Prediction of frost events using machine learning and IoT sensing devices

Ana Laura Diedrichs*, Facundo Bromberg*, Diego Dujovne†, Keoma Brun-Laguna‡,Thomas Watteyne‡
*Universidad Tecnológica Nacional, Mendoza, Argentina.
{ana.diedrichs,facundo.bromberg}@frm.utn.edu.ar
†Universidad Diego Portales (UDP), Santiago, Chile.
{diego.dujovne}@mail.udp.cl
‡Inria, EVA team, Paris, France.
{keoma.brun,thomas.watteyne}@inria.fr
§CONICET, Mendoza, Argentina.

***Abstract*—IoT in Agriculture applications have evolved to solve several relevant problems from producers. Here, we describe a component of an IoT-enabled frost prediction system. We follow current approaches for prediction that use machine learning algorithms trained by past readings of temperature and humidity sensors to predict future temperatures. However, contrary to current approaches, we assume that the surrounding thermodynamical conditions are informative for prediction. For that, a model was developed for each location, including in its training information of sensor readings of all other locations, autonomously selecting the most relevant ones (algorithm dependent). We evaluated our approach by training regression and classification models using several machine learning algorithms, many already proposed in the literature for the frost prediction problem, over data from five meteorological stations spread along the Mendoza Province of Argentina. Given the scarcity of frost events, data was augmented using the Synthetic Minority Oversampling Technique (SMOTE). The experimental results show that selecting the most relevant neighbors and training the models with SMOTE reduces the prediction errors of both regression predictors for all five locations, increases the performance of Random Forest classification predictors for four locations while keeping it unchanged for the remaining one, and produces inconclusive results for Logistic regression predictor. These results demonstrate the main claim of these work: that thermodynamic information of neighboring locations can be informative for improving both regression and classification predictions, but also are good enough to suggest that the present approach is a valid and useful resource for decision makers and producers.**

***Index Terms*—machine learning, Bayesian networks, Random Forest, SMOTE, precision agriculture, frost prediction**

I. Introduction

In Mendoza, Argentina, one of the most relevant wine production regions in Latin America [1], [2], frost events resulted in a loss of 85% of the peach production during 2013, and affected more than 35 thousand hectares of vineyards. Furthermore, research work conducted by Karlsruhe Institute of Technology (KIT) [3] warns that vineyards in Mendoza and San Juan (Argentina) represent the highest risk regions in the world for extreme weather and natural hazards, mainly due to wide diurnal temperature variation with over 20 degrees Celsius in winter as Gonzalez et al. describe on [4]. This reality quantifies one of the aspects that a frost event can generate, but the socio-economical consequences do hit not only producers, but also transport, commerce and general services, which take long recovery periods. Plants and fruits suffer from frost events as a consequence of water icing inside the internal tissues present in the trunk, branches, leaves, flowers and fruits. However, water content and distribution is different among them, generating different damage levels. The most sensible sections are leaves and fruits. Leaves provide photosynthesis surface, while fruits collect nutrients and water from the plant. Individual damage levels can be assessed by studying the effects of freezing those parts under controlled conditions, but an integral plant view is necessary to measure the economical loss at the end of the harvest period.

Frost events are difficult to predict given that they are a localized phenomenon. Frost can be result in partial damage in different areas of the same crop field, with the capacity to destroy the entire production in a matter of hours: even if the damage is not visible just after the event, the effects can surface at the end of the season, both reducing the quantity and quality of the harvest.

There are several countermeasures for frost events, which include air heaters by burning gas, petrol or other fuels, removing air using large fans distributed along the field or turning on sprinklers. However, each of these countermeasures are expensive each time they are used. As a consequence, it is critical to predict frost events with the highest possible accuracy, so as to initiate the countermeasure actions at the right time, reducing the possibility of false negatives (a frost event was not predicted and it happened) or false positives (a frost event was predicted and did not happen). In the first case, the production or part of it may be lost. In the second case, the burned fuel will be useless. Both situations lead to reduced yield or complete production loss.

Given the small amount of frost events during the year, available data is scarce to build an accurate forecasting system, which defines an unbalanced dataset problem. The more data machine learning models have, the better they can improve their accuracy. This is a relevant problem in regions where the meteorological data is not continuous or it has a short history. In these cases, there is a low amount of data to build a predictive model with high accuracy and/or precision.

This work is part of an Internet of Things-based system

aimed to predict frost events in Peach orchards, as described on the work from Watteyne et al. [5]. This system is composed of three stages. On the first stage, historical meteorological data is gathered from a number of selected internet-enabled weather stations. On the second stage, the data collected from the weather stations is used to train a frost-prediction engine. Finally, on the third stage, local field data is collected from a network of IoT sensors and further inserted on the model generated by the prediction engine to provide a frost forecast output.

In this paper, we describe the second stage of the IoT frost prediction system. Instead of using traditional formula-based predictions which are general for wide regions, we propose to use a different approach: We use machine learning algorithms for regression based on Bayesian Networks and Random Forest, and for classification based on Random Forest (RF), Logistic regression and Binary Trees, all ran over a balanced training set augmented with new samples produced using the SMOTE (Synthetic Minority Oversampling Technique)[6] technique. This technique increases the rate of frost detection (sensitivity).

