.

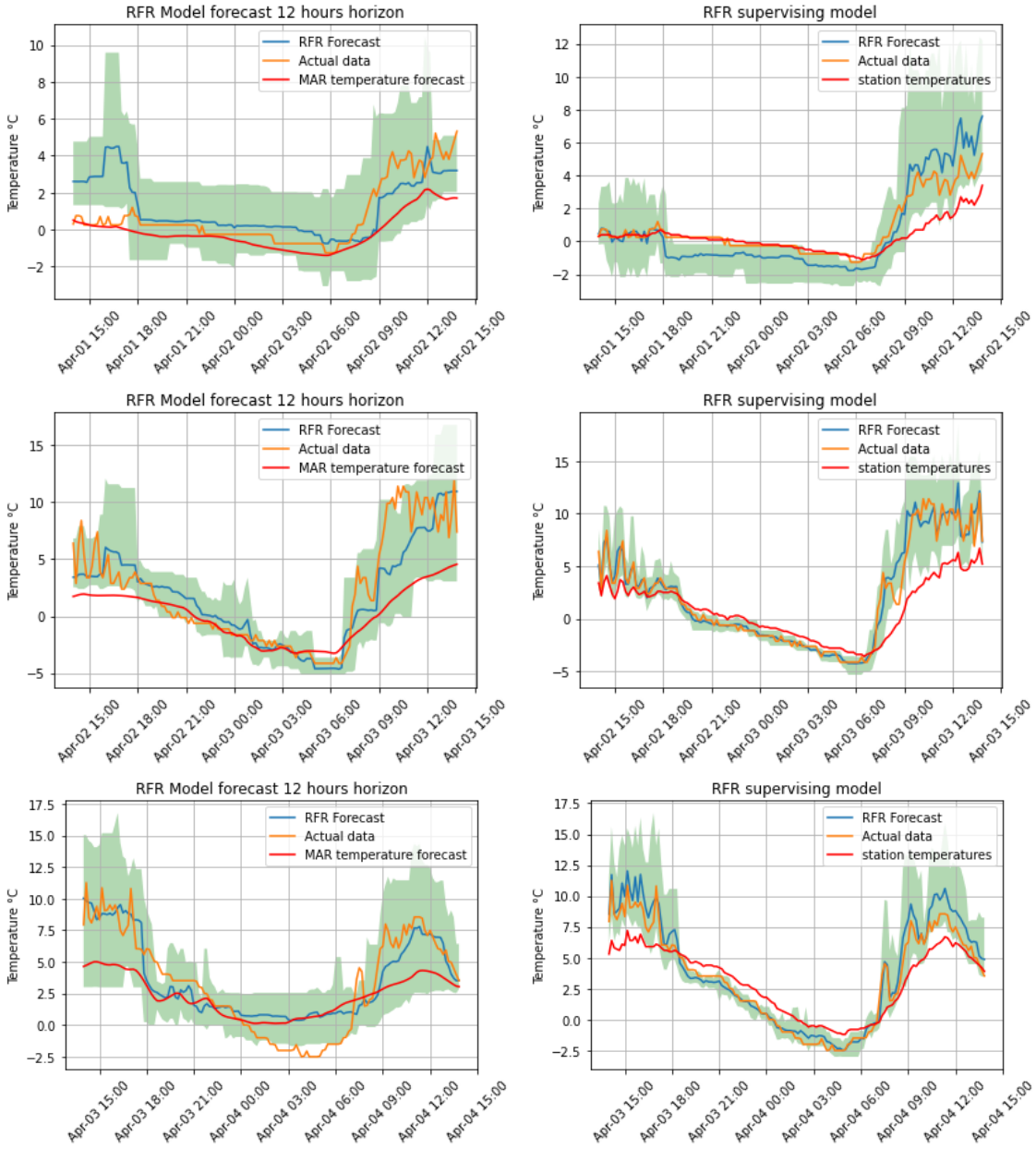

Figure 5.4: Forecasts and supervisions for the nights of April 1, 2 and 3, 2022, by the RFR models with a 95% prediction interval. For the first, second and third night, the actual data comes from the recording of respectively the chip $n°3$, 1 and 2.

