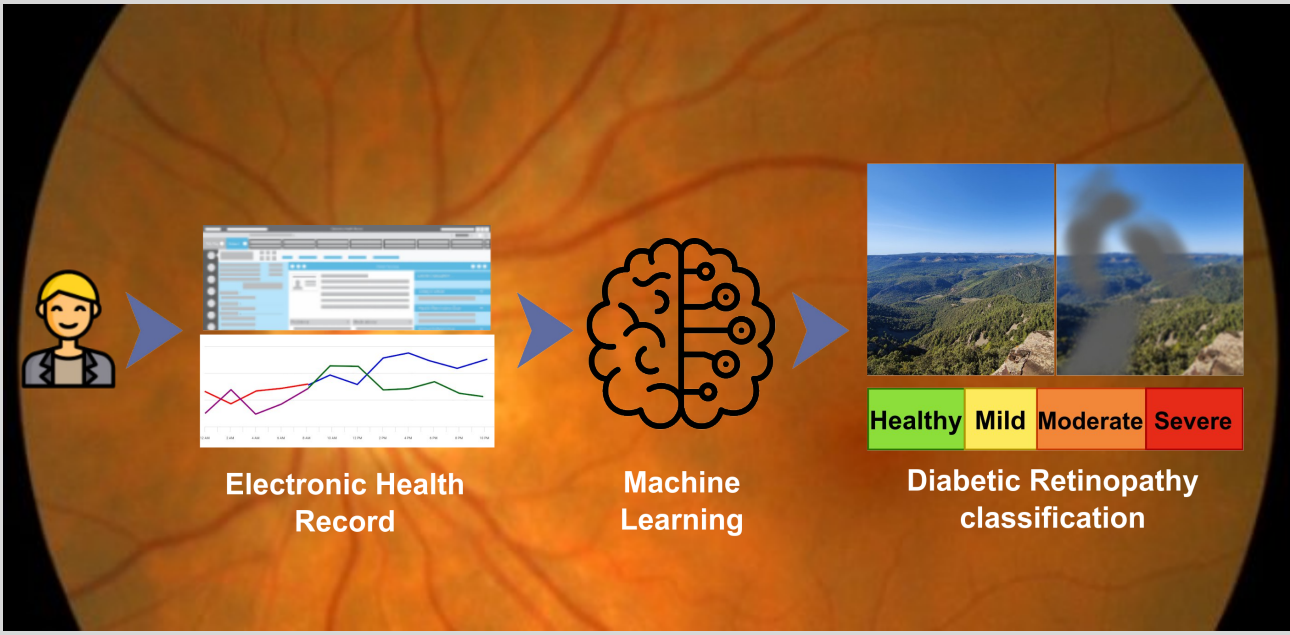
FUZZY-BASED MACHINE LEARNING METHODS FOR CONTINUOUS DIAGNOSIS AND PROGNOSIS OF DIABETIC RETINOPATHY

Jordi Pascual Fontanilles

UNIVERSITAT ROVIRA i VIRGILI

Fuzzy-based machine learning methods for continuous diagnosis and prognosis of Diabetic Retinopathy

JORDI PASCUAL FONTANILLES



DOCTORAL THESIS
2024

UNIVERSITAT ROVIRA I VIRGILI
FUZZY-BASED MACHINE LEARNING METHODS FOR CONTINUOUS DIAGNOSIS AND PROGNOSIS OF DIABETIC RETINOPATHY
Jordi Pascual Fontanilles

UNIVERSITAT ROVIRA I VIRGILI

DOCTORAL THESIS

Fuzzy-based machine learning methods for continuous diagnosis and prognosis of Diabetic Retinopathy

Author:
Jordi Pascual Fontanilles

Supervisor:
Dr. Aïda Valls Mateu

Departament d'Enginyeria Informàtica i Matemàtiques
ITAKA

Tarragona
2024

UNIVERSITAT ROVIRA I VIRGILI
FUZZY-BASED MACHINE LEARNING METHODS FOR CONTINUOUS DIAGNOSIS AND PROGNOSIS OF DIABETIC RETINOPATHY
Jordi Pascual Fontanilles

UNIVERSITAT ROVIRA i VIRGILI

Departament d'Enginyeria Informàtica i Matemàtiques

Av. Països Catalans, 27
43007 Tarragona
Tel. +34 977 55 95 95
Fax. +34 977 55 95 97

FAIG CONSTAR que aquest treball, titulat "Fuzzy-based machine learning methods for continuous diagnosis and prognosis of Diabetic Retinopathy", que presenta Jordi Pascual Fontanilles per a l'obtenció del títol de Doctor, ha estat realitzat sota la meva direcció al Departament d'Enginyeria Informàtica i Matemàtiques d'aquesta universitat.

Tarragona, Gener 2024.

La directora de la tesi doctoral,

Dra. Aïda Valls Mateu

UNIVERSITAT ROVIRA I VIRGILI
FUZZY-BASED MACHINE LEARNING METHODS FOR CONTINUOUS DIAGNOSIS AND PROGNOSIS OF DIABETIC RETINOPATHY
Jordi Pascual Fontanilles

Als meus pares, germans, avis i tota la família i amics, qui m'heu acompanyat en tot aquest camí. Sense el vostre amor, suport i inspiració, aquesta tesi no hauria estat possible. Gràcies per creure en mi i per ajudar-me a arribar fins aquí. Us estimo més del que les paraules poden expressar.

UNIVERSITAT ROVIRA I VIRGILI
FUZZY-BASED MACHINE LEARNING METHODS FOR CONTINUOUS DIAGNOSIS AND PROGNOSIS OF DIABETIC RETINOPATHY
Jordi Pascual Fontanilles

Acknowledgements

The author has been supported by a predoctoral FI grant from the Generalitat de Catalunya and Fons Social Europeu, 2023 FI-3 00036. The work was also funded by project PI21/00064 from Instituto de Salud Carlos III (ISCIII) and co-funded by the European Union, projects 2023PFR-URV-114 and 2022PFR-URV-41 from Universitat Rovira i Virgili (URV) and ITAKA funding 2021-SGR-00114 from AGAUR.

First and foremost, I would like to thank my supervisor, Dr. Aïda Valls, for her invaluable advice, help and support during my PhD studies. Your guidance and knowledge have been fundamental and have helped me become a better researcher. Also, my special thanks to Dr. Toni Moreno and all the fellow members of the ITAKA research group for all your support and the time we spent together.

Finally, I would like to express my appreciation to all my family and friends for their encouragement and support throughout all my studies. I am grateful for your unwavering belief in me and your constant motivation to pursue my dreams.

UNIVERSITAT ROVIRA I VIRGILI
FUZZY-BASED MACHINE LEARNING METHODS FOR CONTINUOUS DIAGNOSIS AND PROGNOSIS OF DIABETIC RETINOPATHY
Jordi Pascual Fontanilles

List of Terms

AA Arithmetic Average.

AI Artificial Intelligence.

BA Balanced Accuracy.

BMI Body Mass index.

CDSS Clinical Decision Support System.

CKDEPI Chronic Kidney Disease Epidemiology Collaboration.

CNN Convolutional Neural Network.

DL Deep Learning.

DM Diabetes Mellitus.

DR Diabetic Retinopathy.

DT Decision Tree.

DTW Dynamic Time Warping.

EHR Electronic Health Record.

EVOL Evolution time of Type-2 DM in years.

FDT Fuzzy Decision Tree.

FG Fuzzy sample Generation.

FN False Negative.

FP False Positive.

x

FRF Fuzzy Random Forest.

HbA1c Concentration of glycated hemoglobin present in the bloodstream.

HTAR Control of arterial hypertension.

MA Microalbuminuria.

MTSC Multivariate Time Series Classification.

O Oversampling.

OC Occupancy.

OWA Ordered Weighted Averaging.

RF Random Forest.

TN True Negative.

TP True Positive.

TSC Time Series Classification.

TTM Treatment for Type-2 DM.

WBA Weighted Balanced Accuracy.

List of Figures

List of Tables

UNIVERSITAT ROVIRA I VIRGILI

Abstract

Escola Tècnica Superior d'Enginyeria
Departament d'Enginyeria Informàtica i Matemàtiques

Doctor of Philosophy

Fuzzy-based machine learning methods for continuous diagnosis and prognosis of Diabetic Retinopathy

by Jordi Pascual Fontanilles

Disease diagnosis and prognosis may be supported by Clinical Decision Support Systems (CDSS) that take advantage of existing data of medical knowledge and patient's information. Such systems are built using diverse Artificial Intelligence and Machine Learning techniques, and can be effective in reducing manual time-consuming tasks, analysing patients health records or supporting non-expert clinicians in a field. This work focuses on Diabetic Retinopathy (DR), a severe complication of Diabetes Mellitus, a chronic, widespread disease. 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 current screening procedure is based on images of the eye-fundus, which requires a time-consuming and costly use of non-mydriatic cameras. On the contrary, Retiprogram is a CDSS based on Fuzzy Random Forests (FRF) to help in the early diagnosis of Diabetic Retinopathy using patients' clinical data.

In this PhD thesis, the goal is to study how a Fuzzy Random Forest classification model can take advantage of data in conditions of dynamic changes. Diabetic Retinopathy is used as the main case study, although methods are also validated in other datasets. The first contribution is to improve the current results of a binary FRF classifier by taking advantage of the data of the new patients which are treated at the hospital. We propose to modify the set of trees that compose the FRF, which allows updating the model without retraining the base model from scratch. The results of this updating process show a clear improvement in the classification performance, specially in the detection of positive cases.

The second contribution adapts the binary FRF classification procedure to the case of ordinal multiclass, where the order between the set of classes is relevant. This is particularly helpful in medical diagnosis to detect the severity of the disease, such as in DR, where ophthalmologists differentiate between different severity degrees of retinopathy. The work is focused on the prediction stage of the FRF. When a new instance arrives, the rules' activation is done with the usual fuzzy operators, but the aggregation of the outputs given by the different rules and trees has been redefined. In particular, the proposal manages the conflicting cases where different classes are predicted with similar support. The support of the classes is calculated using the conjunctive/disjunctive Ordinal Weighted Averaging operator, which permits to model the concept of majority agreement.

A third contribution in this thesis is focused on the detection of DR on long-term diabetic patients. Due to the continuous controls and medications, long-term diabetic people improve some clinical factors, which makes it much harder to predict the appearance and progression of the Diabetic Retinopathy disease. The thesis proposes a method to exploit the Electronic Health Record history data to construct a temporal dataset for DR. It is able to handle missing data and unequally data distribution in time. Moreover, we propose a novel technique to make use of short EHR series to minimize class imbalance, which consists on using longer time series to synthetically complete short time series with a fuzzy-based approach. Several state-of-the-art classification methods for time series have been evaluated. The results indicate that the TapNet classifier is the best one for DR grading with this unusually short time series.

Contents

UNIVERSITAT ROVIRA I VIRGILI
FUZZY-BASED MACHINE LEARNING METHODS FOR CONTINUOUS DIAGNOSIS AND PROGNOSIS OF DIABETIC RETINOPATHY
Jordi Pascual Fontanilles

Chapter 1

Introduction

1.1 Motivation

Clinical Decision Support Systems (CDSS) are computerized tools to assist clinicians in making good decisions during the clinical decision-making process. CDSS can be used for a variety of tasks, including alerting clinicians to potential adverse events or drug interactions that may affect patient safety, providing personalised treatment recommendations based on individual patient characteristics and preferences, and helping clinicians manage complex clinical pathways by providing step-by-step guidance and decision support during the care process. CDSS can be designed using a variety of Artificial Intelligence (AI) and machine learning techniques, including natural language processing, deep learning models, and predictive modeling. These tools can help clinicians to make decisions by providing them with relevant information. They can also be effective in reducing manual time-consuming tasks, such as searching for clinical guidelines, performing complex calculations or analysing patients health records. By automating these tasks and providing relevant information at the point of care, CDSS can help clinicians make more informed decisions with less effort, potentially leading to improved patient outcomes, increased clinician satisfaction and cost reduction for the health centres. CDSS can also be effective in supporting non-expert clinicians in a field by providing them with relevant information and guidance based on their different area of expertise, for instance, in primary health centres.

