

# Iterative Update of a Random Forest Classifier for Diabetic Retinopathy

Jordi PASCUAL-FONTANILLES <sup>a,1</sup>, Aida VALLS <sup>a</sup>, Antonio MORENO <sup>a</sup> and Pedro ROMERO-AROCA <sup>b</sup>

<sup>a</sup> *ITAKA-Intelligent Technologies for Advanced Knowledge Acquisition Dept. d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili Avda. Paisos Catalans, 26, 43007, Tarragona, Spain*

<sup>b</sup> *Servei d'Oftalmologia, Hospital Universitari Sant Joan de Reus, Institut d'Investigació Sanitària Pere Virgili (IISPV), Universitat Rovira i Virgili, Tarragona, Spain*

**Abstract.** Random Forests are well-known Machine Learning classification mechanisms based on a collection of decision trees. In the last years, they have been applied to assess the risk of diabetic patients to develop Diabetic Retinopathy. The results have been good, despite the unbalance of data between classes and the inherent ambiguity of the problem (patients with similar data may belong to different classes). In this work we propose a new iterative method to update the set of trees in the Random Forest by considering trees generated from the data of the new patients that are visited in the medical centre. With this method, it has been possible to improve the results obtained with standard Random Forests.

**Keywords.** Random Forest, Dynamic learning models, Health Care, Diabetic Retinopathy.

## 1. Introduction

Building decision support systems for medical diagnosis is hard. In the last decade it has become common to use classifiers based on Machine Learning for this task. Usually these systems face a trade-off between sensitivity and specificity.

In this paper we address the problem of Diabetic Retinopathy (DR) classification. As a consequence of diabetes, the blood vessels of the eye may break and generate small blood spots, hemorrhages and exudates. These lesions produce vision loss and may even cause blindness if they are not detected and treated at an early stage. The risk of developing DR can be calculated from some clinical data of the patient, including blood analysis results. Our goal is to improve a DR classification model based on a Fuzzy Random Forest used in the Retiprogram system [1][2]. This model is being tested by a group of ophthalmologists at Hospital Sant Joan de Reus. The general results are good (with a sensitivity

---

<sup>1</sup>Corresponding Author: Jordi Pascual-Fontanilles. E-mail: jordi.pascual@urv.cat

and a specificity over 75%), but there are still many miss-classifications. Errors are mainly due to the inherent ambiguity of the training examples (very similar patients can belong to different classes) and to the high unbalance between both classes (more than 90% of diabetic patients do not develop DR).

We propose a novel method to take advantage of the data of the new patients which are treated at the hospital. When enough data is gathered, these new examples will be used to perform an update on the random forest model, with the aim of improving its performance. The method also reuses the training examples that the random forest is not able to classify correctly. Even though it has been tested on a fuzzy random forest, the method is suited to be used with standard random forests.

To obtain some experimental results of the proposed method, we used an extended version of the original dataset used in [2]. Data is split in different sets to simulate the arrival of new data to the hospital. A weighted balanced accuracy is used as the performance metric. Several tests have been performed to check the performance of the model during the multiple iterations of the updating method. A study of the best parameters has also been made.

The rest of the paper is organized as follows. Section 2 presents other approaches used to update random forests, consisting on adding weights to the decision trees, or on building trees dynamically from streaming data. In Section 3, we introduce the proposed method for iterative update of the set of trees. In Section 4, we present the dataset and we discuss the obtained experimental results. Finally, Section 5 presents the conclusions and the future work.

## **2. Related work**

The dynamic improvement of classification models based on Random Forests has been studied in the literature. Two different approaches can be found: adding weights to the RF or building online RFs.

Concerning the former approach, weights may be added in four different ways. The first one consists of adding weights at the last step of the RF classifier, when a voting procedure is made to find the majority class ([3], [4], [5]). Each tree on the ensemble has a weight which corresponds to its accuracy. The accuracy is obtained by obtaining the performance of the tree on their out-of-bag samples.

There are also some variations of these methods. For instance, Dogan and Birant [6] proposed initializing all the weights to the same value, and reward the best performing trees on the validation set formed by the out-of-bag samples. Zhukov et al. [7] added a pruning step to replace the worst decision tree so the ensemble can handle concept drift. Decision trees also have a sliding window of stored samples. Each time a new sample has to be evaluated, similar samples from the sliding window are used to recompute the weights based on their errors.