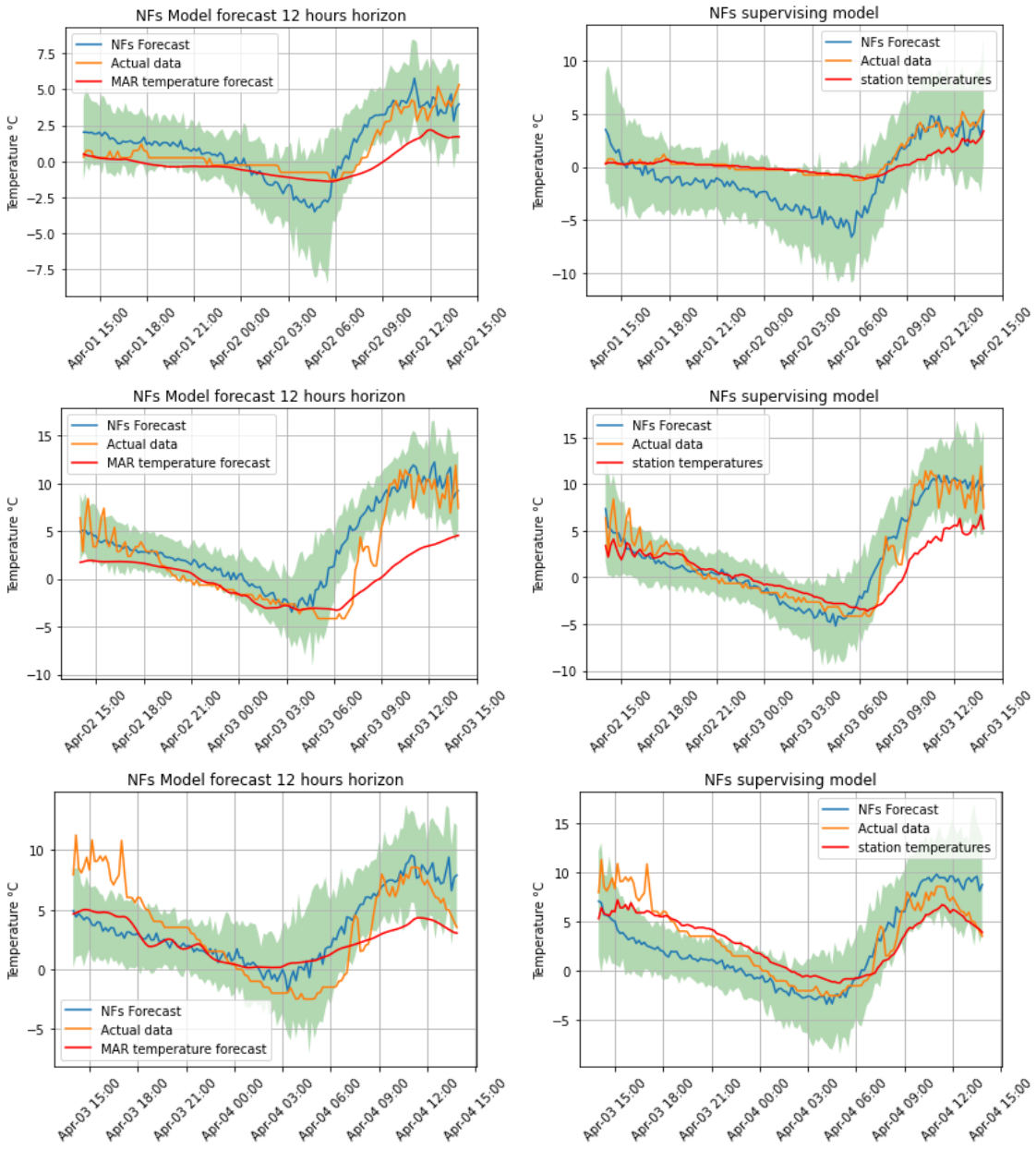

Figure 5.5: Forecasts and supervisions for the nights of April 1, 2 and 3, 2022, by the NFs models with a 95% prediction interval. For the first, second and third night, the actual data comes from the recording of respectively the chip $n°3$, 1 and 2.