CDSS for diagnosis assistance are specialized tools that are constructed for supporting a specific disease. Chronic widespread diseases are of great interest due to the amount of people they can help. This is the case of Diabetes Mellitus (DM), a chronic metabolic disorder that affects millions of people worldwide. DM occurs when the body either does not produce enough insulin (a hormone that regulates

blood sugar levels) or cannot use the insulin it produces effectively. This leads to high blood glucose levels, which over time can cause serious complications such as Diabetic Retinopathy (DR), kidney disease, heart disease, and nerve damage. According to recent estimates by the World Health Organization (WHO), the number of people with diabetes has increased from 180 million in 1980 to 422 million people in 2014 (Roglic, 2016). The prevalence is expected to rise significantly in the coming years due to increasing obesity rates, sedentary lifestyles, and ageing populations. This highlights the urgent need for innovative solutions that can improve the management and prevention of diabetes.

Diabetic Retinopathy is a severe complication of Diabetes Mellitus, leading to vision loss or even blindness if left untreated. This condition occurs when high blood sugar levels damage the tiny blood vessels within the retina, causing them to leak fluid or blood, swell, or form scar tissue that can lead to permanent vision impairment or loss. The world prevalence of Diabetic Retinopathy on diabetic people was estimated to be about 22.27% in 2020 (Teo, Tham, et al., 2021). Early detection of Diabetic Retinopathy is crucial for timely intervention and preserving patients' quality of life (Romero-Aroca, Riva-Fernandez, et al., 2016).

The current diagnostic methods for this condition are often time-consuming and costly. Ophthalmologists typically rely on the use of non-mydriatic cameras to obtain images of DM patients. These procedures are expensive and require specialized equipment as well as trained personnel for accurate assessment. Moreover, due to the huge number of DM patients, these images can only be taken every two years or more because there are not enough resources to perform this test annually. As deterioration of the eye can be rapid, this image-based screening procedure is not sufficient to prevent vision loss in some cases. Additionally, clinical and analytical data can also provide evidence on the patients' DR status. Lately, new studies are trying to develop a CDSS for DR using data instead of eye-fundus images.

The development of a CDSS specifically designed for diagnosing Diabetic Retinopathy has the potential to revolutionize eye care by providing medical professionals with accurate, timely information on which to base their decisions. This is particularly beneficial for family physicians, who may lack the required expertise and training to decide if their patients require eye-fundus screening.

By leveraging advances in Artificial Intelligence and machine learning, such a CDSS could analyse relevant diagnostic data, extract key features, and classify them into the different stages of Diabetic Retinopathy.

This could lead to better patient outcomes, reduced healthcare costs, and improved overall satisfaction of the patients. Furthermore, it has the potential to

increase accessibility to specialized care in remote or underserved areas, as a CDSS can be easily integrated into existing telemedicine platforms.

1.2 Research framework

This doctoral thesis is framed in a research project carried out by the ITAKA research group (University Rovira i Virgili) and the Ophthalmology Unit of Hospital Universitari Sant Joan de Reus. This project is funded by the Spanish research institution called "Instituto de Investigación Carlos III" and FEDER funds. The main goal of this project is to develop tools for the diagnosis of Diabetic Retinopathy. The project has two main research lines with the main goal of assessing the patients' risk of developing DR. One research line consists on analysing the eye-fundus image of patients to assess the patient's risk of developing DR. The software that performs this analysis is named MIRA (J. d. l. Torre, Valls, et al., 2020; Romero-Aroca, Verges-Puig, et al., 2020). The other research line has the same objective, but using just the clinical data of the patient that can be gathered from their Electronic Health Record (EHR). The work in this line started in 2015 and leaded to the development of a software tool called Retiprogram (Saleh, Błaszczyński, et al., 2018; Romero-Aroca, Valls, et al., 2019). The aim of the system is to improve the diagnosis of DR in Primary Health Care centres, reducing the workload of ophthalmologist services. By helping the non-specialists in ophthalmology to determine the DR risk, they can determine who needs an eye-fundus test in order to focus their use only on critical patients. The workflow of the clinicians that use this CDSS that combines both Mira and Retiprogram is depicted in Figure 1.1.

DM patients usually visit the primary health care centres to control their illness. General clinicians are not necessarily experts in detecting DR, thus, they can benefit from Retiprogram to diagnose DR in early stages. If a patient has a low risk of DR, they can program a new DR control after three years, reducing the number of interventions for the patient and the costs for the health care centre. In contrast, if the risk is moderate, high, or Retiprogram has doubts, an eye-fundus image is taken using a non-mydriatic camera. Then, MIRA software is used to automatically analyse the image. If the risk is low, a new visit is scheduled after one or two years. Otherwise, the patient is derived to an ophthalmologist to confirm the diagnosis and assign the proper treatment for the patient.

In the following subsections, both components, MIRA and Retiprogram are introduced in more detail.

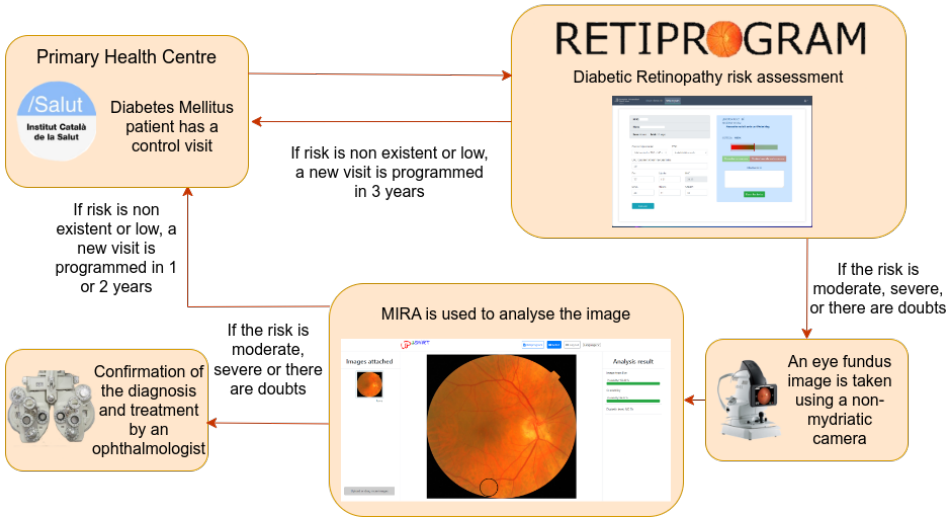


FIGURE 1.1: Workflow for the CDSS to diagnose DR

1.2.1 DR risk assessment from eye fundus images

The DR risk assessment problem is commonly approached as an image analysis problem, consisting on classifying eye-fundus images. Dubey and Dixit (Dubey and Dixit, 2022) and Atwany et al. (Atwany, Sahyoun, et al., 2022) reviewed the recent developments on these kinds of systems. Even DR risk classification task is the most frequent, some studies also perform segmentation tasks, such as the identification of some eye structures relevant for DR (f.i. optical disc, optical nerve or blood vessels), or the location of DR lesions (f.i. microaneurisms or exudates). In the majority of cases, machine learning or deep learning methods are used for the image analysis.

In the case of MIRA, it uses deep learning models to grade DR according to the ETDRS standard classification (Wilkinson, Ferris, et al., 2003): no DR ($DR = 0$), mild ($DR = 1$), moderate ($DR = 2$) and severe ($DR = 3$). They are ordered from the best to the worst medical conditions. Two deep learning models are used. The first one is used to detect the quality of the eye-fundus image that has been taken (Khalid, Rashwan, et al., 2024). When these kinds of images are captured, many problems might arise. For instance, unfocused images, reflections, artifacts, etc. If the quality of the image is not good enough, it is discarded, as it cannot be used to diagnose DR. The second one is a classifier network. Given an eye-fundus

image, it analyses it, and classifies the image according to the ETDRS classification categories for DR (J. d. l. Torre, Valls, et al., 2020; Romero-Aroca, Verges-Puig, et al., 2020). The model has received additional improvements, for instance, in (Escorcia-Gutierrez, Cuello, et al., 2023) transfer learning was applied to the build the deep learning model to improve the grading of DR. MIRA is being tested at the Hospital Universitari Sant Joan de Reus by the members of the ophthalmologist service. The Graphical User Interface of MIRA can be seen in Figure 1.2.

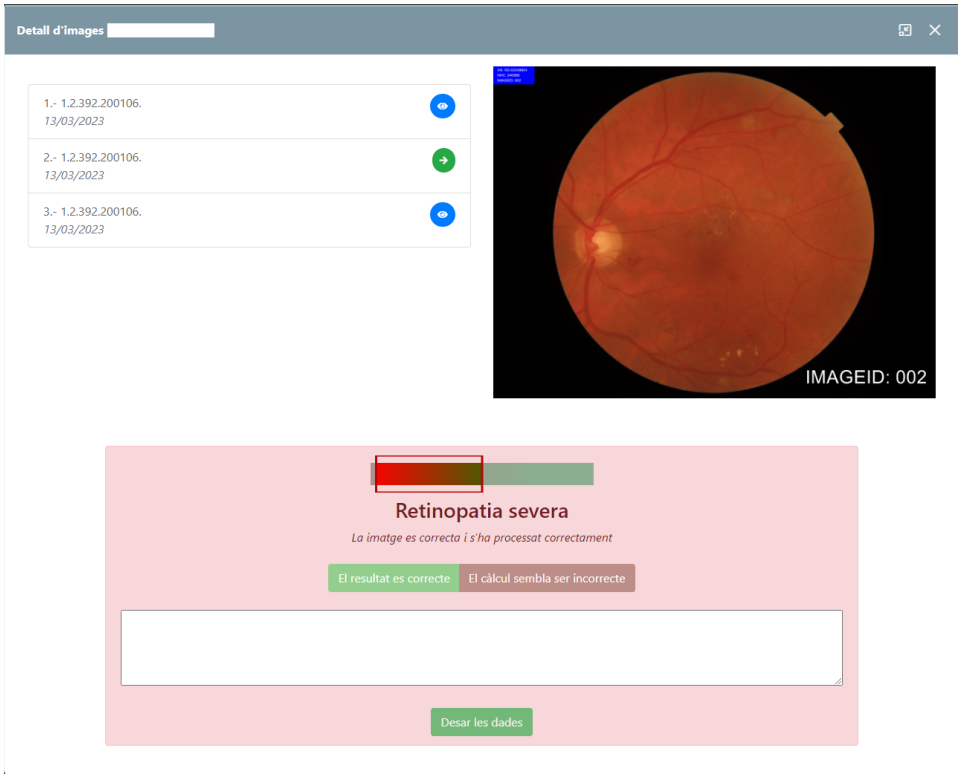


FIGURE 1.2: Graphical User Interface of Mira

1.2.2 DR risk assessment from clinical data

Although not as common as image analysis, in the literature, some approaches using clinical information available in the Electronic Health Record to diagnose DR can also be found. Sun and Zhang used the first hospitalization EHR data of diabetic patients to create a DR dataset (Sun and D. Zhang, 2019). They filtered the available variables to preserve the relevant medical ones for the DR. Some of

the most sensitive variables they found were unsaturated iron binding capacity, bilirubin, and glycosilated serum protein. They compared how several machine learning binary classifiers performed whether feature engineering is applied or not to the dataset . They obtained the best results when applying feature engineering using a random forest classifier. Some other studies in the literature analyse how different kinds of classifiers perform on EHR-based DR datasets (Y. Zhao, X. Li, et al., 2022; Tsao, Chan, et al., 2018; O. I. Ogunyemi, Gandhi, et al., 2019; O. Ogunyemi and Kermah, 2015). They use techniques such as Random Forests, XGBoost, logistic regression, support vector machines or k-nearest neighbours. There is not a consensus in which is the best classifier for the Diabetic Retinopathy disease, as there are different results on each study.

