POLITECNICO DI TORINO
Repository ISTITUZIONALE

Machine Learning Methods for Leaf Wetness Prediction using
Agrometeorological Data

Original
Machine Learning Methods for Leaf Wetness Prediction using
Agrometeorological Data / Colucci, G.P., Filipescu, E., Scatozza, F., Trinchero, D.. - In: IEEE TRANSACTIONS ON AGRIFOOD ELECTRONICS.. - ISSN 2771-9529. - ELETTRONICO. - (In corso di stampa).

Availability:
This version is available at: 11583/3012963 since: 2026-07-11T15:25:04Z

Publisher:
IEEE

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©9999 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

10 August 2026

Machine Learning Methods for Leaf Wetness Prediction using Agrometeorological Data

Giovanni Paolo Colucci, Elena Filipescu, *Student Member, IEEE,* Fabio Scatozza, and Daniele Trinchero, *Senior Member, IEEE*

***Abstract*—This paper addresses the estimation of leaf wetness by means of machine learning methods applied to agrometeorological data. The novel contribution lies in the application of the models to the top and bottom sides of the leaf, separately. For this purpose, data have been acquired in the field using weather stations equipped with several sensors: double-sided capacitive leaf wetness, air temperature and relative humidity, soil temperature and volumetric water content, wind speed and rainfall. Leaf wetness measurements have been used as ground truth, whereas the other sensor data as features. Results demonstrate the applicability of the method, which has also been tested on edge. The dataset has been made publicly available under a Creative Commons license.**

***Index Terms*—Machine Learning, Smart Agriculture, Agrometeorology, Weather Stations, Proximal Sensors, Leaf Wetness Dataset**

I. Introduction

Leaf wetness is an important agrometeorological parameter used to describe the presence of water on the surface of a leaf. It is often used in agriculture, since continuous periods of leaf wetness are directly associated with the spread of fungal and bacterial diseases [1]. A comprehensive assessment of this phenomenon must account for water accumulation on the adaxial (upper) and abaxial (lower) surfaces of the leaf, since they may exhibit different behaviors [2]. In particular, the surrounding environment may affect the two wetness durations differently because of factors such as the presence or absence of trees and grassy soil, as well as slope characteristics, soil texture, and sun exposure.

Monitoring this quantity has become essential for smart agriculture applications. To this end, several proximal sensing solutions have been proposed in recent years. Resistive and capacitive approaches are most commonly employed to transduce the presence of water on an artificial leaf surface into a measurable signal. The former measures the variation of the sensor resistance induced by the presence of water on its surface. The latter measures the dielectric constant between two electrodes, which varies, depending on the amount of water on the sensor surface [3].

While these sensors are suitable for in-situ deployments and provide good accuracy, they introduce additional complexity for weather stations, with higher manufacturing costs, increased energy consumption, and more frequent maintenance requirements. For this reason, leaf wetness is sometimes indirectly estimated by means of simple empirical models. For example, threshold-based models detect leaf wetness when another parameter, usually the Relative Humidity (RH) of the air, exceeds a fixed value; a common rule in this case is to consider RH $> 90\,\%$ [4]. Another widely employed model is the Dew Point Depression (DPD), computed as the difference between the measured temperature and the dew point temperature. The leaf is considered wet when the DPD remains between two values, referred to as the wetness onset and offset, respectively [5].

Higher accuracy in leaf wetness prediction can be achieved by employing machine learning together with additional agrometeorological parameters [6]. Several approaches have already been proposed in the literature, however, none of them explicitly considers the difference between the two sides of the leaf, as they use commercial leaf wetness sensors, typically one-sided. In this paper, we investigate the application of different machine learning models to estimate leaf wetness on the two surfaces of the leaf, using a dataset collected over several years of measurements in the field, by means of double-sided capacitive leaf wetness sensor combined with other agrometeorological parameters used as features. The problem under analysis is formulated as a binary classification, with two labels: wet and dry; each hourly observation is treated as an independent instance. The classification is carried out by considering the two sides of the leaf separately. Additionally, the portability of the machine learning models to low-power edge devices is assessed, in order to guarantee functionality in scenarios where leaf wetness must be processed directly on the sensing node or when data transmission is not possible. Finally, the dataset is made available as an online resource [7].

II. State of the Art

In literature, several machine learning methodologies have been proposed to predict leaf wetness, typically by testing different datasets and algorithms. As early as 1997, researchers tried to apply neural networks to leaf wetness, by collecting data with custom hardware equipment [8]. In general, the difficulty of building datasets through continuous and prolonged measurements has represented a research limitation until the recent years. Nowadays, an important data source is represented by climatological models, such as MERRA-2 [9] and ERA-5 [10]. These datasets have been used to develop a leaf wetness estimator in Alabama and California, thanks to observations obtained by leaf wetness sensors and machine learning algorithms [11]. Regional weather stations have also

The authors are with the Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Torino, Italy (email: giovanni.colucci@polito.it; elena.filipescu@polito.it; fabio.scatozza@polito.it; daniele.trinchero@polito.it).

been used as input sources, but these installations usually lack dedicated leaf wetness sensors. To overcome this problem, researchers have nearby deployed custom sensors that have been used as models ground truth. For example, authors in [12] have exploited a leaf wetness sensor based on a resistive grid unit, to analyze different modeling approaches: regression and classification with different time steps. Also, authors in [13] have exploited regional weather stations in South Korea, to develop a regression model for daily leaf wetness duration estimation. Nonetheless, climatological models and regional weather stations have been merged to feed the same regression model and estimate daily leaf wetness duration [14]. Satellite imaging is an alternative data source. For example, narrow-band radiance observations from the GEO-KOMPSAT-2A have been used to develop a model that does not require in-situ meteorological measurements [15].