Traditional approaches provide analytical solutions which rely on several calibration parameters from the field, providing low prediction accuracy. Although Machine Learning was the natural evolution line of the solution, existing solutions from the literature did not provide enough accuracy for the producers to make well-funded decisions. The novelty of this paper is twofold: First, we provide IoT-based Machine Learning models which combine data from nearby IoT-enabled meteorological stations sharply improving frost prediction performance, and second, the model provided is amenable for integration into an IoT system within the field to provide localized frost prediction, which greatly increases the value for the producers, enabling them to take action and mitigate frost events in a more efficient way.

We chose these machine learning algorithms because they are widely used for decision support applications. Logistic regression and Binary Trees were chosen for comparison against recent competitors [7], whereas RF was chosen because it ensures no-overfitting and has demonstrated very good performance in classification problems. But RF is like a black-box model, which means that is difficult for end-users to understand how the model makes decisions. In contrast, Bayesian networks provide a complete framework for inference and decision making by modeling relationships of cause-consequence between the variables. A nice property of a Bayesian network is that it can be queried: We can ask for the probability of an event given the current evidence (sensor values). In real IoT applications, it could happen that a sensor breaks or looses the connection with the gateway, forcing the network to deal with the problem of processing results with incomplete information to make decisions. Bayesian networks are also one of the most commonly used methods to modeling uncertainty.

We are interested in evaluating the possibility of building good predictors only with temperature and relative humidity variables. These sensors are very common in most of the IoT platforms or data loggers used for environmental measurements. There are situations where sensor networks cannot have access to a gateway or central server; so we want to know, not only if temperature and humidity values are enough to get an accurate predictor, but also if the sensor's neighbors are informative (or not) for the prediction, as a first step to prototype a in-network frost forecast.

Finally, we build a regression model to predict the minimum temperature for the following day using historical information from previous days including a set of variables such as temperature and humidity from itself and from neighboring sites. It is important to have an accurate prediction at least 24 hours ahead because of the many logistic issues farmers must resolve to apply countermeasures against frost (gasoline stock for heaters, permit of irrigation to feed sprinklers, temporal employees schedule). The rest of the paper is organized as follows: Section II locates this work within the IoT-based Frost prediction system. In section III we discuss previous works on daily minimum temperature (frost) prediction. We introduce Bayesian networks in section IV-A, Random Forest in IV-B and Logistic Regression in IV-C. In section V we describe our scenario of interest and the datasets to be used to train the models. Then, we explain our experimental setup in VI, followed by the results section VII. Finally, we share our findings and future work in section VIII

II. The IoT Frost Prediction System

The IoT Frost Prediction System, described on Figure 1, is composed by three integrated stages:

- A group of **internet-enabled weather data collection devices** composed by weather stations which are first used to provide historical data to the prediction engine, and second, to generate new data to continuously improve the prediction model.
- A **prediction engine** based on the techniques and results described in detail on this paper, which is trained using the data gathered from the weather stations.
- A **frost forecast module** based on the model obtained from the prediction engine. This module obtains the data from an IoT-based sensor network different from the weather stations. The IoT-based sensor network provides first, intra-field data from strategically selected locations as input data to the model, and second, is used to calibrate temperatures on the field with those captured by remote sensing devices.

III. Related work

Current frost detection methods can be classified from the data processing they use to generate the forecast: empirical, numerical simulation and machine learning.

A. Empirical Methods

Empirical methods are based on the use of algebraic formulas derived from graphical statistical analysis of a number of selected parameters. The result is the minimal expected temperature, such as the work from Brunt et al. [8] which is

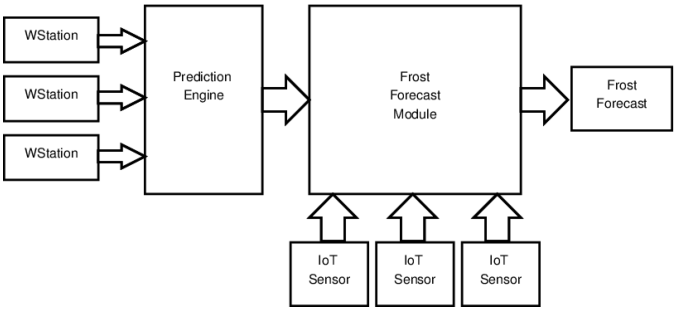

Fig. 1: The IoT-based frost prediction system

applied in [9] and the work from Allen et al. [10]. A complete review of classical frost prediction methods can be found in Burgos et al. [11], where the common pattern among them is the estimation of the minimal temperature during the night. Furthermore, Burgos et al. highlight the work from Smith [12] and Young [13], comparing the minimal prediction accuracy. As a matter of fact, the United States National Weather Service has thoroughly used Young's equation with specific calibration to local conditions and time of the year for frost forecasting.

The Allen method, created in 1957, is still recommended by the Food and Agriculture Organization (FAO) from the United Nations to predict frost events. This formula requires the dry and wet bulb at 3PM of the current day as an estimation of relative humidity and dew point, together with atmospheric pressure and temperature.

All the former models must be adapted to local conditions by calculating a number of constants that characterize each geographical location. The result is the prediction of the minimal temperature for the current night only. A number of these formulas suffer restrictions since they are indicated only for radiative (temperature-based) frost events.