Retiprogram is also based on a machine learning approach that uses data from EHRs of patients to predict the risk of developing DR. It uses a Fuzzy Random Forest classifier. The algorithm takes into account various factors of the current patient's conditions, including some analytical data given in the last blood analysis. Some of them are age, gender, blood pressure, and diabetes duration. It also considers the use of medications for the treatment of diabetes. It was developed as a binary classification system, able to distinguish the negative ($DR = 0$) and positive ($DR = 1$) classes.

Retiprogram is the starting point for this thesis, and we aim to improve the existing model in several ways. The main causes of misclassifications are the general ambiguity of the problem (very similar patients can belong to different classes), and the high imbalance between classes, as most of the diabetic people will not suffer from DR. Patients with progression to the worst classes are also a minority in comparison with the ones that have a mild degree. Consequently, the availability of diabetic patient's data with DR is scarce. Because of this under-representation, the classification models have more difficulty to correctly identify and distinguish the positive classes.

This system is tested at the Hospital Universitari Sant Joan de Reus since 2018 by the members of the ophthalmology service. The Graphical User Interface of the Retiprogram can be seen in Figure 1.3. The results obtained by Retiprogram are quite good, with a specificity of 84%, sensitivity of 80% and a precision of 81% (Valls, Moreno, et al., 2023). We aim to keep improving the system, mainly focusing on positive patients.

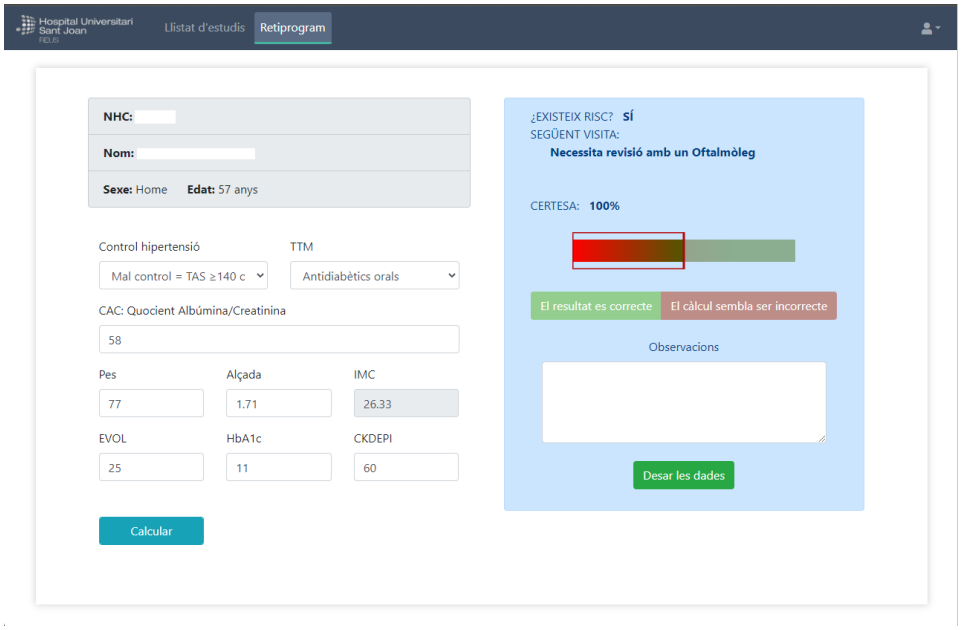


FIGURE 1.3: Graphical User Interface of the Retiprogram

1.3 Thesis objectives

The current performance of the Retiprogram system to assess diabetic patients' risk of developing DR is quite good, with a sensitivity and specificity over 80%. Despite this, it still has room for improvement, which corresponds to the objectives of this thesis, which are outlined below. As new data has been gathered with the use of the Retiprogram system, a first step towards the improvement of the system is to use the new data that arrives at the system to update the model. Instead of training from scratch the model, which would require an extensive validation of the new model, we want to use the data to modify the existing model to obtain higher quality classification results. Additionally, our focus would be on the detection of the positive patients.

Another improvement we want to develop is to detect DR into more accurate categories, according to the ETDRS standard classification. By determining the DR risk as an ordinal multiclass classification problem, a better distribution of resources among the patients who need them the most can be achieved. Moreover, better treatments can be used on the individuals who require them, and avoided on the ones who do not require them.

The last study we want to perform on this thesis is how Retiprogram performs

on long-term DM patients. When a patient is diagnosed of DR, he or she usually starts some treatments in order to improve some clinical factors, therefore, for the patients under treatment it is more challenging to distinguish their DR grade only observing the values of a unique visit. Our hypothesis is that a retrospective analysis could be more adequate to have an overall view of the patient's conditions evolution and could improve the grading of DR when enough data is available.

The main objectives of this thesis can be summarised as follows:

1. The current CDSS, Retiprogram, is based on a Fuzzy Random Forest. This is a static model, it does not evolve with time. When used, new data can be gathered from the medical professionals. By introducing new mechanisms, we aim to dynamically update the classification model using the new gathered data to increase its performance.

2. Transition from the current binary classifier to a multiclass classifier that can determine the current state of a diabetic patient and predict the evolution of their Diabetic Retinopathy.

3. Develop new mechanisms to analyse how risk factors for patients change over time based on data from a sequence of visits, with the aim of determining how the state of Diabetic Retinopathy may evolve in order to take corrective measures.

1.4 Contributions

The main contributions of this thesis are the following:

1. To achieve the first objective of the thesis, a novel method was proposed to iteratively update the set of trees in a Fuzzy Random Forest by considering trees generated from small sets of new examples. We introduced the use of the weighted balance accuracy to determine which trees are better. By using this weighted average between sensitivity and specificity combined with the novel updating method for Random Forests, the detection of positive examples was improved. This contribution is described in the Chapter 3 of this dissertation. It has been published in the following papers:

    - Jordi Pascual-Fontanilles, Aida Valls, Antonio Moreno, and Pedro Romero-Aroca (Oct. 2021). "Iterative Update of a Random Forest Classifier for Diabetic Retinopathy". In: *Artificial Intelligence Research and Development*.

Vol. 339. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 207–216. DOI: 10.3233/FAIA210136

- Jordi Pascual-Fontanilles, Aida Valls, Antonio Moreno, and Pedro Romero-Aroca (Dec. 2022). "Continuous Dynamic Update of Fuzzy Random Forests". In: *International Journal of Computational Intelligence Systems* 15 (1), pp. 1–16. DOI: 10.1007/S44196-022-00134-0
- Jordi Pascual-Fontanilles (Mar. 2022). "Dynamic update of Fuzzy Random Forests to improve classification of Diabetic Retinopathy". In: *7th URV Doctoral Workshop In Computer Science And Mathematics*. ISBN: 9788413650333

2. To adapt the Fuzzy Random Forest for the ordinal multiclass case, we redefined the aggregation procedure of the outputs given by the different rules and trees on a Fuzzy Random Forest. A new procedure is proposed to manage conflicting cases where different classes are predicted with similar support in the ordered multiclass case is proposed. We also used the Ordered Weighted Averaging (OWA) operator to model the concept of majority agreement in the calculation of the prediction confidence. This work was developed during a research stay at the Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague under the supervision of Dr. Lenka Lhotská, head of the Cognitive systems and neurosciences department. These contributions are explained in Chapter 4, and they were published in the following paper, presented in the 24th International Conference of the Catalan Association for Artificial Intelligence (CCIA-2022):

   - Jordi Pascual-Fontanilles, Lenka Lhotska, Antonio Moreno, and Aida Valls (Oct. 2022). "Adapting a Fuzzy Random Forest for Ordinal Multi-Class Classification". In: *Artificial Intelligence Research and Development*. Vol. 356. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 181–190. DOI: 10.3233/FAIA220336

3. Considering the third objective of this thesis, we developed a novel method to exploit the EHR retrospective data to construct a dataset with temporal series of Diabetic Retinopathy assessments for some patients. The proposed approach also addresses the missing data and unequal data distribution in time. Moreover, we propose a novel technique to make use of short EHR series to minimize class imbalance, which consists on using longer time series

to synthetically complete short time series with a fuzzy-based approach. The new dataset created using those methods was used to train several state-of-the-art multivariate time series classifiers. These works are described in the Chapter 5, and published in the following papers:

- Jordi Pascual-Fontanilles, Aida Valls, Antonio Moreno, and Pedro Romero-Aroca (Oct. 2023b). "Challenges in the Exploitation of Historical Clinical Data for the Classification of Diabetic Retinopathy Patients". In: *Artificial Intelligence Research and Development*. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 204–207. DOI: 10.3233/FAIA230683
- Jordi Pascual-Fontanilles, Aida Valls, and Pedro Romero-Aroca (June 2023a). "A fuzzy-based method to boost short time-series to solve class imbalance in health care data". Paper presented at The 20th International Conference on Modeling Decisions for Artificial Intelligence
- Jordi Pascual-Fontanilles, Aida Valls, and Pedro Romero-Aroca (2023c). "Multivariate data binning and examples generation to build a Diabetic Retinopathy classifier based on temporal clinical and analytical risk factors". In: *Artificial Intelligence in Medicine*. Submitted

4. I have also participated in other works that were derived from the project. They were leaded by the ophthalmologists team and focused on the dissemination of our work to medical specialists. My involvement included the use of the Retiprogram system, data curation and preparation, methodology design and testing:

   - Pedro Romero-Aroca, Raquel Verges, Jordi Pascual-Fontanilles, Aida Valls, Josep Franch, Joan Barrot, Xavier Mundet, Alex La Torre, Antonio Moreno, Ramon Sagarra, Josep Basora, Eugeni Garcia-Curto, and Marc Baget-Bernaldiz (Oct. 2023). "Effect of Lipids on Diabetic Retinopathy in a Large Cohort of Diabetic Patients after 10 Years of Follow-Up". In: *Journal of Clinical Medicine 2023, Vol. 12, Page 6674* 12 (20), p. 6674. ISSN: 2077-0383. DOI: 10.3390/JCM12206674
   - Aida Valls, Antonio Moreno, Jordi Pascual-Fontanilles, Julian Cristiano,

Domenec Puig, and Pedro Romero-Aroca (Nov. 2023). "RETIPROGRAM and MIRA software". In: *INTELIGENCIA ARTIFICIAL Y OFTALMOLOGÍA: ESTADO ACTUAL EN CATALUÑA*. vol. 31. 4. Òrgan de la Societat Catalana d'Oftalmologia, pp. 206–213. ISBN: 978-84-19264-38-1

- Pedro Romero-Aroca, Marc Baget, Aida Valls, Eugeni Garcia-Curto, Jordi Pascual-Fontanilles, and Ramon Sagarra (Nov. 2023). "Prediction and progression of diabetic retinopathy". In: *INTELIGENCIA ARTIFICIAL Y OFTALMOLOGÍA: ESTADO ACTUAL EN CATALUÑA*. vol. 31. 4. Òrgan de la Societat Catalana d'Oftalmologia, pp. 256–263. ISBN: 978-84-19264-38-1

1.5 Awards

During the development of this thesis, we have won the following research award:

1. We received the best poster award in the 23rd International Conference of the Catalan Association for Artificial Intelligence (CCIA 2021). The award-winning work is entitled "Iterative update of a Random Forest classifier for Diabetic Retinopathy".

