POLLEN CLASSIFICATION USING CLASSICAL ML ALGORITHMS ON FLUORESCENCE AND SCATTERING DATA

Mihai Boldeanu[1], Corneliu Burileanu[1], Horia Cucu[1], Luminita Marmureanu[2]

Automatic pollen classification is a very new interdisciplinary field. While most research is focused on using deep neural networks for pollen classification, with promising results, the applicability of such algorithms on large historical data bases of pollen measurements is difficult due to the amount of computing required. This paper aims to address this problem by identifying classical ML algorithms capable of pollen classifications with low computational requirements. We propose, a comprehensive comparative analysis, of multiple methods, using multiple methods for dimensionality reduction. We apply a comprehensive hyper-parameter search to find the best configuration for each method. Finally, model ensembling is used to create more robust classifications. The performance of all classifiers is tested using 4 pollen dataset both before and after the hyperparameter tuning process.

Keywords: pollen classification, Rapid-E, Decision Trees, Discriminant Analysis, Naive Bayes

1. Introduction

Pollen is one of the main causes of seasonal allergies in humans, according to [1]. In recent years, the number of diagnosed allergies has been steadily increasing, with the causes still a matter of debate. This has made the need for better pollen monitoring a more pressing matter. Much progress has been made in increasing the number of pollen monitoring stations without much work into making the systems fully automatic. The current standard operating procedure includes Hirst traps [2], which are a human-centric method of pollen monitoring. Requiring a lot of human interaction for sampling preparations, pollen counting, and further pollen grain species classification. This

[1] PhD Student, Faculty of Electronics, Telecommunications and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: mihai.boldeanu@upb.ro

[1] Professor, Faculty of Electronics, Telecommunications and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: corneliu.burileanu@upb.ro

[1] Assoc. Prof., Faculty of Electronics, Telecommunications and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: horia.cucu@upb.ro

[2] Senior Researcher, National Institute of Research and Development for Optoelectronics, INOE 2000, Măgurele, Romania

approach involves a waste of resources and cannot meet the need for more pollen monitoring stations.

Much work has gone into developing machine-learning models capable of classifying pollen in microscope images. This sort of approach requires a lot of processing power with graphical processing units (GPUs) usually involved to run effectively. In this work, we explore several classical machine learning approaches used for classification to find a more suitable approach for low-power/computational less intensive scenarios.

The constraint on the amount of processing done to classify pollen can be assessed from two main perspectives. Firstly, the vast amount of data that would have to be processed daily when classifying data from automatic particle analyzers, capable of capturing tens of thousands of samples per minute. Secondly, the impact on the climate that is created by using algorithms and methodologies that require power-hungry hardware such as GPUs.

These constraints limit the use of most computer vision models such as Conv-nets and visual transformers but allows the use of more simple algorithms generally used for classification.

In this work, we will focus on four types of algorithms that are very popular in a multitude of regression and classification tasks. The common thing for the selected algorithms is that they have been previously used for pollen classification and that they are efficiently implemented in the sklearn Python library [3]. The selected algorithms include Decision trees, Multi-Layer Perceptron, Naive Bayes, and Discriminant Analysis. These algorithms have varying degrees of complexity and require very different amounts of time and processing to train. The research was carried out in three stages and involved firstly, obtaining a baseline for these algorithms and, secondly, a hyper-parameter search to find the best configuration for each algorithm on the task of classification on data from an automatic particle analyzer, Rapid-E (http://www.plair.ch/Rapid-E.html, accessed 01.04.2022). Finally, model ensembling was used to apply the best hyper-parameter configuration on multiple models trained on sub-sets of data to give a final performance edge.

This paper is structured as follows. Section 2 describes the most popular approaches for pollen classification that rely on lightweight algorithms designed to run on CPU only. In Section 3, available data sets of pollen data from Rapid-E devices are presented alongside a few feature extraction/engineering steps required to use with our data. Section 4 presents the results of the best-performing models and the improvements obtained by doing a hyperparameter search and model ensembling. Finally, conclusions are drawn in Section 5 along with a discussion on further research steps.

2. Related Work

Pollen classification using computers was first proposed as early as 1968 by [4]. While the field of automatic pollen monitoring initially developed slowly due to the limitation of the hardware of the era in recent years we have seen a burst of methodologies applied to the task of pollen classification and monitoring.