Chapter 6

Conclusions and Future Work

In Belgium, each year comes with its share of new vine enthusiasts willing to start the adventure of wine production, whether as an amateur or as a professional. Along the process between the first vine plant to the first bottle of wine, they will have to go through a long path full of pitfalls. From bud freezing to vine disease, they will have to face many issues that necessitate for them to be ready and well equipped to tackle all those obstacles.

These last years, techniques related to artificial intelligence, specifically Machine Learning (ML), have brought enormous improvements in many different fields. Nowadays, more and more programming systems have adopted the hype and based their core functioning on these underlying technologies. Agriculture did not escape to this trend and is now more than ever in the need of a deep change to tackle one of the 21^{th} century biggest challenges.

In this work, we proposed to alleviate the bud freezing issue of early Spring by means of the combination of the deterministic Regional Atmospheric Model (MAR) and ML techniques, especially we studied and compared four different methods to draw a performance overview of the discussed techniques. The problematic has been handled through two temporal approaches, a foresighted one that forecasts the temperatures more than one day ahead and a live supervising used as emergency solution. As a matter of fact, air temperature is subject to chaotic phenomena which make any forecast uncertain, leading us toward a back-up solution to supervise the latter.

The four techniques are a Multi-Layers Perceptron (MLP), a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) cells, a Random Forest Regressor (RFR) and a generative model based on Normalizing Flows (NFs). All those methods rely on the principle of machine learning and have each a benefit to bring through their use in this problematic. The data comes from two vineyards situated in Lincé, a small village in the region of Liège, Belgium.

The models have been evaluated and compared on the basis of the Mean Absolute Error (MAE) and a specifically suited score for weather forecasting called the Continuous Ranked Probability Score (CRPS). Especially, since the freezing events occur overnight, the models are mainly evaluated with the MAE on their errors between sunset and sunrise. To assess the confidence of each model, prediction intervals are also drawn from them and were evaluated through the Interval Score

(IS).

Except with NFs, the forecasting models show very satisfying results for the time horizons going from 12 to 36 hours ahead. The best performer, the LSTM model, achieved to predict the temperatures during night with a MAE below 0.5° Celsius and a standard deviation of 0.64°C. However, we did not manage to obtain satisfying results for forecasts further ahead. The best model for a time horizon ranging from 24 to 48 hours ahead generated on average, during the night, an absolute error of $1.27°C$ with a standard deviation of $\sim 1.20°C$.

Since the lowest temperatures in the critical periods are usually reached around six in the morning, our best forecasting model allow wine growers to be warned in the evening two days before it actually occurs. That is, it leaves a sufficient amount of time to organize a workforce and to pre-install the candles all over the threatened parts of the vineyards.

In addition, we also investigated temperature live supervision of the vineyards by means of the same techniques. Indeed, a weather station has been installed in each vineyard hence offering a direct monitoring of the weather factors within the culture. Again, the most accurate results are generated by the LSTM model which accomplished a MAE of $0.23°C$ with a standard deviation of $0.27°C$.

Upon those predictions, the confidence of the models have also been addressed to leave to the wine growers the choice to rely or not on the forecasting. Unfortunately, the best model for forecasting revealed to be the worst for assessing its confidence in its predictions. Nevertheless, the RFR model appear to be a good trade-off between those two metrics by having a night forecast MAE of $0.69°C$ with a standard deviation of $0.90°C$ and a IS for a 75, 85 and 95% centered interval of respectively 3.88, 4.57 and 7.14 (see equation 5.2).

Finally, in order to resort to the least possible amount of captors in the weather stations and thus reducing the wine growers' expenses when buying one, we explored different subsets of input sequences successively narrowed down till the very basic factors. The results show that only with the temperature, the relative humidity and the corresponding dew point, the MLP model achieved a MAE of $0.25°C$ and a standard deviation of $0.38°C$. In comparison to the LSTM supervising model, only the standard deviation is significantly impacted but still, the accuracy remains very satisfying.

Although the combination of those two temporal models already offers some time to wine growers as well as more restful nights, further improvements can be carried out. First of all, the model predicts the temperature but does not adopt a position on the actual status of the buds and if they are freezing or not. Some exploration need to be done on that subject, although it remains very difficult to affirm if a bud really froze or not in practice.