More details about the awards can be found in appendix A.

1.6 Thesis organization

The rest of the document is organised in the following chapters:

- **Chapter 2**: Background
  In this chapter, we explain how the Retiprogram system was designed. It is based on a Fuzzy Random Forest model trained with clinical and analytical data from patients, and is able to assess the risk of diabetic patients on developing Diabetic Retinopathy. This model is the basis for the novel methods that have been researched for this thesis. We also explain the evaluation metrics that are used throughout the thesis to evaluate the obtained results.

- **Chapter 3**: Dynamic improvement of a Fuzzy Random Forest
  In this chapter, we present a new method to dynamically update the Retiprogram model using new data that arrives at the system from time to time.

The results are validated on a Diabetic Retinopathy dataset and on a dataset about the current occupancy of an office room.

- **Chapter 4**: Adapting a Fuzzy Random Forest for ordinal multiclass classification
  This chapter presents an adaptation for the Retiprogram so it can handle multiclass classification for ordinal classes. This is the case of Diabetic Retinopathy, which has multiple ordered positive categories. The addition of ordered multiclass classification allows to more accurately classify patients. The method has been tested on the Diabetic Retinopathy data, as well as on a work burnout dataset to validate the results.

- **Chapter 5**: Improving DR detection on long-term patients using temporal historical data
  In this chapter, a new approach to detect Diabetic Retinopathy on long-term diabetic patients is presented. We analyse how long-term diabetic patients are much harder to assess by the Retiprogram. To alleviate this issue, we propose a new method to perform the risk assessment using the retrospective information of the patients. The method includes the processing step of patients data from their Electronic Health Record as a time series. To compensate for the data imbalance, we have also proposed a method to generate new feasible positive examples from time series that were too short to be considered for training the classifier. The results have been validated on a Diabetic Retinopathy dataset obtained from the Electronic Health Records of Catalan diabetic patients from 2010 to 2021.

- **Chapter 6**: Conclusions and future work
  This chapter gathers the work done in the thesis, discusses the contributions, and presents the conclusions of the Ph.D. thesis. Finally, several lines of future research work are proposed.

- **Appendix A**: Awards
  The awards obtained by the work in this Ph.D. dissertation are presented in this appendix.

Chapter 2

Background

In this chapter, we present the design of the Retiprogram system. It is based on a Fuzzy Random Forest model trained with clinical and analytical data from patients, and it is able to assess the risk of diabetic patients on developing Diabetic Retinopathy. This model was developed previously to this thesis, and it is the basis for the novel methods that have been developed. We also explain in this chapter the evaluation metrics that are used throughout the thesis to validate the methods proposed.

2.1 Retiprogram

The Retiprogram Clinical Decision Support System was developed during two Spanish research projects by members of the ITAKA research group jointly with the group of ophthalmology of Hospital Sant Joan de Reus since 2015. Retiprogram considers several attributes obtained from the clinical data of patients in order to distinguish between the positive and negative DR classes. Many classification models were tested in this problem, with the best results obtained using Fuzzy Random Forests (Romero-Aroca, Valls, et al., 2019). Random Forests (RF) are ensemble learning methods that combine multiple Decision Trees (DT). They have shown great performance in classification tasks in comparison with other techniques (Fernández-Delgado, Cernadas, et al., 2014). In domains with uncertainty or imprecision, we can use Fuzzy Random Forests (FRF), an extension of Random Forests that makes use of fuzzy logic. Their ensemble is constructed using Fuzzy Decision Trees (FDT), where the rules are defined on fuzzy linguistic variables, and many rules can be activated at different levels depending on the inputs. The use of fuzzy input attributes allows the system to reason in a way that is closer to humans. The linguistic variables are also more interpretable by medical experts than using

numerical conditions. Moreover, they avoid reasoning with precise numerical data when the problem does not require it. In order to assess the risk of developing Diabetic Retinopathy, doctors reason qualitatively on the attribute values (e.g., age: child/young/old, body mass: underweight/normal/overweight, hypertension: good control/bad control, etc.). A difference of one year or one kilogram makes no difference in the diagnosis, which is done at a more general scale of measurement (with labels). However, the input data is precise and numerical, so fuzzification is a proper procedure to move from the numerical scale to the linguistic scale of measurement.

For the construction of Retiprogram, nine relevant risk factors for DR diagnosis were selected by experts. They consist of six numerical and three categorical variables (Romero-Aroca, Riva-Fernandez, et al., 2016; Romero-Aroca, Valls, et al., 2019). They are the following:

- **Age**: the current age of the patient in years.
- **Body mass index** (BMI): it is a value derived from the mass and height of a person, and provides a measure to objectively discuss weight problems. It is calculated as the body mass divided by the square of the body height.
- **Duration of Type-2 DM** (EVOL): the number of years that have passed since a patient was diagnosed of Type-2 DM. It indicates the evolution time of Type-2 DM for the patient.
- **HbA1c**: concentration of glycated hemoglobin present in the bloodstream. It is a form of hemoglobin that is chemically linked to sugar, which can be used to detect an excessive amount of sugar in the bloodstream. A high concentration of HbA1c is one of the indicatives of DM.
- **Chronic Kidney Disease Epidemiology Collaboration** (CKDEPI): it is an approximation measure for the glomerular filtration rate of the kidney, which is the amount of fluid the kidney can filter in a given amount of time. If CKDEPI is low, it indicates problems in the kidney function.
- **Microalbuminuria** (MA): it measures an increase in the level of albumin in the urine. It indicates a disease in the kidney, as a healthy kidney filters albumin.
- **Gender**: the sex of the patient.

- **Treatment of Type-2 DM** (TTM): the prescribed treatment for DM by the medical specialists, such as a regulated diet, oral antidiabetics or insulin medication.
- **Control of arterial hypertension** (HTAR): it assesses whether the patient has good or bad control of arterial hypertension. A bad control of arterial hypertension means a persistent high blood pressure, which is a major risk factor for health.

The numerical attributes (i.e., Age, BMI, EVOL, HbA1c, CKDEPI and MA) were fuzzified with the ophthalmologists' expertise, defining appropriate linguistic labels for each relevant attribute (Romero-Aroca, Valls, et al., 2019). Fig. 2.1 shows the linguistic labels defined for the CKDEPI variable, and Fig. 2.2 the linguistic labels defined for the Age variable.

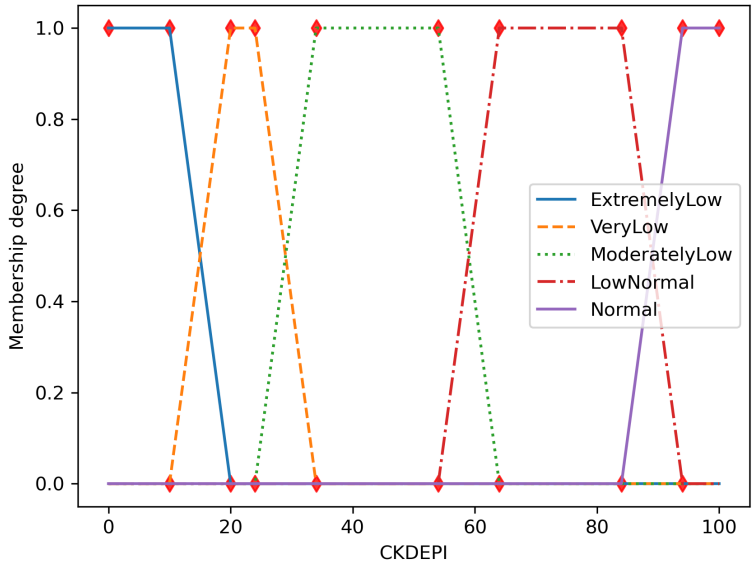


FIGURE 2.1: Definition of linguistic labels for CKDEPI

With the information provided by the available evidence, the ophthalmologist determines the degree of DR, which is the target attribute. According to the ETDRS standard classification (Wilkinson, Ferris, et al., 2003), the possible degrees of DR are: no DR ($DR = 0$), mild ($DR = 1$), moderate ($DR = 2$) and severe ($DR = 3$). They are ordered from the best to the worst medical conditions. As Retiprogram considers a binary classification problem, there are just two possible degrees of DR: $DR = 1$ means a high risk of suffering from Diabetic Retinopathy (i.e., the positive

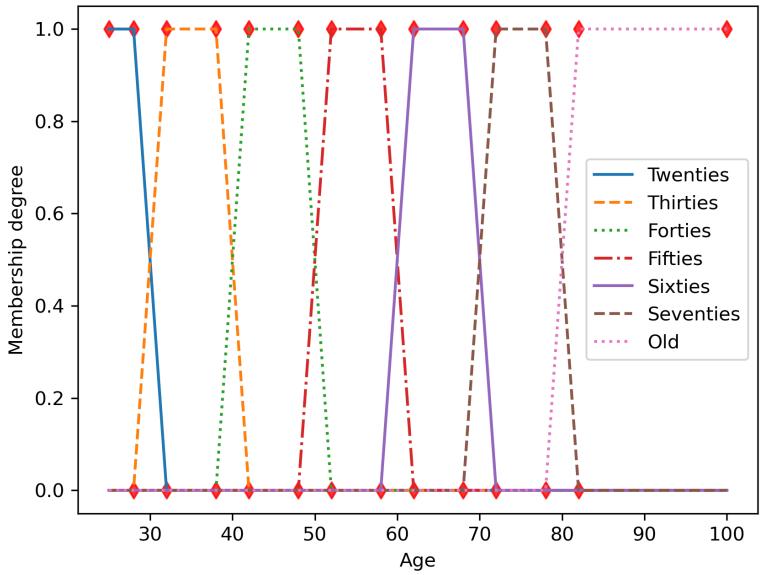


FIGURE 2.2: Definition of linguistic labels for Age

class). In this system, the higher levels of DR (i.e., $DR = 2$ and $DR = 3$) are coded as $DR = 1$. $DR = 0$ remains the same, meaning a low risk of developing DR (i.e., the negative class).

Decision rules can be created using fuzzy variables. They allow assigning a patient to a DR class when he or she fulfills all the conditions of the rule. For instance, a fuzzy rule for the positive class could be: $(Age$ is $Seventies)$ and $(TTM$ is $Insulin)$ and $(HbA1c$ is $8to9)$ and $(HTAR$ is $badControl)$, then $DR = 1$. An example of a fuzzy rule for the negative class could be: if $(EVOL$ is $5to10)$ and $(MA$ is $low)$ and $(CKDEPI$ is $ModeratelyLow)$, then $DR = 0$.

