IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 10, No. 1, March 2021, pp. 257~264
ISSN: 2252-8938, DOI: 10.11591/ijai.v10.i1.pp257-264

Plant disease prediction using classification algorithms

Maria Morgan[1], Carla Blank[2], Raed Seetan[3]
[1,2]Department of Mathematics and Statistics, Slippery Rock University, USA
[3]Department of Computer Science, Slippery Rock University, USA

Article Info

Article history:

Received Feb 8, 2020
Revised Jul 21, 2020
Accepted Feb 17, 2021

Keywords:

Classification
Mushroom
Plant disease
Prediction
Soybean

ABSTRACT

This paper investigates the capability of six existing classification algorithms (artificial neural network, naïve bayes, k-nearest neighbor, support vector machine, decision tree and random forest) in classifying and predicting diseases in soybean and mushroom datasets using datasets with numerical or categorical attributes. While many similar studies have been conducted on datasets of images to predict plant diseases, the main objective of this study is to suggest classification methods that can be used for disease classification and prediction in datasets that contain raw measurements instead of images. A fungus and a plant dataset, which had many differences, were chosen so that the findings in this paper could be applied to future research for disease prediction and classification in a variety of datasets which contain raw measurements. A key difference between the two datasets, other than one being a fungus and one being a plant, is that the mushroom dataset is balanced and only contained two classes while the soybean dataset is imbalanced and contained eighteen classes. All six algorithms performed well on the mushroom dataset, while the artificial neural network and k-nearest neighbor algorithms performed best on the soybean dataset. The findings of this paper can be applied to future research on disease classification and prediction in a variety of dataset types such as fungi, plants, humans, and animals.

Corresponding Author:

Raed Seetan
Department of Computer Science
Slippery Rock University, USA
Email: raed.seetan@sru.edu

1. INTRODUCTION

The main goal of this paper is to test the accuracy and compare the results of existing classification algorithms in predicting edibility in mushrooms and classifying diseases in soybean plants. While the mushroom and soybean datasets used in this paper have many differences, they are similar in that they are datasets of either numerical or categorical attributes, while many similar studies have been conducted on datasets of images instead of raw measurements [1-4]. The objective of the analysis conducted in this paper is to make suggestions to agricultural researchers, or disease researchers in general, on classification methods that perform well, in terms of disease prediction and classification accuracy, on datasets with raw measurements. While image datasets have been tested with the classification algorithms presented here, many researchers still prefer to take and record measurements by hand when studying plants or fungi. Soybeans and mushrooms are very important to humans; thus, it is important to have accurate methods to predict whether or not different variations are safe for human consumption and predict the presence of any diseases that can affect them.

There are both poisonous and edible mushrooms. According to The Audubon Society Field Guide of North American Mushrooms, there is no single characteristic to distinguish between edible mushrooms and poisonous mushrooms [5-6]. One must be certain a mushroom is one of the edible varieties, otherwise, the mushroom should be considered poisonous. Since various types of mushrooms are consumed by humans, it is important to establish some guidelines to determine if a mushroom is edible or not. In this paper we will attempt to train existing classification algorithms that can be used to classify mushrooms, given a dataset of raw measurements, as either edible or poisonous.

Soybeans are processed for their oil and meal [7]. Soybean oil is used in many foods that humans consume daily such as margarine, baked breads, canned tuna, and fried food. Soybean meal is used in food for many farm animals such as poultry, pork, and cattle. Soybeans are an important crop because the oil is directly put into food that humans consume, and the meal is fed to the animals that are widely consumed by humans. There are various diseases that affect soybean crops, in this paper we will attempt to train existing classification algorithms that can be used to classify soybeans plants as having a particular disease, based on a dataset of raw measurements pertaining to the soybean plants.

Discovering applications and techniques for predicting disease presence and classifying diseases is very important when it comes to agriculture. Diseases in crops can have a serious impact on the crop yield [8]. Because diseases will more than likely damage a large number of crops in a growing cycle, farmers can benefit from classification of crop diseases and risk factors that may lead to these diseases. A forecasting system has been developed to predict disease outbreak in strawberry plants in Florida, where 15% of US berries are produced and all berries grown in the winter [9]. The forecasting system, called the Strawberry Advisory System (SAS), helps farmers by predicting the disease incidence recommending fungicide applications [9]. This system has reduced production costs by eliminating unnecessary fungicide applications, while not risking the crop yield. As Richard Strange noted, almost 10% of global food production is lost due to plant disease [10]. These losses can be minimized if accurate methods are developed for predicting and classifying disease.

