Prediction of Frost Events using Bayesian networks and Random Forest

Ana Laura Diedrichs, Facundo Bromberg, Diego Dujovne, Keoma Brun-Laguna, Thomas Watteyne

Prediction of frost events using Bayesian networks and Random Forest

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*—In a couple of hours, farmers can lose everything because of frost events. Handling frost events is possible by using a number of countermeasures such as heating or removing the surrounding air among crops. Given the socio-economical implications of this problem, involving not only loss of jobs but also valuable resources, there have been a number of efforts to design a system to predict frost events, but with partial success: either they are based on formulas needing many unknown coefficients, or the prediction performance is poor. In this paper we propose a new approach, which instead of using the sensor's past information for prediction, as state of the art methods do, we assume that frost prediction in one location could be improved by using the information of the most relevant neighboring sensors.**

However, given the small amount of frost events during the year, available data, even after incorporating information of neighboring sensors, it is not enough to build an accurate frost forecasting system, which defines an unbalanced dataset problem. In order to overcome this disadvantage, we propose to use machine learning algorithms such as Bayesian networks and Random Forest where the training set includes new samples using SMOTE (Synthetic Minority Oversampling Technique). Our results show that selecting the most relevant neighbors and training the models with SMOTE increases significatively the frost detection rate of the predictor, turning the results into a useful resource for decision makers.

***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. 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 a result into partial damage in different levels in the same crop field, but a frost event can 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 data to build a predictive model with high accuracy and/or precision.

In this paper, we propose to use a different approach: We use machine learning algorithms based on Bayesian Networks and Random Forest, over a balanced training set augmented with new samples produced using the SMOTE (Synthetic Minority Oversampling Technique)[4] technique. This technique increases the rate of frost detection (sensitivity) and decreases

the observed root mean square error.

We chose Random Forest (RF) and Bayesian networks (BN) because they are widely used algorithms for decision support applications and both can resolve classification and regression problems. RF 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 brakes 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 the 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 h 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. In section II we discuss previous works on daily minimum temperature (frost) prediction. We introduce Bayesian networks in section III-A and RF in III-B. We describe in section IV our scenario of interest and datasets to be used to train the models. Then, we explain our experimental setup in V, follows by the results section VI. Finally, we share our conclusions in section VII

II. 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.[5] which is applied in [6] and the work from Allen et al. [7]. A complete review of classical frost prediction methods can be found from Burgos et al.[8], 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 [9] and Young [10], comparing the minimal prediction accuracy. As a matter of fact, the United States National Weather Service has throughly 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.[11] 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.[12] 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 the frost prediction[13], [14], [6], [15], 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.[16], 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.[17]. 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. [6] 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.[18] and Verdes et al.[19] 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. [20] 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 78% 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.

III. Machine learning methods

A. Bayesian networks

The work of Aguilera et al.[21] on Bayesian networks (BN) for environmental modeling mentions the benefits of using BN in terms of 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 represent the variables, and each directed edge from the set of edges A represents direct, i.e., non-mediated, probabilistic influence. In an alternative of the graph it can also represent cause-effect from one variable to the other.

The DAG defines a factorization of the joint probability distribution over random variables $V = X_1, X_2, .., X_M$, also known as global probability distribution, into a set of local probability distributions, one for each variable. The factorization form is given by the Markov property of BN, which states that every random variable X_i directly depends only on its parents π_{X_i}, $P(X_1, ..., X_M) = \prod_{i=1}^{M} P(X_i|\pi_{X_i})$. We define likelihood as the probability of observed data D given a model M with parameters θ,

$$L(\theta) = P(D|\theta, M) = P(x_1, x_2, .., x_m|\theta) \tag{1}$$

and maximum likelihood as $\hat{L} = \hat{\theta}_{ML} = \arg\max_\theta L(\theta)$

The Markov blanket of a node X_i, also denoted as $MB(X_i)$, is composed by the parents of X_i, the children of X_i, and the other parents for these children. The Markov blanket implies a set of nodes that probabilistically separates X_i from the rest of the graph, and therefore includes all the knowledge needed to infer any probabilistic information of X_i. Given its graphical representation, a graph can be produced completely by its Markov blankets, so many structure learning procedure learn Markov blanket (directly or indirectly). A blanket example can be observed on Figure 1.