The second option consists on weighting the samples. Kim et al. [8] proposed weighting the samples according to the complexity of classifying them correctly. The trees also have weights, which are computed using the ones on the samples already classified. Yang and Yin [9] considered finding the weight of the decision trees as an optimization problem. For a determined number of epochs, both the weights on the samples and the decision trees are updated to optimize the model.

The third possibility, proposed by Zhong et al. [10], assigns weights to the leaves of the decision trees. That is, each of the rules of the ensemble has a weight based on some performance metric. For regression problems, this method obtains better results than just weighting the decision trees. Finally, there are weighting methods that use different weights for each of the possible output classes. For instance, Zhu et al. [11], Livieris et al. [12] and Utkin et al. [13] use this approach to compensate unbalanced datasets.

The second kind of methods for the dynamic improvement of RFs are based on online random forests. In this case, the training data arrives and is processed in an online manner, that is, in streams. Because the new training samples are not available from the beginning, the methods are not focused on improving an existing model, but on adapting the current one.

Incremental decision trees are one type of online learning methods. For instance, Kalles and Morris [14] and Utgoff et al. [15] proposed variants of ID3 which are incremental. Pecori et al. [16] also proposed using Very Fast Decision Trees to train an ensemble of incremental decision trees, using streaming data. Safari et al. [17] combined online bagging techniques with extremely randomized forests to build an online random forest. The trees on this random forest grow when new data is fed into the model. A new branch on the tree is created when enough samples are on a node, and are good enough to classify new samples. Similar proposals of online random forests include Mondrian Forests [18] and Adaptive Random Forests [19]. They are also based on growing the trees with the arrival of new training samples. Some incremental approaches also drop some members in the ensemble, and create new ones. Although their objective is to handle concept drifts, and their focus is on processing streams of data.

These two approaches to the dynamic construction of random forests are very different. On the one hand, the weighting methods are applied during the training of the model. They do not need new data because they use the out-of-bag training samples. These approaches are able to improve the results of a standard random forest. Updating of the weights using new data is rarely performed as the effect of a few new incoming samples may not be appropriate. Moreover, the core of the model is not changed, that is, no new rules are learnt.

On the other hand, the main drawback of online random forests is that they need more data than standard random forests to achieve a similar performance. They are not designed to update and improve an existing model, but to construct it in time. They are well suited for applications which have to process continuous streams of data.

### **3. Proposed method**

This paper proposes a novel approach. Instead of modifying the weights of the components of the model or to incrementally build the trees, we propose to change the set of trees that compose the random forest model. This change will be done after collecting a sufficiently large set of new cases that can be used as examples for improving the model. Therefore, it can not be considered a method for streaming data, as some of the works described in the previous section. The proposed architecture is illustrated in Fig. 1. It is composed of three steps.