The remainder of this paper is structured as: section 2 discusses the literature review works. Section 3 presents our research method. Section 4 discusses the results of our paper. Section 5 provides conclusion and recommendations for further studies.

2. LITERATURE REVIEW

To date, most studies of this type have used images of plants or fungi as the datasets which classification algorithms are tested on. Previous studies have found that decision trees are widely used because of their ease of interpretation, support vector machines (SVM) and artificial neural networks (ANN) are typically the most accurate, and k-nearest neighbor (KNN) and naïve bayes are not the best classification algorithms for agriculture but they are easy to train and thus have been used in many plant and fungi disease classification studies [11].

The six classification methods chosen for comparison in this study were based on the literature reviewed prior to beginning the experiment. In November 2018, a study was published in which 3 classification algorithms were tested to compare their accuracy in predicting diseases in plants, based on a dataset of plant leaf images. This study found that the decision tree algorithm performed better than ANN and naïve bayes [1]. Another study, published in March 2018, compared the classification accuracy of predicting loss caused by grass grub insect using the following techniques: decision tree, random forest, neural networks, gaussian naïve bayes, SVMs, and KNN [12]. The dataset used in [12] was comparable to the data used in this study because it was a dataset of real recorded values, instead of images. However, the main difference between our proposed study and [12] is that our study is focused on predicting the presence of disease and classifying the types of diseases; while the main goal of [12] is to predict the loss of crops due to disease.The March 2018 study found that neural networks, random forest, and gaussian naive bayes were the most accurate in predicting diseases in crops. Finally, a study published in February 2019 compared the accuracy of SVM and ANN classification algorithms in predicting diseases in plants using a dataset of images, this study found that ANN was the most accurate algorithm [2].

In this study, we will compare the accuracy of six different classification algorithms in predicting diseases in soybean plants and edibility in mushrooms: artificial neural network (ANN), naïve bayes, k-nearest neighbor (KNN), support vector machine (SVM), decision tree, and random forest. The results of this study will be compared to those mentioned in the literature review of similar studies that have been done.

3. RESEARCH METHOD

The purpose of this study is to assess the capability of six existing classification algorithms (artificial neural network, naïve bayes, k-nearest neighbor, support vector machine, decision tree and random forest) in classifying and predicting diseases in soybean and mushroom datasets. In this section, we will discuss our methodology starting with data preparation, then introducing the classification methods, and finally evaluation metrics. In the next section, we will discuss the experiments results.

3.1. Data preparation

The mushroom dataset, obtained from UCI machine learning repository, contains 8,124 hypothetical samples of 23 species of gilled mushrooms in the Agaricus and Lepiota families with 22 categorical attributes [6, 13]. The species are classified as edible or poisonous. Any mushroom that cannot be categorized as edible is considered poisonous, regardless of whether it is poisonous. For the purpose of our comparison in this study between the mushroom and soybean datasets, the poisonous classification will be treated as the disease being present and the edible classification will be treated as the disease not being present. The attributes of the mushroom dataset are: cap-shape, cap-surface, cap-color, bruises, odor, gill-attachment, gill-spacing, gill-size, gill-color, stalk-shape, stalk-root, stalk-surface-above-ring, stalk-surface-below-ring, stalk-color-above-ring, stalk-color-below-ring, veil-type, veil-color, ring-number, ring-type, spore-print-color, population, and habitat.

The soybean dataset, also obtained from UCI machine learning repository, contains 307 observations from soybean plants infected with 19 different diseases and 35 categorical attributes [14] and [13]. The diseases present in the soybean dataset are diaporthe-stem-canker, charcoal-rot, rhizoctonia-root-rot, phytophthora-rot, brown-stem-rot, powdery-mildew, downy-mildew, brown-spot, bacterial-blight, bacterial-pustule, purple-seed-stain, anthracnose, phyllosticta-leaf-spot, alternarialeaf-spot, frog-eye-leaf-spot, diaporthe-pod-&-stem-blight, cyst-nematode, 2-4-d-injury, and herbicide-injury. The 35 categorical attributes present in the soybean dataset are date, plant-stand, precip, temp, hail, crop-hist, area-damaged, severity, seed-tmt, germination, plant-growth, leaves, leafspots-halo, leafspots-marg, leafspot-size, leaf-shread, leaf-malf, leaf-mild, stem, lodging, stem-cankers, canker-lesion, fruiting-bodies, external decay, mycelium, int-discolor, sclerotia, fruit-pods, fruit spots, seed, mold-growth, seed-discolor, seed-size, shriveling, and roots.

