An Empirical Multi-classifier for Coffee Rust Detection in Colombian Crops

David Camilo Corrales[1,3(✉)], Apolinar Figueroa[2], Agapito Ledezma[3], and Juan Carlos Corrales[1]

[1] Grupo de Ingeniería Telemática, Universidad del Cauca, Campus Tulcán, Popayán, Colombia
{dcorrales,jcorral}@unicauca.edu.co
[2] Grupo de Estudios Ambientales, Universidad del Cauca, Carrera 2 no. 1A-25 - Urbanización Caldas, Popayán, Colombia
apolinar@unicauca.edu.co
[3] Departamento de Ciencias de la Computación e Ingeniería, Universidad Carlos III de Madrid, Avenida de la Universidad 30, 28911, Leganés, Spain
ledezma@inf.uc3m.es

Abstract. Rust is a disease that leads to considerable losses in the worldwide coffee industry. In Colombia, the disease was first reported in 1983 in the department of Caldas. Since then, it spread rapidly through all other coffee departments in the country. Recent research efforts focus on detection of disease incidence using computer science techniques such as supervised learning algorithms. However, a number of different authors demonstrate that results are not sufficiently accurate using a single classifier. Authors in the computer field propose alternatives for this problem, making use of techniques that combine classifier results. Nevertheless, the traditional approaches have a limited performance due to dataset absence. Therefore we proposed an empirical multi-classifier for coffee rust detection in Colombian crops.

Keywords: Coffee rust · Classifier · Multi-classifier · Dataset

1 Introduction

Rust is the main disease that attacks the coffee crop and it causes losses up to 30% in susceptible varieties of Arabica Coffee species in Colombia. This disease is found in most of the world's coffee-producing countries, and it was first reported in Colombia in 1983 in the department of Caldas. Since then, it spread rapidly through other coffee departments in the country [1]. Thenceforth The National Centre for Coffee Research (Cenicafé) supplied Colombian coffee farmers with the Castillo variety. This variety incorporates genetic attributes of rust resistance, which improves grain size, quality and productivity compared to the Caturra variety. However, despite of disease resistant plants availability, three-quarters of the area in Colombia planted with coffee varieties is still susceptible. Meaning that plants

O. Gervasi et al. (Eds.): ICCSA 2015, Part I, LNCS 9155, pp. 60–74, 2015.
DOI: 10.1007/978-3-319-21404-7_5

are vulnerable to rust attack, depending on environmental conditions and crop agronomy [2].

Since coffee rust has led to considerable losses in the industry worldwide, recent Brazilian supervised learning researchers have focused on detection of the incidence of the disease using simple classifiers as decision trees, support vector machines and bayesian networks [3, 4, 5, 6, 7, 8]. They made use of numerical values of the infection rates which were mapped into two categories (or classes). The first option of the binary infection rate was with value 1 for infection rates equal or greater than 5 percentage points (pp) and 0 otherwise. The second option was created, with value 1 for infection rates equal or greater than 10 pp, and 0 otherwise.

Meanwhile experts in computer science demonstrated that using a simple classifier is not accurate enough [9]. This indicates that approaches such as those mentioned above [3, 4, 5, 6, 7, 8], which address the rust incidence rate detection using simple classifiers, lack of accuracy needed for predictions. Authors in the computer field suggested as an alternative solution to make use of techniques that combine classifier results [10, 11]. Nevertheless the traditional approaches have a limited performance (Bagging, Random subspace, Rotation forest, Stacking, inter alia) due to dataset absence to construct accurate classifiers [5, 6, 12, 13, 14].

Therefore, we proposed an empirical multi-classifier for coffee rust detection in Colombian crops. The remainder of this paper is organized as follows: Section 2 describes the data collection and the algorithms used; Section 3 the algorithms used in the multi-classifier proposed; Section 4 presents results and discussion and Section 5 conclusions and future work.

2 Background

This section describes the data collection process and the generation of the dataset used in experiments, introduces algorithms which comprise the multi-classifier.

2.1 Data Collection

The data used in this work were collected [12] trimonthly for 18 plots, closest to weather station at the Technical Farm (Naranjos) of the Supracafé, in Cajibio, Cauca, Colombia (21°35'08"N, 76°32'53"W), during the last 3 years (2011-2013). The dataset includes 147 examples from the total of 162 available ones. The remaining 15 samples were discarded due to problems in the collection process.

The dataset is composed of 13 attributes that are divided in 3 categories: Weather conditions (6 attributes), Physic crop properties (3 attributes), and crop management (4 attributes). Below are describe the 13 attributes (Table 1):

Table 1. Dataset for Incidence Rate of Rust Detection

Attributes for Incidence Rate of Rust Detection	
Weather Conditions	Relative humidity average in the last 2 months (RHA2M), Hours of relative humidity > 90% in the last month (HRH1M), Temperature variation average in the last month (TVA1M), Rainy days in the last month (RD1M), Accumulated precipitation in the last 2 months (AP2M), Nightly accumulated precipitation in the last month (NAP1M).
Physic Crop Properties	Coffee Variety (CV), Crop age (CA), Percentage of shade (PS).
Crop Management	Coffee rust control in the last month (CRC1M), Coffee rust control in the last 3 months (CRC3M), Fertilization in the last 4 months (F4M), Accumulated coffee production in the last 2 months (ACP2M).