Some notable examples include [5] that used discriminant analysis among other methods to do pollen classification. Bayesian classification methods were used by [6]. Others such as [7] use decision trees and random forests. MLP were used in [8], [5] and [9]. Finally, [6] used all of them to make similar classifications.

These methods were applied to datasets created from microscope images manually saved by trained personnel. While this type of approach gives the best performance on the dataset, it is not a real-world scalable solution, because it involves humans in the preparation of the microscope slides and image creation.

In this work we will focus on data obtained from an automatic particle analyzer, Rapid-E in [10] and [11]. While this type of data is more difficult for modern classification models to learn, have the advantage to be a fully automated data source. This type of data has been previously used for classification using convolutional neural networks in [10] and [12]. While the obtained results are very good, they rely on processing-heavy approaches and are not scalable to a large number of devices working in the wild without powerful workstations with GPUs.

This work plans to address this by applying smaller models that can be run on CPU-only devices.

3. Materials and Methods

In this section are analyzed the available pollen datasets from Rapid-E devices and the appropriate data processing techniques that can be utilized to facilitate classification. This includes an overview of the possible dimensionality reduction techniques that can be applied and also the algorithms used for classification.

3.1. Datasets

Dataset Name	**Number of Classes**	**Number of Samples**	**Minimum Number Samples per Class**
SAU-SRB [a]	14	85 k	985
SAU-LI [a]	11	399 k	16,114
SAU-CH [a]	10	50 k	1,075
MARS [b]	13	105 k	3,020

Table 1. Overview of available datasets from Rapid-E devices. [a] from [10]; [b] from [11].

Pollen Datasets from Rapid-E devices have been proposed in previous studies such as [10] and [11]. These data sets are representative of four countries from Europe: Serbia, Switzerland, Romania, and Lithuania. Data overview is presented in Table 1. The datasets have a different number of samples and classes. The classes present in each dataset are different because they reflect the pollen most common in those geographic regions.

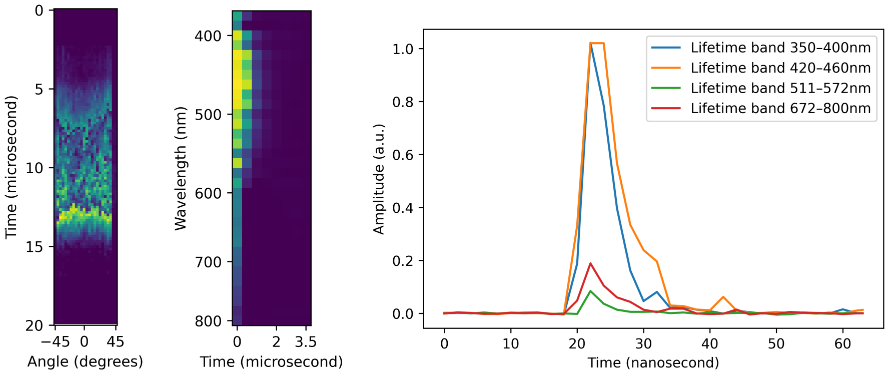

FIGURE 1. From left to right: Scattering image, Fluorescence Spectrum, Fluorescence Lifetime Signal.

All these datasets contain three types of features that describe aerosol particles. The scattering image, Figure 1 a, captures information about the size, shape, and morphology of the particle. The fluorescence spectrum provides information about the chemical makeup of the particle and the lifetime signal gives information about the relative abundance of those chemicals. Figure 1 shows the raw data from the Rapid-E device. The complex unstructured data has to be pre-processed before any classification attempt. While these types of features can be used directly by CNNs, in the case of using simpler models, data pre-processing has to be applied to be able to use such features.

3.2. Feature engineering steps

As seen in Figure 1 the complex and high dimensional data provided by a Rapid-E device needs several pre-processing steps. The features have the following dimensions 120 × 24, 32 × 8 and 4 × 64. Before any dimensionality reduction, the data is flattened. After this, each of the 3 feature vectors is ran through a dimensionality reduction algorithm and finally, the resulting vectors are concatenated.

