International Journal of Agricultural and Environmental Information Systems
Volume 7 • Issue 3 • July-September 2016

Two-Level Classifier Ensembles for Coffee Rust Estimation in Colombian Crops

David Camilo Corrales, Telematic Engineering Group, University of Cauca, Popayán, Colombia and Department of Computer Science and Engineering, Carlos III University of Madrid, Madrid, Spain

Apolinar Figueroa Casas, Environmental Studies Group, University of Cauca, Popayán, Colombia

Agapito Ledezma, Department of Computer Science and Engineering, Charles III University of Madrid, Madrid, Spain

Juan Carlos Corrales, Telematic Engineering Group, University of Cauca, Popayán, Colombia

ABSTRACT

Rust is a disease that leads to considerable losses in the worldwide coffee industry. There are many contributing factors to the onset of coffee rust e.g. Crop management decisions and the prevailing weather. In Colombia the coffee production has been considerably reduced by 31% on average during the epidemic years compared with 2007. Recent research efforts focus on detection of disease incidence using simple classifiers. Authors in the computer field propose alternatives for improve the outcomes, making use of techniques that combine classifiers named ensemble methods. Therefore they proposed two-level classifier ensembles for coffee rust estimation in Colombian crops using Back Propagation Neural Networks, Regression Tree M5 and Support Vector Regression. Their ensemble approach outperformed the classical approaches as simple classifiers and ensemble methods in terms of Pearson's Correlation Coefficient, Mean Absolute Error and Root Mean Squared Error.

KEYWORDS

Classifier, Coffee, Dataset, Ensemble, Rust

1. INTRODUCTION

Coffee rust is a leaf disease caused by the fungus, Hemileia vastatrix. Coffee rust epidemics, with intensities higher than previously observed, have affected a number of countries including: Colombia, from 2008 to 2011; Central America and Mexico, in 2012 - 2013; and Peru and Ecuador in 2013. There are many contributing factors to the onset of these epidemics e.g. the state of the economy, crop management decisions and the prevailing weather, and many resulting impacts e.g. on production, on farmers and labourers income and livelihood, and on food security. Production has been considerably reduced in Colombia (by 31% on average during the epidemic years compared with 2007) and Central America (by 16% in 2013 compared with 2011–12 and by 10% in 2013–2014 compared with 2012–2013) (Avelino et al., 2015).

Since coffee rust has led to considerable losses in the industry worldwide, recent Brazilian supervised learning researchers have focused on estimation of the disease incidence, using simple classifiers as decision trees, support vector machines and bayesian networks (Cintra, Meira, Monard, Camargo, & Rodrigues, 2011; Luaces, Rodrigues, Alves Meira, & Bahamonde, 2011; Luaces et al., 2010; C. Meira, Rodrigues, & Moraes, 2008; C. A. A. Meira, Rodrigues, & Moraes, 2009; Pérez-Ariza, Nicholson, & Flores, 2012). 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

DOI: 10.4018/IJAEIS.2016070103

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 (Li, Zou, Hu, Wu, & Yu, 2013). This indicates that approaches such as those mentioned above (Cintra et al., 2011; Luaces et al., 2011; Luaces et al., 2010; C. Meira et al., 2008; C. A. A. Meira et al., 2009; Pérez-Ariza et al., 2012), 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 (Ghosh, 2002; Ranawana & Palade, 2006) named ensemble methods.

Therefore, we proposed two-level classifier ensembles for coffee rust estimation 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 two-level classifier ensembles 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 two-level classifier ensembles.

2.1. Data Collection

The data used in this work were collected (Corrales et al., 2014) trimonthly for 18 plots (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 37, in the Figure 1), 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 216 available ones. The remaining 69 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 described the 13 attributes (Table 1):

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 (Rivillas-Osorio, Serna-Giraldo, Cristancho-Ardila, & Gaitán-Bustamante, 2011) for a plot with lower or equal area to one hectare. The methodology steps are presented below:

Figure 1. Technical Farm: Los Naranjos (source: authors own elaboration)

Table 1. Dataset for Incidence Rate of Rust Estimation

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).

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, the (Rust Incidence Rate) (RIR) using the following formula, must be computed:

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

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 a rust coffee crops approximation.

2.2. Supervised Learning Techniques

This section provides a short description of base classifiers of the two-level classifier ensembles for coffee rust estimation in Colombian crops (Backpropagation neural network, Regression Tree and Support Vector Machine) and briefly reviews the four main ensemble methods, including: Bagging, Random subspace, Rotation forest and Stacking.

The 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 (Mitchell, 1997). Below are presented the base classifiers of the two-level classifier ensembles for coffee rust estimation. For ease and uniformity of paper the examples for each classifier are focused on coffee rust disease.