BNs can be built manually by drawing the direct cause-effect relationships between the variables using domain expert knowledge, or autonomously with structured learning algorithms that elicitate the network structure completely from input data. There are two major approaches for autonomous structured learning: the score-based approach and the constraint-based approach. The score-based approach assigns a score to each candidate BN to evaluate how well the BN fits the data, typically measured with some version of the likelihood of the data for that DAG, and then searches over the space of DAGs for a structure with maximal score with an heuristic search algorithm. Greedy search algorithms (such as hill-climbing or tabu search) are a common choice, but almost any kind of search procedure can be used. The global distribution of a continuous variable Bayesian network model is a multivariate normal and the local distributions are normal random variables linked by linear constraints. These Bayesian networks are called Gaussian Bayesian networks.

We propose to model a state-based Bayesian network which represents the state of each variable at discrete time intervals; as a consequence, the Bayesian network results into a series of time slices, where each time slice indicates the value of each variable at time t. Our approach involves the learning of Gaussian Bayesian networks from real data using the following score-based structure learning algorithms: Hill Climbing (HC)[22] and Tabu Search (Tabu) from *bnlearn*, provided by R as a package [23], [24]. Maximum Likelihood estimates are used to fit the parameters of a Gaussian Bayesian network, using the regression coefficients for each variable against its parents. We have used the following scores, available from the *bnlearn* package for HC and Tabu:

- The multivariate Gaussian log-likelihood score (loglik-g), $loglik = \log \hat{L}(\theta)$.
- The corresponding Akaike Information Criterion score (aic-g), with $k = 1$.
- The corresponding Bayesian Information Criterion score (bic-g), with $k = \frac{1}{2}\log(D)$, where D is the number of datapoints.
- A score equivalent Gaussian posterior density [25], [26] (bge)

aic-g and bic-g are computed as $2k * nparams(V) - 2 * loglik(V)$, with $nparams$ as the number of parameters θ of the model of V variables.

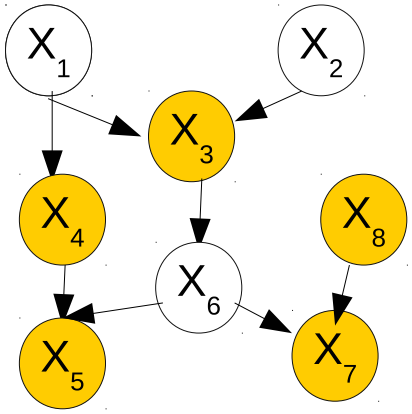

Fig. 1: Bayesian network example for eight variables. The orange nodes are the Markov blanket of X_6, $MB(X_6)$. The distribution of X_6 given its parents is $P(X_6|X_3) \sim \mathcal{N}(\theta_0 + \theta_1 X_3, \sigma^2)$, where $\theta_0, \theta_1, \sigma^2$ are parameters learned/estimated via maximum likelihood method. We factorize this BN as

$$P(X_1, .., X_8) = P(X_1)P(X_2)P(X_3|X_1, X_2)P(X_4|X_1)$$
$$P(X_5|X_4, X_6)P(X_6|X_3)P(X_7|X_6, X_8)P(X_8)$$

B. Random Forest

Random Forest (RF) [27] is a machine learning method that, same as BN, can be applied to 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 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).

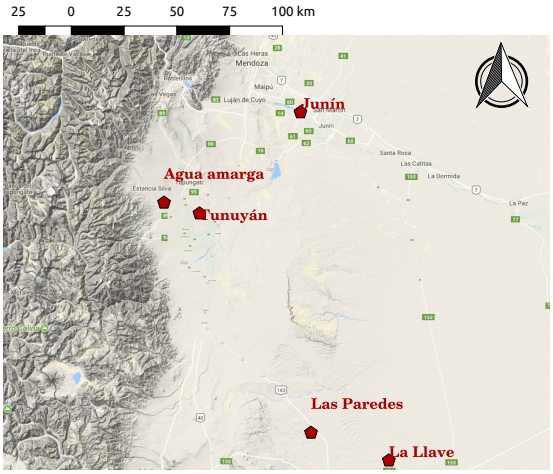

Fig. 2: Map of the DACC's stations located in Mendoza, Argentina.

IV. 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 Figure2, 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)

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

Each location has temperature and relative humidity sensors. The period we consider spans from 2001 until 2016. We create a dataset summarizing for each day the average, minimum and maximum of temperature and humidity, resulting in six variables per location, per day.

Our datasets reflects the type of prediction we intended, where the learned model should accurately predict the temperature of a given day using thermodynamic information from previous days. Each datapoint was constructed concatenating the five thermodynamic variables of each of T previous days, and labeled by the variable we want to predict at the right-most position. In the experiments we considered $T = 1, 2, 3, 4$.