2.2 The binary DR classification model

A Fuzzy Random Forest (FRF) is an extension of Random Forests that makes use of fuzzy logic. This allows them to manage the uncertainty and imprecision of the data. It is composed of a set of Fuzzy Decision Trees (FDT), which can be constructed using several algorithms. In the following subsections, the procedure used to construct the Fuzzy Decision Trees and Fuzzy Random Forests in the Retiprogram is explained. The construction method is based on Yuan and Shaw's induction algorithm (Yuan and Shaw, 1995) with some extensions presented in

(Saleh, Valls, et al., 2016), (Saleh, Błaszczyński, et al., 2018) and (Saleh, Valls, et al., 2019). Once the model has been built, the classification of new examples involves a two-step process. The first step consists of obtaining a prediction from each FDT on the FRF. Then, in a second step, all the predictions are aggregated into the final prediction for the given example.

2.2.1 Fuzzy Decision Tree

In Retiprogram the Fuzzy Decision Tree is constructed from a labelled dataset using the method proposed in (Saleh, Valls, et al., 2016). Each of the examples, u_i, of this dataset belongs to the universe of discourse $U = \{u_1, u_2, ..., u_m\}$. In our case, U represents a set of DM patients. They are described by a set of attributes $A = \{a_1, a_2, ..., a_n\}$. All numeric attributes $a \in A$ have a corresponding linguistic fuzzy partition $T = \{t_1, t_2, ..., t_s\}$ which corresponds to a fixed set of ordered labels. A membership function $\mu_{t_j}(u_i) \in [0, 1]$ is used to determine the degree of membership of the input value x into a certain term t_j of an attribute $a \in A$. A membership of $\mu_{t_j}(u_i) = 0$ represents non-membership to the set; $\mu_{t_j}(u_i) = 1$ represents full membership, whereas $0 < \mu_{t_j}(u_i) < 1$ represents u_i is partially a member of t_j. Following, some measures based on fuzzy evidence are explained. They are used in the induction process to construct a FDT given labelled data, Section 2.2.1.1. Once the FDT has been built, a new example can be classified by using an inference procedure, which is explained in Section 2.2.1.2.

The ambiguity or nonspecificity measure of a possibility distribution π on a set $X = \{x_1, x_2, ..., x_d\}$ is defined in (Yuan and Shaw, 1995) as:

$$g(\pi) = \sum_{i=1}^{d} (\pi_i^* - \pi_{i+1}^*) \ln i, \tag{2.1}$$

where $\pi^* = \{\pi_1^*, \pi_2^*, ..., \pi_d^*\}$ is the permutation of the possibility distribution $\pi = \{\pi(x_1), \pi(x_2), ..., \pi(x_d)\}$ sorted such that $\pi_i^* \geq \pi_{i+1}^*$ for $i = 1, ..., d$, and $\pi_{d+1}^* = 0$.

The ambiguity of a given attribute $a \in A$ is calculated as:

$$Ambiguity(a) = \frac{1}{m} \sum_{i=1}^{m} g(\pi_T(u_i)), \tag{2.2}$$

where π_T is the normalised possibility distribution of μ_T on U:

$$\pi_{t_r}(u_i) = \frac{\mu_{t_r}(u_i)}{\max_{1\leq j\leq s}\{\mu_{t_j}(u_i)\}} \tag{2.3}$$

The **truth level of classification** on a set of classes $C = \{C_1, C_2, ..., C_p\}$ is the possibility of classifying an example u_i into a class $C_k \in C$ given the fuzzy evidence E. Fore each class we calculate:

$$Truth(C_k|E) = \frac{S(E, C_k)}{\max_{1\leq j\leq p}\{S(E, C_j)\}}, \tag{2.4}$$

where $S(E, C_i)$ is the degree of truth for the rule "IF E then C_i". It is computed as a fuzzy subsethood. Given two sets, X and Y, the degree to which X is a subset of Y is computed as follows:

$$S(X, Y) = \frac{M(X \cap Y)}{M(X)} = \frac{\sum_{i=1}^{m} \min(\mu_X(u_i), \mu_Y(u_i))}{\sum_{i=1}^{m} \mu_X(u_i)}, \tag{2.5}$$

where $M(X)$ is the cardinality, or sigma-count, of the fuzzy set X. It is the sum of the membership values of X. In the Retiprogram binary classifier, we are considering all classes to be crisp (i.e., $RD = 0$, $RD = 1$) , therefore, μ_{C_k} will be 0 or 1 in all cases.

Then, the truth level of classification is a possibility distribution on the set U, and $\pi(C|E)$ is the corresponding normalisation using $S(C|E)$.

For a fuzzy partitioning $P = \{E_1, E_2, ..., E_q\}$ on fuzzy evidence F, the **classification ambiguity** can be obtained. It is denoted as $G(P|F)$, and is the weighted average of the classification ambiguity for each subset of the partition:

$$G(P|F) = \sum_{i=1}^{q} W(E_i|F) g(\pi(C|E_i \cap F)), \tag{2.6}$$

where $W(E_i|F)$ is the weight representing the relative size of the subset $E_i \cap F$ in F. That is, $W(E_i|F) = \frac{M(E_i \cap F)}{\sum_{j=1}^{k} M(E_j \cap F)}$.

2.2.1.1 Induction of a Fuzzy Decision Tree

The induction algorithm proposed by (Yuan and Shaw, 1995) is an extension of the ID3 method, which is used for crisp data. The idea is to reduce the classification ambiguity with accumulated fuzzy evidence. The choice of new fuzzy evidence is made according to its reduction of classification ambiguity. In that sense, it follows the same idea as ID3, which performs the same using information entropy as the

induction criterion instead of classification ambiguity. To handle the uncertainty introduced by the use of fuzzy logic, two parameters are introduced:

- **Significance level** (α): is used to determine which pieces of evidence are relevant enough. The induction process discards fuzzy evidence E whose membership degree is lower than α.

$$\mu_{E_\alpha} = \begin{cases} \mu_E(u_i) & \text{if } \mu_E(u_i) \geq \alpha \\ 0 & \text{if } \mu_E(u_i) < \alpha \end{cases} \tag{2.7}$$

- **Truth level threshold** (β): it is used to control the growth of the tree by fixing the minimum truth level of classification of a conclusion given by a rule. Low β values might lead to smaller trees with lower classification accuracy. High β values might instead lead to larger trees with better accuracy. Increasing β over a certain threshold would create a point of diminishing returns.

The selection of both parameters has to be made according to the specific dataset. The induction process consists of the following steps:

Step 1: the best attribute is selected as the root node v. The choice is made using the ambiguity of the attributes, Eq. 2.2.

Step 2: For each linguistic term of v, a branch is created. Examples with at least α support are added in new branches of the attribute v.

Step 3: the truth level of classification (Eq. 2.4) is computed for each branch and for each of the classes.

Step 4: if at least one class C_i has a truth level of classification higher than β, a leaf is created. The information of all classes is maintained in the leaf.

Step 5: if no classes obtain a truth level higher than β, the remaining attributes are tested to check if they would reduce the classification ambiguity.

Step 5.1: if the classification ambiguity (Eq. 2.6) is reduced by any attribute, the one with the smallest ambiguity is selected as a new decision node in that branch. The process is repeated from Step 2 until no further growth is possible.

Step 5.2: if no attribute reduces the classification ambiguity, the branch is terminated in a leaf.

2.2.1.2 Classification procedure in a Fuzzy Decision Tree

Once the FDT has been completed using the described induction procedure, we obtain a hierarchical structure with r branches from the root node to the leaves. Each branch corresponds to a different fuzzy rule with one or more premises consisting of linguistic variables defined on fuzzy sets. When rules are learned automatically from examples, in the leave nodes, we can store a rule support value for each possible class. In this case, it corresponds to the truth level of classification calculated on the induction.

To classify a new example for the binary classification case, the Mamdani inference procedure is used. When a new instance is classified, each rule R provides a decision support value for each of the available classes, obtained from the product of the rule premises activation $\mu_R(u_i)$ and the rule support for each class. Therefore, for each class C_k we obtain a tuple with r decision support values, one for each rule: $D_{k,1}, D_{k,2}, ..., D_{k,r}$.

To decide which is the final class assigned to the example, the overall support received by each class must be taken into account. To merge the values provided by all rules, in (Saleh, Valls, et al., 2019) several aggregation operators were analysed, and the use of the Choquet fuzzy integral with a fuzzy measure based on the distorted probability was proposed. Then, the support value is normalized using the truth level threshold β used for constructing the rules. The maximum amount of support allowed is 1. So, for the k-th class, we have the support calculated using Eq. 2.8.

$$D_k = \min\left(1, \frac{ChoquetIntegral(D_{k,1}, D_{k,2}, ..., D_{k,r})}{\beta}\right) \tag{2.8}$$

An additional step was introduced to the usual Mamdani inference procedure in (Saleh, Błaszczyński, et al., 2018). A threshold value δ_1 was added for binary classification to determine if the FDT had a clear consensus on determining the winner class, Eq. 2.9. To avoid mistakes, the label *Unknown* was introduced. When the difference between the two decision support values is lower than δ_1, we assume that the FDT is not sure and, hence, the label *Unknown* is assigned.

$$Class(u_i) = \begin{cases} \text{"Unknown"} & \text{if } |\mu_{c_0}(u_i) - \mu_{c_1}(u_i)| < \delta_1 \\ argmax_{C_k \in \{C_0, C_1\}}(\mu_{C_k}(u_i)) & \text{otherwise} \end{cases} \tag{2.9}$$

2.2.2 Fuzzy Random Forest

A Fuzzy Random Forest is a set of Fuzzy Decision Trees whose individual classifications are aggregated to determine a class for each given example. In Section 2.2.2.1 the Fuzzy Random Forest construction process is explained, and Section 2.2.2.2 explains the fusion of the outputs of each FDT to obtain the final output class of the FRF.

2.2.2.1 Construction of a Fuzzy Random Forest

The process used to construct a FRF using the approach of (Saleh, Błaszczyński, et al., 2018) using the FDT presented in 2.2.1 is the following:

- **Step 1**: bootstrap aggregating, also named bagging, is used to select a random subset of the training examples. The distribution of examples among both classes is kept balanced (i.e., there are the same number of negative and positive examples), and a bootstrap size of two-thirds of the training dataset is used to train each FDT in the forest.
- **Step 2**: each FDT is trained using its corresponding bootstrapped examples. When splitting a tree node, a random subset of the attributes is checked instead of the whole set of attributes. The parameter determining the number of attributes is γ.

A large enough number of FDT has to be trained, n. In (Saleh, Błaszczyński, et al., 2018) parameters were fixed through experimental testing as $\gamma = 2$ and $n = 100$.

It is a common problem in medical datasets to have an imbalance toward the negative class. That is, there are much more examples belonging to the negative class than to the positive one. This issue needs to be taken into consideration when developing machine learning models. Because of this overrepresentation of the negative class, models might not learn appropriately the patterns of positive examples. In Random Forests, the imbalance problem is tackled by the bagging step that creates the data subsets to train the FDTs. Because each FDT receives a balanced subset, the data imbalance will inherently not be a concern for the ensemble.

2.2.2.2 Fusion in a Fuzzy Random Forest

Once all the FDTs have made a prediction about the output class, as explained in 2.2.1.2, all the predictions on the ensemble are aggregated to decide the final class assignment and its support. An ensemble formed by n FDTs has a set of n predicted classes, each with a support value: $(P_1, S_1), (P_2, S_2), ..., (P_n, S_n)$

In binary classification, the final class assignment is made similarly to the selection of the class in a FDT, with a comparison of the support obtained by the two classes of retinopathy, in this case, the votes (i.e., the number of trees that assign the class C_k, denoted as v_k). In (Saleh, Błaszczyński, et al., 2018) the parameter δ_2 was introduced to detect cases where the difference in votes between the two classes is not significant. If the difference in votes is lower than δ_2, the *Unknown* category is returned by the classification model to avoid mistakes. So, the final class B is obtained as follows:

$$B = \begin{cases} "Unknown", & \text{if } |v_0 - v_1| < \delta_2 \\ argmax_{C_k \in \{C_0, C_1\}}(v_k), & \text{otherwise} \end{cases} \tag{2.10}$$

2.2.2.3 Degree of support

Together with the predicted class, B, the FRF calculates a decision support value for the prediction. This support is obtained from the corresponding support values S_i given by each Decision Tree. An arithmetic average is used as the aggregation operation.