An interesting topic that could be very useful if well studied is the development of the buds. As the seasons pass, each individual bud enters in different life cycles and successively acquires and lose the capacity to withstand cold. If one can say the current resistance of the buds, this could improve the system by auto-adjusting the threshold temperature responsible for triggering an alarm, hence avoiding useless interventions.

As it has been shown, the confidence interval of the best performer in terms of punctual temperature forecasting is the LSTM model. Unfortunately, this model revealed to be bad when assessing its confidence in the prediction. Consequently, some alternatives need to be investigated to tackle this issue and create a model simultaneously accurate and reliable in terms of its confidence.

To conclude, in front of a real issue encountered by wine growers in Belgium, we set up a series of steps from the collection of data to the implementation of multiple ML techniques in order to propose a realistic solution without involving a costly deployment. With only a year of data gathered within a vineyard, we are now able to accurately foresee freezing events. In addition, with the help of a weather station, forecasts are never taken for granted and are constantly reconsidered with a supervision model, hence ensuring a high reliability for avoiding bud freezing.

Appendix A

Variance of the Bayesian layers

As the uncertainty was introduced in NN-based models through Bayesian layers, different variance values were tested to eventually obtain more accurate confidence intervals. Curiously, the two types of network react in a different way. On figure A.2, the night errors for different values of sigma [1]has been plotted and as one can see, the MLP models are much more sensitive to the increasing sigma than the LSTM models.

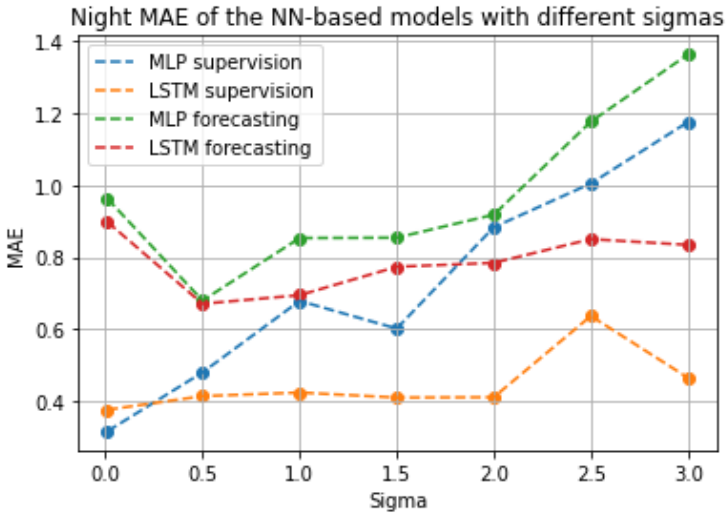

Figure A.1: Night errors of the two NN-based models for different values of sigma.

With no variance, the two models for each role (forecasting and supervising) have very similar night errors, but as the value of sigma increases, one can see the curves coming apart. Indeed, the LSTM models appear to be barely influenced in comparison with the MLP models.

[1]The variance in the Bayesian layers are obtained from Gaussian distributions. Therefore the sigma represents the value of this variance.

Conversely, if the focus is now put on the ISs of the different centered prediction interval sizes with respect to the value of the variance, the opposite pattern can be observed. As the value of sigma increases, the ISs diminish with again a stronger sensitivity for the MLP models.

