Journal of Industrial Electronics Technology and Application Vol. No. ( )

ISSN(Online) :

A Study on Comparative Analysis of Machine Learning Algorithms Using the Leaf Dataset

Phuc Hong Ngo [1], Donghwoon Kwon [2], *

[1] Department of Mathematics and Computer Science, Beloit College, Beloit, USA

[2] Department of Mathematics, Computer Science, and Physics, Rockford University, Rockford, USA

Abstract. Machine learning is widely used for classification in various fields. In this research, we compared and analyzed the performance of three popular machine learning classifiers such as KNN, SVM, and ANN, using the leaf dataset. The original dataset was preprocessed, and the feature selection technique was used to divide the preprocessed dataset into two different types of the dataset. According to experiments, the ANN classifier showed 76.18% of accuracy when all the features were employed, and it outperformed other classifiers. However, when partial features were employed, the SVM classifier showed 73.31% of accuracy that outperformed other classifiers.

Keywords; Machine Learning; KNN; SVM; ANN

1. Introduction

Plants play a crucial role in bolstering life on Earth. The oxygen released from photosynthesis largely accounts for the maintenance of the atmosphere. Plants are also an important source and ingredient of food, medicines, and industrial products. However, according to a recent report, 40 percent of the world’s plant species are reported to be at risk of extinction [1]. Therefore, maintaining and having plant databases, as well as classifications, are critical for plant protection. Plants could be classified by their flower, stems, fruits, etc. However, we decided to consider leaf attributes for plant recognition in this paper because of their complexity in the 3D structure; specifically, the set of attributes range from the solidity to smoothness of the leaf.

A variety of classifiers and algorithms have been developed and proposed for leaf recognition [2-3]. Nevertheless, we employed popular and fundamental machine learning algorithms such as k-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Artificial Neural Network (ANN) because the employed dataset does not seem very complicated. Furthermore, the dataset used in this research came from the

Kaggle site due to the limitation of creation and acquisition [4]. The adopted dataset is called the leaf dataset, and it is the .csv file format. It is composed of multi labels, i.e., a total of 36 labels. Not only 36 labels, but it also includes other attributes like eccentricity, aspect ratio, elongation, solidity, etc. [4]. According to evaluating machine learning classifiers with the dataset, the ANN classifier outperformed the other ones when employed 15 features, but the SVM classifier showed the highest accuracy when adopted 11 features out of 15 features.

This paper is organized as follows: Section 2 discusses the relevant literature of methods of leaf classification. Section 3 provides a brief introduction to each classifier used in the paper. Section 4 contains the results acquired from experiments, and Section 5 discusses the conclusion.

2. Related Work

The first common thing observed from most research papers is that they included an image pre-processing step consisting of cropping, gray scaling, noise removal, etc. Besides, they focused on reducing the computational load and enhancing the accuracy in the phase of distinctive feature extraction [2-3][5-8]. Yet, since the leaf dataset employed in this research is text-based, the image pre-processing step is not applicable.

The second common thing we observed in [2-3] and [5-8] is that KNN, SVM, ANN, and/or a hybrid classifier were used. Furthermore, the classifiers' accuracy with the image-based leaf dataset is pretty much high. For instance, the accuracies ranging from 90% to 100% were reported in [5], 94.5% of the accuracy was reported in [7], and the accuracies ranging from 85.46% to 93.17% were reported in [8].

Based on the literature review described above, we could have the following research questions; Do KNN, SVM, ANN classifiers also work well with the text-based dataset? Which classifier returns the best accuracy?

3. Methodologies

To find answers to the research questions established in Section 2, we built the KNN, SVM, and ANN classifiers with the following hardware specifications and Python modules: Intel i7-8750H CPU 2.20GHz × 12 with 16 GB of RAM, Pandas, Scikit-Learn, Matplotlib, and Numpy.

A. Dataset Pre-processing

The original leaf dataset is composed of both image and text-based data, but we decided to adopt the text-based one. There are 16 features in the text-based dataset, and Table 1 below describes each feature in the dataset.

TABLE I. DATASET FEATURES [4]

No.	Features	Description	No.	Features	Description
1	Class	Represents leaf species.	9	Maximal Indentation Depth	Measures a degree of maximum indentation depth by sampling at 1-degree intervals.
2	Specimen Number	Counts the number of each class.	10	Lobedness	Indicates how lobed a leaf is.
3	Eccentricity	The eccentricity of the ellipse that has values ranging from 0 to 1.	11	Average Intensity	Calculates the mean of the intensity leaf image.
4	Aspect Ratio	Indicates the degree of an elongated shape. The closer the value is to zero, the more elongated the shape is.	12	Average Contrast	The standard deviation of the intensity leaf image.
5	Elongation	Calculates the region of a leaf using the diameter and has a value from 0 to 1.	13	Smoothness	Measures how relatively smooth the intensities in a given region is.
6	Solidity	Measures the degree of convexity.	14	Third Moment	Measures the intensity of histogram’s skewness.
7	Stochastic Convexity	Extends the notion of convexity in terms of topology	15	Uniformity	Measures intensity levels.
8	Isoperimetric Factor	Has a value 0 through 1, and as it reaches the circular region, it has a value of 1.	16	Entropy	Measures intensity randomness.