The attribute values in the mushroom dataset were coded numerically. An R program was written to replace these numeric values with their true values. The description of this dataset from the UCI Machine Learning Repository discussed the one attribute (stalk_root) where values were missing. This was verified using Microsoft Excel because of the simplicity of the dataset. A decision was made to run each of the classification algorithms on two versions of the mushroom dataset; one version with all attributes included and another version with the stalk_root attribute removed. The purpose of creating these two versions was to investigate whether or not the attribute with missing values would skew the results of the classification algorithms.

The attribute values in the soybean dataset were coded numerically as well, an R program was written to replace these numeric values with their true values. The soybean dataset was then examined for missing values, also using an R program. There were nine attributes (hail, severity, seed_tmt, leaf_mild, lodging, shriveling, fruiting_bodies, fruit_spots, seed_discolor) where 10% or more of the values were missing. A decision was made to run each of the six classification algorithms on two versions of the soybean dataset, one version with all attributes included and another version with these nine attributes removed. The purpose of creating these two versions was the same as the justification for the same method in the mushroom dataset, to investigate whether or not these attributes with missing values would skew the results of the classification algorithms. Upon further investigation of the soybean dataset, it was discovered that there existed only one data point for the 2-4-d-injury class and that most of the attribute values were missing for this one data point. This data point was removed from both versions of the soybean datasets in order to avoid skewing the results of the classification algorithms.

3.2. Classification methods

Six different classification techniques were tested in this study to build classification models for predicting diseases in soybeans and edible or poisonous features of mushrooms. The classification algorithms were all trained using 10-fold cross validation and were executed using functions in Weka [15]. With 10-fold cross validation, the rows within the datasets are randomly reorganized and split into 10 folds of equal size [16]. With each iteration of the classification model training process, one fold is used as the test dataset and the remaining 9 folds are used as the training datasets. This process repeats 10 times until each fold has been used as the test dataset. The resulting classification model is an average of the 10 iterations of the training process. The following 6 classification methods were used in this study:

3.2.1. Artificial neural network

Artificial neural networks (ANN) are built to resemble the way a human brain thinks. ANNs contain multiple weighted connections between inputs and outputs, these weights are adjusted when building the model on the training data in order to correctly predict class labels based on the input data object [17]. In this study, ANNs were built using the multilayer perceptron algorithm in Weka. The multilayer perceptron algorithm builds an ANN through a process called backpropagation. In this process, weights are assigned to each data object in the input layer of the ANN. These weights are then re-assigned as necessary in one, or multiple, hidden layers of the ANN in order to minimize the mean squared error between the class label predicted by the ANN and the true class label of the given data object. The process is called backpropagation because these adjustments to the weights are done in the backwards direction starting at the output layer, which contains the class labels, and going back through all of the hidden layers to the first hidden layer [18]. The multilayer perceptron algorithm was executed using a learning rate of 0.3, a momentum of 0.2, and training time of 500.

3.2.2. Naïve bayes

Naïve bayes is a probabilistic classification method that uses bayes theorem. The naïve bayes classifier takes a set of features from a dataset and determines the probability of each feature occurring in each class within the data [19]. For each row of data, the values of the attributes are used to calculate the posterior probability for each class within the dataset, the row of data is then assigned to the class with the highest posterior probability. This method is referred to as naïve because it assumes that all features of the dataset are independent of one another, which is an assumption that is likely untrue and thus naïve. Despite this assumption not being true in all cases, naïve bayes has been shown to be a successful classifier in large datasets. The naïve bayes algorithm was executed using the naivebayes classifier in Weka. The naïve bayes classifier in Weka uses estimator classes. A batch size of 100 was used without kernel estimation or supervised discretization.