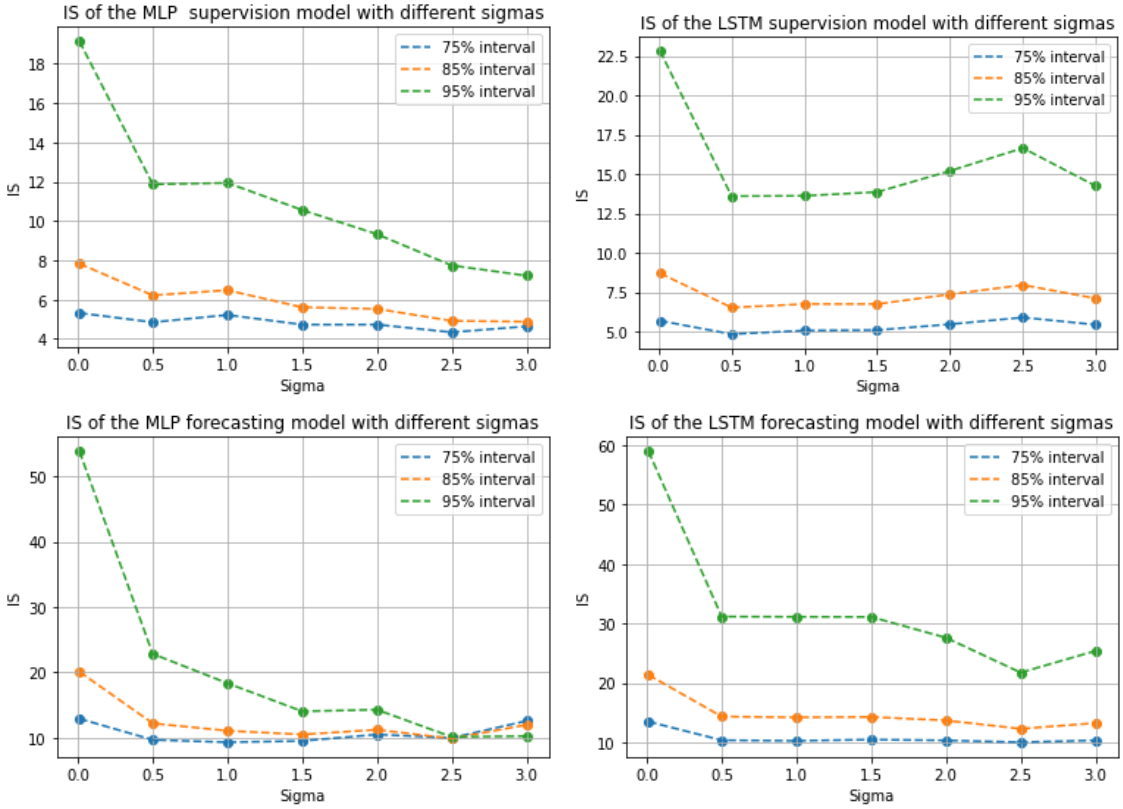

Figure A.2: ISs for different centered prediction intervals depending on the variance.

Therefore a reasonable value for sigma different from zero that seems simultaneously optimal for all models both in terms of night error and prediction interval accuracy is 0.5.

Appendix B

Factor importance for MAR data

In the same optic as with the supervising models, the performances of the forecasting models with different subsets of the initial input sequence has been studied. On the list below are gathered the different subset compositions, where the variables in red are not included and the ones in green are. All the other features (see table 5.1) are not removed in this experiment.

1^{st} **combination** : [Pressure, Wind speed, Wind gust, Wind cosine, Wind sine, Solar radiation, Rain, Cloud high, Cloud medium, Cloud low]

2^{nd} **combination** : [Pressure, Wind speed, Wind gust, Wind cosine, Wind sine, Solar radiation, Rain, Cloud high, Cloud medium, Cloud low]

3^{rd} **combination** : [Pressure, Wind speed, Wind gust, Wind cosine, Wind sine, Solar radiation, Rain, Cloud high, Cloud medium, Cloud low]

At first, based on the less correlated features of the MAR output variables with the targeted temperatures (see figure 3.8), only the rain and the factors related to the wind were removed. Surprisingly, it seems that the LSTM model performs better without those during night but has the same accuracy when considering all records (see table B.1). On the other hand, the other model performances remain unchanged.

As variables are progressively removed, the error and CRPS increase alongside. The same behavior as in table 5.8 is also observed here, the MLP model reveals to be more robust against variable removal than the three others.

	1^{st} combination			2^{nd} combination			3^{rd} combination		
	MAE	CRPS	Night MAE	MAE	CRPS	Night MAE	MAE	CRPS	Night MAE
MLP	1.312	1.040	0.560	1.290	1.036	0.574	1.338	1.074	0.612
LSTM	1.236	0.998	0.377	1.264	1.039	0.477	1.325	1.135	0.672
RFR	1.407	1.068	0.660	1.461	1.126	0.812	1.523	1.189	0.841
NFs	1.483	1.059	1.000	1.612	1.152	1.328	1.655	1.169	1.230

Table B.1: Error of the forecasting models in the village vineyard with a 12 hours horizon over three metrics (temperatures expressed in $°C$) for different subsets of the initial inputs.