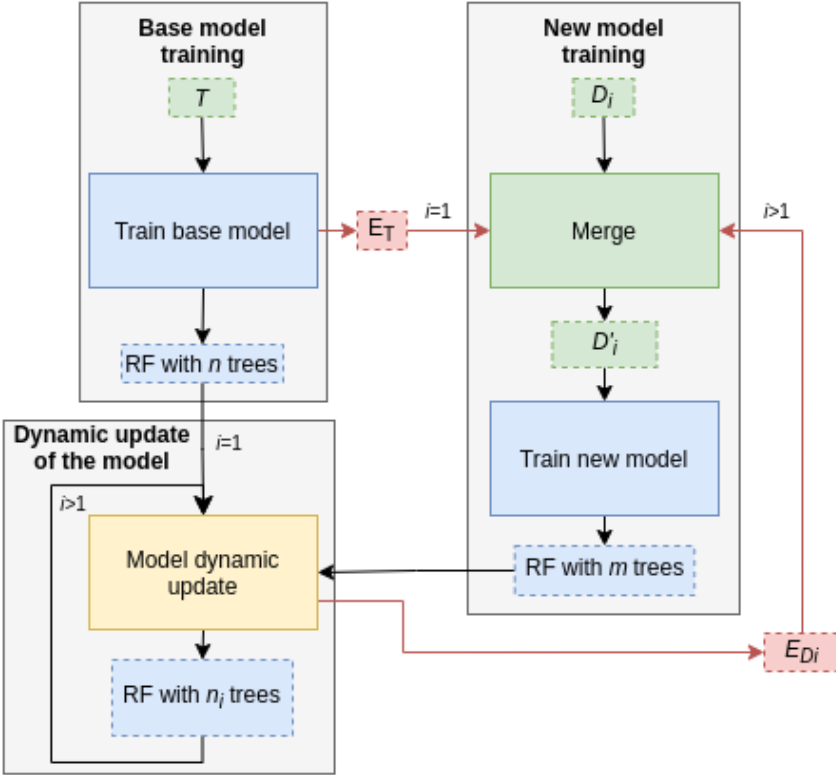


Figure 1. Architecture of the iterative learning of Random forests

1. **Base model training:** The first stage consists on training the base model with a large training dataset,  $T$ , which contains labelled examples. With a learning algorithm for Random Forests, we obtain  $n$  decision trees, where  $n$  is a large number, usually more than 100. During the construction process, the out-of-bag samples are used to compute two metrics for each tree, the specificity and the sensitivity, which are stored on each of the trees. The obtained classification model should be validated with a testing dataset in order to ensure its good performance (this stage is not shown in Fig. 1). Optionally, after creating the base random forest, the training dataset can be used for testing and the samples that are not correctly classified are stored in a file  $E_T$ .
2. **New model training:** Every time enough new patients data  $D_i$  have been gathered, a new training iteration  $i$  is performed. First, the dataset  $D'_i$  is generated. For the first training iteration  $i = 1$ , optionally, the  $D_i$  samples and the  $E_T$  errors can be merged in a single dataset. With the  $D'_i$  samples we train a new random forest but with a low number of trees, because the size of the new training set is small. We get  $m$  new decision trees, with  $m \lll n$ . Their out-of-bag samples are also used to compute the aforementioned metrics for each of these new trees.
3. **Dynamic update:** The base model is updated in this step.

The  $m$  new decision trees trained from the  $D'_i$  samples are added to the base model. From the  $n + m$  trees of the new model, the worst ones are removed. The number of decision trees to remove is fixed to a certain percentage  $p$ . The quality metric used to sort the trees is the balanced accuracy, which is defined as an average between specificity and sensitivity, with a weighting factor  $\alpha$ .

$$BA = \alpha \cdot sensitivity + (1 - \alpha) \cdot specificity \quad (1)$$

The resulting random forest with  $n_i$  trees is taken as the new model to be used by the clinicians until a new set of cases is available, and a new iteration starts. Optionally, the errors of the updated random forest model on the  $D'_i$  dataset may be also retrieved and stored in  $E_{D_i}$ . If so, in the next iteration, the new samples  $D_i$  are merged with those error cases  $E_{D_i}$  in order to enlarge the training dataset of the subsequent iteration.

The use of the sets of wrongly classified examples  $E_x$  is optional. The merging of these error examples with the new ones has two purposes. On the one hand, to increase the size of the training set  $D'_i$  and, on the other hand, to show again this wrongly classified cases to the learning model in order to be able to build new rules that cover them appropriately. In that way, the model is learning from the past errors. In the next section, the effect of using errors will be studied.

#### 4. Experiments

The diabetic retinopathy risk detection problem has been used to test the proposed iterative method for updating a Random Forest. It is a binary classification problem with two labels. DR=1 means a high risk of suffering from diabetic retinopathy (i.e. positive class), whereas DR=0 means a low risk (i.e. negative class). The experiments have been performed using real data from a total of 25912 diabetic patients. This data includes 9 different attributes, 6 numerical and 3 categorical. The target attribute is the label of the class DR=0 or DR=1.

The data is split in three different datasets: training, validation and testing. The training dataset,  $T$ , is used to train the base RF model. It is used to create the model with the largest number of trees (100 trees), hence, it is the dataset with more samples. The validation set is used to simulate the new data that would arrive from the diabetic patients from time to time. We split the validation set in chunks of 800 samples. Each of them is used in a different iteration  $i$  during the dynamic updating process. From each of this smaller new training data sets,  $D_i$ , the system generates 20 new trees. Then, the dynamic updating stage is done, obtaining the Random Forest model  $RF_i$ . Finally, the testing set is used after each iteration to check the performance of the new random forest  $RF_i$ . Note that the samples from this testing dataset are not included in the error sets, thus, the model is never trained using these samples.

Table 1 shows the splitting of the data among the three datasets. It can be seen that the datasets are highly unbalanced towards patients without diabetic retinopathy.