In this sense, the class was defined as, the Incidence Rate of Rust (IRR). IRR is calculated by following a unique methodology in Colombian coffee crops collection developed by Cenicafé [2] for a plot with area lower or equal of one hectare. The steps of the methodology are presented below:

1. The farmer must be standing in the middle of the first furrow and he has to choose one coffee tree and pick out the branch with greater foliage for each level (high, medium, low); the leaves of the selected branches are counted as well as the infected ones for rust.

2. The farmer must repeat the step 1 for every tree in the plot until 60 trees are selected. Take in consideration that the same number of trees must be selected in every furrow (e.g. if plot has 30 furrows, the farmer selects two coffee trees for each furrow).

3. Finished the step 1 and 2, the leaves of the coffee trees selected (LCT) are added as well as the infected leaves of rust (ILR). Later it must be computed the Incidence Rate of Rust (IRR) using the following formula:

$$IRR = \frac{ILR}{LCT}\ 100 \tag{1}$$

The collection process and IRR computation spend large amount of money and time, for this reason the IRR samples are limited (trimonthly for 18 plots). This process and its samples are considered very important, since it provides coffee crops rust approximation.

2.2 Supervised Learning Techniques

To evaluate the empirical multi-classifier for coffee rust detection in Colombian crops were determined the three following base classifiers: Backpropagation neural network (BPNN), Regression Tree (M5) and Support Vector Regression (SVR). This section provides a short description of base classifiers mentioned above and briefly reviews the four main ensemble methods, including: Bagging, Random subspace, Rotation forest and Stacking.

Base Classifiers

These classifiers learn by examples that map input vectors into one of several desired output classes. That is, a pattern classifier can be created through the training or learning process. The learning process of creating a classifier is to calculate the approximate distance between input–output examples and make correct output labels of the training set. This process is called the model generation phase. When the model is generated, it can classify an unknown instance into one of the learned classes in the training set [15]. Below are presented the base classifiers of the empirical multi-classifier for coffee rust detection

Backpropagation neural network

Backpropagation neural network (BPNN) is a feed forward neural network used to capture the relationship between the inputs and outputs [16]. The neural network is trained using backpropagation algorithm [17], where the error in the output layer is propagated backwards to adjust the weights in the hidden layers. The error in neuron p in the hidden layer is obtained using

$$\delta_p = O_p(1 - O_p)\sum_q \delta_q W_{pq}(n+1) \tag{2}$$

The error δ_p is used to adjust the weights connecting to neuron p in the hidden layer. This process is repeated for all the hidden layers. Application of all inputs once to the network and adjusting the weights is called an epoch [18]. In this work, a three layer feedforward neural network is used with a learning rate $\alpha = 0.3$ and momentum applied to the weights during updating of $\beta = 0.2$.

Regression Tree

M5 algorithm constructs a regression tree by recursively splitting the instance space. The splitting condition is used to minimize the intra-subset variability in the values down from the root through the branch to the node. The variability is measured by the standard deviation of the values that reach that node from the root through the branch [19]. The standard deviation reduction (SDR) is calculated as follows

$$SDR = sd(T) - \sum_i \frac{|T_i|}{|T|} \, x \, sd(T_i) \tag{3}$$

Where T, is the set of examples that reach the node, T_i are the sets that are resulted from splitting the node according to the chosen attribute and sd is the standard deviation. We defined the minimum proportion of the variance on all the data that needs to be present at a node of 0.001.

Support Vector Regression

Support Vector Regression (SVR) is a supervised learning algorithm based on statistical learning theory and structural risk minimization principle [20, 21]. It can be expressed as the following equation:

$$f(x) = w^t \varphi(x) + b \tag{4}$$

Where $\varphi(.)$ is a non-linear mapping which takes the input data points into a higher dimensional feature space, w is a vector in the feature space and b is a scalar threshold [22]. For our multi-classifier was used a Gaussian radial basis function, soft margin parameter $C = 5.0$ and insensitive cost function parameter $\epsilon = 0.01$.

Ensemble Methods

An ensemble of classifiers is a collection of several classifiers whose individual decisions are combined in some way to classify the test examples [23]. In the literature, there are a number of ensemble methods, e.g. Bagging, Random subspace, Rotation forest, Stacking, Cascading, Boosting, etc. Next are presented the four main ensemble methods.

Bagging

In its standard form, the bagging (Bootstrap Aggregating) algorithm [24] generates M bootstrap samples $T_1, T_2, \ldots, T_M$ randomly drawn (with replacement) from the original training set T of size n. From each bootstrap sample T_i (also of size n), a base classifier C_i is induced by the same learning algorithm. Predictions on new observations are made by taking the majority vote of the ensemble C^* built from $C_1, C_2, \ldots, C_M$. As bagging resamples the training set with replacement, some instances may be represented multiple times while others may be left out. Since each ensemble member is not exposed to the same set of instances, they are different from each other. By voting the predictions of each of these classifiers, bagging seeks to reduce the error due to variance of the base classifier [25].