Although sensors represent the primary source of ground truth, alternative approaches have been explored. As an example, the output of the Penman–Monteith model [16], capable of determining leaf wetness, has been used to train machine learning algorithms with the purpose of reducing the number of input variables [17]. More recently, a novel approach combining terahertz spectroscopy and machine learning has been proposed. The authors have defined the leaf wetness as the weight of a pattern of water droplets at the surface of the leaf, and two different algorithms have been tested: decision tree and neural network [18].

In the study [19], researchers have built a classifier capable to predict the source of wetness among three different labels: dew, irrigation and rain. To this purpose, a frequency domain analysis, based on the implementation of the short-time Fourier transform, has enhanced data representation to feed two neural networks [19].

The performance of machine learning models has been compared with that of classical empirical methods. Authors in [20] have reported better scores for the former, concluding also that, when dealing with large geographical areas, a sub-regional partitioning of the dataset improves the accuracy. In the case of peculiar microclimate conditions, empirical models have achieved adequate performance, as demonstrated in [6].

None of the aforementioned studies has analyzed the difference between the two sides of the leaf, even though experimental observations have proved that they exhibit different behaviors [2, 21]. Therefore, in this paper we apply model predictions to both sides of the leaf, separately. An additional contribution is the publication of an open-access dataset, which features measurements from double-sided capacitive leaf wetness sensors deployed in vineyards across different regions of Italy.

III. Materials and Methods

A. Dataset Acquisition

The dataset has been created using the iXemWine platform [22], developed by our team to collect agrometeorological data measured by weather stations deployed across the Italian territory since 2018. The weather stations are equipped with the following sensors: air temperature and relative humidity (Sensirion SHT31), rainfall (Pronamic tower rain), temperature and volumetric water content of the soil (Senseca HD3910), wind speed and wind gust (Navis WSS 100), and an in-house manufactured double-sided capacitive leaf wetness sensor, designed applying a microstrip approach which leads to a multi-layered PCB, where each capacitance is measured by a digital converter [21]. Data are collected from stations deployed alternatively near the crops or outside the canopy. The leaf wetness sensor is tilted down 30°; the soil sensor is buried 30 cm in the ground, installed horizontally and rotated 45° around its axis; the Sensirion SHT31 sensor is inserted in a dedicated Stevenson screen. As an example, three weather station deployments are shown in Fig. 1.

Figure 1: Examples of weather stations used to acquire the dataset; the leaf wetness sensor can be recognized in the lower part of the pole.

Table I: Location, elevation and characteristics of the sites selected to acquire the dataset.

Site	Latitude (DD)	Longitude (DD)	Elevation (m)	Characteristics
1	39.108	8.712	125	valley by the sea
2	38.987	8.577	33	sea side
3	43.077	11.468	242	medium hilly
4	43.018	13.746	258	high hilly
5	44.722	8.076	187	medium hilly
6	45.533	7.854	434	medium mountain
7	45.559	7.835	418	mountain valley

Among the available locations, a subset of seven sites has been selected, in order to provide a representative set of agrometeorological scenarios. In Table I, the sites are listed reporting latitude and longitude in decimal degrees, elevation in meters and environment type; the map in Figure 2 shows the positions of the sites.

Sites 1 to 4 are characterized by a Mediterranean climate. In particular, sites 1 and 2 are located in southern Sardinia, in a maritime environment with sandy soil. Sites 3 and 4 are instead located in central Italy, in Tuscany and Marche, respectively and lie at medium altitude in a hilly environment. Sites 5 to 7 are characterized by a continental climate, in the Piedmont region. Site 5 is located in the territory where Barbaresco wine is produced, at a medium altitude. Sites 6 and 7 are located in northern Piedmont, at the highest elevation among the considered sites, in a mountain environment characterized by colder and windier conditions.

Figure 2: Map of Italy: the red triangles represent the position of the weather stations used to acquire the dataset.

Table II: List of features with their dynamic ranges.

Feature	Dynamic Range
Temperature	-40 to $125\,°\mathrm{C}$
Relative Humidity	0 to $100\,\%$
Soil Temperature	-40 to $60\,°\mathrm{C}$
Soil Volumetric Water Content	0 to $60\,\%$
Wind gust	0 to $50\,\mathrm{m/s}$
Rain Fall	0 to $720\,\mathrm{mm/h}$
Day of Year	1 to 365
Hour of Day	0 to 23

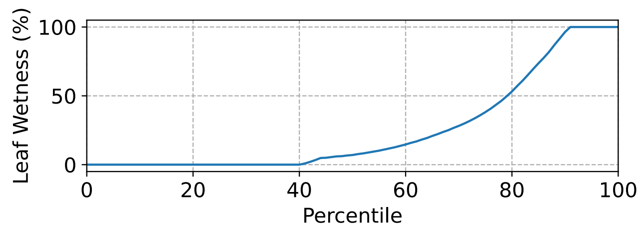
B. Dataset Characterization and Features Selection

Since agrometeorological data may comprehend several parameters, acquired with different timesteps, a first task has been dedicated to the characterization and selection of dataset features.