Dimensionality reduction (DR), as the name suggests, is reducing the number of random variables using various mathematical methods from statistics and machine learning. DR is often used to reduce the problem of managing and manipulating large data sets. DR techniques generally use linear transformations in determining the intrinsic dimensionality of the manifold as well as extracting its principal directions. In this work, some different dimensionality reduction methods were tried including Principal component analysis (PCA), Independent component analysis (ICA), Gauss Random Projections (GRP), and Sparse Random Projections (SRP). All of the methods were applied with 5 levels of granularity for the resulting number of components. This was done to find the best representation for each of the algorithms used for classification.

PCA is a process for finding several orthogonal vectors bases, the principal components, that are used to create a new representation of data in a lower dimensional space while keeping as much of the variability of the original data. PCA is generally used as an exploratory tool on unknown datasets or as a tool for preprocessing high-dimensional data to better work with some predictive or classification models.

ICA is another approach used for representing high dimensional data but in this case, the goal is to decompose a multivariate signal into independent non-Gaussian signals. The independent components are found by maximizing the statistical independence of the estimated components. The two broadest definitions of independence for ICA are minimization of mutual information and maximization of non-Gaussianity.

Random projection is another dimensionality reduction technique. It is a simple and computationally efficient way to reduce the dimensionality of data by trading a controlled amount of error for faster processing and smaller models. In this work we looked at two flavors of random projections:

GRP reduces the dimensionality by projecting the original input space on a randomly generated matrix where components are drawn from a distribution.

SRP reduces the dimensionality by projecting the original input space using a sparse random matrix. Sparse random matrices are an alternative to dense Gaussian random projection matrices that guarantees similar embedding quality while being much more memory efficient and allowing faster computation of the projected data.

3.3. Selected algorithms

Because the main goal of this paper is to find the algorithms that perform the best with limited computing resources we will investigate the algorithms listed below.

Naive Bayes is a family of simple ”probabilistic classifiers” based on applying Bayes’ theorem with strong (naive) independence assumptions between the features. They are among the simplest Bayesian network models but coupled with kernel density estimation, they can achieve high accuracy levels. Naive Bayes classifiers are highly scalable, requiring several parameters linear in the number of variables (features/predictors) in a learning problem.

Discriminant analysis encompasses methods that can be used for both classification and dimensionality reduction. Linear discriminant analysis (LDA) is particularly popular because it is both a classifier and a dimensionality reduction technique. Quadratic discriminant analysis (QDA) is a variant of LDA that allows for the non-linear separation of data.

Decision tree learning or induction of decision trees is one of the predictive modeling approaches used in statistics, data mining, and machine learning. It uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

Multilayer Perceptron is a class of feedforward artificial neural network (ANN). The term MLP is used ambiguously, sometimes loosely to mean any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons (with threshold activation). Multilayer perceptrons are sometimes colloquially referred to as basic neural networks, especially when they have a single hidden layer. An MLP consists of at least three layers of nodes: an input layer, a hidden layer, and an output layer. This work will not use networks that rely on convolutions.

3.4. Hyper-parameter optimization

Hyper-parameters are algorithm parameters used to control the learning process.

For each of the algorithms presented in the previous sub-section, a search over the hyper-parameter space is done to find the best configuration for the task of pollen classification. The search is done by making a 3-fold validation of each hyper-parameter value to avoid problems caused by differences in datasets.

For the Gaussian Naive Bayes and Quadratic Discriminant Analysis present in the sklearn library only 1 hyper-parameter is available for GNB we have variable smoothing and for QDA we have the regularization parameter. This means that the hyper-parameter search is simple as we only change one value for each trial.

For the Decision Tree and MLP present in sklearn we have a rather large number of possible hyper-parameters to optimize and this requires a different approach. For DT we selected the following hp as a search space: the criterion used to split a batch of samples can be either gini or entropy, the splitter decision can be made using the best split or a random split, the maximum allowed depth to which a tree can grow was selected from 5, 10, 15, 20, and None, with None meaning the tree can grow as large as needed. Finally, the minimum number of samples required to allow a split in the tree branches allowed goes from the default value of 2 to 4 and 8.

In the case of the MLP algorithm, the number of hyper-parameters includes: the number of hidden layers and their size, the activation function used after each layer, the learning rate and a regularization parameter alpha are used to prevent overfitting.

3.5. Model Ensembles

A very powerful method for improving the performance of weak classifiers is the use of ensembling. This procedure works by using multiple weak learners to train on small subsets of data and then at inference time, they are used together to get a classification by averaging or polling all the separate classifications.