Random subspace

The random subspace method (RSM) is an ensemble construction technique, in which the base classifiers $C_1, C_2, \ldots, C_M$ are trained on data sets $T_1, T_2, \ldots, T_M$ constructed with a given proportion of attributes picked randomly from the original set of features F. The outputs of the models are then combined, usually by a simple majority voting scheme. The author of this method suggested to select about 50% of the original features. This method may benefit from using random subspaces for both constructing

and aggregating the classifiers. When the data set has many redundant attributes, one may obtain better classifiers in random subspaces than in the original feature space. The combined decision of such classifiers may be superior to a single classifier constructed on the original training data set in the complete feature space. On the other hand, when the number of training cases is relatively small compared with the data dimensionality, by constructing classifiers in random subspaces one may solve the small sample size problem [25, 26].

Rotation forest
Rotation forest [27] refers to a technique to generate an ensemble of classifiers, in which each base classifier is trained with a different set of extracted attributes. The main heuristic is to apply feature extraction and to subsequently reconstruct a full attribute set for each classifier in the ensemble. To this end, the feature set F is randomly split into L subsets, principal component analysis (PCA) is run separately on each subset, and a new set of linear extracted attributes is constructed by pooling all principal components. Then the data are transformed linearly into the new feature space. Classifier C_i is trained with this data set. Different splits of the feature set will lead to different extracted features, thereby contributing to the diversity introduced by the bootstrap sampling [25].

Stacking
Stacking is an ensemble technique that uses a meta-learner for determining which classifiers are reliable and which are not. Stacking is usually employed to combine models built by different inducers. The idea is to create a meta-dataset containing a tuple for each tuple in the original dataset. However, instead of using the original input attributes, Stacking uses the classifications predicted by the base-classifiers as the input attributes. The target-attribute remains as in the original training-set. A test instance is first classified by each of the base-classifiers. These classifications are fed into a meta-level training-set to produce a meta-classifier. The meta-classifier that has been produced combines the different predictions into a final prediction. In order to avoid over-fitting of the meta-classifier, the instances used for training the base-classifiers should not be used to train the meta-classifier. Thus the original dataset should be partitioned into two subsets. The first subset is reserved to form the meta-dataset while the second subset is used to build the base-level classifiers. Consequently, the meta-classifier predictions reflect the true performance of base-level learning algorithms [28].

3 Empirical Multi-classifier for Coffee Rust Detection

The empirical multi-classifier for coffee rust detection is based on Cascade Generalization method, where are used sequentially a set of classifiers, at each step performing a modification of the original dataset [29]. In this manner, the main idea is focused on the use of multiple classifiers in such a way that each of the classifiers (BPNN, M5 and SVR) covers a different part of the dataset. All of this with the objective to integrate the classification results and produce the final classification (Fig. 1). In addition, we used the interquartile range and k-mean algorithms to improve the performance in the dataset.

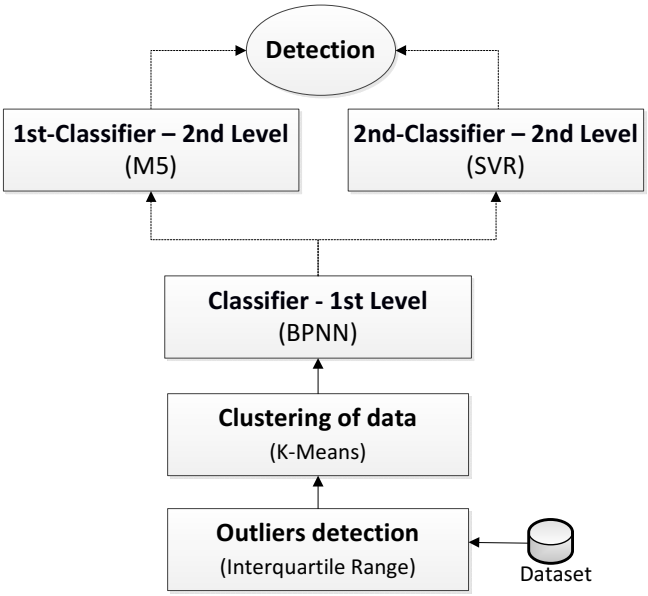

Fig. 1. Workflow for coffee rust detection

The *outliers detection module*, detects and removes the values that have an abnormal behavior of an attribute into the dataset through Interquartile Range Method (IR). This method compute the quartile Q_1 , Q_2 , Q_3 , which split a sort dataset in four parts [30], after, computing the interquartile range (IQR), which is the difference among the Third quartile (Q_3) and First quartile (Q_1). The IQR is a measure of noise for the data set. Points that are beyond the quartiles by half IQR's will be deemed potential outliers [31]. Below is presented the mathematic representation:

$$x < q_1 - 1.5\,x\,IQR \;\vee\; x > q_3 + 1.5\,x\,IQR;\; \forall\, x \in \mathbb{R} \tag{5}$$

Where, x is the observation to evaluate and $q_1 - 1.5\,\mathrm{x}\,\mathrm{IQR}$ denote the lower inner fences and $q_3 + 1.5\,\mathrm{x}\,\mathrm{IQR}$ the upper inner fences. Hence points beyond these fences are potential outliers. In this case the Interquartile Range Method was applied to the class - Incidence Rate of Rust (IRR). The outcomes obtained to apply IR, presented as lower inner fence -2.0156 and upper inner fence 11.84, thus, the last 9 observations (139 - 147) are removed.