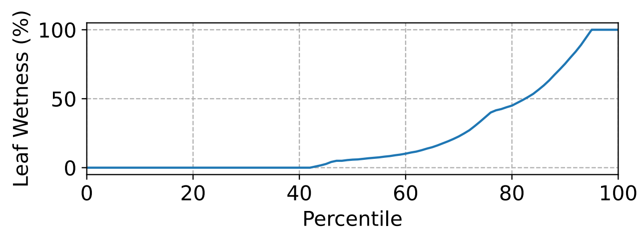
Normally, agrometeorological data are used as input to plant disease models, that work with a minimum time step of one hour. Therefore, the raw sensors data have been resampled on an hourly base. Mean values have been used for all measurements, except rainfall, for which the hourly intensity has been computed, and wind gust, for which the hourly maximum has been selected. Both wind speed and wind gust have been tested as potential features; however, wind gust has been preferred, as its correlation with leaf wetness was found to be more significant. Furthermore, to enrich the feature space, two variables have been added: the day of the year (1 to 365) and the hour of the day (0 to 23), both represented as integer values. This approach is often preferred to the use of raw timestamps, because it helps to better describe seasonal and daily trends. Table II summarizes the adopted features with their corresponding dynamic ranges, derived from sensor datasheets. The low dimensionality of the dataset is clearly evident.

To exclude data originated by sensor failures, the raw dataset has been preliminarily analyzed. For each weather station, rolling-window means, rolling-window standard deviations and z-score have been computed to detect outliers. Samples identified as anomalous have been visually inspected, resulting in 54 643 samples retained out of a total of 54 706.

Figure 3 reports the percentile distribution of the leaf wetness values for the top and bottom sides of the leaf. Both sides exhibit a similar percentile behavior: the data are not fairly distributed around the $50\,\%$ of leaf wetness, with the median value being lower; the variation from 0 to $100\,\%$ is quick, especially for the values larger than $50\,\%$. On the contrary, the bottom side has a lower amount of samples at $100\,\%$, and the percentile curve is less regular with a couple of cusps.



(a) Top side

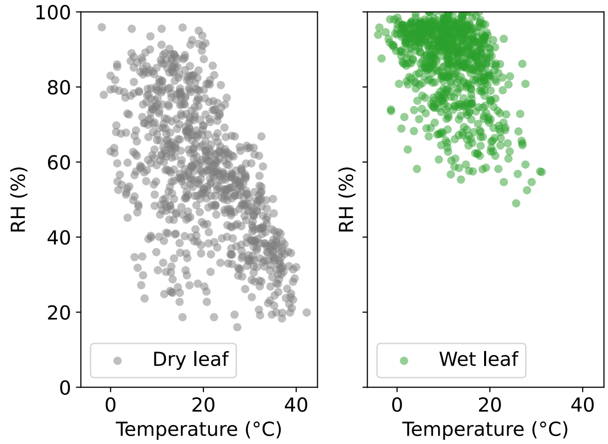


(b) Bottom side

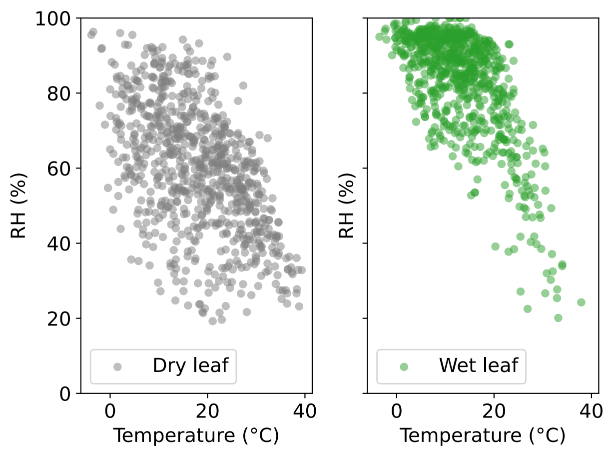
Figure 3: Dataset percentile distribution of leaf wetness sensors measurements for the adaxial and abaxial surfaces.

Since the aim is to solve a binary classification problem, a threshold value must be defined to label each sample as wet or dry. According to the scientific literature, a threshold associated with visual inspection is $20\,\%$ [6]. In this paper, this value has been adopted, although results have been compared to alternative choices: $30\,\%$ and $40\,\%$. From the data presented in Fig. 3, it can be observed that $20\,\%$ approximately corresponds to the 65th and 68th percentiles for the top and bottom sides, respectively.

Before training the algorithms, a visual analysis of the dataset has been performed. Figure 4a shows a scatter plot of dry and wet samples for the top side of the leaf, against temperature and relative humidity. Some peculiar characteristics can be observed: below $60\,\%$ relative humidity and above $25\,°\mathrm{C}$, wet samples are rarely present. This aligns with the common expectation that leaf wetness on the top side is primarily influenced by air relative humidity [23]. Conversely, dry samples are more uniformly distributed across the plot and appear less trivial to predict; although, at very high values of relative humidity, a predominance of wet samples can be observed. Regarding the bottom side, the scatter plot in Fig. 4b reveals a similar overall distribution, except that wet samples also occur at lower values of relative humidity. This suggests that, compared to the top side, wetness on the bottom side may be affected by additional factors. Visually, other features appear to provide less discriminative information: soil data usually exhibit a low variance due to the smoother dynamics of underground conditions, while wind gust and especially rainfall generate sparse data, as these phenomena occur less frequently.



(a) Leaf top side



(b) Leaf bottom side

Figure 4: Scatter plots showing the distribution of samples labeled as dry (left) and wet (right), in the temperature and relative humidity domain, with a leaf wetness threshold rule of 20 %, for top and bottom leaf surfaces.

C. Training Set Preparation

The dataset has been shuffled and split into training and test sets with the following proportion: 80 % and 20 %. The training set is not balanced around the target variable: the top side is composed by 36 % of wet samples and 64 % of dry samples; the bottom side is composed by 32 % of wet samples and 68 % of dry samples. Machine learning classifiers trained on imbalanced datasets may become biased toward the dry class, reducing the sensitivity to the wetness events, and this would be critical in agronomic applications. Therefore, to obtain models that better generalize the two labels, the training set has been balanced by randomly undersampling the dominant class, resulting in equal numbers of samples for each class.