3.2.3. k-nearest neighbor

The k-nearest neighbor (KNN) algorithm assigns class labels to rows within a dataset based on the class labels of training data that are similar [17]. The KNN algorithm works by searching the training data for k training tuples that are closest to the test data tuple and assigns the test tuple a class label based on the class labels of those closest training tuples. The closeness of a training tuple to a test tuple is determined using a distance function, such as Euclidean distance. KNN was implemented in Weka for this experiment using the instance based learner (IBK) algorithm. The IBK algorithm was executed using the Euclidean distance function, a batch size of 100, and k=1.

3.2.4. Support vector machine

Support vector machine (SVM) is a supervised machine learning algorithm used in classification and regression. SVMs were first presented by Vladimir Vapnik and his coworkers, Bernhard Boser and Isabelle Guyun, at the computational learning theory (COLT-92) conference [20]. In this algorithm, training data is transformed to a higher dimension. A line or hyperplane separates the classes of data from each other. The line or hyperplane are found using support vectors. Support vectors are the points closest to the hyperplane. SVMs are highly accurate, which makes up for the slow speed associated with them. In this study, SVMs were built using the sequential minimal optimization (SMO) algorithm in Weka. The SMO algorithm uses the complexity parameter, also known as the C parameter, to control the flexibility of the process in drawing the line between classes [21]; the C parameter used was 1.0. The PolyKernel default was used, which separates the classes by a curved line [21].

3.2.5. Decision tree

A decision tree is a structure that contains internal nodes that denote attributes, branches that denote the outcome of a test on an observation and leaf nodes that denote the class label [17]. The top node of this tree-like structure is the root node. In order to determine the class of an observation, the decision tree is followed, starting at the root, moving down to the leaf nodes. The decision tree algorithm was implemented in Weka for this study using the J48 decision tree algorithm. The J48 algorithm was executed using a batch size of 100, the minimal of objects of 2, without using unpruned trees, a confidence interval of 0.25, subtree raising and without binary splits [22].

3.2.6. Random forest

A random forest is a collection of decision trees. Each decision tree within the random forest generates a class prediction; the class with the largest number becomes the prediction of the random forest

[23]. In order for this algorithm to be efficient, the individual models must not be correlated or should have a low correlation. There are two methods used to ensure that the individual decision tree models are not too closely correlated with each other. One method is bagging. Each individual tree selects a random sample from the dataset with replacement [23]. The second method is random linear combinations of the attributes. This method uses new attributes that are a linear combination of the existing attributes [17]. This also helps to reduce correlation between classifiers. The random forest algorithm in Weka was used in this study. The random forest algorighm uses the numFeatures value of 0, which selects the number of attributes considered at each split point. The algorithm was executed with a bag size percent of 100%, which creates a new random sample the same size as the training sample. The NumIterations value was 100, which sets the number of bags or iterations to 100.

3.3. Performance evaluations

The following seven measures were used to evaluate the performance of the six classification algorithms on the soybean and mushroom datasets, these measures were selected based on their use in a similar study which used classification functions in Weka for plant disease detection on a dataset of plant images [4].

Accuracy: A percentage calculated by dividing the number of correctly classified data points by the total number of data points and multiplying by 100.

Mean absolute error: The mean absolute error (MAE) is calculated by taking the sum of the absolute errors divided by the number of non-missing data points.

True positive rate: The TP rate is calculated by dividing the number of true positive classifications by the sum of the number of true positive classifications and the number of false negative classifications. (TP/(TP+FN)).

False positive rate: The FP rate is calculated by dividing the number of false positive classifications by the sum of the number of false positive classifications and the number of true negative observations. (FP/(FP+TN)).

Precision: Precision is calculated by dividing the number of true positive classifications by the sum of the number of true positive classifications and the number of false positive classifications. (TP/(TP+FP)).

Recall: Recall is calculated by dividing the number of true positive classifications by the sum of the number of true positive classifications and the number of false negative classifications. (TP/(TP+FN)).

F - Measure: The F-Measure is calculated by multiplying the precision and recall, dividing this value by the sum of the precision and recall, and finally multiplying this number by two. (2*((precision*recall)/(precision+recall))).