On the other hand, the *clustering of data module*, builds clusters from data set (leaving out the IRR class) with K-means algorithm (k=3). The k-means algorithm partitions a set of data into a number k of disjoint clusters by looking for inherent patterns in the set. Let us suppose that X represents the available set of samples. Each sample can be represented by an m-dimensional vector in the Euclidean space R^m. Thus, in the following, $X = \{x_1, x_2, \ldots, x_3\}$ will represent a set of n samples, where the generic sample xi is a m-dimensional vector [32]. Each cluster is a subset of X and contains samples with some similarity. The distance between two samples provides a measure of similarity: it shows how similar or how different two samples are.

In the k-means approach, the representative of a cluster is defined as the mean of all the samples contained in the cluster [32].

Once the k-means is used each cluster is transformed in a data training set. The basic idea is to know the meaning of the clusters and define an expert classifier for each cluster. Next is explained the interpretation of the clusters through Bayesian Network and Decision Tree.

We used a Bayesian network [33] to build a conditional probability distribution for clusters generated by k-means (k=3) and four main attributes: coffee rust control in the last month (CRC1M), coffee rust control in the last 3 months (CRC3M), fertilization in the last 4 months (F4M) and incidence rate of rust (IRR), as can be seen in Table 2.

Table 2. Percentage of conditional probability distribution for clusters generated by k-means

Cluster	Coffee rust control in the last month		Coffee rust control in the last 3 months		Fertilization in the last 4 months		Incidence Rate of Rust	
	Yes	Not	Yes	Not	Yes	Not	<7.18 %	≥7.18%
C_1	45.3%	54,7%	98.3%	1.7%	32.8%	67.2%	75%	25%
C_2	1%	99%	27.9%	72.1%	20.2%	79.8%	66.66 %	33.33 %
C_3	88.6%	11.4%	58.8%	41.2%	99.1%	0.9%	98.70%	1.30%

For cluster C_1, CRC1M and CRC3M have opposite probability distribution (54.7% indicates that it was not done a coffee rust control in the last month while 98.3 % shows that it was done a coffee rust control in the last 3 months), which means that C_1 contains contradictory instances. For another part, for C_2, the attributes of CRC1M, CRC3M and F4M present a similar behavior, indicating that it was not done rust controls and fertilizations on coffee crops (probability distribution of "Not" values: 99%, 72.1% and 79.8% respectively), whereas probability distribution of CRC1M, CRC3M and F4M attributes of C_3 indicates the use of coffee rust controls and fertilizations (probability distribution of "Yes" values: 88.6%, 58.8% and 99.1% respectively); for this reason the Incidence Rate of Rust is less than 7.18% (probability distribution of IRR < 7.18 % = 98.70%).

To test out the outcomes obtained by Bayesian Network, we used a C4.5 Decision Tree (pruning the irrelevant attributes) [34], as can be seen in Fig. 2.

In Fig. 2, the C4.5 decision tree accounted 3 attributes: Coffee rust control in the last month (CRC1M), Hours of relative humidity > 90% in the last month (HRH1M), and Crop Age (CA) in months. The distribution of instances is founded in the leaves. In this sense the rule obtained for cluster C_1 (Fig. 2) does not contain the necessary attributes to know the meaning of C_1 (CRC1M and HRH1M); unlike of C_2 where conditions promote the appearance of rust, because there was not done a coffee rust control in the last month, high hours of relative humidity > 90% in the last month (> 341 hours), and older crops (age > 50 months). Finally C_3 can be interpreted as the youngest crops (age < 48 months) that are resistant to rust without regard for conditions of relative humidity and coffee rust controls performed.