2.3 Evaluation measures

Evaluation measures are used to assess the performance of a classification model on a given dataset. There are several types of evaluation measures that can be used, and some of them are specific to the binary or multiclass classification cases. In the following subsections, the evaluation metrics that are used throughout the document to evaluate the performance of the different proposals are explained.

2.3.1 Confusion matrix

A confusion matrix is a table that represents the performance of a classification model. It shows how many instances were correctly and incorrectly classified. The columns of the table represent the instances of the predicted classes, whereas the

rows represent the instances of the ground truth classes. In the binary case, there are four possible outcomes, which are the following ones:

- **True Positive** (TP): the instance is correctly classified as belonging to the positive class.
- **True Negative** (TN): the instance is correctly classified as belonging to the negative class.
- **False Positive** (FP), Type I error: the instance is incorrectly classified as belonging to the positive class.
- **False Negative** (FN), Type II error: the instance is incorrectly classified as belonging to the negative class.

They are represented in a two by two confusion matrix, as can be seen in Table 2.1.

TABLE 2.1: Confusion matrix for binary classification

		Predicted class	
		Class 0	**Class 1**
Ground truth	**Class 0**	TN	FP
	Class 1	FN	TP

Similarly, for the multiclass case, a confusion matrix M can also be represented as an n by n confusion matrix, where $n = \#classes$. Each element in the matrix can be denoted as $M_{i,j}$, where i is the ground truth class and j the predicted class. The main diagonal of the matrix, when $i = j$, are the correctly predicted instances. An example of a multiclass confusion matrix where $n = 4$ is depicted in Table 2.2.

TABLE 2.2: Confusion matrix for multiclass classification

		Predicted class			
		Class 0	**Class 1**	**Class 2**	**Class 3**
Ground truth	**Class 0**	0, 0	0, 1	0, 2	0, 3
	Class 1	1, 0	1, 1	1, 2	1, 3
	Class 2	2, 0	2, 1	2, 2	2, 3
	Class 3	3, 0	3, 1	3, 2	3, 3

To calculate class-wise metrics, a multiclass confusion matrix can be considered as a binary confusion matrix for a specific class, also known as a one-vs-all matrix. Let us consider c the class for which we want to obtain the one-vs-all matrix. The binary confusion matrix would be obtained as indicated in Eq. 2.11.

$$\begin{aligned} TP_c &= \sum M_{i,j}, \text{ such that } i = c \text{ and } j = c \\ TN_c &= \sum M_{i,j}, \text{ such that } i \neq c \text{ and } j \neq c \\ FP_c &= \sum M_{i,j}, \text{ such that } i \neq c \text{ and } j = c \\ FN_c &= \sum M_{i,j}, \text{ such that } i = c \text{ and } j \neq c \end{aligned} \tag{2.11}$$

2.3.2 Binary evaluation metrics

Several metrics can be used to evaluate binary classification models. One of the most commonly used indices is accuracy, Eq. 2.12. It measures how closely the predictions match the ground truth class labels. It also provides an overall idea of whether the model is properly predicting both classes. Furthermore, it could be misleading in cases with imbalanced datasets, where one of the classes has many more instances than the other one. In those cases, its results should be analysed together with other metrics.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.12}$$

The sensitivity, or recall, measures the proportion of positive examples the classifier was able to correctly predict, Eq. 2.13. It is quite important in the medical domain, as it is focused on the FN, which is the most undesirable case, because the model is predicting an unhealthy patient being healthy. As it only evaluates the positive examples, imbalanced data does not affect it.

$$Sensitivity = Recall = \frac{TP}{TP + FN} \tag{2.13}$$

Specificity is similar to sensitivity, but it measures the proportion of negative examples being correctly predicted, Eq. 2.14.

$$Specificity = \frac{TN}{FP + TN} \tag{2.14}$$

One option to summarise the performance of a binary classifier when dealing with imbalanced data is the balanced accuracy, Eq. 2.15. Because it is the arithmetic mean of sensitivity and specificity, the class imbalance does not affect it.

$$BA = \frac{Sensitivity + Specificity}{2} \tag{2.15}$$

The precision quantifies the proportion of positive examples that are correctly identified, Eq. 2.16. It is useful to detect if the model has a high amount of FP. In the

medical domain, it is not as critical as sensitivity, but it should also be minimised to avoid considering healthy patients as unhealthy.

$$Precision = \frac{TP}{TP + FP} \tag{2.16}$$

F1-Score summarises both precision and recall by computing their harmonic mean, Eq. 2.17. If one of them has a bad performance, the overall F1-Score will be penalised.

$$F1\text{-}Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{2.17}$$

2.3.3 Multiclass evaluation metrics

For the multiclass case, where there are more than two classes, the binary classification metrics are adapted to deal with more classes. Initially, metrics are computed for each of the possible classes by using the one-vs-all matrices, and secondly, they are aggregated.

A first aggregation is done using a micro-average. In this case, the same equations as in the binary case are used, and we take each possible outcome (i.e., TP, TN, FP, FN) as the sum of those outcomes for each of the classes. Micro-averaging is computing the proportion of correctly classified examples out of all examples; hence, it is considered as the accuracy. The same result would be obtained by micro-F1, micro-Precision and micro-Recall. In Eq. 2.18 the multiclass average is computed as the micro-F1. It does not take into account the different size each class might have, and consequently, it might not be the most appropriate metric for imbalanced datasets.

$$micro\text{-}F1 = \frac{2 \cdot \sum_{c \in C} TP_c}{2 \cdot \sum_{c \in C} TP_c + \sum_{c \in C} FP_c + \sum_{c \in C} FN_c} \tag{2.18}$$

A second possible aggregation is the macro-average. It consists on calculating the arithmetic mean of each of the class-wise values for a given metric. For instance, to calculate the macro-Precision, we would use Eq. 2.19. For imbalanced data, because the macro-average treats each class with equal significance, it is a good choice to evaluate the classifiers.

$$macro\text{-}Precision = \frac{\sum_{c \in C} Precision_c}{p} \tag{2.19}$$

Another aggregation method is the weighted-average. It takes the support of each class as its weight when computing the mean of the measures of each class. Support is the proportion of examples each class has over the total number of examples. In the imbalanced case, it will give more weight to the classes with more examples. An example of the weighted-Recall is shown in Eq. 2.20.

$$weighted\text{-}Recall = \sum_{c \in C} (Recall_c \cdot \frac{|C|}{m}) \tag{2.20}$$

When the classes are ordered, another metric that can be used is the quadratic weighted kappa, κ. It is an index of agreement between the predictions and the ground truth labels that accounts for the proximity between the classes. It is a relevant metric for ordinal multiclass problems because it allows to define different penalisation for mistakes between classes depending on the distance between them (J. D. L. Torre, Puig, et al., 2017). In medical decision support, a short difference between the correct class and the predicted one is crucial in order to not affect the health of the patient. Hence, the aim is to minimise it as much as possible. Following the kappa interpretation of Landis and Koch (Landis and Koch, 1977), a kappa in the interval $[0.61 - 0.8]$ describes a substantial strength of agreement between the predictions and the ground truth. Values greater than 0.8 indicate an almost perfect agreement, which would be a strong indicator of a good medical decision support system. It involves three different matrices. The matrix of predictions (x_{ij}); the matrix with the ground truth (m_{ij}); and the matrix of weights (w_{ij}). The latter contains the distance of disagreement between the predictions and the ground truth squared. The calculation is shown in Eq. 2.21.

$$\kappa = 1 - \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} w_{ij} x_{ij}}{\sum_{i=1}^{k} \sum_{j=1}^{k} w_{ij} m_{ij}} \tag{2.21}$$

2.4 Conclusions

In this chapter, we have presented the Clinical Decision Support System Retiprogram, which is the preliminary work for this thesis. First, the usage of fuzzy logic for the diagnosis of DR and the reasons for using it are motivated. Second, the clinical and analytical variables chosen by the medical experts to create this CDSS are explained. Then, the algorithm to construct the binary classifier based on a Fuzzy Random Forest is explained. It includes the induction process of the Fuzzy

Decision Trees that compose the Fuzzy Random Forest, as well as the classification process for new patients. Finally, the measures that will be used throughout the thesis to evaluate the performance of the new classification models have been defined.

UNIVERSITAT ROVIRA I VIRGILI
FUZZY-BASED MACHINE LEARNING METHODS FOR CONTINUOUS DIAGNOSIS AND PROGNOSIS OF DIABETIC RETINOPATHY
Jordi Pascual Fontanilles

Chapter 3

Dynamic improvement of a Fuzzy Random Forest

3.1 Introduction

In dynamic environments, after developing and deploying a classification system, the institution (e.g., company, hospital) may gather additional labelled data over time. This occurs in scenarios like Retiprogram, where physicians use the recommendation given by the system during the diagnosis, but they finally take the decision about the DR condition on each patient and introduce their diagnostic decision in the system. Therefore, additional labelled examples can be collected as new patients arrive. The challenge is, then, how to enhance the performance of a binary classifier by leveraging these newly acquired samples.

In this chapter, we propose a novel methodology that takes advantage of the additional data arriving at decision support systems based on Fuzzy Random Forests. When a sufficient amount of new information has been gathered, the new data will be used to update and refine the initial FRF model with the aim of improving its performance. Furthermore, this method may also reuse training examples that the FRF was unable to classify correctly. Although initially tested on an FRF, the methodology presented can be applied to standard Random Forests as well.

The proposed updating model for FRF has been evaluated using two different datasets. The first one is the DR dataset, which focuses on identifying Diabetic Retinopathy in patients based on their medical history. The second dataset used is related to the detection of occupancy within an office room (Candanedo and Feldheim, 2016). The objective here is to determine whether a given office room is

occupied or not, which can be achieved by analysing various environmental variables such as temperature, light levels, and CO2 concentration. The reasoning with environmental variables can be done in a qualitative way on a fuzzy scale rather than a numerical one. In this case, the data is also highly imbalanced towards the negative class (i.e., unoccupied rooms). The dataset has been split to simulate the arrival of new labelled data into the system. In both cases, the number of additional examples is assumed to be scarce as the frequency of use of the system is low.

Experimental results were obtained on both datasets, with numerical input variables being fuzzified. The output of the FRF classifier is determined based on which class has a higher activation level. This approach allows for the use of standard metrics to evaluate the performance of the model. In this case, weighted balanced accuracy was employed as the primary metric due to its effectiveness in addressing the trade-off between sensitivity and specificity that arises from class imbalance. Various tests were conducted to assess the performance of the model during multiple iterations of the updating method. The results demonstrated significant improvements when utilizing the proposed algorithm, highlighting its potential applicability for other classification tasks involving imbalanced datasets.

The rest of the chapter is organized as follows: Section 3.2 presents other approaches used to update Fuzzy Random Forests, consisting of adding weights to the decision trees or building trees dynamically from streaming data. In Section 3.3, we introduce the proposed method for iteratively updating the trees in a FRF. In Section 3.4, we present the datasets and discuss the obtained experimental results. Finally, Section 3.5 presents the conclusions and the lines of future work.