Bibliography

[1] Masahisa Abe. Hokkaido – japan's new wine mecca. page 2.

[2] Agosta, C., Amory, C., Kittel, C., Orsi, A., Favier, V., Gallée, H., van den Broeke, M. R., Lenaerts, J. T. M., van Wessem, J. M., van de Berg, W. J., Fettweis, and X. Estimation of the antarctic surface mass balance using the regional climate model mar (1979–2015) and identification of dominant processes. *The Cryosphere*, 13(1):281–296, 2019.

[3] Md Manjurul Ahsan, M. A. Parvez Mahmud, Pritom Kumar Saha, Kishor Datta Gupta, and Zahed Siddique. Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance. *Technologies*, 9(3):52, July 2021.

[4] Jehad Ali, Rehanullah Khan, Nasir Ahmad, and Imran Maqsood. Random forests and decision trees. *International Journal of Computer Science Issues(IJCSI)*, 9, 09 2012.

[5] Marine Andrieu. *Pour le vin belge, le réchauffement climatique est positif*, 2022.

[6] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. Time series analysis: Forecasting and control. *Wiley Series in Probability and Statistics (fifth edition)*, 2016.

[7] Leo Breiman. Arcing classifiers. *The Annals of Statistics*, 26(3):801–824, 1998.

[8] Marc De Brouwer. Les gelées de printemps dans les vignobles belges et voisins, quelles mesures de protection ? 2017.

[9] J.M. Cadenas, M.C. Garrido, R. Martínez-España, and M.A. Guillén-Navarro. Making decisions for frost prediction in agricultural crops in a soft computing framework. *Computers and Electronics in Agriculture*, 175:105587, August 2020.

[10] G. Candille and O. Talagrand. Evaluation of probabilistic prediction systems for a scalar variable. *Quarterly Journal of the Royal Meteorological Society*, 131(609):2131–2150, 2005.

[11] Liya Ding, Kosuke Noborio, and Kazuki Shibuya. Frost forecast using machine learning - from association to causality. 159:1001–1010.

[12] Sébastien Doutreloup, Coraline Wyard, Charles Amory, Christoph Kittel, Michel Erpicum, and Xavier Fettweis. Sensitivity to convective schemes on precipitation simulated by the regional climate model mar over belgium (1987–2017). *Atmosphere*, 10:34, 01 2019.

[13] Jonathan Dumas, Antoine Wehenkel, Damien Lanaspeze, Bertrand Cornélusse, and Antonio Sutera. A deep generative model for probabilistic energy forecasting in power systems: normalizing flows. *Applied Energy*, 305:117871, January 2022.

[14] SPF Economie. La viticulture belge en chiffres. 2017.

[15] SPF Economie. *Un quart de viticulteurs belges en plus en 2020*, 2020.

[16] X. Fettweis, B. Franco, M. Tedesco, J. H. van Angelen, J. T. M. Lenaerts, M. R. van den Broeke, and H. Gallée. Estimating the greenland ice sheet surface mass balance contribution to future sea level rise using the regional atmospheric climate model mar. *The Cryosphere*, 7(2):469–489, 2013.

[17] Xavier Fettweis, Coraline Wyard, Sébastien Doutreloup, and Alexandre Belleflamme. Noël 2010 en belgique : Neige en flandre et pluie en haute-ardenne. *Bulletin de la Société Géographique de Liège*, 68:97–107, 09 2017.

[18] Hugh Fraser, Ken Slingerland, Kevin Ker, K. Helen Fisher, and Ryan Brewster. Reducing cold injury to grapes through the use of wind machines. *CanAdvance Project - ADV*, 12 2009.

[19] Peter B Gibson, William E Chapman, Alphan Altinok, Luca Delle Monache, Michael J DeFlorio, and Duane E Waliser. Training machine learning models on climate model output yields skillful interpretable seasonal precipitation forecasts. *Communications Earth & Environment*, 2(1):1–13, 2021.

[20] Tilmann Gneiting and Adrian E Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378, March 2007.

[21] Tilmann Gneiting, Adrian E. Raftery, Anton H. Westveld, and Tom Goldman. Calibrated Probabilistic Forecasting Using Ensemble Model Output Statistics and Minimum CRPS Estimation. *Monthly Weather Review*, 133(5):1098–1118, May 2005.