2.2.1. Backpropagation Neural Network

Backpropagation neural network (BPNN) is a feed forward neural network used to capture the relationship between the inputs and outputs (Poh, 1991). The neural network is trained using backpropagation algorithm (Haykin, 2003), where the error in the output layer is propagated backwards to adjust the weights in the hidden layers. In the backpropagation training algorithm the error in the output neuron q is given by

$$\delta_p = \mathrm{o}_q\left(1- o_q\right)\left(t_q - o_q\right) \tag{2}$$

Where o_q and t_q are the actual and desired outputs of neuron q in the output layer, respectively. The weight from neuron p in the hidden layer to neuron q in the output layer is adjusted using

$$W_{pq}\left(n+1\right) = W_{pq}\left(n\right) + \cdot \delta_q o_p \tag{3}$$

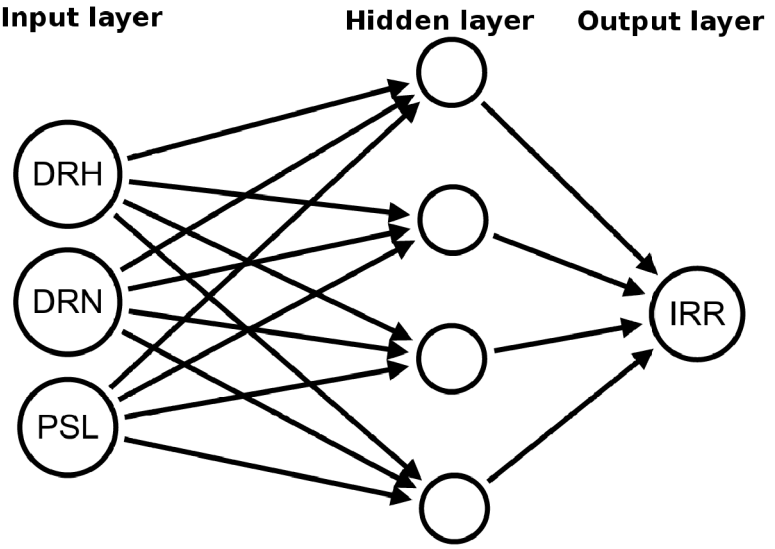
Where $\cdot$ is the learning rate coefficient, $0 < \cdot < 1$, and $W_{pq}\left(n\right)$ and $W_{pq}\left(n+1\right)$ are the weights before and after adjustment, respectively. The error in the output layer is propagated backwards to adjust the weights in the hidden layers. The error in neuron p in the hidden layer is obtained using

$$\delta_p = O_p\left(1-O_p\right)\sum_q \delta_q W_{pq}\left(n+1\right) \tag{4}$$

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. In the backpropagation training algorithm, the network weights are adjusted for certain number of epochs to map the relationship between inputs and outputs (Suhasini, Palanivel, & Ramalingam, 2011). In this work, a three layer feedforward neural network is used with a learning rate $\cdot = 0.3$.

Figure 2 shows an Artificial Neural Network (ANN) that detects the Incidence Rate of Rust (IRR) in coffee one month beforehand. This ANN is comprised of input, hidden and output layers. The input layer receives the number of consecutive days with relative humidity above 85% (DRH), number of consecutive days with rain at night (DRN), and the percentage of shade in the lot (PSL).

Figure 2. Incidence rate of rust detection through artificial neural networks (source: authors own elaboration)



The hidden layer further solves problems when the data are not linearly separable, this way different hidden layers can be used depending on the organization of the data. Finally, the output layer calculates the Incidence Rate of Rust (Corrales, Corrales, & Apolinar, 2015).

2.2.2. Decision Tree

Decision trees (DT) are algorithms that allow the approximation of functions of discrete and continuous values, beginning with the generation of a set of rules. DTs classify the values that attributes (called instances) can take in a tree, where each node represents an instance of an attribute, and each branch descending from the node corresponds to one of the possible values that the attribute can take (Mitchell, 1997). In other words, the DTs seek the instance that has the closest relationship with the target variable, dividing the dataset into subsets according to attribute. The procedure is repeated until the established criterion is satisfactorily met (Meyfroidt, Güiza, Ramon, & Bruynooghe, 2009).

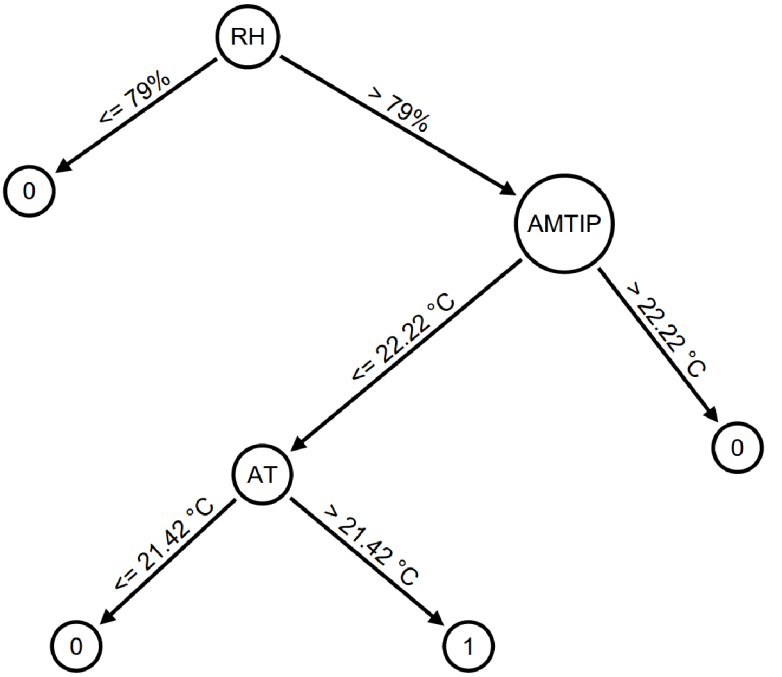
We use a variant of DT for numerical classes called Regression Tree (M5) with a minimum proportion of the variance on all the data that needs to be present at a node of 0.001.