Features 3 (Eccentricity) through 9 (Maximal Indentation Depth) and features 10 (Lobedness) through 16 (Entropy) refer to a shape and texture, respectively.

Furthermore, since the specimen number feature is not necessary, this feature is eliminated so that there are a total of 15 features. As mentioned earlier, the class feature consists of a total of 36 classes (labels).

Based on the preprocessed dataset, two possible scenarios come to the fore. The first scenario is to employ all 15 features, and the second scenario is to adopt some critical features out of 15 for feature selection. Note that the dataset is split into two categories: training and testing based on the rule of 80/20. Various feature selection techniques are considered to select critical features, e.g., VarianceThreshold, SelectKBest, SelectFromModel, etc. Among them, we choose SelectKBest by our subjective judgment. This technique is for univariate feature selection, which means that since non-informative features could be added to the dataset, it is used to select the informative features based on the ANOVA F-value calculation. Thus, the following 11 features are selected according to the highest ANOVA F-value: Aspect Ratio, Isoperimetric Factor, Solidity, Elongation, Stochastic Convexity, Eccentricity, Maximal Indentation Depth, Average Contrast, Smoothness, Average Intensity, and class. Therefore, experiments are conducted with these two different datasets; that is, with all 15 features or 11 key features.

B. KNN

KNN is an algorithm that adopts the lazy learning method, which means the classifier does not make any generalization of the dataset prior to the query phase and store all the training data to answer classification or regression problems.

Each instance of the dataset represents a neighbor in a multidimensional space. To make classifications, it first needs to choose a distance metric to calculate the distance from the query to other neighbors and then take the major vote between *k* nearest points to decide the label of the query.

One significant advantage of this method is the simplicity of implementation without the need for optimization and training [9]. The model could also be easily modified to answer different problems due to the fact that the target function is calculated locally. However, KNN is vulnerable to noise, which deteriorates the accuracy. Thus, choosing the suitable *k* and normalizing the dataset could tremendously improve the performance [10]. The following Fig. 1 below illustrates the overview of KNN.

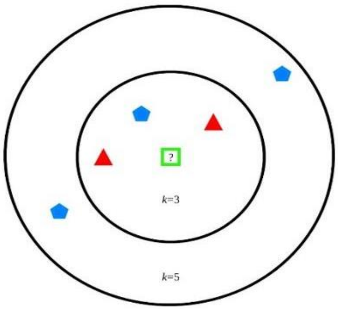

Fig 1. The overview of KNN. The green square is a test sample and will be assigned to red triangles if *k=3*. Otherwise, if *k=5*, it will be assigned to blue pentagons

C. SVM

The idea behind SVM is to determine the largest hyperplane to separate vectors of different classes on a multi-dimensional plane. The margin or size of the hyperplane is defined to be the distance between the nearest data points of different labels.

If a dataset cannot be linearly separated, the soft margin approach could be alternatively used; that is, a classifier considers the balance between increasing the hyperplane's width and committing more training errors [12]. In a non-linear problem, the kernel trick could be applied to map original inputs into a higher-dimensional space [12]. With such a transformation, finding a hyperplane in the higher dimension could help make a decision boundary.

Note that SVM is one of the most robust classifiers, but it has a complex algorithm structure and slow training speed. The trained model is also difficult to interpret.

D. ANN

ANN is a bio-inspired classifier that attempts to mimic the human brain's biological neural network [13]. The basic structure of ANN comprises an input layer and output layer with one more many hidden layers. Each layer has nodes called neurons, and they are interconnected through weighted edges denoting the strength of the neurons.

The data is first passed from the input layer to calculate the weighted sum on its inputs. The result is then evaluated, and the weights are adjusted so that the result is able to get closer to the expected training output. This process is called the feed-forward and back-propagation method.