B. Numerical simulation methods

Numerical simulations are widely used to predict weather behavior. Prabha et al.[14] have shown the use of Weather Research and Forecasting (WRF) models for the study of two specific frost events in Georgia, U.S. The authors used the Advanced Research WRF (AWR) model with a 1km resolution scaling to the region of interest with a set of initial values, land use characteristics, soil data, physical parametrization and for a specific topography map resolution. The resulting model obtains accuracies between 80% and 100% and a Root Mean Square Error (RMSE) between 1.5 and 4 depending on the use case.

Wen et al. [15] also base their study on WRF; however, the authors integrate a number of weather observations from the MODIS database as inputs, composed by multispectral satellite images. Wen et al. highlight that the model improves when they include local model observations. This model predicts caloric balance flows, such as net radiation, latent heat, sensible heat and soil heat flow.

Although this is a valuable modeling tool, numerical simulations and empirical formulas require a number of measurements and parameters which are not always available to the producer, such as solar radiation and soil humidity at different depths.

C. Machine learning methods

There have been several pioneering efforts to apply machine learning techniques to frost prediction [16], [17], [9], [18]. However, newer approaches have taken advantage of the evolution of machine learning techniques and massive data processing facilities to obtain higher accuracy on their results.

Maqsood et al. [19], provides a 24-hour weather prediction south of Saskatchewan, Canada, creating seasonal models. The authors used Multi-Layered Perceptron Networks (MPLN), Elman Recurrent Neural Networks (ERNN), Radial Basis Function Networks (RBFN) and Hopfield Models (HFM), all trained with temperature, relative humidity and wind speed data.

Another example of applied machine learning to frost prediction is the work from Ghielmi et al.[20]. In this work, the authors build a minimal temperature prediction engine in north Italy. The aim of this work is to predict spring frost events, using temperature at dawn, relative humidity, soil temperature and night duration from weather stations. Ghielmi et al. considers input data from six sources to an MPLN and compares the behavior with Brunt's model and other authors.

Eccel et al. [9] has also studied minimal temperature prediction on the Italian Alps using numerical models combined with linear and multiple regression, artificial neural networks (ANNs) and Random Forest. The most relevant finding from this publication is the ability of the Random Forest method to provide the most accurate frost event prediction.

Ovando et al. [21] and Verdes et al. [22], build a frost prediction system based on temporal series of temperature-correlated thermodynamic variables, such as dew point, relative humidity, wind speed and direction, cloud surface among others using neural networks.

Lee et al. [7] use logistic regression and decision trees to estimate the minimal temperature from eight weather variables for each station in South Korea, for frost events between 1973 and 2007, with the following results: average recall values between 75% and 80% and false alarm rate of (in average) between 22% and 28%.

We can observe that the currently proposed Machine Learning based methods for frost prediction concentrate on the use of a single weather station to provide input to the model without considering variables from other neighboring weather stations. All the former proposals have used long periods of captured data for training purposes, ranging from 8 to 30 years, highlighting the local nature of the frost phenomena. It is also noticeable from the literature that the most relevant parameters found by these as inputs to the models are temperature and relative humidity.

IV. Machine learning methods

A. Bayesian networks

The work of Aguilera et al. [23] on Bayesian networks (BN) for environmental modeling stresses the benefits of using BN for inference, knowledge discovery and decision making

applications. BN is a type of probabilistic graphical model, whose set of random variables and conditional independences among them can be represented as a directed acyclic graph (DAG), whose set of nodes $V = X_1, X_2, .., X_M$ represent the random variables, and each directed edge represents a direct, i.e., non-mediated, probabilistic influence between the target variable and the originating variable, referred to as parent. This results in each random variable to be independent of all its non-descendents given its parents; a property known as the *Markov property*. If we denote by π_{X_i} the set of parents of variable X_i, the Markov property results in the following factorization of the joint probability distribution $P(X_1, ..., X_M)$: $P(X_1, ..., X_M) = \prod_{i=1}^{M} P(X_i|\pi_{X_i})$. This factorization is the main advantage of BNs, as it can result in practice in drastic dimensionality reductions, from M down to the cardinality of the largest parents' set.

BNs can be built autonomously using structure learning algorithms (SLA) that elicitate both the network structure as well as its numerical parameters, completely from input data. In this work, we take the score-based approach for structure learning, that as its name implies, assigns a score to each candidate BN structure to evaluate how well it fits the data, and then searches over the space of DAGs for a structure with maximal score using an heuristic search algorithm. Typical score measures are some variation of the lilkelihood, defined as the probability of the (observed) data D given a Bayesian network BN consisting of a DAG M and numerical parameters θ, i.e., $L(\theta) = P(D|\theta, M) = P(x_1, x_2, .., x_m|\theta)$. Greedy search algorithms (such as hill-climbing or tabu search) are a common choice for the search procedure, but almost any kind of search procedure can be used.

We propose to model a state-based, Gaussian Bayesian network which on one hand models both the local distributions of the factorization as normal distributions linked by linear constraints, and on the other represents the state of each variable at discrete time intervals; resulting into a series of time slices, with each indicating the value of each variable at time t.