**Table 1.** Diabetic retinopathy patients data

Dataset	Training	Validation	Testing	Total
<i>DR=0 samples</i>	12885 (90%)	6379 (80%)	2514 (70%)	21778
<i>DR=1 samples</i>	1431 (10%)	1621 (20%)	1082 (30%)	4134
<i>Total samples</i>	14316	8000	3596	25912

To obtain the experimental results, we used the aforementioned datasets to build a fuzzy random forest. In this case, the rules use fuzzy variables, which have been defined from the numerical attributes [1]. The fact that the rules are fuzzy does not introduce any change in the methodology for updating the RF model. The training algorithm for fuzzy RF is explained in [2].

The aim of the DR classifier is to improve the detection of patients with high risk of developing diabetic retinopathy, that is, improve the sensitivity of the random forest. For this particular application, it is preferred to misclassify non-DR patients as having the disease (FP), than the other way around (FN). This is due to the very bad consequences of not detecting DR on time, which produces a degradation of the vision that may even cause total blindness.

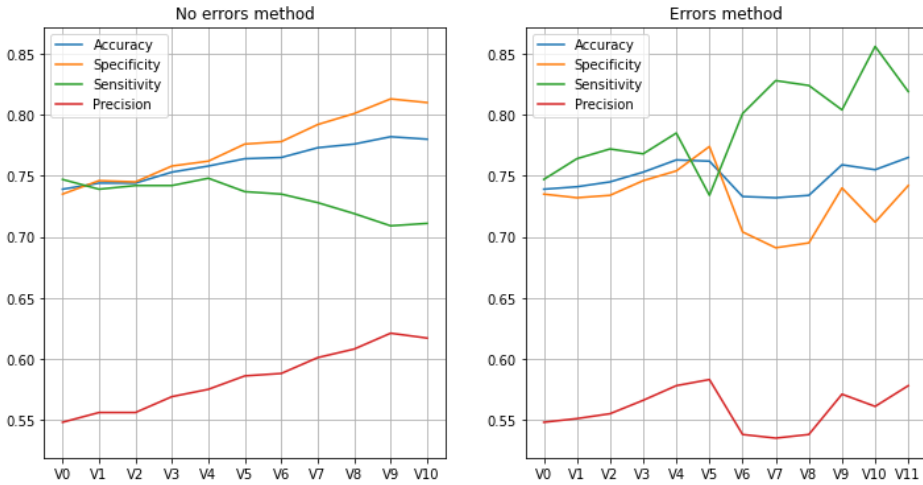
The method has two parameters: the percentage of trees changed at each iteration,  $p$ , and the balancing factor in the calculation of the accuracy,  $\alpha$ . We have tested different values of  $p$  from 5% to 15%. After some experiments, the percentage was fixed to  $p = 10\%$ . A lower  $p$  value did not produce significant changes on the random forest model. With a higher  $p$ , the random forest suffered too many changes and was highly unstable. To determine the  $\alpha$  value, diverse tests were performed with  $\alpha$  from 0.5 to 0.7. Values higher than 0.5 were selected in order to prioritize the sensitivity over the specificity. We have also compared the performance of the updating methodology when using the misclassified patients (errors in the training datasets) and without using those patients' data (no errors).

The obtained results are presented in Fig. 2, Fig. 3 and Fig. 4. Note that the errors method execution performs one iteration more than the no errors one. This is due to the additional samples obtained from the errors, which permit to create one more training set.

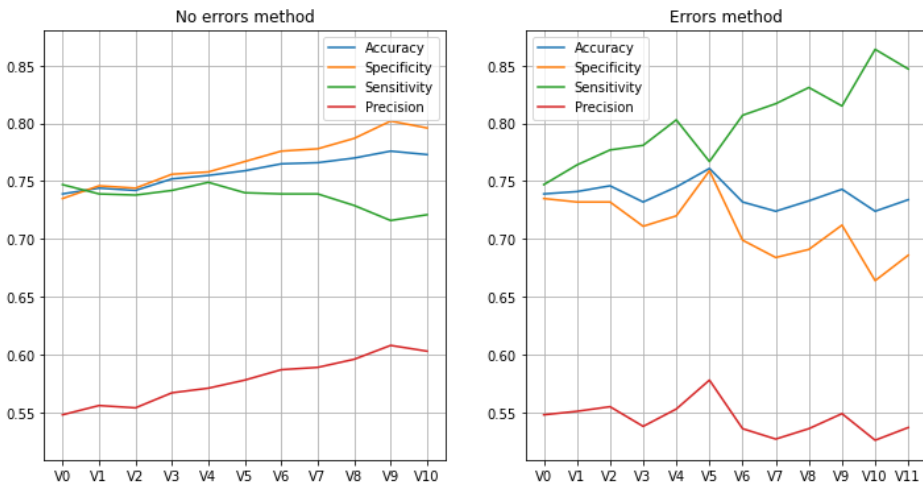
In these 3 figures, we can see similar behaviour on the errors and no errors methods when changing  $\alpha$ . The method that does not use errors slightly increases all the performance metrics except for the sensitivity, which slightly decreases. Even though this behaviour could be desired in some applications, in our case, it is critical to improve the sensitivity, as said before.