4. RESULTS AND DISCUSSION

The algorithms were tested on both variations of the mushroom dataset, one with the attribute with missing values removed and one with all attributes included. The measures previously described were reported for each classification algorithm. The results for both variations of the mushroom dataset were similar, so the reported results are from the version of the dataset with all attributes included. All of the six algorithms tested performed extremely well on the mushroom dataset with almost all accuracy values at 100%. The naïve bayes algorithm performed the worst on this dataset with an accuracy of 95.83%, which is still a good accuracy level. Table 1 shows for results for the mushroom dataset.

Table 1. Results for mushroom dataset

Parameter	Classification Method					
	ANN	Naïve Bayes	KNN	SVM	Decision Tree	Random Forest
Accuarcy	100.00%	95.83%	100.00%	100.00%	100.00%	100.00%
MAE	0.00	0.04	0.00	0.00	0.00	0.00
TP-Rate	1.00	0.96	1.00	1.00	1.00	1.00
FP-Rate	0.00	0.04	0.00	0.00	0.00	0.00
Precision	1.00	0.96	1.00	1.00	1.00	1.00
Recall	1.00	0.96	1.00	1.00	1.00	1.00
F-Measure	1.00	0.96	1.00	1.00	1.00	1.00

The algorithms were tested on two variations of the soybean dataset, one with any attributes that contained 10% or more missing values removed and one with all attributes included. The results for both variations of the soybean dataset were similar, so the reported results are from the version of the dataset with all attributes included. All of the six algorithms, except for decision tree, performed well on the soybean

dataset with accuracy values falling in the range of 89.22-91.83%. The decision tree algorithm had a reported accuracy of 82.68% for the soybean dataset, so this was the worst-performing algorithm in classifying diseases in the soybean dataset. Table 2 shows for results for the soybean dataset.

Table 2. Results for soybean dataset

Parameter	Classification Method					
	ANN	Naïve Bayes	KNN	SVM	Decision Tree	Random Forest
Accuarcy	91.18%	90.20%	91.83%	89.22%	82.68%	89.54%
MAE	0.01	0.01	0.01	0.1	0.02	0.04
TP-Rate	0.91	0.9	0.92	0.89	0.83	0.9
FP-Rate	0.91	0.01	0.01	0.01	0.02	0.01
Precision	0.91	0.92	0.92	0.89	0.82	0.9
Recall	0.91	0.9	0.92	0.89	0.83	0.9
F-Measure	0.91	0.9	0.92	0.89	0.83	0.89

In comparing the results for both datasets, ANN and KNN performed best on the soybean dataset, while all methods other than naïve bayes performed at 100% accuracy on the mushroom dataset. It is to be expected that most of the classification methods would perform best on the mushroom dataset because there are only two classes present in this dataset, while the soybean dataset that was tested has 18 classes. As mentioned in theliterature review section, naïve bayes and KNN are not typically used in agricultural studies. This is an interesting point to consider because naïve bayes was the only algorithm that did not produce 100% accuracy in the mushroom dataset, but also interesting to note because KNN was one of the best performing algorithms in the soybean dataset [11]. Although the performance of KNN on the soybean dataset seem to conflict with the previous literature, the results show that ANN was one of the top performing algorithms in the soybean dataset compared to the other algorithms, and this confirms what was found in the February 2019 study mentioned in the literature review section [2].

Further comparison of the results for both datasets revealed another major difference between the two datasets. The mushroom dataset is balanced, with the observations being equally distributed among the two classes, poisonous and edible. The balanced distribution of the mushroom dataset is shown in Figure 1. The soybean dataset, however, is imbalanced among the disease classes. As shown in Figure 2, there are 4 classes that contain a much higher percentage of the observations compared to the other classes. The disease classes with this high percentage of observations, in Figure 2 (D1 through D4), are phytophthora-rot, brown-spot, alternarialeaf-spot, and frog-eye-leaf-spot. Because the soybean dataset is imbalanced, measures other than accuracy needed to be considered to determine if the imbalance of the dataset was skewing the results for each classification algorithm. In the case of imbalanced datasets with a large number of values, the precision and recall values can be evaluated to determine the performance of the classification algorithm [24]. As shown in the results for the classification algorithms tested on the soybean dataset in Table 2, all of the precision and recall values are close to 1. Referring back to the parameter evaluations section of this paper, this indicates that the number of true positive classifications are much larger than the number of false negative and false positive classifications. If the classification algorithms were being skewed by the imbalance in the dataset, our precision and recall values would be much lower. Thus, although this is a major difference between our datasets, the imbalance feature of the soybean dataset did not have an adverse effect on the results of each classification algorithm. In future studies of this kind, datasets that are imbalanced may need additional data preparation techniques as to not skew the results of the classification algorithms. Two potential techniques for handling imbalanced datasets are oversampling and undersampling [25]. In oversampling, synthetic data is generated so that additional observations are present in the minority class and the distribution of the data is more equal among the classes. In undersampling, observations are removed from the majority classes to make the distribution of data more equal among the classes.