Since input features represent different physical phenomena with different scales, the Min-Max scaler has been used for feature normalization. Notably, sparse features such as rainfall and wind have a minimum value of zero, and the resulting scaled samples remain sparse. Afterward, the training set has been used for hyperparameters tuning on each model, by means of grid search with a 5-fold cross-validation strategy.

D. Classification Algorithms

Distinct models have been tested to assess their performance on the leaf wetness prediction task. In the following, a brief description of the selected algorithms is provided. All operations have been performed using the Python packages pandas for data analysis [24] and scikit-learn for machine learning models [25].

Linear Discriminant Analysis (LDA) This algorithm aims to find a linear combination of features that better separate the target classes with a linear decision surface. This is achieved by considering the data distribution of each class and using Bayes' theorem [26].

Decision Tree (DT) Decision rules are encoded using a binary tree structure in which leaves are used for final predictions. The dataset is partitioned in a recursive way. For each node of the tree, a feature and a threshold value are selected: the criterion is to minimize an impurity function, such as the entropy [27].

Random Forest (RF) This model is an ensemble of trees that are trained on a random subset of the original dataset, and optionally by randomly selecting a single portion of features for performing splits on each node of the tree. Final decision can be taken by voting, or like in the case of scikit-learn, by averaging [28, 25].

k-Nearest Neighbor (k-NN) In this algorithm, no internal model is built, instead, classification is performed through majority vote by querying the nearest samples in the dataset. For this reason, a distance metric shall be selected; the Euclidean one is often used. Furthermore, different internal data structures are defined, in order to efficiently compute the nearest neighbor [29].

Support Vector Machine (SVM) This model builds a hyperplane that maximizes the distance, for each class, from the nearest training data samples for each class. These small subsets of nearest samples are the support vectors. Data can be computed in a higher dimensional space, by applying the so-called kernel trick, allowing the algorithm to deal with non-linear problems [30, 31].
The Linear SVM is a special case of SVM, using a linear kernel function mostly for computational efficiency when the dataset can be linearly separable.

E. System Architecture

The proposed system architecture is designed as a pipeline composed of four stages: data acquisition, cleaning, scaling and classification. In the first stage data are collected by the sensors; optionally, data aggregation is performed when the acquisition frequency exceeds one sample per hour. Additionally, the time features (day of the year and hour) are included as integer values in the feature vector. Data cleaning ensures features integrity. Depending on the characteristics of each feature, threshold-based or statistical methods are applied. Furthermore, since no temporal dependency is assumed between samples, an error on a single observation does not propagate in the future, since it affects only the current sample. Before reaching the two classifiers, one for the top side and one for the bottom side of the leaf, data is scaled using the Min-Max method with parameters computed from the training set. In the final stage, the feature vector is fed into the proposed models and the probability for the two labels is obtained. A decision rule is then applied to convert probability into a binary state.

IV. Results

The scores computed using the test set are accuracy, balanced accuracy, precision, recall and F1-score [32]. Metrics that describe a single label are individually reported for both dry and wet classes. Instead, for the cross-validation computed during the grid search, only the accuracy is reported for simplicity. For further information, tables with grid search results are reported in Section A.

A. Cross-Validation and Hyperparameters

Grid search results are useful for selecting the optimal hyperparameters for each model. In addition, cross-validation ensures that the observed outcomes are not the result of a specific hyperparameter combination or an arbitrary split of the training and test sets. In general, as it is shown in Appendix A, all models obtain a relatively low standard deviation across different folds and hyperparameters selection, suggesting a good stability.

Results for DT are reported in Table A1. In this case, two impurity functions have been evaluated with different values of minimum samples split. This parameter controls the tree depth and avoids overfitting by limiting the minimum node size required to perform a split in the tree. Across the parameter combinations scores are similar, with low standard deviations, suggesting that the cross-validation results are not accidental. The same findings apply to DT trained using the wetness of the bottom side of the leaf.

For the RF, both the number of used estimators and the minimum samples split parameter for the trees have been investigated. Results in Table A2 show that increasing the number of trees does not impact performance, thus this number has been kept small. Moreover, since a smaller value of minimum samples split provides the best accuracy, it has been set to 20.

Regarding k-NN, cross-validation has been performed on different values of k, which defines the number of neighbors queried when voting to classify a new sample. Results are reported in Table A3, showing that no particular benefit is obtained by increasing k; consequently, it has been set to 3. This might happen for several reasons: data could be already well separated, or sparse features could make the neighborhood less informative.

For both SVM and Linear SVM, a grid search has been performed to inspect different values of the regularization parameter C, which determines the trade-off between margin size and misclassification penalty. Additionally, in SVM the Radial Basis Function (RBF) is used as kernel, thus gamma parameter is evaluated too. Since this parameter influences the region of similarity, it must be tuned in order to avoid overfitting. Results for Linear SVM are reported in Table A4: increasing C initially improves the scores, but after a certain point further reduction of the margin has little effect. On the other hand, SVM results in Table A5 prove that the RBF kernel improves the model, yielding scores comparable to other nonlinear models, with a more consistent increase in accuracy as C is raised. Two values for gamma have been tested, 0.1 and 1, obtaining the best results for the second.