In the case of the decision trees, we have a special class of ensemble in the form of Random Forest which takes advantage of the way decision trees are built to train large numbers of them on subsets of data and subsets of features from the data. This decreases overfitting and increases overall performance. The main advantage of Random Forrest is that all the trees in a forest can be trained in parallel as they are independent classifiers.

For ensembling the other types of algorithms, more general approaches have to be used such as Bagging Classifiers which train multiple instances of a model on different subsets of data. This was applied to the GNB, QDA, and MLP.

4. Results and Discussions

The Results and Discussions section is divided into three parts. In the first part, the basic results for each of the DR methods, the number of components, and the classification algorithms are presented. In the next part are analyzed these results and the knowledge gained to perform a hyper-parametric search for the most an efficient method of dimensional reduction, and the number of components for each algorithm.

In the last part, the best-performing models are used to create ensembles that are further tested and compared.

4.1. Using Default Hyper-parameters as baseline

The baseline run was compiled by combining the results of 320 different experiments over 4 datasets, 4 dimensionality reduction methods, 5 levels of reduction, and 4 classification algorithms. The results of these experiments are presented in Table 2.

The columns in Table 2 are the dimensionality reduction approach used and the number of components, while the rows are the data sets and the performance of each classification model type.

To have a better view of the impact of each of the variables in the baseline run Figure 2 shows a breakdown of accuracy and time required for training the different models split along different parameters. While in general cases a

	D.R.	PCA					ICA					GRP					SRP				
DS	Algo	10	20	50	100	200	10	20	50	100	200	10	20	50	100	200	10	20	50	100	200
RO	DT	34	36	35	36	34	34	33	31	32	29	27	31	32	34	34	29	29	31	32	34
	MLP	57	60	62	64	61	58	60	62	63	59	49	57	60	61	59	49	56	60	60	60
	GNB	32	29	27	27	25	30	31	30	29	29	27	26	26	27	26	25	24	28	28	27
	QDA	41	42	43	43	41	41	42	43	43	41	36	39	39	41	41	36	38	39	41	41
SR	DT	46	47	46	46	46	44	43	44	43	40	39	43	43	45	46	42	43	45	45	46
	MLP	66	68	68	66	60	66	70	73	74	76	58	65	69	71	71	59	65	69	70	71
	GNB	35	39	38	37	35	33	33	29	33	33	27	30	29	30	30	29	28	29	30	30
	QDA	34	34	38	44	47	38	40	42	43	44	42	43	43	41	41	44	42	42	42	42
LI	DT	44	45	46	46	46	43	45	43	42	36	36	41	42	44	46	39	41	44	45	46
	MLP	65	70	73	74	77	67	68	67	64	58	58	64	67	65	65	60	64	66	67	64
	GNB	32	35	33	30	30	38	41	40	38	34	34	37	38	39	39	37	36	38	39	38
	QDA	38	40	42	43	44	34	34	38	44	47	46	47	44	43	45	47	45	44	43	45
CH	DT	54	53	53	56	57	51	49	45	48	49	41	46	48	47	50	45	47	49	51	52
	MLP	70	71	67	68	74	71	70	67	70	74	60	64	65	66	66	61	64	67	66	68
	GNB	55	53	46	47	49	55	55	49	48	51	46	47	45	46	46	46	44	43	47	45
	QDA	60	60	55	52	53	60	60	55	52	53	52	52	50	50	50	52	50	50	50	53

Table 2. Baseline experiments split by dimensionality reduction method, number of components, and the algorithm used for classification applied on all of the publicly available data-sets.

model once trained is just used for inference there is the option of continuously training the model and in such scenarios a small train time is advantageous.

The initial baseline experiments indicate that PCA as dimensionality reduction brings the best results. The number of components shows an increase in accuracy with an increase in the number of components for DT and MLP up to 100 components with a little gain after that but a major increase in training time. For the simple algorithms such as QDA and GNB, we find the best option for the number of components as 20 and 10 respectively, with the algorithm struggling with dimensionality higher than that.

When comparing the algorithms over the entire range of experiments we see that MLP has the best performance but with the highest train time associated. While the DT and QDA have similar performance but with drastically different times required for training. The GNB has the lowest train time but also the lowest accuracy. This learner could be useful in ensembles but not on its own.

The results of the baseline run can be used as a starting point for the hyper-parameter search and allow us to have a good metric to compare against to see that we are improving classification.