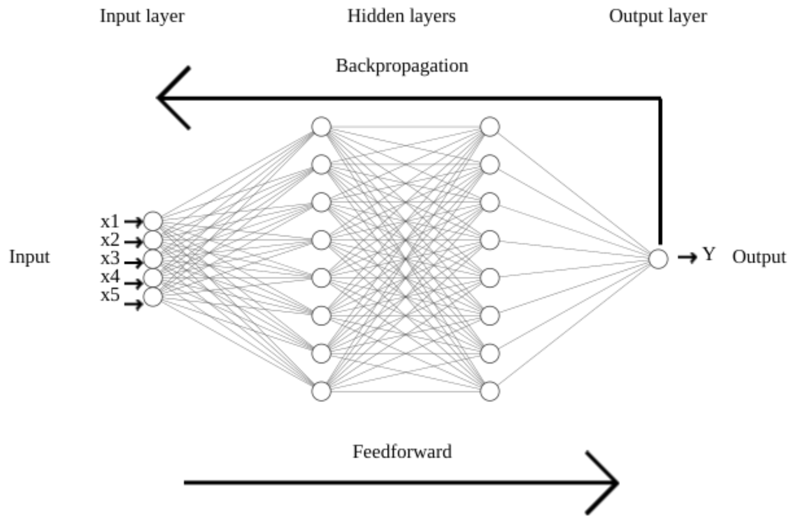

Fig 2. The structure of ANN. It is an interconnected graph of nodes between all three layers.

ANN is a simple statistical model that can effectively solve the predictive query. However, one must pay attention is the existing noise in the data, and it sometimes takes a long time for model training [12]. Fig. 2 above depicts a structure of ANN. Note the ANN model in this research consists of one input, one hidden, and one output layer.

4. Evaluation

To evaluate both datasets, we built KNN, SVM, and ANN classifiers. Keep in mind that each classifier has different hyperparameter settings depending on the datasets because the primary purpose of the experiments is to obtain the highest accuracy. See the following Tables 2 and 3 for the hyperparameters used in each classifier.

TABLE II. HYPERPARAMETER SETTINGS

Classifier	Dataset	Hyperparameters
KNN	with all 15 features	n_neightbors: 4, p: 1, weight: ‘distance’
	with 11 selected features	n_neighbors: 2, p: 1, weight: ‘distance’
SVM	with all 15 features	C: 100000, kernel: ‘poly’, cache_size: 500, degree: 2, class_weight: ‘balanced’, gamma:‘scale’
	with 11 selected features	C: 100000, kernel: ‘rbf’, cache_size: 500, class_weight: ‘balanced’, gamma: ‘scale’

ANN	with all 15 features	hidden_layer_sizes: 70, activation: ‘relu’, alpha: 0.001, max_iter: 2200, solver: ‘lbfgs’, learning_rate_init: 0.0001, learning_rate: ‘adaptive’
	with 11 selected features	hidden_layer_sizes: 64, activation: ‘relu’, alpha: 0.001, max_iter: 1500, solver: ‘lbfgs’, learning_rate_init: 0.0001, learning_rate: ‘adaptive’

TABLE III. EXPLANATION OF THE HYPERPARAMETERS

Hyperparameters	Description
n_neighbors	The number of neighbors to take into consideration upon queries
p	Power parameter for the Minkowski distance. p either equals 1 or 2 that corresponds to Manhattan distance and Euclidean distance
weight	The weight function to calculate the influence of each neighbor. If weights equal ‘uniform’, all points are equivalent. Otherwise, the closer the point, the more impact it has on the vote with ‘distance’.
C	The regulation parameter. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly.
kernel	It is the kernel function used in the classifier. It includes ‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, or ‘precomputed’
cache_size	The size of the kernel cache in MB
degree	Degree of ‘poly’ kernel function
class_weight	Set the parameter C of class *i* to class_weight [*i*] * C for SVC. If not given, all classes are supposed to have weight one. The ‘balanced’ mode uses the values of *y* to automatically adjust weights inversely proportional to class frequencies.
gamma	Kernel coefficient for ‘rbf’, ‘poly’ and ‘sigmoid’. If gamma = ‘scale’ is passed, it uses 1 / (the number of features * input variance) as the value of gamma. If it is ‘auto’, it uses 1 / the number of features.
hidden_layer_sizes	The number of neurons in each hidden layer
activation	Activation function for hidden layer. The option includes ‘identity’, ‘logistic’, ‘tanh’, or ‘relu’.
alpha	L2 penalty parameter
max_iter	The maximum number of iterations or epochs
solver	The solver for weight optimization. It could be chosen from ‘lbfgs’, ‘sgd’, or ‘adam’

Based on the hyperparameter settings mentioned in Tables 2 and 3 above, each classifier returns the following accuracies shown in Fig. 3.