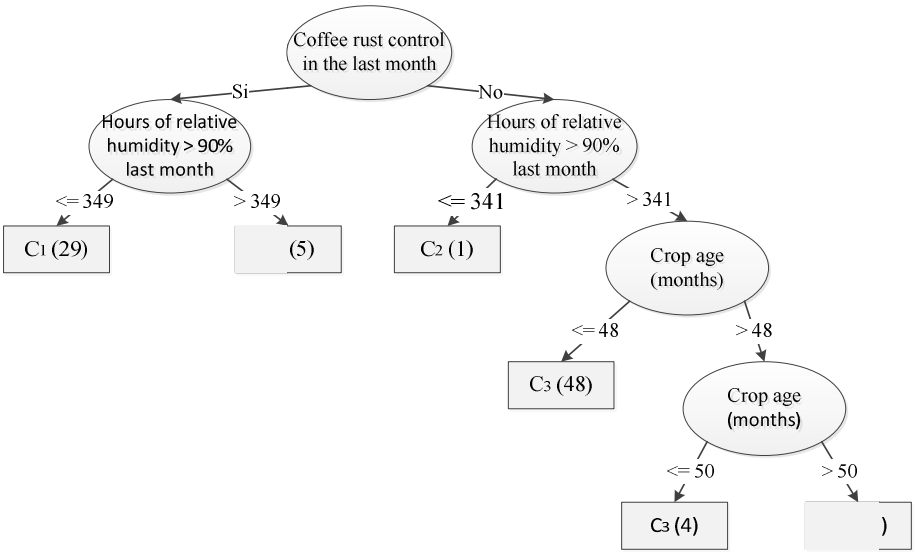

Fig. 2. C4.5 decision tree for clusters generated by k-means

Based on the foregoing, we interpret the C_2 cluster as the cases where conditions induce high losses in crops caused by rust (IRR ≥7.18%), while C_3 cluster presents the cases with low risk of losses in crops (IRR < 7.18%).

To choose the k suitable, the k-means algorithm was tested with$k = 2, 3, 4$. The clusters are displayed with CLUSPLOT graphical library, which creates a bivariate plot visualizing a partition (clustering) of the data [24]. All observation (instances) are represented by points in the plot, using Principal Components Analysis (relative to the first two principal components) [27]. The clusters $\{C_1, C_2, \ldots, C_n\}$; $k = n$, are again represented as ellipses, which are based on the average and the covariance matrix of each cluster; and their size is such that they contain all the points of their cluster (Fig. 3).

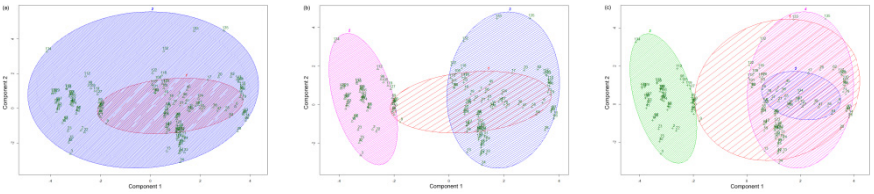

Fig. 3. k-means algorithm with k = 2, 3, 4 displayed with CLUSPLOT

The Fig. 3a presents the k-means algorithm with $k = 2$, where the points of C_1 (red color) has misclassified in correspondence to C_2 (blue color), in this manner we cannot differentiate the C_1 from the C_2. On the other hand, $k = 3$ (Fig. 3b), C_2 (pink color) and C_3 (blue color) are completely distinct. However, most of the points of C_1 (red color) was misclassified as C_3. Lastly, $k = 4$ (Fig. 3c), C_1 (red color), C_2 (blue color) and C_4 (pink color) are overlapped, while the C_3 (green color) differ of the other clusters.

In this sense, we define the k-means algorithm with $k = 3$ (Fig. 3b), because its clusters C_2 (pink color) and C_3(blue color) are completely distinct. C_1 (red color) has misclassified points due to its belonging to C_2 and C_3, besides the contradictory instances found by Bayesian Network and C.4 Decision Tree. For this reasons, the observations of C_1 were deleted (29 instances) to avoid the noise. The new dataset has 109 instances, 52 of C_2 and 57 of C_3.

When the clusters are defined we create the three classifiers (In Fig. 1 *classifier - 1st level,* and *classifiers - 2nd level*):

The *classifier - 1st level* use a Backpropagation neural network (BPNN) responsible for deciding which classifier of second level will detect the incidence rate of rust (IRR). The *1st-classifier-2nd level* trains a regression tree (M5) with instances of C_2 to detect the incidence rate of rust greater than 7.18%; whereas the *2nd-classifier-2nd level* trains a support vector regression (SVR) with instances of C_3 with aim to detect the incidence rate of rust less than 7.18%.

4 Experimental Results

This section reports a number of experiments carried out to select the base classifiers used to detect the coffee rust incidence rate. Here we compare the results obtained by the empirical multi-classifier against classical approaches as: simple classifiers and ensemble methods.

4.1 Selection of Base Classifiers

The families of supervised learning algorithms assessed were: Support Vector Machines, Neural Networks, Bayesian Networks, Decision Trees, and K nearest neighbors. The selection criteria of these classifiers are based on previous surveys which show that are the most suitable for classification and predictions tasks [35, 36], especially in the detection of crops diseases and pest [37]. With the dataset introduced in Section 2.1, we used a 10-fold cross validation to estimate the scores reported in the following figure and tables.

Classifier - 1st Level

We tested the most relevant algorithms of supervised learning for classification tasks as Support Vector Machine (SVM), Backpropagation Neural Network (BPNN), Naive Bayes (NB), C4.5 Decision Tree, and K nearest neighbors (K-NN) [37] to choose the *classifier - 1st level,* computing precision, recall and F- measure as seen in Table 3.