In our experiments, we considered different learning scenarios, including hill climbing (HC) [24] and Tabu Search (Tabu) for the search procedure, both provided by R as in packages [25], [26]; as well as maximum likelihood for fitting the parameters of the Gaussian Bayesian network, using the regression coefficients for each variable against its parents. As scores, we considered:

- The multivariate Gaussian log-likelihood score (loglik-g),
- The corresponding Akaike Information Criterion score (aic-g),
- The corresponding Bayesian Information Criterion score (bic-g),
- A score equivalent Gaussian posterior density (bge)

B. Random Forest

Random Forest (RF) [27] is a machine learning method that, same as BN, can be applied to both regression and classification problems. The RF algorithm is a very well-known ensemble learning method which involves the creation of various decision trees models. Each tree is built as follows [28]:

1) Build the training set for RF by sampling the training cases at random with replacement from the original data. About one-third of the cases are left out of the sample. This *oob* (out-of-bag) data is used to get a running unbiased estimate of the classification error as trees are added to the forest. It is also used to get estimates of variable importance.
2) If there are M input variables, a number $m << M$ is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node (decide the parent's node and leaves). The value of m, also known as $mtry$, is held constant during the forest growing.
3) the best split of one of the m variables is calculated using the Gini importance criteria.
4) Each tree is grown until there are no more m variables to add to the tree. The algorithm continuous until $ntree$ constant number of trees were created. No pruning is performed.

The RF algorithm can be used for selecting the most relevant features from the training dataset by evaluating the Gini impurity criterion of the nodes (variables).

C. Logistic Regression

Binary logistic regression is a generalized linear model with the Bernoulli distribution, which is a particular case of the binomial distribution where $n = 1$. We define frost events as a binary variable ($Y = 0$ for no frost occurrence and $Y = 1$ for frost event), so the logistic regression equation for frost events probability $p(x_i)$ can be defined as:

$$logit[p(x_i)] = exp(\beta_0 + \beta' x_i) \tag{1}$$

where β' is the coefficient corresponding to independent variable x_i ; $i = 1..N$ is the number of variables, and β_0 is the intercept of the linear term. The β' and β_0 parameters can be calculated using maximum likelihood estimates. The probability of frost events, $Y = 1$, is defined as:

$$P(Y = 1) = \frac{exp(\beta_0 + \beta' x_i)}{1 + exp(\beta_0 + \beta' x_i)} \tag{2}$$

V. Our datasets

We worked with data from *Dirección de Agricultura y Contingencias Climáticas* (DACC) [29], from Mendoza, Argentina. DACC provided data from five meteorological stations located in Mendoza province in Argentina as depicted on Figure 2, which are listed below:

- Junín ($33°6'\ 57.5''$ S,$68°29'\ 4''$ W)
- Agua Amarga ($33°30'\ 57.7''$ S,$69°12'\ 27''$ W)
- La Llave ($34°38'\ 51.7''$ S, $68°00'\ 57.6''$ W)
- Las Paredes ($34°31'\ 35.7''$ S,$68'°25'\ 42.8''$ W)
- Tunuyán ($33°33'\ 48.8''$ S,$69°01'\ 11.7''$ W)

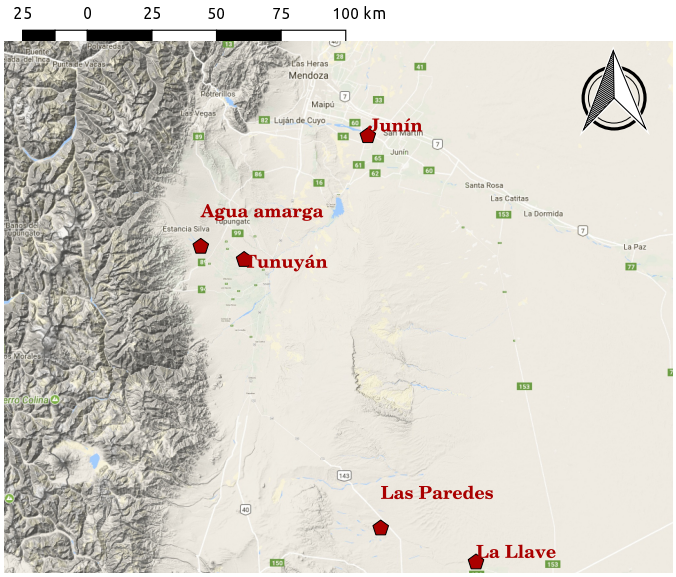

Fig. 2: Map of the DACC's stations located in Mendoza, Argentina.

Name	Location	Heigth (m)
Junín	Junín	653
Agua Amarga	Tunuyán	970
La Llave	San Rafael	555
Las Paredes	San Rafael	813
Tunuyán	Tunuyán	869

For each location, we have data from temperature and relative humidity sensors spanning a period from 2001 until 2016. Our dataset uses this information to record, for each day, the average, minimum and maximum of both temperature and humidity, resulting in six variables per location, per day sampled.

Our datasets reflect the type of prediction intended, where the learned model should accurately predict the minimum temperature for the next day using information from previous days. It is therefore built based on lagged variables, with a lag of $T = 1, 2, 3, 4$ previous days. This resulted in each labeled data point containing T times 6 variables corresponding to the T previous days, plus, for the training dataset only, the label variable reporting the next day minimum temperature at the location of the station. For classification models, we discretized the label variable like frost or not frost, defining a frost event as below zero degree Celsius, and consider the frost event as the positive class.