Figure 3 shows a case study carried out in (Cintra et al., 2011) to detect the appearance of rust in coffee, making use of a decision tree and taking into account the attributes of relative humidity (RH), average maximum temperature for the rust incubation period (AMTIP), and average temperature (AT). Each attribute is represented by an intermediate node depending on its value. It will take different paths to reach a leaf node. The leaf nodes of the tree correspond to a particular category of the target variable (class), in this case 0 (when there is no rust) and 1 (when rust is detected).

2.2.3. Support Vector Machine

Support vector machine (SVM) is a method proposed by Vladimir Vapnik for solving classification and regression problems. The basic operation begins with a set of points, which in turn contains two subsets of points, in which each one belongs to one of two possible classes (Vapnik, 1995). Based on this, the support vector machine searches for the greatest distance (maximum margin) that separates the classes by way of a hyperplane, in order to build a model that is able to predict whether a new point (of unknown class) belongs to one class or the other. This method is applied as long as the set of points are linearly separable (Ayodele, 2010). Otherwise the input data must first

Figure 3. Detection of coffee rust using decision tree (source: authors own elaboration)



be transformed to a new space, usually to a higher dimension where classes are linearly separable. The process of transformation of spaces is performed implicitly by a kernel function which allows all the calculations to be performed in the input space. This process is called kernel trick (Karimi, Prasher, Patel, & Kim, 2006).

For our two-level classifier ensembles approach was used a variant of SVM for numerical classes called Support Vector Regression (SVR) (Weimin & Leping, 2008) with a Gaussian radial basis function, soft margin parameter $C = 5.0$ and insensitive cost function parameter $\epsilon = 0.01$.

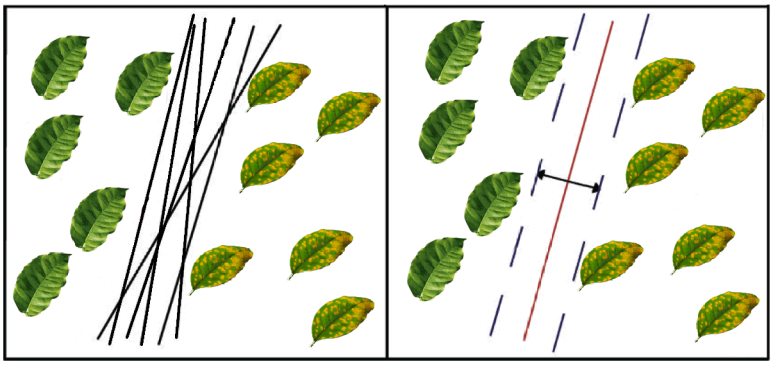
The following is an example of linearly separable binary SVM for the classification of leaves infected with rust. Figure 4 presents the possible hyperplanes that separate healthy leaves and infected leaves. In this context, SVM is searching for the hyperplane that optimally separates the two leaf types (Corrales, Corrales, et al., 2015).

On the other hand, an ensemble of classifiers is a collection of several classifiers whose individual decisions are combined in some way to classify the test examples (Skurichina & Duin, 2002). In the literature, there is a number of ensemble methods, e.g. Bagging, Random subspace, Rotation forest, Stacking, Cascading, Boosting, etc. Next are presented the four main ensemble methods.

2.2.4. Bagging

In its standard form, the bagging (Bootstrap Aggregating) algorithm (Corrales, Ledezma, & Corrales, 2015b) 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

Figure 4. (a) Possible hyperplanes (b) Optimal hyperplane using SVM (source: authors own elaboration)



other. By voting the predictions of each of these classifiers, bagging seeks to reduce the error due to variance of the base classifier (Marqués, García, & Sánchez, 2012).

2.2.5. 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. Moreover, 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 (Marqués et al., 2012; Tin Kam, 1998).

2.2.6. Rotation Forest

Rotation forest (Corrales, Ledezma, & Corrales, 2015a) 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 (Marqués et al., 2012).

2.2.7. 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 (Menahem, Rokach, & Elovici, 2009).

3. CONSTRUCTING TWO-LEVEL CLASSIFIERS ENSEMBLES

The ensemble approach for coffee rust estimation here proposed 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 (Figure 5). In addition, we used the interquartile range and k-mean algorithms to improve the performance in the dataset in data analysis module.

3.1. Data analysis

This module examines data which deviate so much from other data (Corrales, Ledezma, et al., 2015a) through outliers analysis sub-module, besides Data Analysis module (DA) defines the amount of base classifiers of the ensemble approach for coffee rust estimation supported by cluster analysis sub-module. This module is handled by an expert in data analysis such as: data scientist, a data analyst, a data miner, etc.