Table 3. Precision, Recall and F-measure for SVM, ANN, BN, DT and K-NN

	Supervised learning algorithms				
Measures	**SVM**	**BPNN**	**NB**	**C4.5**	**K-NN**
Precision	99.3%	99.3%	88.6%	96.3%	97.9%
Recall	99.3%	99.3%	87.4%	96.3%	97.8%
F- measure	99.3%	99.3%	87.5%	96.3%	97.8%

The supervised learning algorithms in Table 3 present values greater than 88.6% for Precision, 87.4% for Recall and 87.5% for F-measure, which indicates low false positives and good outcomes to classify new instances. However, these measures are insufficient to choose the *classifier - 1st level* inasmuch as SVM and BPNN have the same values (Precision, Recall and F-measure 99.3%). For this reason we used the ROC Curve for evaluating the SVM and BPNN such as shows in Fig. 4.

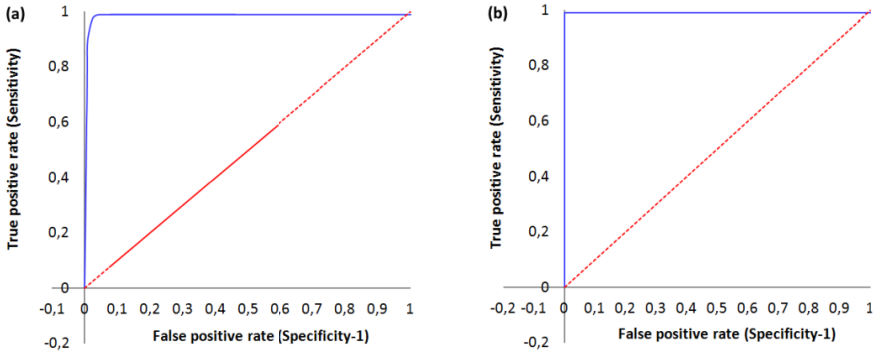

Fig. 4. Roc curves for (a) SVM (b) BPNN

In this regard, the curves showed a good performance of SVM (Fig. 4a) and BPNN (Fig. 4b) to classify correctly new instances, with an area under a curve of 99.1% for SVM and 100% of BPNN. Based on the above, the Backpropagation neural network was selected as *classifier - 1st level.*

Classifiers-2nd Level

We tested the main learning algorithms for prediction tasks as Support Vector Regression (SVR), Multilayer Perceptron (MP), Radial Basis Function Network (RBFN), K Nearest Neighbors Regression (K-NN R) and Regression Tree (M5) [37] to choose the *Classifiers - 2nd level* through Pearson's Correlation Coefficient (PCC), Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as seen in Table 4 and Table 5.

Table 4. Comparison: SVR, MP, RBF, K-NN R, M5 for selection of *1st-classifier-2nd* level

Measures	Supervised learning algorithms				
	SVR	**MP**	**RBF**	**K-NN R**	**M5**
PCC	0.43	0.39	-0.19	0.32	0.51
MAE	3.17%	3.62%	3.50%	3.22%	1.83%
RMSE	3.81%	4.83%	4.36%	4.04%	2.16%

In Table 4, the M5 regression tree is approaching a directly proportional relation (PCC = 0.51) among Incidence Rate of Rust Detected (IRRP) and Incidence Rate of Rust Real (IRRR). In that manner M5 presents the least difference among IRRP and IRRR with MAE = 1.83% and RMSE = 2.16%. In accordance with the above, the M5 regression tree was selected as *1st-classifier-2nd level.*

Table 5. Comparison: SVR, MP, RBF, K-NN R, M5 for selection of 2nd-classifier-2nd level

Measures	Supervised learning algorithms				
	SVR	**MP**	**RBF**	**K-NN R**	**M5**
PCC	0.36	0.32	0.27	0.37	0.08
MAE	1.20%	1.22%	1.24%	1.31%	1.49%
RMSE	1.73%	1.91%	1.74%	1.86%	2.08%

In Table 5, the K-NN R and SVR algorithms have the highest value for positive correlations (PCC= 0.37 and PCC = 0.36 respectively), however, SVR present the least difference among IRRP and IRRR (MAE = 1.20% and RMSE = 1.73%) respect to K- NN R (MAE = 1.31% and RMSE = 1.86%). Based on the above, SVR was selected as *2nd-classifier-2nd level.*

4.2 Evaluation of the Empirical Multi-classifier for Coffee Rust Detection

This section presents the results obtained by the empirical multi-classifier against simple classifiers and classical ensemble methods.

Empirical Multi-classifier vs. Simple Classifiers

Table 6 compares the outcomes obtained by the empirical multi-classifier against simple classifiers as Support Vector Regression (SVR), Back Propagation Neural Network (BPNN), and Regression Tree (M5) which were tested in [12] with the same dataset.