As explained before, we considered two cases. One in which we take only the variables of the station, referred to as *local* or *config-local*, and one in which we considered variables from all other stations as well, referred to as *all* or *config-all*. To illustrate, we present some example rows for different cases of the training dataset. In the examples, we denote by $x^t_{i,j}$, the value for the j-th variable, with $j = 1, \ldots, 6$, of the i-th station, with $i = 1, \ldots, 5$, of t days in the past. For the case of station $i = 2$, $T = 1$ and local, an example row of the training dataset would contain data for $[x^1_{2,1}, x^1_{2,2}, x^1_{2,3}, x^1_{2,4}, x^1_{2,5}, x^1_{2,6}, Y_2]$ where Y_i is the minimum temperature for the next day at station i. For the case of station $i = 2$, $T = 1$ and config-all, an example row of the training dataset would contain data for $[x^1_{1,1}, x^1_{1,2}, ..., x^1_{1,6}, x^1_{2,1}, x^1_{2,2}, ..., x^1_{2,6}, Y_2]$. Finally, for the case of $i = 2$, $T = 2$ and config-local, the example row would contain: $[x^1_{2,1}, x^1_{2,2}, ..., x^1_{2,6}, x^2_{2,1}, x^2_{2,2}, ..., x^2_{2,6}, Y_2]$.

VI. Experiment setup

We conducted several experiments to validate our central claim that measurements in nearby locations can help improve prediction of frost events. For that, we compare our approach over several classification and regression algorithms. We considered regression models that predict the minimum temperature for the next day, namely Bayesian network (BN) and Random Forest (RF). We train the BN models using HC and Tabu from *bnlearn* R package [26], [25] with their default values for all the selected scores. To contrast our approach with previous works [7], we train classification models using binary trees (recursive partitioning (Rpart) [30][31] and C5.0 without boosting [32]), logistic regression and RF [33]. C5.0 incorporates boosting and a feature selection approach (winnowing). For the purpose to adapt to a binary tree similar to the one presented in [7], we didn't train C5.0 with boosting, and for Rpart we chose an entropy split.

In order to train the regression and classification models, we split the dataset in train and test sets. The train set is used by the algorithms to fit the model parameters to the data, while the test set is used to validate the performance of the models under unseen conditions. In our experiments, we performed a 68% split in the first part of the dataset for the training phase and the rest for testing purposes.

To tackle the scarcity of frost events, we evaluated not only the original datasets, but also datasets with an augmented number of frost events, by using the SMOTE re-sampling methodology [34], [35]. SMOTE involves a combination of minority class over-sampling, balanced by a majority class under-sampling, resulting in a dataset of the same number of data points. In our experiments we chose a three-time oversampling of the minority class, that triples the number of frost events.

The method used for selection of the optimal values of user-given parameters in the training phase is cross-validation on a rolling basis, best known as time-series cross-validation [36]. The range of values for each parameter used for tuning are listed on Table VI. This method consists in K training and testing events over the same dataset, reported through the average of the performance measurement over all cases. The k-th case, $k = 1, \ldots, K$, consists in a training range over the first $((k-1) \times horizon) + windowSize$ data points of the original dataset, and testing over the following $horizon$ data points. Note that the tested data points of one fold are then included as part of the next training dataset fold. In our experiments we used $windowSize = 300$, and $horizon = 100$. We implemented this procedure using the caret R package [35].

VII. Results

The experimental results consist on a number of different performance measures computed over the test set, which

Algorithm	Parameter	Values
Random forest	mtry*	2,4,6,8,..60
Random forest	ntree	500
Bayesian networks	SLA	tabu, hc
Bayesian networks	score	loglik-g, aic-g, bic-g, bge
C5.0	trials	1
C5.0	model	tree
C5.0	winnow	true,false
Rpart	complexity parameter	0.1,0.2,0.3,0.4,0.5
Rpart	split	entropy

(*) $mtry$ maximum value depends on the number of variables in the dataset for each experiment. For example, in local config with $T = 1$ the dataset has only 6 variables, so $mtry$ values to try would be 2,4 and 6.

TABLE I: Tuning parameters for classification and regression models.

	Reference	
Prediction	True Positive (TP)	False Positive (FP)
	False Negative (FN)	True Negative (TN)

TABLE II: Confusion matrix for a binary classifier

differ for classification and regression models. For regression models, we report the MAE (Mean Absolute Error) and RMSE (Root Mean Square Error). If Y_{real} denotes the actual minimum temperature as reported in the test set, and Y_{pred} denotes the predicted value of the model, the metrics are defined as follows:

- $RMSE = \sqrt{\frac{1}{n}\sum (Y_{pred} - Y_{real})^2}$
- $MAE = \frac{1}{n}\sum_{t=1}^{n} |Y_{pred} - Y_{real}|$

For classification models, all performance measurements considered are computed over a data structure that summarizes prediction quality of classification problems called the confusion matrix. Table VII shows a schema of a confusion matrix for a binary classifier. Based only on the four quantities defined, namely, true and false positives, denoted TP and FP respectively, and true and false negatives, denoted TN and FN respectively; we can compute the following metrics:

- Sensitivity: $\frac{TP}{TP+FN}$, also known as true positive rate, probability of detection or recall. Higher values of sensitivity indicate that it is a good predictor of the positive class.
- Precision: $\frac{TP}{TP+FP}$ reflects how accurately is the predictor for predicting the positive class. The higher is this value the lower the chances of false positives.
- F1-score: $F1 = \frac{2 \cdot precision \cdot sensitivity}{precision + sensitivity}$, which represents a balance between precision and sensitivity.