4.2. Using Best Hyper-parameters

When doing the hyper-parameter search we have to find the best performance over all 4 datasets for a specific algorithm. This means that good visualizations for the hyper-param search results have to be employed. In the case of GNB and QDA where only one configurable parameter exists and the training time is really fast, the search is quite easy. The entire search space can be traversed and the optimum found can be seen in Figure 3 for GNB and Figure 4 for QDA.

For GNB the value selected for variable smoothing does not appear to matter that much after a certain threshold. The selected value was 0.01 because it was the region with good results on all datasets.

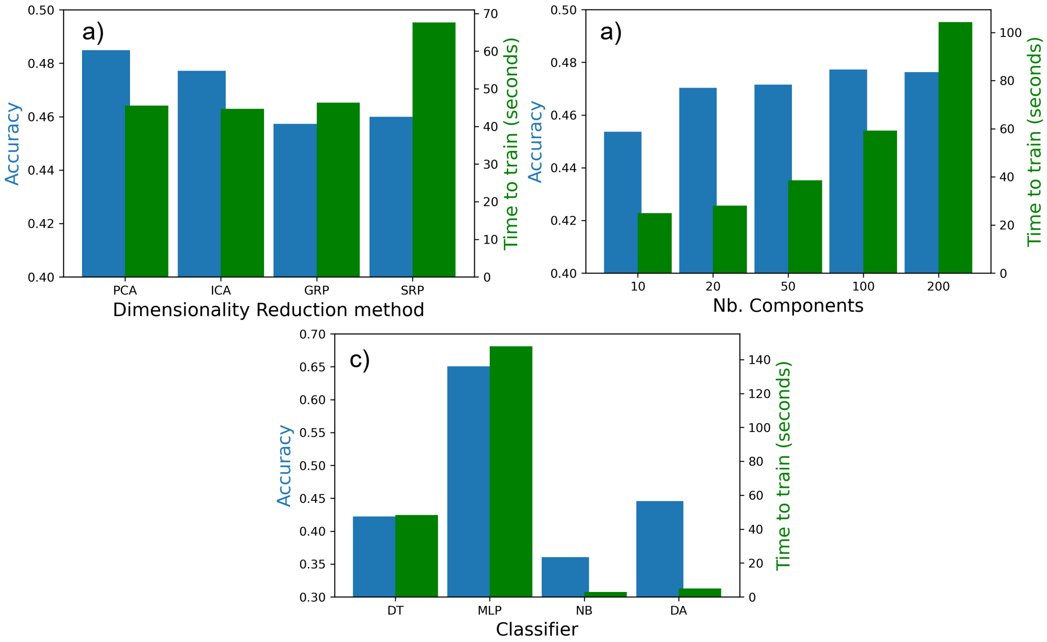

Figure 2. Baseline experiments aggregated by a) dim.red. method, b) number components for dim.red. and c) the type of classfier.

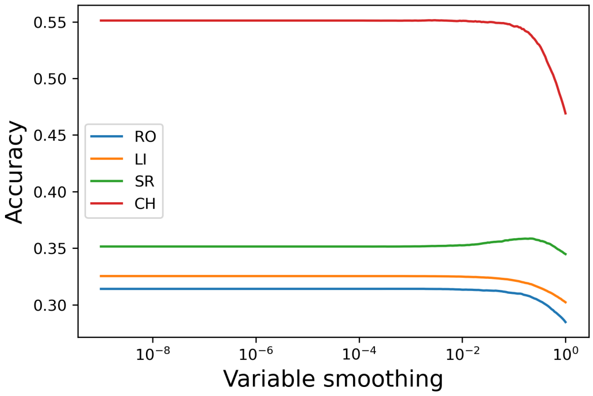

Figure 3. Classification accuracy versus smoothing parameter for GNB model. Each line is representative of one dataset.

The QDA model had an obvious peak in performance on all datasets. The value selected for the regularization parameter was 0.35, as this was the value that gave the best results on all datasets. The peak accuracy was obtained at different values of the regularization parameter for each dataset, as can be seen in Figure 4.