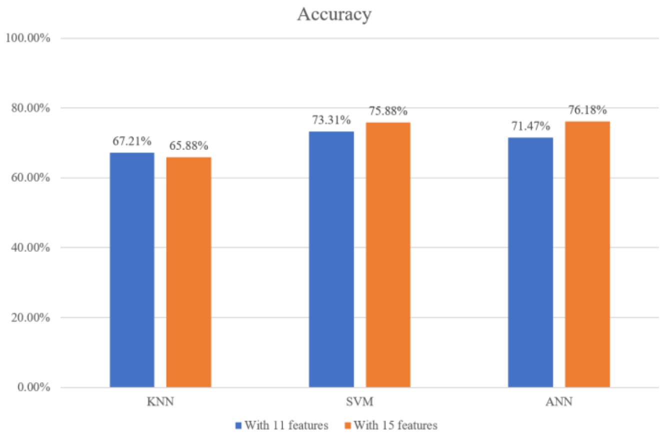

Fig 3. The accuracy of each classifier

According to Fig. 3, the ANN classifier with all 15 features returns the highest accuracy compared to other classifiers, and the KNN classifier with 15 features returns the lowest accuracy. Interestingly, when we evaluated each classifier with 11 key features, the SVM classifier showed the highest accuracy. One reason we can think of these two different accuracies is probably related to the nature of the dataset; in other words, the diverse and non-linear nature of the dataset. As mentioned in the methodology section earlier, the SVM classifier is one of the most robust classifiers when the dataset is not linearly separated and has a large set of labels or classes. Furthermore, another interesting observation is that all three classifiers do not show higher accuracies than using the image-based dataset; that is, three classifiers are more powerful when utilizing image-based datasets than when utilizing text-based datasets.

5. Conclusion

In this research, we evaluated three machine learning classifiers, i.e., KNN, SVM, and ANN, using the text-based leaf dataset. After tuning hyperparameters and applying cross-validation, we observed that ANN showed the highest accuracy with 76.18% when all features were employed. We also observed that SVM with 15 features obtained a slightly lower accuracy than the ANN classifier due to the multilabel classification task and the imbalanced dataset. However, when 11 features out of 15 features were adopted, the SVM classifier showed a slightly better accuracy than the ANN classifier. The paper showed that we could acquire a relatively decent result with only simple tuned classifiers. However, future studies are needed to improve the classification's performance and provide a more in-depth analysis due to several limitations.

References

[1] A. Antonelli et. al., “State of the World’s Plants and Fungi,” Diss. Royal Botanic Gardens (Kew); Sfumato Foundation, 2020.

[2] M. Lukic, E. Tuba, and M. Tuba, “Leaf recognition algorithm using support vector machine with Hu moments and local binary patterns,” 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI), pp. 485-490, Jan. 26, 2017.

[3] K. B. Lee, and K. S. Hong, “An implementation of leaf recognition system using leaf vein and shape,” International Journal of Bio-Science and Bio-Technology, 5(2), pp.57-66, April, 2013.

[4] Leaf Dataset retrieved from https://archive.ics.uci.edu/ml/datasets/Leaf.

[5] J. Chaki, and R. Parekh, “Plant leaf recognition using shape based features and neural network classifiers,” International Journal of Advanced Computer Science and Applications, 2(10), 2011.

[6] S. G. Wu et. al, “A leaf recognition algorithm for plant classification using probabilistic neural network,” IEEE international symposium on signal processing and information technology, pp. 11-16. IEEE, Dec. 15, 2007.

[7] C. A. Priya, T. Balasaravanan, and A. S. Thanamani, “An efficient leaf recognition algorithm for plant classification using support vector machine,” International conference on pattern recognition, informatics and medical engineering (PRIME-2012), pp. 428-432. IEEE, Mar. 21, 2012.

[8] X. Gu, J. X. Du, and X. F. Wang, “Leaf recognition based on the combination of wavelet transform and gaussian interpolation,” International Conference on Intelligent Computing, pp. 253-262. Springer, Berlin, Heidelberg, Aug. 23, 2005.

[9] D. J. Hand, “Principles of data mining,” Drug safety, 30(7), 621-622, Jul., 2007.

[10] S. M. Piryonesi, and T. E. El-Diraby, “Role of data analytics in infrastructure asset management: Overcoming data size and quality problems. Journal of Transportation Engineering, Part B: Pavements,” Journal of Transportation Engineering, Part B: Pavements / Volume 146 Issue 2, Journal of Transportation Engineering, Part B: Pavements 146, no. 2 (2020): 04020022, Jun. 1, 2020.

[11] P. N. Tan, M. Steinbach, and V. Kumar, “Introduction to data mining,” Pearson Education India, 2016.

[12] W. S. Sarle, “Neural networks and statistical models,” 1994.

[13] J. Ren, “ANN vs. SVM: Which one performs better in classification of MCCs in mammogram imaging”, Knowledge-Based Systems, Vol. 26, pp. 144-153, 2012.