Figures 3, 4, 5, 6, and 7 synthesize all the results we obtained from the experiments.

Figure 4 shows results for MAE (top) and RMSE (bottom), for the two regression models: Bayesian networks (left) and Random Forest (right). As shown, the config-all case (labeled *all*, and shown in the lower box) presents the best performance in all cases, with a lower value for both errors, both algorithms, and all locations, proving the benefit of adding information of neighboring stations. These figures also show that for both config-all and local configurations, the MAE and RMSE are below 3°C, a small value when compared to the large thermal variability in Mendoza during winter days that can reach up to 20°C [4].

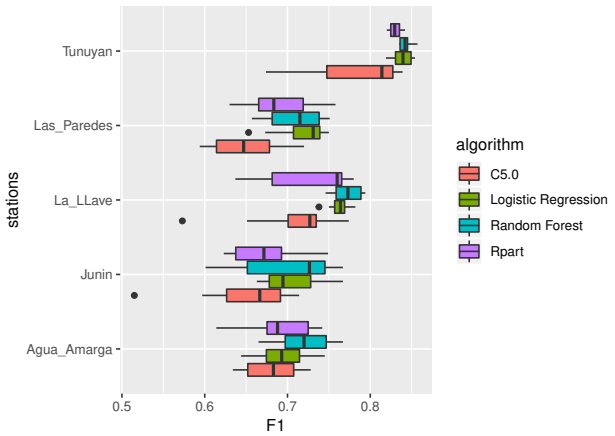

Fig. 3: Performance of classification algorithms

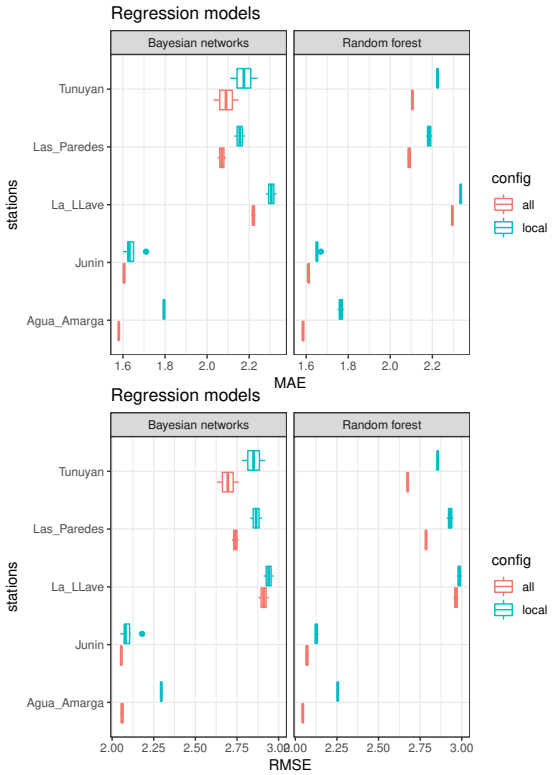

Fig. 4: Regression results: local vs all

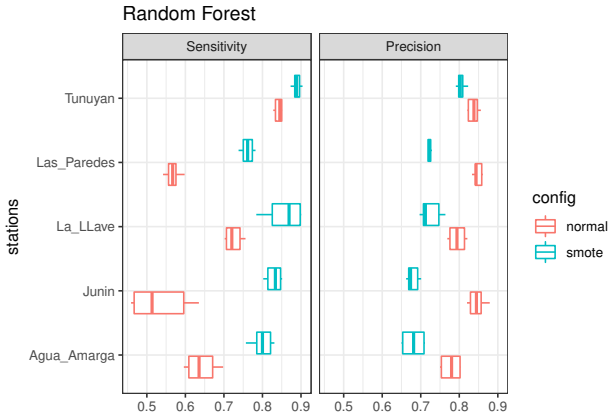

Fig. 5: Random Forest: SMOTE vs normal

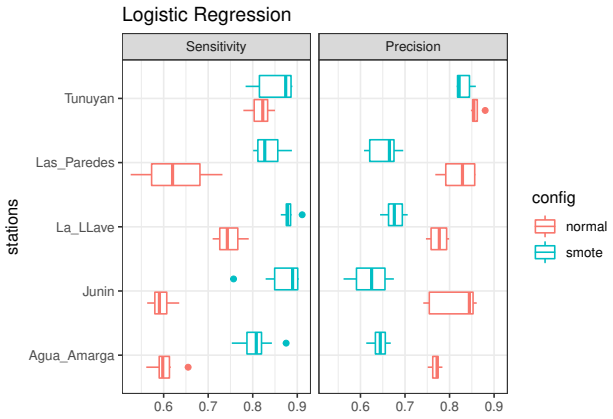

Fig. 6: Logistic Regression: SMOTE vs normal

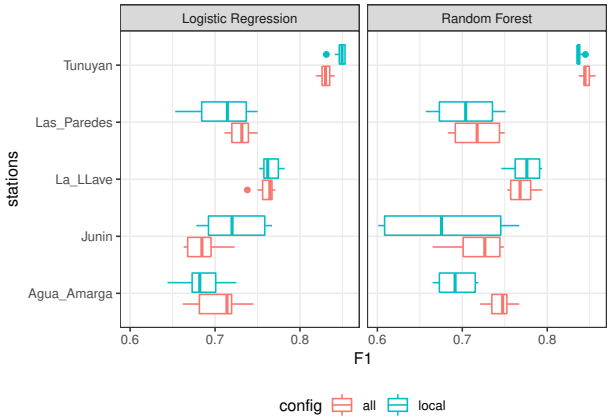

Fig. 7: Random Forest and Logistic Regression models: local vs all

Figure 3 shows a comparison between the F1-score for all four classification algorithms. As observed, RF and Logistic regression outperforms both C5.0 and Rpart in all cases. Thus, to simplify the following analysis, we decided to filter them out.