When investigating the results of the hyper-param search for DT multiple graphics are required to get an overall picture of what parameters give the

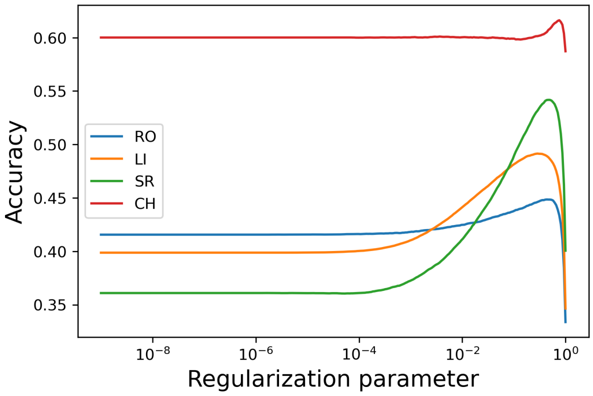

FIGURE 4. Classification accuracy versus smoothing parameter for QDA model. Each line is representative of one dataset.

greatest increase. In Figure 5 we can see that the criterion used for selection has a minimal effect on performance, and the minimum number of samples, when making a split, has a moderate effect. The most powerful impact is made by the maximum depth of the tree and the splitting policy. The depth of the tree can cause overfitting when the tree becomes too large and can underfit when it is too small. The splitting policy can add some overfitting but it improves performance.

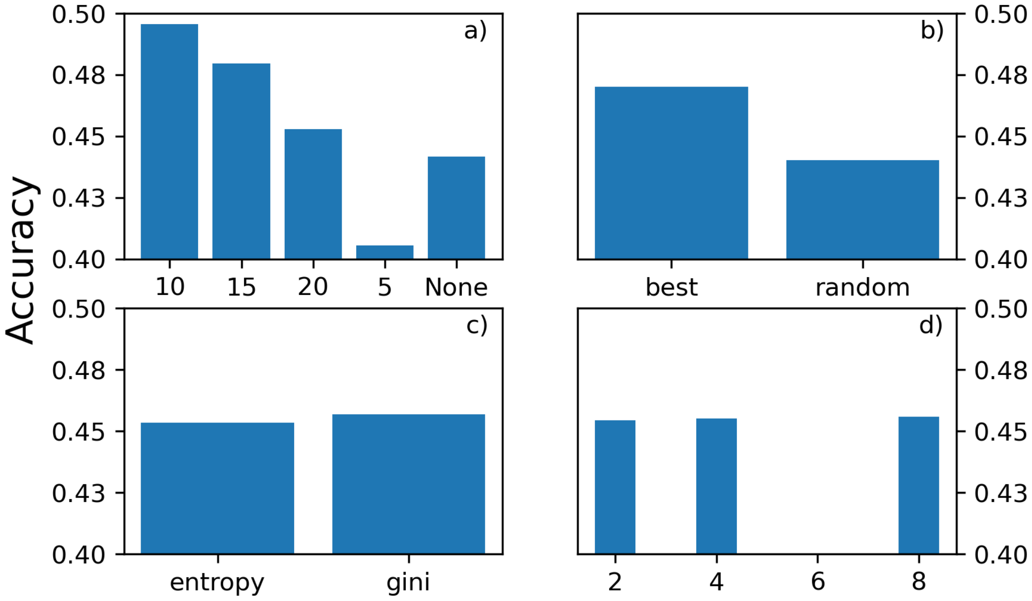

FIGURE 5. Classification accuracy versus a) tree depth, b) data split, c) criterion and d) data points per leaf for DT model.

For the MLP the hyper-parameter search was the most difficult because of the higher base train time of this type of model and a large number of hyper-parameters available for tuning. We focused on just a handful of parameters to have a similar number of experiments as for the other model types.

In Figure 6-a the impact of the activation function shows that the best performing one is the Rectified Linear Unit; Figure 6-b gives us a range between 1E-3 and 1E-4 for the best performing learning rate; Figure 6-c shows that the regularization strength is not very important for this type of model/task;

Finally, we looked at the number of hidden layers and their size in 6-d. We looked in 2 directions at the number of neurons per layer comparing the result of 10, 50, and 100 neurons, and we found the best result on the 50-neuron version, this allows us to make a smaller model than using the baseline 100-neuron model. We also compared the number of layers with network with 1, 2, and 3 hidden layers all of size 10, here we found that increasing the number of layers had a detrimental effect on classification.