[22] E. P. Grimit, T. Gneiting, V. J. Berrocal, and N. A. Johnson. The continuous ranked probability score for circular variables and its application to mesoscale forecast ensemble verification. *Quarterly Journal of the Royal Meteorological Society*, 132(621C):2925–2942, 2006.

[23] Hans Hersbach. Decomposition of the Continuous Ranked Probability Score for Ensemble Prediction Systems. *Weather and Forecasting*, 15(5):559–570, October 2000.

[24] Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. 2005.

[25] Tim Januschowski, Yuyang Wang, Kari Torkkola, Timo Erkkilä, Hilaf Hasson, and Jan Gasthaus. Forecasting with trees. *International Journal of Forecasting*, page S0169207021001679, December 2021.

[26] D. E. Johnson and G. S. Howell. Factors influencing critical temperatures for spring freeze damage to developing primary shoots of concord grapevines. *Am. J. Enol. Vitic*, 32, 1981.

[27] C. Kittel, C. Amory, C. Agosta, A. Delhasse, S. Doutreloup, P.-V. Huot, C. Wyard, T. Fichefet, and X. Fettweis. Sensitivity of the current antarctic surface mass balance to sea surface conditions using mar. *The Cryosphere*, 12(12):3827–3839, 2018.

[28] David Kreuzer, Michael Munz, and Stephan Schlüter. Short-term temperature forecasts using a convolutional neural network — an application to different weather stations in germany. *Machine Learning with Applications*, 2:100007, 2020.

[29] Hyojin Lee, Jong A. Chun, Hyun-Hee Han, and Sung Kim. Prediction of Frost Occurrences Using Statistical Modeling Approaches. *Advances in Meteorology*, 2016:1–9, 2016.

[30] Yadvinder Malhi, Janet Franklin, Nathalie Seddon, Martin Solan, Monica G. Turner, Christopher B. Field, and Nancy Knowlton. Climate change and ecosystems: threats, opportunities and solutions. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 375:1794, 01 2020.

[31] R. Meenal, Prawin Angel Michael, D. Pamela, and E. Rajasekaran. Weather prediction using random forest machine learning model. *Indonesian Journal of Electrical Engineering and Computer Science*, 22(2):1208, May 2021.

[32] Michelle Renée Mozell and Liz Thach. The impact of climate change on the global wine industry: Challenges & solutions. *Wine Economics and Policy*, 3(2):81–89, 2014.

[33] Institut Royal Météorologique. 2020 : l'année la plus chaude depuis le début des observations à uccle, 2020.

[34] Thornton P. Recalibrating food production in the developing world: Global warming will change more than just the climate. *CCAFS Policy Brief*, 6, 2012.

[35] Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs Bergmann, and Roland Vollgraf. Multivariate Probabilistic Time Series Forecasting via Conditioned Normalizing Flows, January 2021. arXiv:2002.06103 [cs, stat].

[36] Eliad Sasoni. Innovations in frost protection. 11 2018.

[37] P.R. Shukla, J. Skea, R. Slade, A. Al Khourdajie, R. van Diemen, D. McCollum, M. Pathak, S. Some, P. Vyas, R. Fradera, M. Belkacemi, A. Hasija, G. Lisboa, S. Luz, J. Malley, and (eds.). Climate change 2022: Mitigation of climate change. contribution of working group iii to the sixth assessment report of the intergovernmental panel on climate change. *Cambridge University Press*, 2022.

[38] David Silver, Aja Huang, Christopher Maddison, Arthur Guez, Laurent Sifre, George Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 01 2016.

[39] S.F. Sopelsa Neto, N.F.and Stefenon, L.H. Meyer, R. Bruns, A. Nied, L.O. Seman, G.V. Gonzalez, V.R.Q. Leithardt, and K.-C. A Yow. Study of multilayer perceptron networks applied to classification of ceramic insulators using ultrasound. *Applied Sciences*, 11(1592), 2021.

[40] Doutreloup Sébastien, Bois Benjamin, Pohl Benjamin, Zito Sébastien, and Richard Yves. Climatic comparison between belgium, champagne, alsace, jura and bourgogne for wine production using the regional model mar. *Oeno One*, 56, 2022.