B. Test Dataset Evaluation

Metrics computed on the test dataset are reported separately in Tables III and IV, corresponding to the top and bottom sides, respectively. Linear models such as Linear SVM and LDA achieve sufficient accuracy scores; however, they exhibit low precision for the wet class, in contrast to the higher values for the dry class. Conversely, recall is higher for the wet label and lower for the dry one. Non-linear models, on the other hand, have higher scores, with RF performing slightly better than the others. They also exhibit a smaller difference in precision between the two labels and a more stable recall. In general, higher precision for the dry label than for wet label, means that models have a slightly tendency to overestimate wet samples compared to dry ones. However, the generally good recall for non-linear models, prove that the vast majority of actual wetness events are successfully detected. A possible explanation is that models are tricked by transitional periods of wetness, during which, for example, high relative humidity could be measured just before wetness is actually detected by the sensor. However, since leaf wetness is used for disease forecasting, overestimating wetness is safer, as underestimation could result in a missed detection of an infection period.

The difference in performance between linear and non-linear models is more pronounced for the bottom leaf surface than for the top. This suggests that the relationship between agrometeorological features and wetness on the abaxial surface is characterized by higher non-linearity, making the linear decision boundary less effective. In Fig. 5, the decision boundary in the temperature and relative humidity space of the LDA classifier is depicted. Relative humidity discriminates leaf wetness better than temperature; however, the decision boundary for the bottom side has a steeper slope compared to the top one. This result suggests that in this case, relative humidity alone is less efficient in prediction; therefore, other features may contribute more significantly to reducing prediction error.

The DT model allows visual inspection of the internal decision rules. In Figs. 6a and 6b, the first three levels of the trees are shown for the top and bottom sides, respectively.

Table III: Models scores computed over the test set for the top side, using a leaf wetness threshold rule of 20 %.

Model	Accuracy	Balanced Accuracy	Precision (Wet)	Precision (Dry)	Recall (Wet)	Recall (Dry)	F1 (Wet)	F1 (Dry)
LDA	0.7932	0.8139	0.6531	0.9218	0.8845	0.7434	0.7514	0.8230
DT	0.9144	0.9136	0.8563	0.9494	0.9106	0.9165	0.8826	0.9327
RF	0.9264	0.9290	0.8652	0.9644	0.9378	0.9202	0.9001	0.9418
k-NN	0.9176	0.9215	0.8474	0.9624	0.9350	0.9080	0.8891	0.9344
Linear SVM	0.8066	0.8191	0.6780	0.9113	0.8617	0.7765	0.7589	0.8385
SVM	0.9055	0.9067	0.8362	0.9488	0.9109	0.9025	0.8719	0.9251

Table IV: Models scores computed over the test set for the bottom side, using a leaf wetness threshold rule of 20 %.

Model	Accuracy	Balanced Accuracy	Precision (Wet)	Precision (Dry)	Recall (Wet)	Recall (Dry)	F1 (Wet)	F1 (Dry)
LDA	0.7182	0.7445	0.5333	0.8886	0.8151	0.6740	0.6447	0.7665
DT	0.9346	0.9340	0.8687	0.9680	0.9323	0.9356	0.8994	0.9515
RF	0.9458	0.9471	0.8853	0.9765	0.9504	0.9437	0.9167	0.9599
k-NN	0.9229	0.9283	0.8332	0.9722	0.9428	0.9137	0.8846	0.9421
Linear SVM	0.7236	0.7455	0.5398	0.8849	0.8046	0.6865	0.6461	0.7732
SVM	0.8932	0.8931	0.7928	0.9481	0.8929	0.8933	0.8399	0.9199

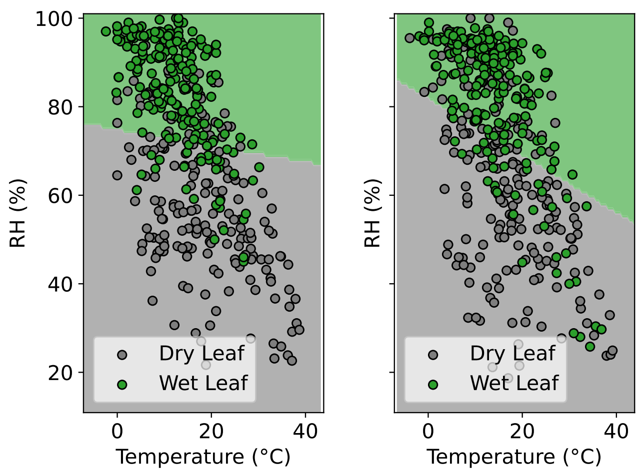


Figure 5: Decision boundary for the LDA classifier in the temperature and relative humidity domain, for the top leaf side (left), and bottom leaf side (right) using a leaf wetness threshold rule of 20 %

.

Orange nodes are mainly composed of dry samples, while blue nodes mainly contain wet samples. In both cases, the initial split is based on relative humidity. For the top side, the following features are equally exploited: relative humidity, soil volumetric water content, soil temperature, wind gust, and day of year. For the bottom side, the soil volumetric water content is exploited alone at the second level, with other features employed at subsequent levels. This aligns with the observation that the bottom side of the leaf is more influenced by soil moisture due to dew distillation [33].