A known effect of SMOTE on the performance of classification algorithms is to increase the sensitivity while reducing precision, a pattern followed for our data, as shown by Figures 5 and 6 for algorithms RF and Logistic regression, respectively, where SMOTE results (top box) are large for sensitivity (left column) for all five locations, while for precision (right column) are clearly lower.

F1-score results for logistic regression and RF models, are shown in Figure 7, showing a comparison between cases config-all (top box) and local (bottom box). We can see that the behavior between the stations is different (Figures 3, 7), due the fact they are apart from each other in different micro-climate zones: Agua Amarga and Tunuyán are in Uco Valley, Junín in East Valley, Las Paredes and La Llave are in South Valley. We can observe that Tunuyán station presents less variability in its F1 results. Therefore, adding information from neighbors does not help to improve F1 metric to all the stations. For RF, config-all helps to Tunuyan, Las Paredes, Junín, and Agua Amarga to increase the average of the F1. Agua Amarga improves F1 score with config-all for both models: RF and Logistic regression, and RF has a better performance. In the case of Logistic Regression, Junín and Tunuyán slightly increase the average F1 by using config-local. Despite that we show that our results regarding recall are similar or better than the previous approach from Lee et al. [7], where the recall value is between 0.7 and 0.8, there are two main differences between both studies. First, the work from Lee et al. uses a longer period: 40 years of weather data for training; and second, both scenarios are geographically different. Finally, we observe that farmers would prefer a higher recall value in presence of lower precision, because the cost of crop losses may be higher than the price of frost protection even including false positives. In this case, a combination of config-all and SMOTE could help to build a better fitted forecast engine.

VIII. Conclusion and Future Work

In this paper we have created an forecasting engine which is part of an IoT-enabled frost prediction system, which gathers environmental data to predict frost events using machine learning techniques. We have shown that our prediction capability outperforms current proposals in terms of sensitivity, precision and F1.

In particular, the application of SMOTE during the training phase has shown an improved performance in terms of recall in both RF and Logistic Regression models.

We have also observed that, in specific relevant cases, the inclusion of neighbor information helps to improve the precision or recall of the forecasted classification model. On the other hand, regression models have less error by including neighbor information. In these cases, including the spatial relationships, there is a resulting improvement in model performance. We hope to contrast this approach with other scenarios in the future.

IX. Acknowledgments

We want to thank to the Dirección de Agricultura y Contingencias Climáticas (DACC) for sharing data with us to make this work possible and the support from the STIC-AmSud program through the PEACH project (16STIC-08). This project was also possible because of contribution from the following Universidad Tecnológica Nacional (UTN, Argentina) funds: EIUTIME0003601TC: “Aprendizaje automático aplicado a problemas de Visión Computacional”, PID UTN 25/J077 “Predicción localizada de heladas en la provincia de Mendoza mediante técnicas de aprendizaje de máquinas y redes de sensores”, and EIUTNME0004623: “Peach: Predicción de hEladas en un contexto de Agricultura de precisión usando maCHine learning”.

References

[1] L. Saieg, “Casi 35 mil hectáreas afectadas por heladas,” November 2016. [Online; posted 20-November-2016], http://www.losandes.com.ar/article/casi-35-mil-hectareas-de-vid-afectadas-por-heladas.

[2] A. Barnes, “El Nino hampers Argentina’s 2016 wine harvest,” May 2017. [Online; posted 23rd-May-2017], http://www.decanter.com/wine-news/el-nino-argentina-2016-wine-harvest-305057/.

[3] M. Lehné, "Winemakers lose every year millions of dollars due to natural disasters," April 2017. [Online; posted 26th-April-2017], http://www.kit.edu/kit/english/pi_2017_051_winemakers-lose-billions-of-dollars-every-year-due-to-natural-disasters.php.

[4] F. G. Antivilo, R. C. Paz, M. Echeverria, M. Keller, J. Tognetti, R. Borgo, and F. R. Juñent, "Thermal history parameters drive changes in physiology and cold hardiness of young grapevine plants during winter," *Agricultural and Forest Meteorology*, vol. 262, pp. 227 – 236, 2018.

[5] T. Watteyne, A. L. Diedrichs, K. Brun-Laguna, J. E. Chaar, D. Dujovne, J. C. Taffernaberry, and G. Mercado, "Peach: Predicting Frost Events in Peach Orchards Using IoT Technology," *EAI Endorsed Transactions on the Internet of Things*, 2016.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[7] H. Lee, J. A. Chun, H.-H. Han, and S. Kim, "Prediction of Frost Occurrences Using Statistical Modeling Approaches," *Advances in Meteorology*, vol. 2016, 2016.