In part of our experiments, we analyze the impact of humidity and temperature on the result. To achieve this goal, we generated datasets containing only temperature information (*dacc-temp*) and another one containing information on all the sensors (*dacc*). Also, we considered another dataset containing only data from the Spring season in Mendoza, Argentina (*dacc-spring*), corresponding to: August, September, October and November.

V. Experiment setup

Our experiments involved several steps as depicted on Figure 3. In order to train the machine learning models (RF and BN), we split the dataset in train and testing sets. The first one is used by the algorithms to fit the parameters to the data, while the second one is used to validate the real behavior of the models under unseen conditions. We setup to split 68% of the first part of the dataset for the training phase and the rest for testing purposes.

On the training phase we considered two setups: the original dataset and one using the SMOTE. SMOTE involves a combination of minority class over-sampling and majority class under-sampling. We choose a three time over-sampling of the minority class (an 300%), e.g if we have only 100 datapoints with frost days, the oversampling technique will be create 300 frost datapoints and take another 300 from the majority class. In this case, the days in T=0 with frost events (meaning temperature below zero) were labeled in our dataset prior to the use of SMOTE, to obtain a balanced train-set to train the models. To achieve this goal, we used the R package *unbalanced* [30].

For the Bayesian structure learning process we define a white and a black list of variable relationships to model our prediction problem, with the white list indicating the relationships that the structure learning algorithm must include, and the black list indicating the forbidden ones. For our prediction problem, the white list includes all edges running between each thermodynamic variable during the previous day into the corresponding variable at of the same node at theprediction date, resulting in 5 edges for each day included in the prediction, assuming only one prediction variable. The black list instead, contains edges between variables at different locations which correspond to the prediction date. All other edges not included in either list are elicited from the data with the structure learning algorithm.

We then train the Bayesian network models using HC and Tabu from *bnlearn* R package [24], [23] with their default values for all the selected scores and the following experiment parameters:

- No restarts where considered for HC and Tabu.
- Equivalent sample size in BGe (iss) is 10.
- The prior ϕ matrix formula to use in the BGe score: Heckerman method is used which computes the posterior Wishart probability.

We use the *randomForest* R package [31] to implement RF experiments. We tried different values of *mtry* (from 10 to 20 in steps of 1) and *ntrees* (500,1000,1500,2000,2500), where *mtry* is the number of variables which are selected for the decision tree split node, and *ntree* is the maximum number of generated trees.

The rf-local configuration involves only the local variables not neighbors, as the local configuration for BN. In this case the *mtry* is setup by default as the squared root of the number of input variables.

VI. Results

In order to analyze how well the models predict the temperature, we choose RMSE and r^2 as regression metrics. Given

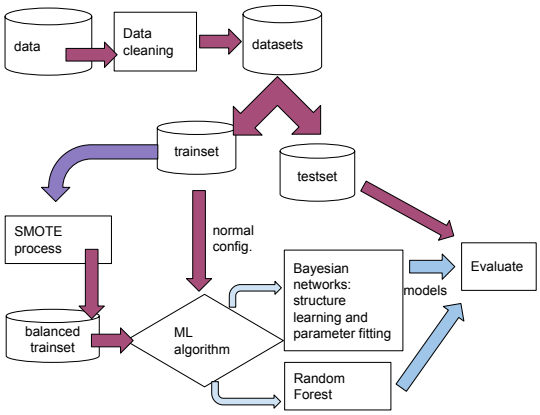

Fig. 3: Experiment work-flow diagram.

	Reference (observed value)	
Predicted	True Positive (TP)	False Positive (FP)
	False Negative (FN)	True Negative (TN)

Fig. 4: Confusion matrix for a binary classifier

the test set, Y_{real} is a vector with the real values from the test set (the minimum temperature of one location that we want to predict) and Y_{pred} a vector with the predicted values from a model. We use the following metrics to analyze the results:

- RMSE: root mean square error, $RMSE = \sqrt{\frac{1}{n}\sum (Y_{pred} - Y_{real})^2}$
- r squared or Pearson coefficient of correlation.

We analyze how well the models perform in terms of frost prediction by using a confusion matrix, which is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. We define a frost event as below zero degree Celsius to build the confusion matrix, and we consider the frost event as the positive class. We discretized the values of Y_{pred} and Y_{real} vectors to analyze the results as a binary classifier.