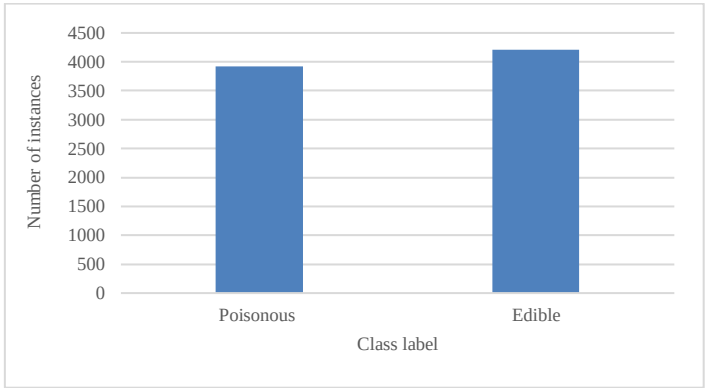

Figure 1. Mushroom dataset class distribution

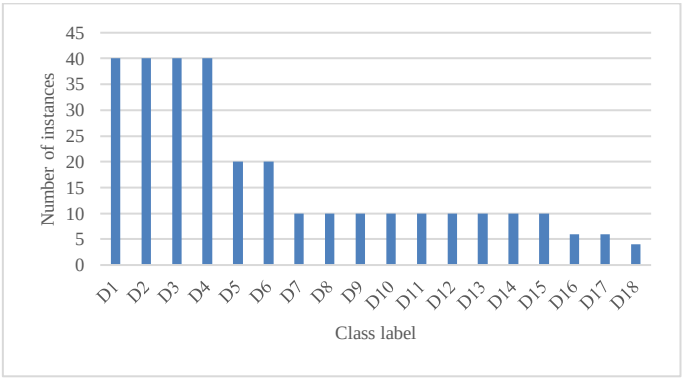

Figure 2. Soybean dataset class distribution

5. CONCLUSION

In this paper, we tested ANN, naïve bayes, KNN, SVM, decision tree, and random forest classifiers to predict disease presence in a mushroom dataset and classify disease in a soybean dataset. In the mushroom dataset, we found that all classifiers, except for naïve bayes, performed at 100% accuracy. This is a likely result given a dataset with only two classes. In the soybean dataset, we have shown that ANN and KNN are the best classifiers in terms of accuracy, but that ANN is likely the better choice since KNN classification is not typically used for plant datasets. We also showed that the imbalance of the soybean dataset did not affect the results of the classification methods, likely because a large amount of data is present. In the mushroom dataset, we used classification to determine if a disease was present or not (edible or poisonous) and in the soybean dataset, we used classification to determine which disease was present. The purpose of these experiments was to come up with classification methods that can be used on datasets for plants or fungi that contain real measurements instead of images. The findings in this paper can be repeated on similar fungi or plant datasets but may also be extended to training classification algorithms for predicting disease presence or disease classification in human or animal datasets with raw measurements.

REFERENCES

[1] G. Prem, *et al*, "Plant Disease Prediction Using Machine Learning Algorithms," *International Journal of Computer Applications*, vol. 182, no. 25, pp. 1–7, 2018. DOI: 10.5120/ijca2018918049.
[2] N. Kanaka Durga & G. Anurhada, "Plant Disease Identification Using SVM and ANN Algorithms," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 7, no. 5S4, 2019.
[3] H. Al-Hiary, *et al.*, "Fast and Accurate Detection and Classification of Plant Diseases," *International Journal of Computer Applications*, vol 17, no. 1, 2011. DOI: 10.5120/2183-2754.
[4] R. Ramya, *et al.*, "A Review of Different Classification Techniques in Machine Learning Using Weka for Plant Disease Detection," *International Research Journal of Engineering and Technology (IRJET)*, vol 5, no.5, 2018.
[5] G. H. Lincoff, The Audubon Society field guide to North American mushrooms. Alfred A. Knopf; distributed by Random House, 1981.