[8] D. Brunt, *Physical and dynamical meteorology*. Cambridge University Press, 2011.

[9] E. Eccel, L. Ghielmi, P. Granitto, R. Barbiero, F. Grazzini, and D. Cesari, "Prediction of minimum temperatures in an alpine region by linear and non-linear post-processing of meteorological models," *Nonlinear processes in geophysics*, vol. 14, no. 3, pp. 211–222, 2007.

[10] R. L. Snyder and C. Davis, "Principles of frost protection," *Long version–Quick Answer FP005) University of California*, 2000.

[11] J. J. Burgos, *Las heladas en la Argentina*. No. 632.1, Ministerio de Agricultura, Ganadería y Pesca, Presidencia de la Nación,, 2011.

[12] J. W. Smith, *Predicting Minimum Temperatures from Hygrometric Data: By J. Warren Smith and Others*. US Government Printing Office, 1920.

[13] F. Young, "Forecasting minimum temperatures in Oregon and California," *Monthly Weather Rev*, vol. 16, pp. 53–60, 1920.

[14] T. Prabha and G. Hoogenboom, "Evaluation of the Weather Research and Forecasting model for two frost events," *Computers and Electronics in Agriculture*, vol. 64, no. 2, pp. 234–247, 2008.

[15] X. Wen, S. Lu, and J. Jin, "Integrating remote sensing data with WRF for improved simulations of oasis effects on local weather processes over an arid region in northwestern China," *Journal of Hydrometeorology*, vol. 13, no. 2, pp. 573–587, 2012.

[16] R. J. Kuligowski and A. P. Barros, "Localized precipitation forecasts from a numerical weather prediction model using artificial neural networks," *Weather and forecasting*, vol. 13, no. 4, pp. 1194–1204, 1998.

[17] I. Maqsood, M. R. Khan, and A. Abraham, "Intelligent weather monitoring systems using connectionist models," *Neural, Parallel, and Scientific Computations*, vol. 10, no. 2, pp. 157–178, 2002.

[18] I. Maqsood, M. R. Khan, and A. Abraham, "Neurocomputing based Canadian weather analysis," in *Second international workshop on Intelligent systems design and application*, pp. 39–44, Dynamic Publishers, Inc., 2002.

[19] I. Maqsood, M. R. Khan, and A. Abraham, "An ensemble of neural networks for weather forecasting," *Neural Computing & Applications*, vol. 13, no. 2, pp. 112–122, 2004.

[20] L. Ghielmi and E. Eccel, "Descriptive models and artificial neural networks for spring frost prediction in an agricultural mountain area," *Computers and electronics in agriculture*, vol. 54, no. 2, pp. 101–114, 2006.

[21] G. Ovando, M. Bocco, and S. Sayago, "Redes neuronales para modelar predicción de heladas," *Agricultura Técnica*, vol. 65, no. 1, pp. 65–73, 2005.

[22] P. F. Verdes, P. M. Granitto, H. Navone, and H. A. Ceccatto, "Frost prediction with machine learning techniques," in *VI Congreso Argentino de Ciencias de la Computación*, 2000.

[23] P. Aguilera, A. Fernández, R. Fernández, R. Rumí, and A. Salmerón, "Bayesian networks in environmental modelling," *Environmental Modelling & Software*, vol. 26, no. 12, pp. 1376–1388, 2011.

[24] D. Margaritis, "Learning Bayesian network model structure from data," tech. rep., Carnegie-Mellon Univ. Pittsburgh PA School of Computer Science, 2003.

[25] R. N. Marco Scutari, "bnlearn: Bayesian Network Structure Learning, Parameter Learning and Inference," 2015. R package version 4.2.

[26] M. Scutari, "Learning Bayesian Networks with the bnlearn R Package," *Journal of Statistical Software*, vol. 35, no. 3, pp. 1–22, 2010.

[27] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[28] L. Breiman and A. Cutler, "Random Forests Leo Breiman and Adele Cutler website." [Online, last visited November 29th], https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm.

[29] A. Gobierno de Mendoza, "Dirección de Agricultura y Contingencias climáticas," April 2017. [Online 27th-November-2017], http://www.contingencias.mendoza.gov.ar.

[30] T. M. Therneau, E. J. Atkinson, *et al.*, "An introduction to recursive partitioning using the RPART routines," *Rpart package vignettes.*, 2018.

[31] T. Therneau, B. Atkinson, and B. Ripley, "rpart: Recursive Partitioning. R package version 4.1-13," *URL: http://cran.r-project.org/web/packages/rpart/index.html*, 2018.

[32] M. Kuhn, S. Weston, N. Coulter, and R. Quinlan, "C50: C.50 decision trees and rule-based models," *R package version 0.1.2, https://CRAN.R-project.org/package=C50*, vol. 50, 2014.

[33] F. original by Leo Breiman, R. p. b. A. L. Adele Cutler, and M. Wiener, "Breiman and Cutler's Random Forests for Classification and Regression," 2015. R package version 4.6-12.

[34] L. Torgo, "DMwR: Functions and data for data mining with R," *version 0.4.1*, 2010.

[35] M. Kuhn, J. Wing, S. Weston, *et al.*, "Package 'caret'," *Classification and regression training*, 2015.

[36] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Information Sciences*, vol. 191, pp. 192–213, 2012.