Given a confusion matrix, Figure 4, for a binary classifier, we can compute the following metrics:

- Sensitivity: $\frac{TP}{TP+FN}$ also known as true positive rate, probability of detection and recall. Higher values of sensitivity indicates that we have 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.
- Accuracy: $(TP + TN)/(TP + TN + FP + FN)$
- Specificity: $TN/(FP + TN)$, probability of detection of the negative class

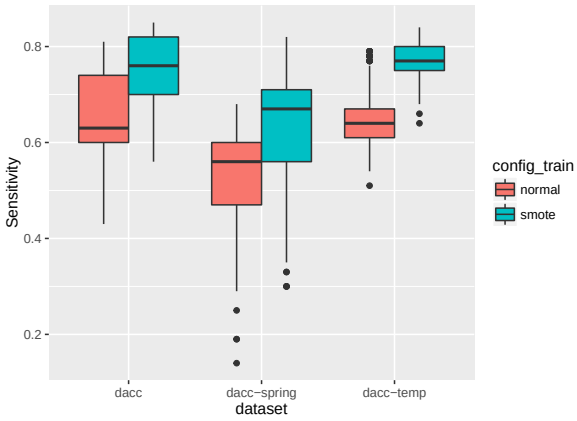

Fig. 5: Sensitivity by dataset in BN experiments

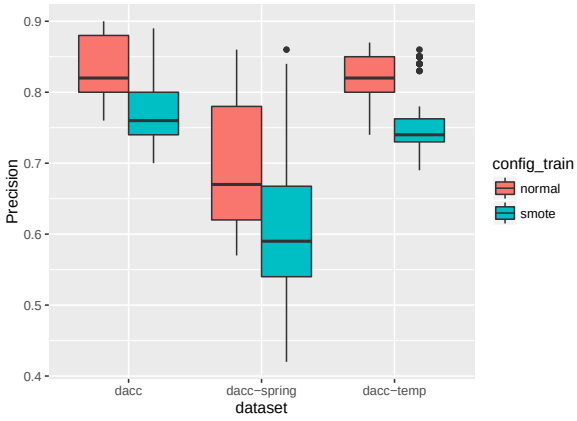

Fig. 6: Precision of the best models selected by scenario

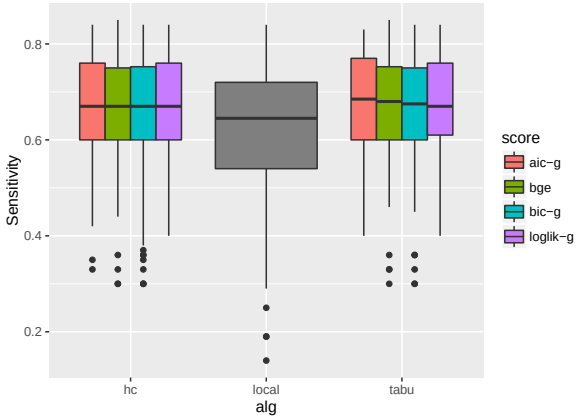

Fig. 7: Sensitivity by Bayesian network structure learning algorithm and score

location	dataset	days	alg	ntree	mtry	score	rmse	r2	sensitivity	accuracy	precision	specificity
Junin	dacc-temp	2	rf	1500	23		2.17	0.9	**0.84**	0.94	**0.7**	0.95
Junin	dacc-temp	2	hc			loglik-g	2.14	0.9	**0.77**	0.94	**0.76**	0.97
Agua Amarga	dacc-temp	3	tabu			bge	2.23	0.89	**0.81**	0.93	**0.73**	0.95
Agua Amarga	dacc-temp	1	rf	2000	11		2.24	0.88	**0.85**	0.93	**0.73**	0.95
Tunuyan	dacc	2	hc			bge	2.59	0.87	**0.85**	0.91	**0.87**	0.94
Tunuyan	dacc-temp	1	rf-local	2500			2.97	0.83	**0.88**	0.9	**0.81**	0.91
La Llave	dacc	3	hc			loglik-g	2.94	0.85	**0.84**	0.9	**0.76**	0.92
La Llave	dacc	3	rf-local	1500			2.97	0.84	**0.86**	0.91	**0.75**	0.92
Las Paredes	dacc	2	tabu			aic-g	2.82	0.84	**0.79**	0.91	**0.73**	0.93
Las Paredes	dacc-temp	1	rf	1500	15		3.08	0.81	**0.79**	0.91	**0.72**	0.93

Fig. 8: Table of the best models for each location in terms of sensitivity and precision. All the models have used SMOTE during the training phase

Figures 5,6,7 synthesize all the results we obtained from the experiments and table 8 groups the best models per scenario.