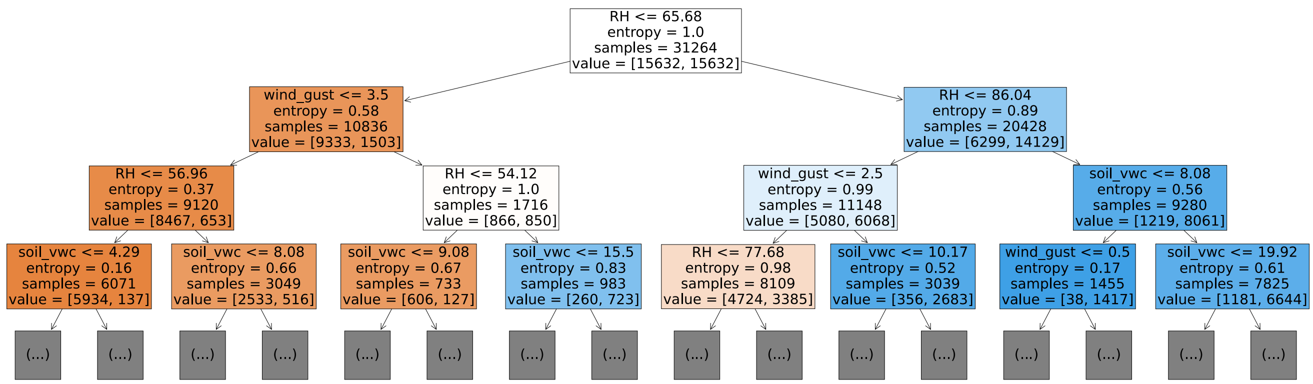
Finally, in order to investigate the effect of different leaf wetness thresholds, the dataset has been altered by computing the binary labels using a 30 % and 40 % thresholds. Table V shows the scores obtained on the test set by the models; in this case, to simplify the comparison, only the balanced accuracy is reported. Increasing the wetness threshold causes some models to lose in accuracy, but the observed drop can be considered acceptable, since the negative impact is moderate and it affects mostly the linear models. This finding suggests that, if a higher wetness threshold is required for any agronomic reason, the proposed models can still be usable.

C. Soil Features Impact

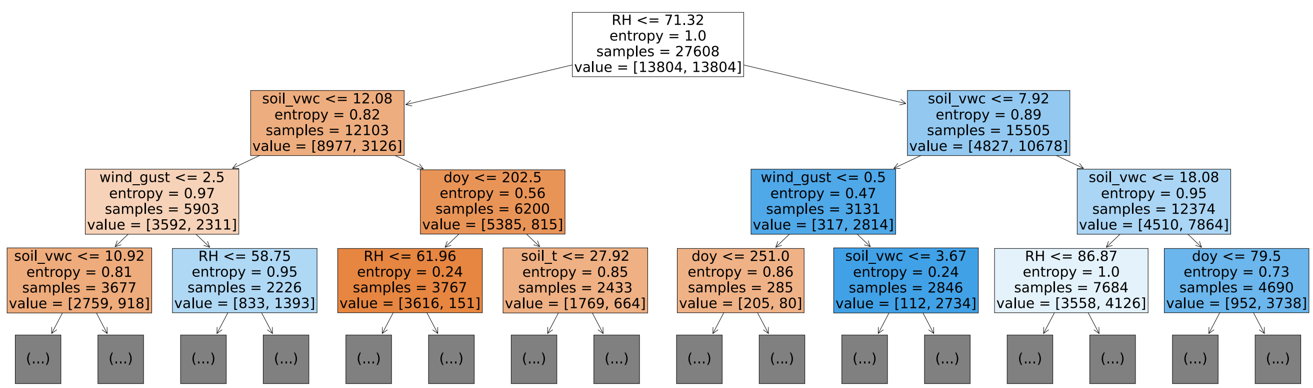
Since typical agrometeorological weather stations do not include soil sensors, the performance of the algorithms was evaluated with and without soil features. Results are reported in Table VI, where, for each model, the balanced accuracy is listed for the following cases: all features, all features but the soil humidity, all features but the soil temperature, all features without any soil measurement. It can be observed that linear models do not benefit from soil features, whereas non-linear models improve by up to 6 % and 13 % for the top and bottom sides, respectively. It can be concluded that for non-linear models, the use of soil features enhances accuracy.

D. Deployment on Edge

All investigated models can be easily deployed in a cloud environment; this is the most common option, since it allows for a simple integration with server architectures. Nonetheless, thanks to the increased computational power of microcontrollers, it is also possible to implement machine learning models on smart devices. For instance, models could run directly on weather stations; to evaluate this, edge deployment tests have been conducted. In particular, the DT model has been ported to a weather station equipped with an ultra low-power STM32L072 microcontroller by STMicroelectronics. In order to do so, the model has been translated in C language using emlearn [34], a python package specifically designed for porting machine learning models to embedded devices. The obtained model has a memory footprint of 15.4 KiB, suitable to be deployed on devices designed for low-power operations. Additionally, the prediction time has been evaluated, obtaining



(a) Top side



(b) Bottom side

Figure 6: First three levels of the DTs for both leaf sides, orange nodes are composed mainly of dry samples; blue ones are composed mainly of wet samples.

Table V: Comparison of the balanced accuracies, computed over the test set, using different values of leaf wetness threshold rule.

Model	Balanced Accuracy Top Side			Balanced Accuracy Bottom Side		
	20 %	30 %	40 %	20 %	30 %	40 %
LDA	0.8139	0.8062	0.7990	0.7445	0.7354	0.7230
DT	0.9136	0.8947	0.8918	0.9340	0.9229	0.9313
RF	0.9290	0.9135	0.9118	0.9471	0.9445	0.9450
k-NN	0.9215	0.8998	0.9045	0.9283	0.9235	0.9183
Linear SVM	0.8191	0.8098	0.8050	0.7455	0.7370	0.7211
SVM	0.9067	0.8918	0.8929	0.8931	0.8911	0.8829

a mean value of 14 µs. This result demonstrates the usability of low-power devices for this specific application.