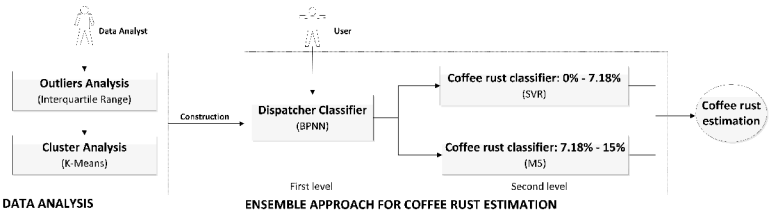
3.1.1. Outliers Analysis

The outlier's analysis sub-module detects and removes the values that have an abnormal behavior of an attribute into the dataset through Interquartile Range Method (IR). This method computes the quartile Q_1, Q_2, Q_3, which split a sort dataset in four parts (McAlister, 1879), 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 (Grubbs, 1969). 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 $\mathrm{q}_1 - 1.5\ \mathrm{x}\ \mathrm{IQR}$ denote the lower inner fences and $\mathrm{q}_3 + 1.5\ \mathrm{x}\ \mathrm{IQR}$ 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

Figure 5. Workflow of two-level classifiers ensembles for coffee rust estimation (source: authors own elaboration)



outcomes obtained to apply IR, presented as lower inner fence -2.0156 and upper inner fence 11.84, thus, are removed 9 observations with values of IRR: 12.41%, 13.01%, 13.1%, 13.5%, 13.6%,13.6%, 14.5%, 16.5%, 20.50% greater than upper inner fence 11.84% as we can see in Figure 6.

3.1.2. Cluster Analysis

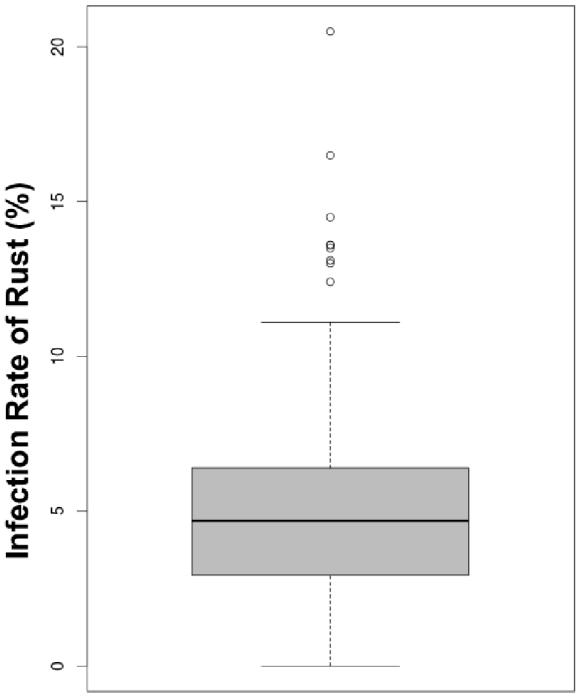
The cluster analysis sub-module, builds clusters from data set (leaving out the IRR class) with K-means algorithm $\left(k = 3\right)$. The k-means algorithm partitions a set of data into a number k of disjoint clusters by looking for inherent patterns in the set. Let us suppose that X represents the available set of samples. Each sample can be represented by an m-dimensional vector in the Euclidean space R^{m}. Thus, in the following, $\mathrm{X} = \left\{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_{\mathrm{n}}\right\}$ will represent a set of n samples, where the generic sample x_{i} is a m-dimensional vector (Mucherino, Papajorgji, & Pardalos, 2009). 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 (Mucherino et al., 2009).

3.1.3. Interpretation of Clusters

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 (Araujo, 2006) to build a conditional probability distribution for clusters generated by k-means $\left(k = 3\right)$ and four main attributes: coffee rust control in the last month

Figure 6. Box-plot of Infection Rate of Rust (source: authors own elaboration)



(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.

For cluster C_1, when it was done a coffee rust control in the last month and 3 months (45.3% - CRC1M = "Yes" and 98.3% - CRC3M = "Yes"), exists a probability distribution unequal and incoherent instances, because if the majority of instances of CRC3M shows that it was done a coffee rust control, mandatorily the instances of CRC1M must present the same behavior.

For another part, for C_2, the attributes of CRC1M, CRC3M and F4M present a similar behavior, which we can interpret: the instances in C_2 has 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) (Quinlan, 1993), as can be seen in Figure 7.

In Figure 7, 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 found in the leaves. In this sense the rule obtained for cluster C_1 (Figure 7) 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.

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%).