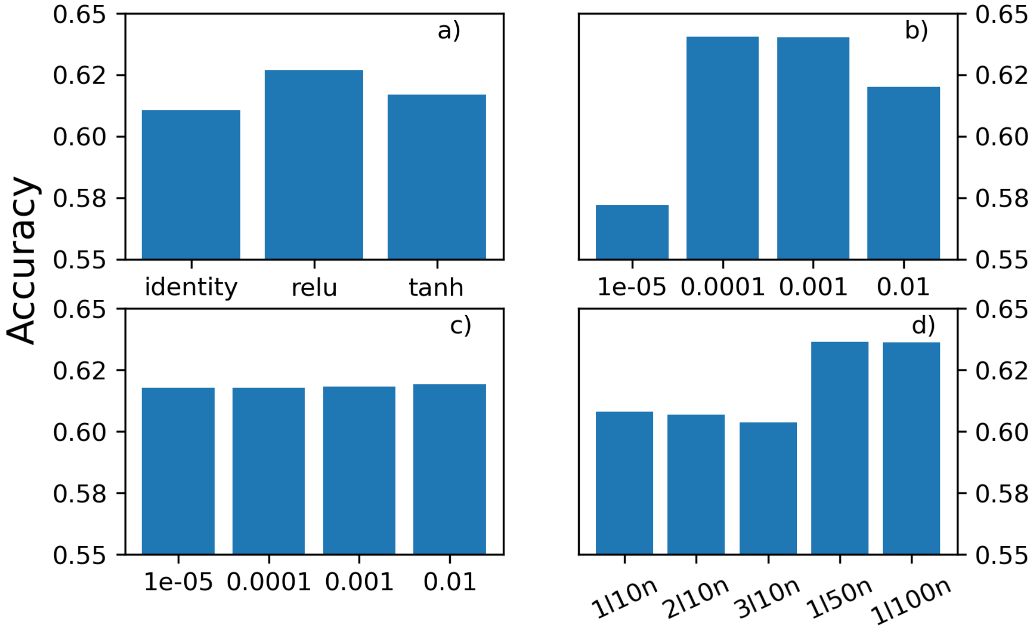

Figure 6. Classification accuracy versus a) activation functions, b) learning rate, c) regularization strength and d) number layer/number neurons for MLP model.

4.3. Using Model Ensembling with Best Hyper-parameters

A final performance boost that can be applied to classical machine learning algorithms is the use of ensembling. Because the training time for such models is not too long, multiple models can be trained in parallel on different subsets of the data. This creates a collection of models that act as a more powerful classifier.

The models used in the ensemble were all built using the insight gained after the hyperparameter search. The case of ensembling the Decision Tree was treated separately because there exists a method for building ensembles, Random Forest, that relies on the trees being non-identical.

Algo	Config/Dataset	RO	LI	SRB	CH
GNB	Best Conf.	31	32	34	54
	Ensemble	31	32	35	54
QDA	Best Conf.	44	48	53	60
	Ensemble	44	48	53	61
DT	Best Conf.	42	51	52	63
	Ensemble	47	56	57	69
	Rand. Forest	42	55	54	63
MLP	Best Conf.	63	71	67	73
	Ensemble	**66**	**74**	**70**	**77**
CNNs	[12]	81	84	80	87

TABLE 3. Classification accuracy for all models in best configuration and as an ensemble of models. Final row is SOTA using convolutionary neural networks

In Table 3 the final results of the best models and best results of ensembling are presented on all data sets. The The main takeaway is that while some algorithms benefit from ensembling and hyper-parameter searches some simple ones don't get much improvement.

The Gaussian Naive Bayes and Quadratic Discriminant analysis did not see any gain after ensembling. This might indicate that the models in the ensemble, all learned the same thing even if trained on different subsets of the data.

On the other hand, Decision trees had a significant improvement over just one model in its best configuration.

5. Conclusions

In this work, we investigate the use of low computational power machine learning models on the task of pollen classification using data from automatic particle monitors.

We compare several machine learning models on the task of pollen classification using data sets obtained on Rapid-E devices from 4 European countries.

We experiment with multiple dimensionality reduction approaches and a different number of components, to find the best-performing combination. This is used to create a baseline performance for all datasets. The classification accuracy varies between datasets because they contain different classes or species of pollen. The types of pollen can have a different degree of similarity between them. The average performance across all datasets is 35% for the NB, 42% for the DT, 44% for the DA, and 65% for the MLP.