[41] Yosuke Tamura, Liya Ding, Kosuke Noborio, and Kazuki Shibuya. Frost Prediction for Vineyard Using Machine Learning. In *2020 Joint 11th International Conference on Soft Computing and Intelligent Systems and 21st International Symposium on Advanced Intelligent Systems (SCIS-ISIS)*, pages 1–4, Hachijo Island, Japan, December 2020. IEEE.

[42] Trang Thi Kieu Tran, Sayed M. Bateni, Seo Jin Ki, and Hamidreza Vosoughifar. A review of neural networks for air temperature forecasting. *Water*, 13(9), 2021.

[43] Trang Thi Kieu Tran, Taesam Lee, and Jong-Suk Kim. Increasing neurons or deepening layers in forecasting maximum temperature time series? 11(10):1072.

[44] Hans Van de Vyver, Bert Van Schaeybroeck, Rozemien De Troch, Lesley De Cruz, Rafiq Hamdi, Cecille Villanueva-Birriel, Philippe Marbaix, Jean-Pascal van Ypersele, Hendrik Wouters, Sam Vanden Broucke, Nicole P.M. van Lipzig, Sébastien Doutreloup, Coraline Wyard, Chloé Scholzen, Xavier Fettweis, Steven Caluwaerts, and Piet Termonia. Evaluation framework for sub-daily rainfall extremes simulated by regional climate models. *JOURNAL OF APPLIED METEOROLOGY AND CLIMATOLOGY*, 60(10):1423–1442, 2021.

[45] Cor Verdouw, Sjaak Wolfert, and Bedir Tekinerdogan. Internet of things in agriculture. *CAB Reviews*, 11:1–12, 12 2016.

[46] Antoine Wehenkel and Gilles Louppe. Unconstrained Monotonic Neural Networks, March 2021. arXiv:1908.05164 [cs, stat].

[47] Daniel Wilks. *Statistical Methods in the Atmospheric Sciences, Volume 91, Second Edition (International Geophysics)*. 01 2005.

[48] Wille, J.D., Favier, V., Dufour, and A. et al. West antarctic surface melt triggered by atmospheric rivers. *Nat. Geosci*, 12(1):911–916, 2019.

[49] Ruiyun Yu, Yu Yang, Leyou Yang, Guangjie Han, and Oguti Move. RAQ–A Random Forest Approach for Predicting Air Quality in Urban Sensing Systems. *Sensors*, 16(1):86, January 2016.

Acronyms

ANN Artificial Neural Network. 10, 17

BNN Bayesian Neural Network. 35, 36, 39

CRPS Continuous Ranked Probability Score. 43, 45, 47, 48, 56, 61, 62

GFS Global Forecast System. 20

IoT Internet of Things. 9

IS Interval Score. 44, 46–49, 56, 57, 60

KDE Kernel Density Estimation. 45

LSTM Long Short-Term Memory. 13, 16, 17, 36, 37, 39, 40, 44–49, 53, 56–59, 61, 62

MAE Mean Absolute Error. 37, 38, 40, 43, 45–49, 56, 57, 62

MAR Regional Atmospheric Model. 3, 7, 20, 21, 23, 25, 26, 28–30, 34, 36, 41–44, 46, 56, 61

MASE Mean Absolute Scaled Error. 16, 17

ML Machine Learning. 1, 3, 6, 9, 10, 16, 41, 56, 58

MLP Multi-Layers Perceptron. 11, 36, 37, 39, 40, 44–49, 52, 56, 57, 59–62

MSE Mean Squared Error. 11, 12

NFs Normalizing Flows. 14, 35–39, 44–49, 55–57, 62

NN Neural Network. 10–13, 35, 36, 39, 46, 47, 49, 59

NNs Neural Networks. 36, 38

RFR Random Forest Regressor. 14, 35–39, 41, 44–49, 54, 56, 57, 62

RMSE Root Mean Square Error. 16, 17

RNN Recurrent Neural Network. 12, 13, 36, 39, 56

RNNs Recurrent Neural Networks. 13

SARIMA Seasonnal AutoRegressive Integrated Moving Average. 17

SPF Service Public Fédéral. 5

SVM Support Vector Machine. 17, 18

UMNN Unconstrained Monotonic Neural Networks. 38, 39