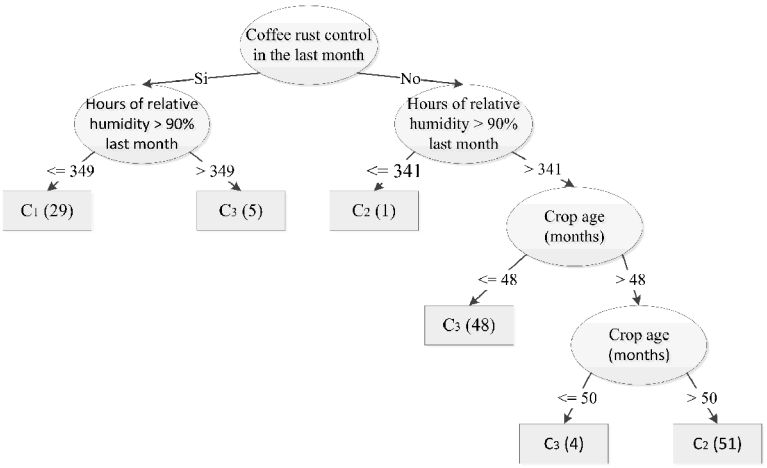
3.3.2. Building the Clusters

To choose the k suitable, the k-means algorithm was tested with $k = 2, 3, 4$ and the Euclidean distance. The clusters are displayed with CLUSPLOT graphic library, which creates a bivariated plot visualizing a partition (clustering) of the data (Corrales, Ledezma, et al., 2015b). All observation (instances) are represented by points in the plot, using Principal Components Analysis (relative to the first two principal components) (Corrales, Ledezma, et al., 2015a). The clusters $\{C_1, C_2, \ldots, C_n\}$; $k = n$, are again represented as ellipses, which are based on the average and the

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

Cluster	CRC1M		CRC3M		F4M		IRR	
	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%

Figure 7. C4.5 decision tree for clusters generated by k-means (source: authors own elaboration)

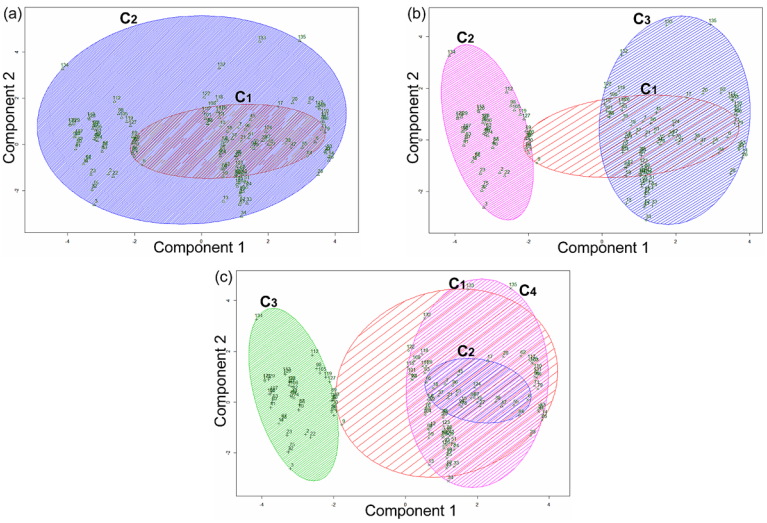


covariance matrix of each cluster; and their size is such that they contain all the points of their cluster (Figure 8).

The Figure 8a presents the k-means algorithm with $k = 2$, where the points of C_1 has misclassified in correspondence to C_2, in this way, we cannot differentiate the C_1 from the C_2. On the other hand, $k = 3$ (Figure 8b), C_2 and C_3 are completely distinct. However, most of the points of C_1 were misclassified as C_3. Lastly, $k = 4$ (Figure 8c), C_1, C_2 and C_4 are overlapped, while the C_3 differ of the other clusters.

In this sense, we define the k-means algorithm with $k = 3$ (Figure 8b), because its clusters C_2 and C_3 are completely distinct. C_1 has misclassified points due to its belonging to C_2 and C_3, besides the incoherent instances found by Bayesian Network and C.4 Decision Tree. For this reasons,

Figure 8. K-means algorithm with k = 2, 3, 4 displayed with CLUSPLOT (source: authors own elaboration)



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.

Once defined the final clusters (C_2 and C_3) we create three base classifiers for construction of the ensemble approach for coffee rust estimation which are explained below.

3.2. Ensemble Approach for Coffee Rust Estimation

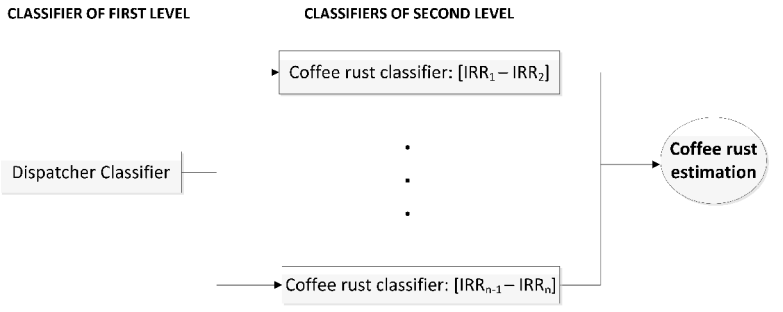
Ensemble approach for coffee rust estimation is constructed based on data analysis module with two-level classifiers ensembles: the first level contains a Backpropagation neural network (BPNN) named dispatcher classifier responsible for deciding which classifier of second level will estimate the incidence rate of rust (IRR). The second level contains expert classifiers to solve a specific task, in this case, we define two classifiers: the first classifier of second level (Figure 5 as coffee rust classifier: 0% - 7.18%) trains a support vector regression (SVR) with instances of C_3 with aim to detect the incidence rate of rust among 0% and 7.18% whereas the second classifier of second level (Figure 5 coffee rust classifier: 7.18% - 15%) trains a regression tree (M5) with instances of C_2 to classify the incidence rate of rust among 7.18% and 15%. The number of classifiers of second level change according to cluster analysis and size of dataset as we can see in Figure 9; seen in this light, we can get n classifiers of second level where each classifier detects the incidence rate of rust (IRR) among $\left[IRR_{n-1} - IRR_n\right]$. Our processed dataset contains 109 instances with IRR values among 0% and 15%.