3.2 Related work

The optimization of classification models based on Random Forests has been studied in the literature. Even though most techniques are not fuzzy, they could also be applied to Fuzzy Decision Trees or Fuzzy Random Forests. We can distinguish two main approaches: adding weights to the Decision Trees, or building online new Decision Trees. The former is summarized in subsection 3.2.1, whereas the latter is presented in subsection 3.2.2. Finally, in 3.2.3 an analysis of these methods is performed.

3.2.1 Adding weights to Random Forests

Weighting some components of the classification model is one possible way to achieve better performance. During the training stage, weights may be added to the model in four different ways. The first one consists of adding weights at the last step of the FRF classifier when a voting procedure is made to find the majority class (Winham, Freimuth, et al., 2013; El Habib Daho, Settouti, et al., 2014; H. B. Li, W. Wang, et al., 2010). Each tree in the ensemble has a weight that corresponds to its accuracy. The accuracy is obtained by calculating the performance of the tree on the out-of-bag samples. Other possibilities consist of changing the weights during the training stage. For instance, Dogan and Birant (Dogan and Birant, 2019) proposed initializing all the weights to the same value and rewarding the best-performing trees on the validation set formed by the out-of-bag samples. Zhukov et al. (Zhukov, Sidorov, et al., 2017) added a pruning step to replace the worst Decision Tree, so the ensemble can handle concept drift. Decision Trees may 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. This first option to add weights on RF is marked in green on Fig. 3.1.

The second option consists of weighting the samples. Kim et al. (Hyunjoong Kim, Hyeuk Kim, et al., 2011) 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 (C. Yang and Yin, 2019) considered finding the weight of the Decision Trees an optimization problem. For a certain number of epochs, both the weights on the samples and the decision trees are updated to optimize the model. The weighted samples are in red on Fig. 3.1.

The third possibility, proposed by Zhong et al. (Zhong, H. Yang, et al., 2020), 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. A similar approach is proposed by Khan et al. (Khan, Shin, et al., 2008), in which they applied weights to the rules on Fuzzy Decision Trees. The rule weight is considered to be the certainty factor, which is computed for each branch of the FDT using the training data. The fuzzy rule with the maximum membership value for a sample to be classified is the one that decides the final class. The leaves of the Decision Trees are shown in purple on Fig. 3.1.

Finally, there are weighting methods that use different weights for each of the possible output classes. For instance, Zhu et al. (Zhu, Xia, et al., 2018), Livieris et al. (Livieris, Kanavos, et al., 2019) and Utkin et al. (Utkin, Kovalev, et al., 2019) use this approach to compensate imbalanced datasets. Output classes are shown in blue on Fig. 3.1.

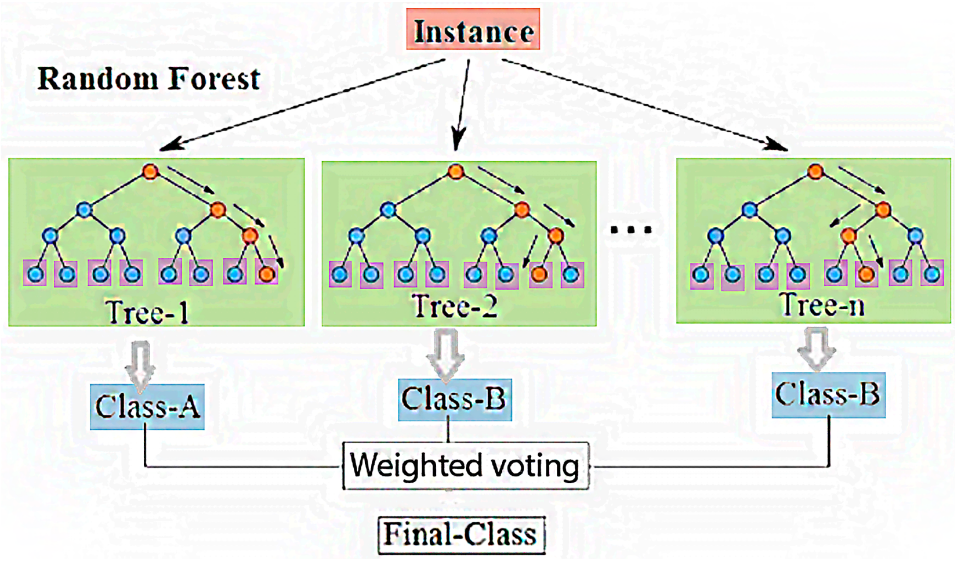


FIGURE 3.1: Diverse ways of adding weights to Random Forests

3.2.2 Online Random Forests

The second kind of method deals with so-called Online Random Forests. They differ from conventional RFs in that they are dynamically constructed and optimized using streams of continuous data. In this case, the training data arrives and is processed continuously. As 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. Gomes et al. (Heitor Murilo Gomes, Barddal, et al., 2017) reviewed several methods for data stream classification using ensemble-based methods and proposed a taxonomy to classify them.

Incremental Decision Trees are one type of online learning method. This type of Decision Tree can be grown in an online fashion, i.e., their rules can be updated using new data examples. For instance, Kalles and Morris (Kalles and Morris, 1996) and Utgoff et al. (Utgoff, Berkman, et al., 1997) proposed variants of ID3,

which are incremental. Regarding fuzzy approaches, Guetova et al. (Guetova, Hölldobler, et al., 2002) proposed a fuzzy incremental variant of ID3. Ichihashi et al. (Ichihashi, Shirai, et al., 1996) also proposed an incremental variant of ID3, Neuro-Fuzzy ID3, in which they considered the membership function as a three-layered neural network. Isazadeh et al. (Isazadeh, Mahan, et al., 2016) and Pecori et al. (Pecori, Ducange, et al., 2020) also proposed different approaches based on Very Fast Decision Trees to train an ensemble of fuzzy incremental Decision Trees using streaming data.

Saffari et al. (Saffari, Leistner, et al., 2009) combined online bagging techniques with extremely randomized forests to build an Online Random Forest. The trees in this Random Forest grow when new data is fed into the model. A new branch on the tree is created when there are enough samples on a node, and they are good enough to classify new samples. Similar proposals for Online Random Forests include Mondrian Forests (Lakshminarayanan, Roy, et al., 2014) and Adaptive Random Forests (Heitor M. Gomes, Bifet, et al., 2017). They are also based on growing the trees' branches (i.e., the rules of the trees) with the arrival of new training samples. Some incremental approaches also drop some members of the ensemble and create new ones. Their objective is to handle concept drifts, and their main focus is on processing streams of data.

3.2.3 Analysis of the related work

The two approaches analysed to improve the 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 out-of-bag training samples. As the optimization is done during the training, the core of the model is not modified a posteriori. Published results show that these approaches are able to improve the performance of a standard Random Forest. Despite the fact that updating the weights using new data is rarely done, these weighting methods could be used to update the core model using new data. However, they would not learn new patterns, as these techniques do not create new rules.

On the other hand, the main drawback of Online Random Forests is that they require 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 while it is being used. They are well suited for applications that have to process continuous streams of data, as they need large amounts of data to be trained.

The main difference between these approaches and the method we propose is that we want to first create a classification system with a FRF, and later this FRF will be dynamically updated from time to time using new small sets of data. Some similarities exist with online methods. Specifically, with incremental approaches, according to the taxonomy proposed in (Heitor Murilo Gomes, Barddal, et al., 2017). They are designed to work on data streaming and are focused on handling concept drift. In contrast, the kinds of problems we aim to solve do not include the use of data streams, and they do not have any concept drift. The characteristics of our datasets are explained in more detail in Section 3.4.1.

3.3 Proposed method

The method we propose to update a FRF using a new set of incoming data consists of modifying the set of trees that compose the FRF model, taking into account new examples that were not initially available. The goal is to try to increase the performance by updating the classification model without retraining it from scratch. This updating process will be performed after collecting a sufficiently large set of new cases that can be used as examples for improving the model. The proposed architecture is illustrated in Fig. 3.2.

Once the base classification model has been constructed (first box in the figure), the dynamic components can be activated when new data collected for updating the trees is available. To improve the results of the base model and to enlarge the collected data in each update iteration, as the first step, the use of previous misclassified examples (i.e., errors) is proposed. Three different ways of dealing with errors are studied. They are called *NoError*, *ErrorLT* (Errors from the Last Test), and *AllData*&*ErrorLT*, depending on which error examples are used during the dynamic updating. *NoError* and *ErrorLT* use the new data that arrived at the system in the current update step. In contrast, *AllData*&*ErrorLT* uses all the new data that arrived at the system in all the previous updating steps. Regarding the misclassified examples, *NoError* does not include them, whereas *ErrorLT* and *AllData*&*ErrorLT* include the misclassified examples of the previous update step.

As a second dynamic component of the methodology, we consider the ensemble voting procedure of the Fuzzy Random Forest. Two techniques are usually applied. The first one is the majority voting, in which the most-voted prediction among the Fuzzy Random Forests in the ensemble is the final answer. The second one

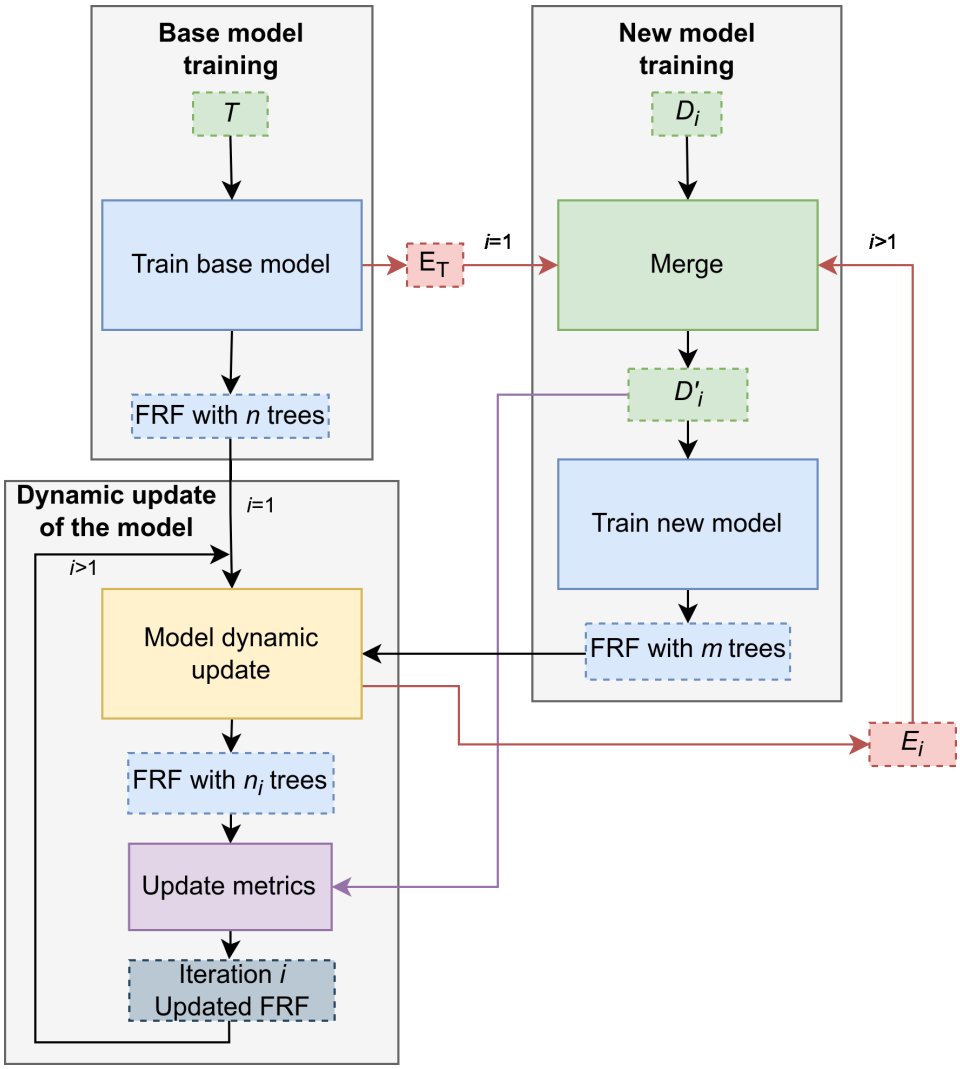


FIGURE 3.2: Architecture of the iterative learning of Fuzzy Random Forests

is weighted voting, in which the vote of each Fuzzy Decision Tree is weighted according to its performance.