[6] G. H. Lincoff, "UCI Machine Learning Repository," University of California, Irvine, School of Information and Computer Sciences, 1981. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/mushroom.
[7] North Carolina Soybeans, "Uses of Soybeans," 2019. [Online]. Available: https://ncsoy.org/media-resources/uses-of-soybeans/.
[8] L. V. Madden, G. Hughes, and M. E. Irwin, "Coupling Disease-Progress-Curve and Time-of-Infection Functions for Predicting Yield Loss of Crops," *Phytopathology*, vol 90, no. 8, 2000. https://doi.org/10.1094/PHYTO.2000.90.8.788.
[9] W. Pavan, C. W. Fraisse, an N. A. Peres, "Development of a web-based disease forecasting system for strawberries," *Computers and Electronics in Agriculuture*, vol 75, no. 1, 2011.
[10] R.N Strange and P. R. Scott, "Plant Disease: A Threat to Global Food Security," *Annual Review of Phytopathology*, vol. 43, no.1, pp. 83-116, 2005.
[11] D. C. Corrales, J. C. Corrales and A. Figueroa-Casas, "Towards detecting crop diseases and pest by supervised learning*," Ing. Univ.*, vol. 19, no. 1, pp. 207-228.
[12] U. Ayub and S. A. Moqurrab, "Predicting crop diseases using data mining approaches: Classification," *2018 1st International Conference on Power, Energy and Smart Grid (ICPESG)*, 2018.
[13] D. Dua, and C. Graff, "UCI Machine Learning Repository," University of California, Irvine, School of Information and Computer Sciences, 2019. [Online]. Available: http://archive.ics.uci.edu/ml
[14] R.S. Michalski and R.L. Chilausky, "UCI Machine Learning Repository," University of California, Irvine, School of Information and Computer Sciences, 1980. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Soybean+(Large)
[15] E. Frank, M. A. Hall, and I. H. Witten, "The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques," Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, Fourth Edition, 2016. [Online]. Available: xxxxxhttps://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf
[16] R. Bouckaert, WEKA Manual for Version 3-7-8, 2013. [Online]. Available: http://statweb.stanford.edu/~lpekelis/13_datafest_cart/WekaManual-3-7-8.pdf
[17] J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques. Amsterdam: Elsevier/Morgan Kaufmann, 2012.
[18] R. Rojas R, "The Backpropagation Algorithm," Neural Networks. Springer, Berlin, Heidelberg, 1996.
[19] D. Soni, "Introduction to Naive Bayes Classification," Towards Data Science, 16 July 2019. [Online]. Available: https://towardsdatascience.com/introduction-to-naive-bayes-classification-4cffabb1ae54
[20] R. Gandhi, "Support Vector Machine - Introduction to Machine Learning Algorithms," Towards Data Science, 7 June 2018. [Online]. Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
[21] J. Brownlee, How To Use Classification Machine Learning Algorithms in Weka, 25 July 2016. [Online]. Available: https://machinelearningmastery.com/use-classification-machine-learning-algorithms-weka/
[22] V. Koblar, Study Programme: Information and Communication Technologies, 2012. [Online]. Available: https://pdfs.semanticscholar.org/94a2/9d5a74ed9ac656de4e55c71fac92d07795ef.pdf
[23] T. Yiu, "Understanding Random Forest," Towards Data Science, 12 June 2019. [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2
[24] S. Lador, "What metrics should be used for evaluating a model on an imbalanced data set?," Towards Data Science, 5 Sep 2017. [Online]. Available: https://towardsdatascience.com/what-metrics-should-we-use-on-imbalanced-data-set-precision-recall-roc-e2e79252ae
[25] W. Bard, "Having an Imbalanced Dataset? Here Is How You Can Fix It," Towards Data Science, 22 Feb 2019. [Online]. Available: https://towardsdatascience.com/having-an-imbalanced-dataset-here-is-how-you-can-solve-it-1640568947e