In this way, the final user types the inputs (weather conditions, physic crop properties, and crop management) and the ensemble approach returns the estimation of infection rate of rust.

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 ensemble approach against classical approaches as: simple classifiers and ensemble methods. 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.

Figure 9. Two-level classifiers ensembles for coffee rust estimation (source: authors own elaboration



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 the suitable learners for classification and predictions tasks (Bhavsar & Ganatra, 2012; Kotsiantis, 2007), especially in the detection of crop diseases.

4.1.1. Dispatcher Classifier

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) (Corrales, Corrales, et al., 2015) to choose the dispatcher classifier, computing precision, recall and F- measure as seen in Table 3.

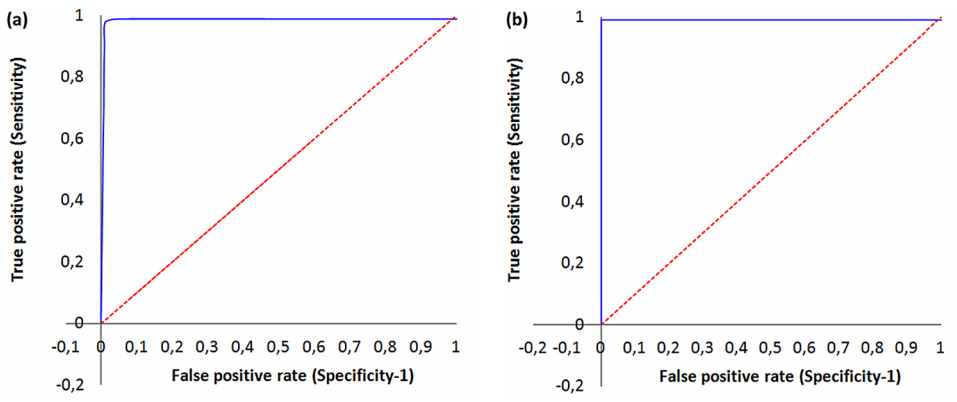
The supervised learning algorithms in Table 3 present values greater than 88.6% for Precision, 87.4% for Recall and 87.5% for F-measure, which indicates low false positives and good outcomes to classify new instances. However, these measures are insufficient to choose the dispatcher classifier considering that 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 Figure 10.

In this regard, the curves showed a good performance of SVM (Figure 10a) and BPNN (Figure 10b) 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 dispatcher classifier.

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%

Figure 10. Roc curves for (a) SVM (b) BPNN (source: authors own elaboration)



4.1.2. Classifiers of Second 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) (Corrales, Corrales, et al., 2015) to choose the Classifiers of second 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.

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 first classifier of second level (1st-classifier-2nd level).

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

4.2. Evaluation of the Ensemble Approach for Coffee Rust Estimation

This section presents the results obtained by the ensemble approach against simple classifiers and classical ensemble methods.

4.2.1. Ensemble Approach vs. Simple Classifiers

Table 6 compares the outcomes obtained by the ensemble approach against simple classifiers as Support Vector Regression (SVR), Back Propagation Neural Network (BPNN), and Regression Tree (M5) which were tested in (Corrales et al., 2014) with the same dataset.

The outcomes obtained by ensemble approach are better than simple classifiers; especially for *2nd-classifier-2nd level* where the instances are closer to each other regard to instances of *1st-classifier-2nd level* (Figure 8 for k = 3).

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

	Supervised learning algorithms				
Measures	**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%

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

	Supervised learning algorithms				
Measures	**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%

Table 6. Comparison of ensemble approach and simple classifiers

Measures	Supervised learning algorithms				
	Ensemble approach		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%

4.2.2. Ensemble Approach vs. Classical Ensemble Methods

Table 7 compares the outcomes obtained by the ensemble approach 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.

The outcomes obtained by ensemble approach 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%).

4.3. Discussion

The results presented previously show an improvement compared to simple classifiers and classical ensemble methods, nevertheless the errors rate remain above 1.20 percentage points of Infection Rate of Rust (IRR), due to dataset is very small and IRR is scattered with few values in a widely range (0% - 20%), as we can see in Table 8:

In this way the initial dataset has 147 instances, of which 9 instances were removed by outlier analysis module, 29 instances by cluster analysis, and the remaining 109 instances were used in the ensemble approach for coffee rust estimation in Colombian crops. The main drawback is the amount of instances; if the available examples are few perhaps the dataset does not represent a sample of the population.