V. Conclusions

This paper presents a comparative analysis of different machine learning models to predict leaf wetness on the top and bottom sides of the leaf, using a novel dataset of in-field agrometeorological measurements. The selected algorithms obtain overall good scores, especially in the case of non-linear models. Moreover, the use of two separate classifiers, one for each side of the leaf, leads to good outcomes. The models effectively capture key differences, such as the significant role of soil-related features in predicting wetness on the bottom side of the leaf. Finally, tests on a low-power device prove that the proposed classification pipeline can be effectively deployed on edge, to perform low-latency leaf wetness prediction. In summary, all non-linear models can be adopted for this task, with RF being the optimal choice and DT being the best trade-off between scores and model complexity, especially when targeting on-edge deployment. Despite the current dataset has been collected in vineyards, since the ground truth of the proposed models is the leaf wetness sensor, the presented results can be used regardless of the crop.

Further investigation could be conducted by improving the dataset: while the adopted features can be commonly derived from agrometeorological weather stations, additional features like wind direction and shortwave solar radiation could also be included. Moreover, geographic based features (e.g., altimetry, slope, geographical orientation) and an increased number of sites for data collection, could help to better generalize the

Table VI: Comparison of the balanced accuracies, computed over the test set, having the models trained with and without soil features.

Model	Top Side Balanced Accuracy				Bottom Side Balanced Accuracy			
	All Features	No Soil vwc	No Soil Temp	No Soil	All Features	No Soil vwc	No Soil Temp	No Soil
LDA	0.8139	0.8139	0.8130	0.8124	0.7445	0.7400	0.7394	0.7395
DT	0.9136	0.8847	0.9059	0.8505	0.9340	0.8767	0.9192	0.8074
RF	0.9290	0.9106	0.9172	0.8732	0.9471	0.8942	0.9330	0.8359
k-NN	0.9215	0.9009	0.9034	0.8613	0.9283	0.8778	0.9038	0.8145
Linear SVM	0.8191	0.8191	0.8160	0.8163	0.7455	0.7411	0.7399	0.7405
SVM	0.9067	0.8865	0.8743	0.8546	0.8931	0.8436	0.8546	0.8024

Table A1: Grid search on DT on both leaf sides, cross-validation scores reported as mean (regular) and standard deviation (italic).

Criterion	Min Samples Split	Accuracy Top Side	Accuracy Bottom Side
gini	20	0.9039 *0.0045*	0.9248 *0.0043*
gini	30	0.9021 *0.0032*	0.9216 *0.0041*
gini	50	0.9003 *0.0028*	0.9165 *0.0038*
entropy	20	0.9089 *0.0048*	0.9305 *0.0015*
entropy	30	0.9066 *0.0078*	0.9267 *0.0030*
entropy	50	0.9017 *0.0048*	0.9211 *0.0027*

models.

VI. Data Availability

The dataset has been uploaded on the Mendeley Data platform, with license CC BY-NC 4.0. The dataset is identified by the following doi: 10.17632/sd4b8vpvyb.1.

Appendix

List of tables reporting the grid search results over the investigated models for hyperparameters tuning. Cross-validation is computed using k-fold, accuracy is reported double row, with mean value and, in italic, standard deviation.

Table A2: Grid search on RF on both leaf sides, cross-validation scores reported as mean (regular) and standard deviation (italic).

Min Samples Split	N. Trees	Accuracy Top Side	Accuracy Bottom Side
20	20	0.9275 *0.0013*	0.9445 *0.0023*
20	80	0.9294 *0.0023*	0.9473 *0.0031*
50	20	0.9163 *0.0028*	0.9313 *0.0023*
50	80	0.9194 *0.0018*	0.9326 *0.0021*
80	20	0.9090 *0.0039*	0.9203 *0.0029*
80	80	0.9098 *0.0039*	0.9230 *0.0026*

Table A3: Grid search on k-NN on both leaf sides, cross-validation scores reported as mean (regular) and standard deviation (italic).

k	Accuracy Top Side	Accuracy Bottom Side
1	0.9142 *0.0024*	0.9291 *0.0014*
3	0.9200 *0.0028*	0.9273 *0.0018*
5	0.9189 *0.0018*	0.9206 *0.0028*
7	0.9174 *0.0029*	0.9175 *0.0037*

References

[1] Laurent Huber and TJ Gillespie. “Modeling leaf wetness in relation to plant disease epidemiology”. In: *Annual review of phytopathology* 30.1 (1992), pp. 553–577.

[2] CA Brewer and WK Smith. “Patterns of leaf surface wetness for montane and subalpine plants”. In: *Plant, Cell & Environment* 20.1 (1997), pp. 1–11.

[3] Gemma Hornero et al. “A novel low-cost smart leaf wetness sensor”. In: *Computers and electronics in agriculture* 143 (2017), pp. 286–292.

[4] M. L. Gleason et al. “Development and validation of an empirical model to estimate the duration of dew periods.” English. In: *Plant Disease* 78 (10 1994), pp. 1011–1016. ISSN: 0191-2917. DOI: 10.1094/PD-78-1011.

[5] TJ Gillespie, B Srivastava, and RE Pitblado. “Using operational weather data to schedule fungicide sprays on tomatoes in southern Ontario, Canada”. In: *Journal of Applied Meteorology and Climatology* 32.3 (1993), pp. 567–573.