Table 6. Comparison of empirical multi-classifier and simple classifiers

Measures	Supervised learning algorithms				
	Empirical multi-classifier		Simple classifiers		
	1st-classifier-2nd level	***2nd-classifier-2nd level***	**SVR**	**BPNN**	**M5**
PCC	0.51	0.36	0.29	0.35	0.22
MAE	1.83%	1.20%	2.28%	2.34%	2.55%
RMSE	2.16%	1.73%	3.38%	3.31%	3.50%

The outcomes obtained by empirical multi-classifier are better than simple classifiers; especially for *2nd-classifier-2nd level* where the instances are closer to each other respect to instances of *1st-classifier-2nd level* (Fig. 3 for k = 3).

Empirical Multi-classifier vs. Classical Ensemble Methods

Table 7 compares the outcomes obtained by the empirical multi-classifier against classical ensemble methods as Bagging, Random subspaces, Rotation forest and Stacking. The classical ensemble methods as Bagging used M5 as base classifier, Random subspaces: K- NN R, Rotation forest: M5, Stacking three base classifiers: BPNN, K- NN R, M5 and SVR as meta-learner. We choose the four ensemble methods as the best outcomes to use the dataset explained in section 2.1.

Table 7. Comparison of empirical multi-classifier and classical ensemble methods

Measures	Supervised learning algorithms					
	Empirical multi-classifier		Classical ensemble methods			
	1st-classifier-2nd level	*2nd-classifier-2nd level*	Bagging	Ran. Sub-spaces	Rot. Forest	Stacking
PCC	0.51	0.36	0.27	0.25	0.24	0.14
MAE	1.83%	1.20%	2.38%	2.38%	2.43%	2.41%
RMSE	2.16%	1.73%	3.34%	3.52%	3.37%	3.43%

The outcomes obtained by empirical multi-classifier are better than classical ensemble methods. Bagging is the ensemble method with better results; nevertheless, simple classifiers as BPNN (PCC = 0.35; RMSE = 3.31%) and SVR (MAE = 2.28%) outperformed the results obtained by Bagging (PCC = 0.27; MAE = 2.38%; RMSE = 3.34%).

5 Conclusions and Future Work

This paper presented an empirical multi-classifier for coffee rust detection in Colombian crops. Our multi-classifier proposal outperformed the classical approaches as simple classifiers and ensemble methods in terms of PCC (0.51 of *1st-classifier-2nd level* and 0.36 of *2nd-classifier-2nd level* respect to 0.35 of BPNN and 0.27 of Bagging), MAE (1.83% of *1st-classifier-2nd level* and 1.20% of *2nd-classifier-2nd level* respect to 2.28% of SVR and 2.38% of Bagging) and RMSE (2.16% of *1st-classifier-2nd level* and 1.73% of *2nd-classifier-2nd level* respect to 3.31% of BPNN and 3.34% of Bagging) which use the same dataset of coffee rust. The limitation encountered during this study was the absence of data from actual coffee crop. Especially in rust incidence rate samples due to the expensive collection process that requires big efforts in money and time. Accordingly, the results obtained on this study are not very precise.

In future studies we intend to tackle the insufficient data using different approaches such as synthetic data and incremental learning. This will allow an existing classifier be updated using only new individual data instances, without having to re-process past instances [38]. Also we will propose the use of weather time series data which are automatically capture each five minutes by weather station. We will analyze its behavior with the rust infection rate.

Acknowledgments. The authors are grateful for the technical support of the Telematics Engineering Group (GIT), Environmental Study Group (GEA) of the University of Cauca, Control Learning Systems Optimization Group (CAOS) of the Carlos III University of Madrid, Supracafé, AgroCloud project of the RICCLISA Program and Colciencias for PhD scholarship granted to MsC. David Camilo Corrales.