Table 7. Comparison of ensemble approach and classical ensemble methods

Measures	Supervised learning algorithms					
	Ensemble approach		Classical ensemble methods			
	1st-classifier-2nd level	*2nd-classifier-2nd level*	Bagging	Ran. Subspaces	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%

Table 8. Number of Instances for Ranges of IRR

Instances	Range of IRR
21	0% - 2.27%
43	2.27% - 4.55%
47	4.55% - 6.83%
21	6.83% - 9.11%
6	9.11% -11.38%
6	11.38% - 13.66%
1	14.5%
1	16.5%
1	20.5%

5. CONCLUSION AND FUTURE WORK

This paper presented an ensemble approach for coffee rust estimation in Colombian crops. Our ensemble approach outperformed the classical approaches as simple classifiers and ensemble methods in terms of Pearson's Correlation Coefficient (0.51 of 1st-classifier-2nd level and 0.36 of 2nd-classifier-2nd level respect to 0.35 of Back Propagation Neural Networks and 0.27 of Bagging), Mean Absolute Error (1.83% of 1st-classifier-2nd level and 1.20% of 2nd-classifier-2nd level respect to 2.28% of Support Vector Regression and 2.38% of Bagging) and Root Mean Squared Error (2.16% of 1st-classifier-2nd level and 1.73% of 2nd-classifier-2nd level respect to 3.31% of Back Propagation Neural Networks 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 examples 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, which update the hypothesis of classifier using new individual data instances, without having to re-process past instances (Schlimmer & Granger, 1986). Also we will propose the use of weather time series data which are automatically capture each five minutes in weather station. We will analyze its behavior with the rust infection rate.

ACKNOWLEDGMENT

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, KNIME® Analytics Platform, Ministerio de Economía y Competitividad de España (Proyecto TRA2011-29454-C03-03. i-Support: Sistema Inteligente Basado en Agentes de Soporte al Conductor) and Colciencias (Colombia) for PhD scholarship granted to MsC. David Camilo Corrales.

REFERENCES

Araujo, B. S. (2006). *Aprendizaje automático: conceptos básicos y avanzados: aspectos prácticos utilizando el software Weka*. España: Pearson Prentice Hall.

Avelino, J., Cristancho, M., Georgiou, S., Imbach, P., Aguilar, L., Bornemann, G., & Morales, C. et al. (2015). The coffee rust crises in Colombia and Central America (2008–2013): Impacts, plausible causes and proposed solutions. *Food Security*, *7*(2), 303–321. doi:10.1007/s12571-015-0446-9

Ayodele, T. O. (2010). Types of Machine Learning Algorithms. In Y. Zhang (Ed.), New advances in machine learning (pp. 20-48). India: In-Tech - intechweb.org.

Bhavsar, H., & Ganatra, A. (2012). A Comparative Study of Training Algorithms for Supervised Machine Learning. *International Journal of Soft Computing and Engineering*, *2*(4), 74–81.

Cintra, M. E., Meira, C. A. A., Monard, M. C., Camargo, H. A., & Rodrigues, L. H. A. (2011, 22-24 November 2011). *The use of fuzzy decision trees for coffee rust warning in Brazilian crops.* Paper presented at the 2011 11th International Conference on Intelligent Systems Design and Applications (ISDA).

Corrales, D. C., Corrales, J. C., & Apolinar, F.-C. (2015). Towards Detecting Crop Diseases and Pest by Supervised Learning. *Ingeniería y Universidad*, *19*(1), 207–228. doi:10.11144/Javeriana.iyu19-1.tdcd

Corrales, D. C., Ledezma, A., & Corrales, J. C. (2015a). A conceptual framework for data quality in knowledge discovery tasks: A proposal. *Journal of Computers*, *10*(6), 396–405. doi:10.17706/jcp.10.6.396-405

Corrales, D. C., Ledezma, A., & Corrales, J. C. (2015b). A survey of data quality issues in knowledge discovery tasks. *Paper presented at the VII Congreso Iberoamericano de Telemática - Workshop TIC-@agro*, Popayán, Colombia.

Corrales, D. C., Ledezma, A., Peña, A., Hoyos, J., Figueroa, A., & Corrales, J. C. (2014). A new dataset for coffee rust detection in Colombian crops base on classifiers. *Sistemas y Telemática*, *12*(29), 9–22. doi:10.18046/syt.v12i29.1802

Ghosh, J. (2002). Multiclassifier Systems: Back to the Future. *Paper presented at theThird International Workshop on Multiple Classifier Systems*. doi:10.1007/3-540-45428-4_1

Grubbs, F. (1969). Procedures for Detecting Outlying Observations in Samples. *Technometrics*, *11*(1), 1–21. doi:10.1080/00401706.1969.10490657

Haykin, S. S. (2003). *Neural networks: a comprehensive foundation*. Prentice Hall.

Karimi, Y., Prasher, S. O., Patel, R. M., & Kim, S. H. (2006). Application of support vector machine technology for weed and nitrogen stress detection in corn. *Computers and Electronics in Agriculture*, *51*(1–2), 99–109. doi:10.1016/j.compag.2005.12.001

Kotsiantis, S. B. (2007). Supervised Machine Learning: A Review of Classification Techniques. *Paper presented at the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering*.