For each of the models, we do an extensive hyper-parameter search to find the best configuration. This gets us an improvement over the baseline with the average classification accuracy across all datasets of 38% for the NB, 52% for the DT, 51% for the DA, and 69% for the MLP.

Finally, we use ensembles to get a final improvement in classification accuracy. This final step gives a performance boost to some model types with an average classification accuracy across datasets of 57% for the DT and 72% for the MLP.

The accuracy results are lower than what can be obtained by using CNNs as in [12], but with our best performer, the MLP ensemble, we obtain results comparable to that of humans, in the task of pollen classification.

While this paper shows that classical ML approaches still hold an important place in the automatic pollen classification landscape future work is required to create high performance and scalable pollen monitoring systems. Future work will focus on finding better dimensionality reduction approaches and better feature engineering that might yield significant improvements over this baseline.

Acknowledgements

This work was supported by the Romanian National Core Program Contract No.18N/2019.

REFERENCES

[1] G J Burbach, L M Heinzerling, G Edenharter, C Bachert, C Bindslev-Jensen, S Bonini, J Bousquet, L Bousquet-Rouanet, P J Bousquet, M Bresciani, A Bruno, G W Canonica, U Darsow, P Demoly, S Durham, W J Fokkens, S Giavi, M Gjomarkaj, C Gramiccioni, T Haahtela, M L Kowalski, P Magyar, G Muraközi, M Orosz, N G Papadopoulos, C Röhnelt, G Stingl, A Todo-Bom, E von Mutius, A Wiesner, S Wöhrl, and T Zuberbier. Ga(2)len skin test study ii: clinical relevance of inhalant allergen sensitizations in europe. *Allergy: European Journal of Allergy and Clinical Immunology*, 64(10):1507–15, October 2009.

[2] Simon Adamov, Natalie Lemonis, Bernard Clot, Benoît Crouzy, Regula Gehrig, Marie-José Graber, Christine Sallin, and Fiona Tummon. On the measurement uncertainty of hirst-type volumetric pollen and spore samplers. *Aerobiologia*, 09 2021.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[4] J.R. Flenley. The problem of pollen recognition. *Clowes MB, Penny JP, eds. Problems of picture interpretation*, page 141–145, 1968.

[5] P Li, WJ Treloar, JR Flenley, and L Empson. Towards automation of palynology 2: the use of texture measures and neural network analysis for automated identification

of optical images of pollen grains. *Journal of Quaternary Science: Published for the Quaternary Research Association*, 19(8):755–762, 2004.

[6] Magdalena K. Sobol and Sarah A. and Finkelstein. Predictive pollen-based biome modeling using machine learning. *PLOS ONE*, 13(8):1–29, 08 2018.

[7] Agnieszka Kubik-Komar, Elżbieta Kubera, Krystyna Piotrowska-Weryszko, and Elzbieta Weryszko-Chmielewska. Application of decision tree algorithms for discriminating among woody plant taxa based on the seasonal characteristics of pollen. *Archives of Biological Sciences*, 67:89–89, 01 2015.

[8] I France, AWG Duller, GAT Duller, and HF Lamb. A new approach to automated pollen analysis. *Quaternary Science Reviews*, 19(6):537–546, 2000.

[9] Y Zhang, DW Fountain, RM Hodgson, JR Flenley, and S Gunetileke. Towards automation of palynology 3: pollen pattern recognition using gabor transforms and digital moments. *Journal of Quaternary Science: Published for the Quaternary Research Association*, 19(8):763–768, 2004.

[10] I. Šaulienė, L. Šukienė, G. Daunys, G. Valiulis, L. Vaitkevičius, P. Matavulj, S. Brdar, M. Panic, B. Sikoparija, B. Clot, B. Crouzy, and M. Sofiev. Automatic pollen recognition with the rapid-e particle counter: the first-level procedure, experience and next steps. *Atmospheric Measurement Techniques*, 12(6):3435–3452, 2019.

[11] Mihai Boldeanu, Cristina Marin, Dragos Ene, Luminiţa Marmureanu, Horia Cucu, and Corneliu Burileanu. Mars: the first romanian pollen dataset using a rapid-e particle analyzer. In *2021 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pages 145–150, 2021.

[12] Mihai Boldeanu, Horia Cucu, Corneliu Burileanu, and Luminița Mărmureanu. Multi-input convolutional neural networks for automatic pollen classification. *Applied Sciences*, 11(24), 2021.