In contrast, the method that uses the errors data is able to significantly increase the sensitivity. The specificity decreases in this case, but, taking into account the high unbalance between classes, in proportion it does not decrease as much as sensitivity increases. Moreover, the accuracy of the model remains close to 0.75 during all the iterations. For this reason, the approach including data from miss-classified patients is the best suited for this application.

Comparing the three results, the value  $\alpha = 2/3$  gives us the best performance. With  $\alpha = 0.6$ , the sensitivity is a bit unstable and takes more time to increase up to 0.8, compared with using  $\alpha = 2/3$ . If more weight is given to sensitivity with  $\alpha = 0.7$ , then specificity decreases too much. So a weight of  $2/3$  for sensitivity and  $1/3$  for specificity seems a good trade-off for this application.



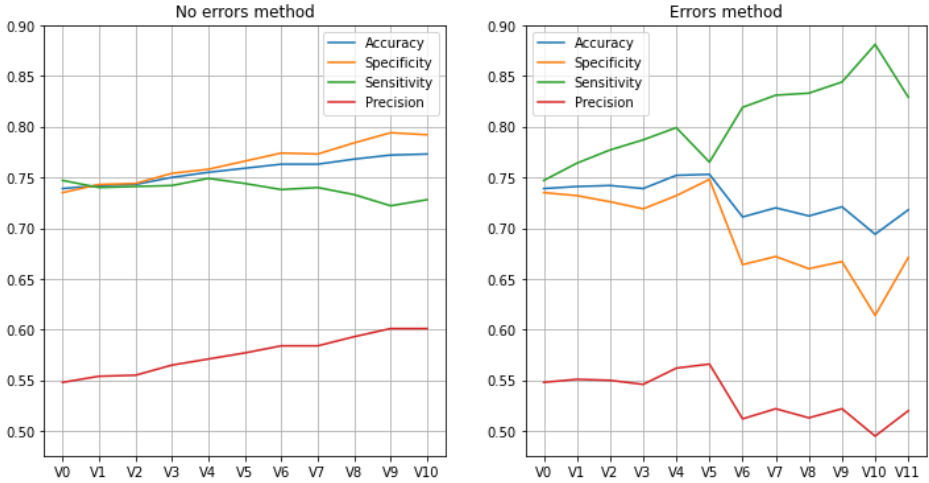
**Figure 2.** Evolution of the metrics.  $p = 10\%$ ,  $\alpha = 0.6$



**Figure 3.** Evolution of the metrics.  $p = 10\%$ ,  $\alpha = 2/3$

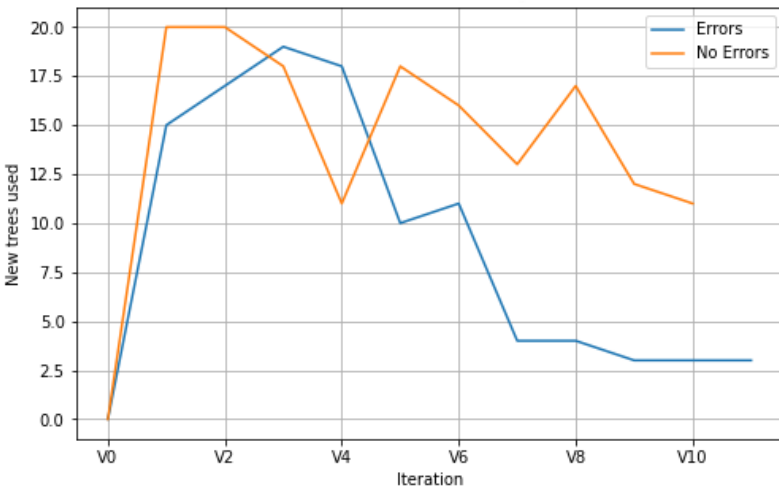
The behaviour of the selected parameters ( $p = 10\%$  and  $\alpha = 2/3$ ) has been further analysed. Fig. 5 shows the number of new generated trees incorporated to the random forest at each iteration. After a few iterations in which the number is quite high, it decreases. It can be seen that the decrease is faster when using the errors data. Reusing the samples in which the random forest is making wrong classifications contributes to a faster learning of a better model.