Li, L., Zou, B., Hu, Q., Wu, X., & Yu, D. (2013). Dynamic classifier ensemble using classification confidence. *Neurocomputing*, *99*(0), 581–591. doi:10.1016/j.neucom.2012.07.026

Luaces, O., Rodrigues, L. H. A., Alves Meira, C. A., & Bahamonde, A. (2011). Using nondeterministic learners to alert on coffee rust disease. *Expert Systems with Applications*, *38*(11), 14276–14283. doi:10.1016/j.eswa.2011.05.003

Luaces, O., Rodrigues, L.H.A., & Meira, C.A.A., Quevedo, J.R., & Bahamonde, A. (2010). *Viability of an alarm predictor for coffee rust disease using interval regression*. Paper presented at the 23rd international conference on Industrial engineering and other applications of applied intelligent systems, Cordoba, Spain (Vol. 2).

Marqués, A. I., García, V., & Sánchez, J. S. (2012). Two-level classifier ensembles for credit risk assessment. *Expert Systems with Applications*, *39*(12), 10916–10922. doi:10.1016/j.eswa.2012.03.033

McAlister, D. (1879). The Law of the Geometric Mean. *Proceedings of the Royal Society of London*, *29*(196-199), 376–367. doi:10.1098/rspl.1879.0061

Meira, C., Rodrigues, L., & Moraes, S. (2008). Análise da epidemia da ferrugem do cafeeiro com árvore de decisão. *Tropical Plant Pathology*, *33*(2), 114–124. doi:10.1590/S1982-56762008000200005

Meira, C. A. A., Rodrigues, L. H. A., & Moraes, S. A. (2009). Modelos de alerta para o controle da ferrugem-do-cafeeiro em lavouras com alta carga pendente. *Pesquisa Agropecuaria Brasileira*, *44*(3), 233–242. doi:10.1590/S0100-204X2009000300003

Menahem, E., Rokach, L., & Elovici, Y. (2009). Troika – An improved stacking schema for classification tasks. *Information Sciences*, *179*(24), 4097–4122. doi:10.1016/j.ins.2009.08.025

Meyfroidt, G., Güiza, F., Ramon, J., & Bruynooghe, M. (2009). Machine learning techniques to examine large patient databases. *Best Practice & Research. Clinical Anaesthesiology*, *23*(1), 127–143. doi:10.1016/j.bpa.2008.09.003 PMID:19449621

Mitchell, T. (1997). *Machine learning*. McGraw-Hill.

Mucherino, A., Papajorgji, P., & Pardalos, P. (2009). Clustering by k-means. In D.-Z. Du (Ed.), *Data Mining in Agriculture* (Vol. 30, pp. 47–56). New York: Springer. doi:10.1007/978-0-387-88615-2_3

Pérez-Ariza, C. B., Nicholson, A. E., & Flores, M. J. (2012). Prediction of Coffee Rust Disease Using Bayesian Networks. *Paper presented at theSixth European Workshop on Probabilistic Graphical Models*, Granada, Spain.

Poh, H. L. (1991). *A neural network approach for marketing strategies research and decision support* [Ph.D Thesis]. Stanford University.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc.

Ranawana, R., & Palade, V. (2006). Multi-Classifier Systems: Review and a roadmap for developers. *Int. J. Hybrid Intell. Syst.*, *3*(1), 35–61. doi:10.3233/HIS-2006-3104

Rivillas-Osorio, C., Serna-Giraldo, C., Cristancho-Ardila, M., & Gaitán-Bustamante, A. (2011). La roya del cafeto en Colombia, impacto, manejo y costos de control. In S. Marín (Ed.), *Avances Tecnicos Cenicafe. Chinchiná - Caldas - Colombia: Cenicafé*.

Schlimmer, J., & Granger, R. Jr. (1986). Incremental learning from noisy data. *Machine Learning*, *1*(3), 317–354. doi:10.1007/BF00116895

Skurichina, M., & Duin, R. P. (2002). Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis & Applications*, *5*(2), 121–135. doi:10.1007/s100440200011

Suhasini, A., Palanivel, S., & Ramalingam, V. (2011). Multimodel decision support system for psychiatry problem. *Expert Systems with Applications*, *38*(5), 4990–4997. doi:10.1016/j.eswa.2010.09.152

Tin Kam, H. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on* Pattern Analysis and Machine Intelligence, *20*(8), 832–844. doi:10.1109/34.709601

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, Inc. doi:10.1007/978-1-4757-2440-0

Weimin, H., & Leping, S. (2008, August 3-4). Weighted Support Vector Regression Algorithm Based on Data Description. *Paper presented at the ISECS International Colloquium on Computing, Communication, Control, and Management CCCM '08*.

David Camilo Corrales is a PhD student in computer science at Universidad del Cauca, Popayán, Departamento de Telemática. He holds a Master degree in Telematics engineering. Apolinar Figueroa Casas received his PhD in Biology at Universidad de Valencia in 1999. His main fields of research are in ecology, nutrient cycles, climate change and environmental impact assessment. Agapito Ledezma is a Professor at the Universidad Carlos III, Madrid. He received his PhD in Computer Science in 2004 at Universidad Carlos III. His fields of interest are in business intelligence, activity recognition, artificial intelligence, data mining. Juan Carlos Corrales received his PhD in computer science at Université de Versailles Saint-Quentin-en-Yvelines, UVSQ, France in 2008. His fields of research are in Geographic information systems, environmental information systems and services.