The best results in terms of sensitivity and recall factors (for Random Forests and Bayesian networks) is generated by the application of SMOTE to the training set, as seen on Figure 5. We have also noticed that, by applying SMOTE to the training set, the training set reduces its size to 50% at most. We found that better than increasing the amount of data to build a larger training set, it is important to add meaningful information to the problem.

The effect of SMOTE on the training results is to increase the sensitivity while reducing precision, as it is observed on Fig. 6. This is due to the fact that the algorithm learns by repetition, so diversity (meaning a variety of frost events) is reduced. An increase on diversity can be obtained using a longer period of training data to provide a wider range of frost events. However, historical data availability from weather stations is not a general rule in practice.

Comparing the datasets, we noticed dacc-spring dataset has the worst performance in average in terms of precision and recall (which can be observed on Figures 5 and 6). As a consequence, the data from the other seasons is relevant to the learning process. Mendoza has a dry and desert-type weather with high temperature span during the day. This could be the reason why the dacc-temp dataset has performed better in most scenarios (see Table 8).

The Bayesian Network experiments did not show statistically significant changes between the scores and the algorithms applied to the input data, as depicted on Figure 7. This means that there is no score and algorithm combination showing a better than average result. To the contrary, local configuration reduces sensitivity, so the consideration is to use multiple sources in Bayesian Networks.

On the table shown on Figure 8, we select the best models per scenario, and for each of them, the best Bayesian Network and the best Random Forest model. When sensitivity is taken into account, Random Forest models have a better prediction capability than Bayesian Networks, however, Bayesian Networks stand out in terms of precision.

Finally, RF has a trend to select local data sources as the best performing one (*rf-local*), while Bayesian Networks tend to use local and neighbor data sources to improve prediction performance. This happens because RF finds the local variables as the most important. In contrast, the structure learning

approach assigns more score to the non-local configuration.

VII. Conclusion and Future Work

In this paper we have created a forecasting 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, recall and accuracy. Furthermore, the proposed system can be applied to decision support systems as a product.

In particular, the application of SMOTE during the training phase has shown in both RF and BN models, an improved performance in terms of recall. The best BN models were competitive in terms of sensitivity and precision.

We also show that, in specific cases, the inclusion of neighbor information helps to improve the accuracy of the forecast model. 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.

Our future work will include testing HC and Tabu using different value configurations, by adding random restarts. Since both are heuristic search algorithms, they stop when the optimum value is found, even if it arrives to a local optimum. Restarts forces the algorithms to change the space search direction for the optimum, increasing the chances of arriving to the global optimum. We are also interested on trying others training phase configuration (cross-validation, other oversampling techniques) and structure learning approaches.

VIII. 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. This project was also possible because of contribution from the following Universidad Tecnológica Nacional (UTN) 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], url: 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], URL: 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] 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.

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

[6] 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.

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

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

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

[10] F. Young, "Forecasting minimum temperatures in oregon and california," *Monthly Weather Rev*, vol. 16, pp. 53–60, 1920.

[11] 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.

[12] 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.

[13] 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.

[14] I. Maqsood, M. R. Khan, and A. Abraham, "Intelligent weather monitoring systems using connectionist models," *NEURAL PARALLEL AND SCIENTIFIC COMPUTATIIONS*, vol. 10, no. 2, pp. 157–178, 2002.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] D. Margaritis, "Learning bayesian network model structure from data," tech. rep., CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 2003.

[23] R. N. Marco Scutari, "bnlearn: Bayesian network structure learning, parameter learning and inference," 2015. R package version 4.2.

[24] M. Scutari, "Learning bayesian networks with the bnlearn R package," *Journal of Statistical Software*, vol. 35, no. 3, pp. 1–22, 2010.

[25] D. Heckerman and D. Geiger, "Learning bayesian networks: a unification for discrete and gaussian domains," in *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pp. 274–284, Morgan Kaufmann Publishers Inc., 1995.

[26] D. Geiger and D. Heckerman, "Learning gaussian networks," in *Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*, pp. 235–243, Morgan Kaufmann Publishers Inc., 1994.

[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 29th November,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] A. D. Pozzolo, O. Caelen, and G. Bontempi, "unbalanced: Racing for unbalanced methods selection," 2015. R package version 2.0.

[31] 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.