In Fig. 6, we represent the values in the confusion matrix. In the no errors method, the random forest is able to improve the true negatives and false positives detection rate. Besides, the true positives and false negatives detection rate slightly decrease. This evolution was expected after analysing Fig. 3. In this case, the updated model is more biased towards classifying in class 0 rather than class



**Figure 4.** Evolution of the metrics.  $p = 10\%$ ,  $\alpha = 0.7$

1. For this reason, it improves the results on detecting the patients without DR (specificity), but slightly decreases the detection of DR patients (sensitivity). In the errors method case, the true positives and false negatives are the ones which are improved on the updated random forest. Even though the negative class detection decreases, considering the high class unbalance, this loss in the negative class is not as important as the gains obtained in the positive class.



**Figure 5.** Evolution of new trees used.  $p = 10\%$ ,  $\alpha = 2/3$



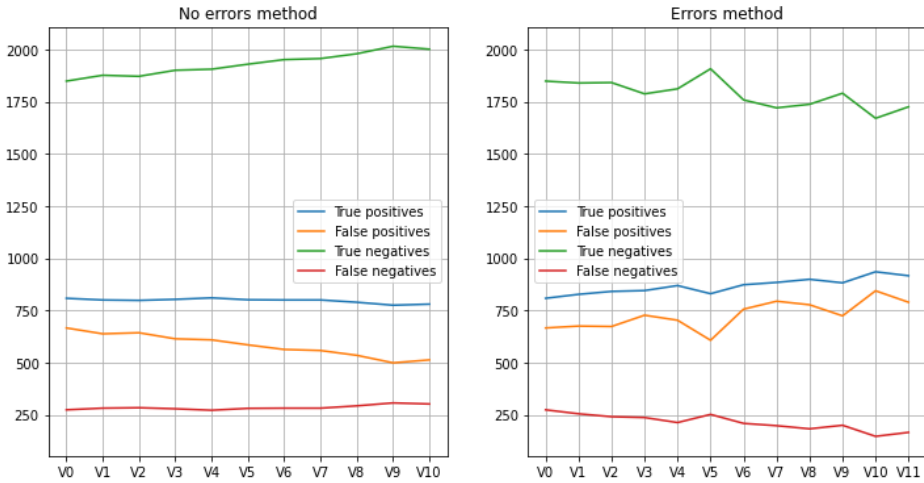


Figure 6. Evolution of the confusion matrix.  $p = 10\%$ ,  $\alpha = 2/3$

### 5. Conclusions and future work

A clinical decision support system for the assessment of diabetic retinopathy risk can help to avoid unnecessary screenings on patients and to reduce the workload of the ophthalmologists. The available resources can also be distributed among the patients, focusing on the ones that really need them.

The method presented in the paper is able to update a random forest in an iterative manner. Given an ill patient, a false negative is more dangerous than a false positive, because the patient would be considered healthy and would not receive the appropriate treatment. Thus, from the two variations of the proposed method, the one using miss-classified samples is the preferred for our use case. This method has been proven to improve the detection of DR=1 patients on a dataset with real data. Even though the detection of DR=0 patients does not improve, given the high unbalance between classes and the inherent ambiguity of this problem, the method seems suitable to improve a random forest model when new data from the patients are available.

As future work, we plan to test the proposed method on other domains. We want to study the possibility of adding weights to the trees. We would like to consider the update of weights of the random forest during the updating process of the model, taking advantage of the new data collected at each iteration. We would also want to study the modifications done to the ensemble, and how they affect the results.

### Acknowledgements

This work has been funded by the research project PI18/00169 from Instituto de Salud Carlos III & FEDER funds. The University Rovira i Virgili also supports this work with project 2019PFR-B2-61. The first author has a pre-doctoral FI

grant (2021 FLB 00139) from Secretaria d'Universitats i Recerca de la Generalitat de Catalunya i del Fons Social Europeu.