References

1. Cristancho, M., Rozo, Y., Escobar, C., Rivillas, C., Gaitán, Á.: Razas de roya (2012)
2. Rivillas-Osorio, C., Serna-Giraldo, C., Cristancho-Ardila, M., Gaitán-Bustamante, A.: La roya del cafeto en Colombia, impacto, manejo y costos de control. Cenicafé (2011)
3. Luaces, O., Rodrigues, L.H.A., Alves Meira, C.A., Quevedo, J.R., Bahamonde, A.: Viability of an alarm predictor for coffee rust disease using interval regression. In: García-Pedrajas, N., Herrera, F., Fyfe, C., Benítez, J.M., Ali, M. (eds.) IEA/AIE 2010, Part II. LNCS, vol. 6097, pp. 337–346. Springer, Heidelberg (2010)
4. Luaces, O., Rodrigues, L.H.A., Alves Meira, C.A., Bahamonde, A.: Using nondeterministic learners to alert on coffee rust disease. Expert Systems with Applications **38**, 14276–14283 (2011)
5. Cintra, M.E., Meira, C.A.A., Monard, M.C., Camargo, H.A., Rodrigues, L.H.A.: The use of fuzzy decision trees for coffee rust warning in Brazilian crops. In: 2011 11th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 1347–1352 (2011)
6. Pérez-Ariza, C.B., Nicholson, A.E., Flores, M.J.: Prediction of Coffee Rust Disease Using Bayesian Networks. In: Andrés Cano, MG.-O., Nielsen, T.D. (eds.) The Sixth European Workshop on Probabilistic Graphical Models. DECSAI, University of Granada, Granada (Spain) (2012)
7. Meira, C.A.A., Rodrigues, L.H.A., Moraes, S.: Modelos de alerta para o controle da ferrugem-do-cafeeiro em lavouras com alta carga pendente. Pesquisa Agropecuária Brasileira **44**, 233–242 (2009)
8. Meira, C., Rodrigues, L., Moraes, S.: Análise da epidemia da ferrugem do cafeeiro com árvore de decisão. Tropical Plant Pathology **33**, 114–124 (2008)
9. Li, L., Zou, B., Hu, Q., Wu, X., Yu, D.: Dynamic classifier ensemble using classification confidence. Neurocomputing **99**, 581–591 (2013)
10. Ranawana, R., Palade, V.: Multi-Classifier Systems: Review and a roadmap for developers. Int. J. Hybrid Intell. Syst. **3**, 35–61 (2006)
11. Ghosh, J.: Multiclassifier systems: back to the future. In: Roli, F., Kittler, J. (eds.) MCS 2002. LNCS, vol. 2364, pp. 1–15. Springer, Heidelberg (2002)
12. Corrales, D.C., Ledezma, A., Peña, A., Hoyos, J., Figueroa, A., Corrales, J.C.: A new dataset for coffee rust detection in Colombian crops base on classifiers. Sistemas y Telemática **12**, 9–22 (2014)
13. Jain, R., Minz, S.: Ramasubramanian: Machine Learning for Forewarning Crop Diseases. Journal of the Indian Society of Agricultural Statistics **63**, 97–107 (2009)
14. Korada, N.K., Kumar, N.S.P., Deekshitulu, Y.: Implementation of Naive Bayesian Classifier and Ada-Boost Algorithm Using Maize Expert System. International Journal of Information Sciences and Techniques (IJIST) **2** (2012)
15. Mitchell, T.: Machine learning. McGraw-Hill (1997)
16. Poh, H.L.: A neural network approach for marketing strategies research and decision support, vol. Ph.D. Thesis. Stanford University (1991)
17. Haykin, S.S.: Neural networks: a comprehensive foundation. Prentice Hall (2003)
18. Suhasini, A., Palanivel, S., Ramalingam, V.: Multimodel decision support system for psychiatry problem. Expert Systems with Applications **38**, 4990–4997 (2011)
19. Bonakdar, L., Etemad-Shahidi, A.: Predicting wave run-up on rubble-mound structures using M5 model tree. Ocean Engineering **38**, 111–118 (2011)
20. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (2000)

21. Vapnik, V.N.: An overview of statistical learning theory. IEEE Transactions on Neural Networks **10**, 988–999 (1999)
22. Balasundaram, S., Gupta, D.: Training Lagrangian twin support vector regression via unconstrained convex minimization. Knowledge-Based Systems **59**, 85–96 (2014)
23. Skurichina, M., Duin, R.P.: Bagging, boosting and the random subspace method for linear classifiers. Pattern Analysis & Applications **5**, 121–135 (2002)
24. Breiman, L.: Bagging Predictors. Machine Learning **24**, 123–140 (1996)
25. Marqués, A.I., García, V., Sánchez, J.S.: Two-level classifier ensembles for credit risk assessment. Expert Systems with Applications **39**, 10916–10922 (2012)
26. Tin Kam, H.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**, 832–844 (1998)
27. Rodriguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation Forest: A New Classifier Ensemble Method. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**, 1619–1630 (2006)
28. Menahem, E., Rokach, L., Elovici, Y.: Troika – An improved stacking schema for classification tasks. Information Sciences **179**, 4097–4122 (2009)
29. Gama, J., Brazdil, P.: Cascade Generalization. Machine Learning **41**, 315–343 (2000)
30. McAlister, D.: The Law of the Geometric Mean. Proceedings of the Royal Society of London **29**, 367–376 (1879)
31. Grubbs, F.: Procedures for Detecting Outlying Observations in Samples. Technometrics **11**, 1–21 (1969)
32. Mucherino, A., Papajorgji, P., Pardalos, P.: Clustering by k-means. In: Du, D.-Z. (ed.) Data Mining in Agriculture, vol. 30, pp. 47–56. Springer, New York (2009)
33. Araujo, B.S.: Aprendizaje automático: conceptos básicos y avanzados: aspectos prácticos utilizando el software Weka. Pearson Prentice Hall, España (2006)
34. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc (1993)
35. Kotsiantis, S.B.: Supervised machine learning: a review of classification techniques. In: Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies, pp. 3–24. IOS Press (2007)
36. Bhavsar, H., Ganatra, A.: A Comparative Study of Training Algorithms for Supervised Machine Learning. International Journal of Soft Computing and Engineering (IJSCE) **2**, 74–81 (2012)
37. Corrales, D.C., Figueroa, A., Corrales, J.C.: Toward detecting crop diseases and pest by supervised learning. Revista Ingeniería y Universidad **19** (2015)
38. Schlimmer, J., Granger Jr, R.: Incremental learning from noisy data. Machine Learning **1**, 317–354 (1986)