Table A4: Grid search on Linear SVM on both leaf sides, cross-validation scores reported as mean (regular) and standard deviation (italic)

C	Accuracy Top Side	Accuracy Bottom Side
0.001	0.7963 *0.0047*	0.7442 *0.0039*
0.01	0.8163 *0.0035*	0.7584 *0.0039*
0.1	0.8194 *0.0031*	0.7609 *0.0047*
1	0.8203 *0.0038*	0.7616 *0.0047*
10	0.8202 *0.0038*	0.7615 *0.0048*

Table A5: Grid search on SVM on both leaf sides, cross-validation scores reported as mean (regular) and standard deviation (italic)

C	Gamma	Accuracy Top Side	Accuracy Bottom Side
10	0.1	0.8485 *0.0035*	0.7923 *0.0043*
100	0.1	0.8670 *0.0030*	0.8173 *0.0040*
1000	0.1	0.8809 *0.0037*	0.8475 *0.0043*
10	1	0.8911 *0.0028*	0.8655 *0.0025*
100	1	0.8994 *0.0021*	0.8789 *0.0033*
1000	1	0.9079 *0.0022*	0.8922 *0.0046*

[6] Jon Detka et al. "Machine learning vs. empirical models: Estimating leaf wetness patterns in a wildland landscape for plant disease management". In: *Agricultural and Forest Meteorology* 362 (2025), p. 110392.

[7] Giovanni Paolo Colucci et al. *Leaf Wetness and Agrometeorological Dataset*. 2025.

[8] LJ Francl and S Panigrahi. "Artificial neural network models of wheat leaf wetness". In: *Agricultural and forest meteorology* 88.1-4 (1997), pp. 57–65.

[9] Ronald Gelaro et al. "The modern-era retrospective analysis for research and applications, version 2 (MERRA-2)". In: *Journal of climate* 30.14 (2017), pp. 5419–5454.

[10] Hans Hersbach et al. "The ERA5 global reanalysis". In: *Quarterly journal of the royal meteorological society* 146.730 (2020), pp. 1999–2049.

[11] Parisa Asadi and Di Tian. "Estimating leaf wetness duration with machine learning and climate reanalysis data". In: *Agricultural and Forest Meteorology* 307 (2021), p. 108548.

[12] Martín Solís and Vanessa Rojas-Herrera. "Approaches for the prediction of leaf wetness duration with machine learning". In: *Biomimetics* 6.2 (2021), p. 29.

[13] Junsang Park et al. "Leaf wetness duration models using advanced machine learning algorithms: Application to farms in Gyeonggi Province, South Korea". In: *Water* 11.9 (2019), p. 1878.

[14] Karam Alsafadi et al. "Prediction of daily leaf wetness duration using multi-step machine learning". In: *Computers and Electronics in Agriculture* 224 (2024), p. 109131.

[15] Ju-Young Shin et al. "Prediction of leaf wetness duration using geostationary satellite observations and machine learning algorithms". In: *Remote Sensing* 12.18 (2020), p. 3076.

[16] PS Rao, TJ Gillespie, and AW Schaafsma. "Estimating wetness duration on maize ears from meteorological observations". In: *Canadian journal of soil science* 78.1 (1998), pp. 149–154.

[17] Ju-Young Shin, Junsang Park, and Kyu Rang Kim. "Emulators of a physical model for estimating leaf wetness duration". In: *Agronomy* 11.2 (2021), p. 216.

[18] Milan Koumans et al. "Physics-assisted machine learning for THz time-domain spectroscopy: Sensing leaf wetness". In: *Scientific Reports* 14.1 (2024), p. 7034.

[19] Riya Saini et al. "Identifying the source of water on plant using the leaf wetness sensor and via deep learning-based ensemble method". In: *IEEE Sensors Journal* 24.5 (2024), pp. 7009–7017.

[20] Gary D Gillespie, Kevin P McDonnell, and Gregory MP O'Hare. "Can machine learning classification methods improve the prediction of leaf wetness in North-Western Europe compared to established empirical methods?" In: *Expert Systems with Applications* 182 (2021), p. 115255.

[21] Elena Filipescu, Giovanni Paolo Colucci, and Daniele Trinchero. "Design and Implementation of a Capacitive Leaf Wetness Sensor Based on Capacitance-to-Digital Conversion". In: *IEEE Transactions on AgriFood Electronics* 2.2 (2024), pp. 244–251. DOI: 10.1109/TAFE.2024.3401252.

[22] iXemLabs. *iXemWine - An open and free agrometeorological platform*. https://ixem.wine. Accessed: 2025-09-25. 2018.

[23] Paulo C Sentelhas et al. "Suitability of relative humidity as an estimator of leaf wetness duration". In: *Agricultural and forest meteorology* 148.3 (2008), pp. 392–400.

[24] Wes McKinney. "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[25] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[26] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[27] Leo Breiman et al. *Classification and regression trees*. Chapman and Hall/CRC, 2017.

[28] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[29] Thomas Cover and Peter Hart. "Nearest neighbor pattern classification". In: *IEEE transactions on information theory* 13.1 (1967), pp. 21–27.

[30] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. "A training algorithm for optimal margin classifiers". In: *Proceedings of the fifth annual workshop on Computational learning theory*. 1992, pp. 144–152.

[31] Alex M Andrew. "An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods by Nello Christianini and John Shawe-Taylor, Cambridge University Press, Cambridge, 2000, xiii+ 189 pp., ISBN 0-521-78019-5 (Hbk,£ 27.50)." In: *Robotica* 18.6 (2000), pp. 687–689.

[32] Oona Rainio, Jarmo Teuho, and Riku Klén. "Evaluation metrics and statistical tests for machine learning". In: *Scientific Reports* 14.1 (2024), p. 6086.

[33] JL Monteith. "Dew". In: *Quarterly Journal of the Royal Meteorological Society* 83.357 (1957), pp. 322–341. DOI: https://doi.org/10.1002/qj.49708335706.

[34] Jon Nordby, Mark Cooke, and Adam Horvath. *emlearn: Machine Learning inference engine for Microcontrollers and Embedded Devices*. Mar. 2019. DOI: 10.5281/zenodo.2589394. URL: https://doi.org/10.5281/zenodo.2589394.