## References

- [1] Romero-Aroca P, Valls A, Moreno A, Sagarra-Alamo R, Basora-Gallisa J, Saleh E, et al. A Clinical Decision Support System for Diabetic Retinopathy Screening: Creating a Clinical Support Application. *Telemedicine and e-Health*. 2019 Jan 1;25(1):31-40.
- [2] Saleh E, Valls A, Moreno A, Romero-Aroca P, Torra V, Bustince H. Learning Fuzzy Measures for Aggregation in Fuzzy Rule-Based Models. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Verlag; 2018. p. 114-27.
- [3] Winham SJ, Freimuth RR, Biernacka JM. A weighted random forests approach to improve predictive performance. *Statistical Analysis and Data Mining: The ASA Data Science Journal*. 2013 Dec 1;6(6):496-505.
- [4] El Habib Daho M, Settouti N, Lazouni MEA, Chikh MEA. Weighted vote for trees aggregation in Random Forest. In: *International Conference on Multimedia Computing and Systems -Proceedings*. IEEE Computer Society; 2014. p. 438-43.
- [5] Li HB, Wang W, Ding HW, Dong J. Trees Weighting Random Forest method for classifying high-dimensional noisy data. In: *Proceedings - IEEE International Conference on E-Business Engineering, ICEBE 2010*. 2010. p. 160-3.
- [6] Dogan A, Birant D. A Weighted Majority Voting Ensemble Approach for Classification. In: *UBMK 2019 - Proceedings, 4th International Conference on Computer Science and Engineering*. Institute of Electrical and Electronics Engineers Inc.; 2019. p. 366-71.
- [7] Zhukov A V., Sidorov DN, Foley AM. Random forest based approach for concept drift handling. In: *Communications in Computer and Information Science*. Springer Verlag; 2017. p. 69-77.
- [8] Kim H, Kim H, Moon H, Ahn H. A weight-adjusted voting algorithm for ensembles of classifiers. *Journal of the Korean Statistical Society*. 2011 Dec 27 ;40(4):437-49.
- [9] Yang C, Yin XC. Diversity-Based Random Forests with Sample Weight Learning. *Cognitive Computation*. 2019 Oct 1;11(5):685-96.
- [10] Zhong Y, Yang H, Zhang Y, Li P. Online random forests regression with memories. *Knowledge-Based Systems*. 2020 Aug 9;201-202:106058.
- [11] Zhu M, Xia J, Jin X, Yan M, Cai G, Yan J, et al. Class weights random forest algorithm for processing class imbalanced medical data. *IEEE Access*. 2018 Jan 3;6:4641-52.
- [12] Livieris IE, Kanavos A, Tampakas V, Pintelas P. A weighted voting ensemble self-labeled algorithm for the detection of lung abnormalities from X-rays. *Algorithms*. 2019 Mar 16;12(3):64.
- [13] Utkin L V., Kovalev MS, Meldo AA. A deep forest classifier with weights of class probability distribution subsets. *Knowledge-Based Systems*. 2019 Jun 1;173:15-27.
- [14] Kalles D, Morris T. Efficient incremental induction of decision trees. *Machine Learning*. 1996 Sep;24(3):231-42.
- [15] Utgoff PE, Berkman NC, Clouse JA. Decision Tree Induction Based on Efficient Tree Restructuring. *Machine Learning*. 1997;29(1):5-44.
- [16] Pecori R, Ducange P, Marcelloni F. Incremental learning of fuzzy decision trees for streaming data classification. In: *Proceedings of the 11th Conference of the European Society for Fuzzy Logic and Technology, EUSFLAT 2019*. Atlantis Press; 2020. p. 748-55.
- [17] Saffari A, Leistner C, Santner J, Godec M, Bischof H. On-line random forests. In: *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops 2009*. 2009. p. 1393-400.
- [18] Lakshminarayanan B, Roy DM, Teh YW. Mondrian Forests: Efficient Online Random Forests. Vol. 27, *Advances in Neural Information Processing Systems*. 2014.
- [19] Gomes HM, Bifet A, Read J, Barddal JP, Enembreck F, Pfahringer B, et al. Adaptive random forests for evolving data stream classification. *Machine Learning*. 2017 Oct 1;106(910):1469-95.