As a third dynamic component, we consider the possibility of updating the metrics of the Random Forest using the new data and, then, using these quality values to update the model and as weights in the weighted voting.

Algorithm 1 shows the methodology for the construction and update of the FRF. The procedure is composed of three steps. The first one consists of the training of

the base classification model, and it is only run once at the beginning. The other two steps are executed iteratively each time we collect a new set of examples, and the model can be updated.

Algorithm 1 FRF iterative update algorithm

Input: T, $method$, $doUpdateWeights$
1: $updatedModel, E_T \leftarrow modelTrain(T)$ // Base model training
2: $i \leftarrow 1$ // Current update iteration
3: $D_i \leftarrow waitForNewData()$ // Data used to update the model
4: $E_{i-1} \leftarrow none$ // Error data with misclassified samples
5: **if** $method$ is *ErrorLT* or $method$ is *AllData&ErrorLT* **then**
6: $E_{i-1} \leftarrow E_T$
7: **end if**
8: **repeat** // Iterative model update process
9: **if** $method$ is *NoError* **then**
10: $D'_i \leftarrow D_i$
11: **else if** $method$ is *ErrorLT* **then**
12: $D'_i \leftarrow merge(D_i, E_{i-1})$
13: **else if** $method$ is *AllData&ErrorLT* **then**
14: $D'_i \leftarrow merge(D_{0..i}, E_{i-1})$
15: **end if**
16: $newModel \leftarrow modelTrain(D'_i)$
17: $updatedModel, E_i \leftarrow modelUpdate(updatedModel, newModel)$
18: **if** $doUpdateWeights$ is *true* **then**
19: $updatedModel \leftarrow updateWeights(updatedModel, D'_i)$
20: **end if**
21: $i \leftarrow i + 1$
22: **until** $(D_i \leftarrow waitForNewData()) == none$ // Repeat if there is new data

The three steps of the proposed methodology are commented on in more detail as follows:

1. **Base model training:** The first step consists of training the base model with a large training dataset, T, which contains labelled examples (line 1). With a learning algorithm for Fuzzy Random Forests, we obtain n FDTs, where n is a large number, usually more than 100. During the construction process, the out-of-bag samples of each FDT are used to compute two metrics for each of them, the specificity and the sensitivity, which are stored on the FDT. Those metrics have two main purposes. The first one is to be used in the weighted voting if this option is selected. They are also used in the third step of the proposed method, the update process. The obtained classification model should be validated with a testing dataset in order to ensure its good

performance (this step is not shown in Fig. 3.2). After creating the base FRF, the training dataset can be used for testing, and the samples that are not correctly classified are stored in a file E_T. Those error samples E_T are used in the following step in the *ErrorLT* and *AllData*&*ErrorLT* versions.

2. **New model training:** Every time enough new samples D_i have been gathered, usually around 200 samples, a new training iteration i is performed (lines 2-16). The merge process generates the dataset used to update the FRF. Its output, D'_i, depends on the method version used:

   - The *NoError* version does not merge anything with D_i (line 10).
   - In contrast, the *ErrorLT* version merges the errors data from previous iterations with D_i (line 12). For the first training iteration $i = 1$, the E_T errors samples are merged. In further iterations, the merged errors samples, E_i, are generated in the third step of the method.
   - The *AllData*&*ErrorLT* version is an extension of *ErrorLT*. It additionally merges the training data from all previous iterations, $D_{0..i}$ (line 14).

   Once the merge process is completed, the D'_i samples are used to train a new FRF (line 16). The difference between the base model and this newly generated one is the number of trees. Because the size of the new training set is small, we train a lower amount of FDTs m, with $m <<< n$, usually around 20 trees. Their out-of-bag samples are also used to compute the aforementioned metrics for each of these new trees. They are also used for the weighted voting and in the third step of the proposed method.

3. **Dynamic update:** The current model is updated in this step (lines 17-20). If the iteration is the first one ($i = 1$), the base model is updated. In further iterations where $i > 1$, the model being updated is the resulting model from the previous iteration $i - 1$. The m new FDTs trained in the previous step are used to update the current model (line 17). To do so, the m FDTs are added to the current model. To improve the performance of the updated FRF, the worst FDTs from those $n_{i-1} + m$ trees are removed. The number of trees being removed is fixed by a certain percentage p. The updated model will have a total of $(1 - p/100) * (n_{i-1} + m)$ FDTs. To sort the trees and keep the best ones, a quality metric is used: weighted balanced accuracy (Eq. 3.1). It is defined as a weighted average between specificity and sensitivity with a weighting factor α.

$$WBA = \alpha \cdot sensitivity + (1 - \alpha) \cdot specificity \tag{3.1}$$

After pruning the worst trees, an additional update of the metrics can optionally be performed (lines 18-20). When this option is selected, the metrics computed using the out-of-bag samples are updated. The training data D'_i of the current iteration i is used to compute the quality metrics for each of the FDTs. Instead of replacing the old metrics, the average between the old and the new metrics is computed: $NewMetric = (CurrentMetric + UpdatedMetric)/2$. This allows a more gradual update of the metrics.

The resulting FRF with n_i trees is set as the current model, and it is taken as the new model to be used until a new set of cases is available and a new update iteration starts.

The errors of the updated FRF model on the D'_i dataset may also be retrieved and stored in E_i as was done for the base model with E_T. In the case of using the *ErrorLT* or *AllData&ErrorLT* versions, in the next iteration, the new samples D_i are merged with those error cases E_i in order to enlarge the training dataset of the subsequent iteration.

The use of the sets of wrongly classified examples E_i is optional. It is only used in the *ErrorLT* and *AllData&ErrorLT* versions of the iterative method. Their use has two purposes. On the one hand, to increase the size of the training set D'_i and, on the other hand, to show these wrongly classified cases again 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 past errors. In the next section, the effects of the diverse configurations of the proposed method are studied.

3.4 Experimental results

In this section, the obtained experimental results are shown. In subsection 3.4.1 the tested datasets are explained and analysed. In 3.4.2 the selection of the method parameters is explained. Subsection 3.4.3 shows the obtained results and analyses them. Finally, in 3.4.4 an in-depth analysis of some results is performed to study how the proposed method modifies the FRF model.

3.4.1 Datasets

Two different datasets have been used to test the proposed iterative method for updating a FRF. The first one is the Diabetic Retinopathy risk detection problem. This is a private dataset from the Hospital Sant Joan de Reus, located in Catalonia, Spain. It is a binary classification problem with two labels: $DR = 1$ means a high risk of suffering from Diabetic Retinopathy (i.e., the positive class), whereas $DR = 0$ means a low risk (i.e., the negative class). The experiments have been performed using real data from diabetic patients. This data includes nine different attributes: six numerical (Age, Evolution time of diabetes, HbA1c, CKDEPI, Microalbuminuria, and Body Mass Index) and three categorical (Sex, Medical Treatment, and Hypertension). The target attribute is the label of the class $DR = 0$ or $DR = 1$.

The second one is the Occupancy (OC) dataset (Candanedo and Feldheim, 2016), in which the occupancy of an office room is predicted. The dataset is publicly available at the UCI Machine Learning Repository. It is also a binary classification problem with two labels. $OC = 0$ means the office room is not occupied (i.e., negative class), whereas $OC = 1$ means the office room is occupied (i.e., positive class). The occupancy dataset has six different attributes: five numerical (Temperature, Humidity, Light, CO2, and Humidity Ratio) and one categorical (Date). The target attribute is the label of the class $OC = 0$ or $OC = 1$.

The data from both problems is split into three different datasets: training, validation and testing. The training dataset, T, is used to train the base FRF 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 in the system from time to time. We split the validation set into chunks of 200 samples. Each of them is used in a different iteration i during the dynamic updating process. From each of these small new training datasets, D_i, the system generates 20 new trees. Then, the dynamic updating step is done, obtaining the FRF model M_i.

Finally, the testing set is used after each iteration to check the performance of the new FRF M_i. Note that the samples from this testing dataset are not included in the error sets; thus, the model is never trained using them.

Tables 3.1 and 3.2 show the splitting of the data among the three datasets for Diabetic Retinopathy and Occupancy problems, respectively. It can be seen that both datasets are highly imbalanced toward the negative class.

To obtain the experimental results, we used the aforementioned datasets to build a FRF for each of the problems. Because of the use of a FRF, the rules use

TABLE 3.1: Diabetic Retinopathy patients data

Dataset	Training	Validation	Testing	Total
DR=0 samples	1376 (72%)	380 (63%)	863 (78%)	2619
DR=1 samples	537 (28%)	222 (37%)	240 (22%)	999
Total samples	1913	602	1103	3618

TABLE 3.2: Office room occupancy data

Dataset	Training	Validation	Testing	Total
OC=0 samples	3064 (78%)	1583 (79%)	1293 (79%)	5940
OC=1 samples	844 (22%)	417 (21%)	336 (21%)	1597
Total samples	3908	2000	1629	7537

fuzzy variables; therefore, the datasets had to be fuzzified. The labels and fuzzy sets for the Diabetic Retinopathy problem have been defined from the numerical attributes (Romero-Aroca, Valls, et al., 2019) by medical experts. The numerical attributes for the occupancy problem were fuzzified using Yuan's algorithm (Yuan and Shaw, 1995). A set of k centres are used to define the membership function for each linguistic label. They start being evenly distributed among the numerical values of the attribute. Then, they are adjusted through an iterative process to reduce the distance between the centres and the numerical values. For each numerical attribute of the occupancy dataset, $k = 5$ linguistic labels have been computed (Very Low, Low, Medium, High and Very High). In both cases, the training algorithm for FRFs that we have used to test the proposed method is the one explained in Chapter 2.

3.4.2 Parameter selection

The main goal of the FRF model update is to improve the sensitivity results on datasets that are highly imbalanced towards the negative class. In such situations, it is hard for the classifiers to detect the positive instances.

In the particular case of the Diabetic Retinopathy disease, doctors want to improve the detection of patients at risk of developing DR, that is, improve the sensitivity of the Random Forest. Therefore, it is preferred to misclassify non-DR patients as having the risk of developing the disease (False Positive) rather than the other way around (False Negative). This is due to the very bad consequences of not detecting DR on time, which produce a degradation of the vision that may even cause total blindness.

The update method has two parameters: the percentage of trees changed at each iteration, p, and the balancing factor in the calculation of the weighted balanced accuracy, α. After performing an empirical experimentation (Pascual-Fontanilles, Valls, et al., 2021), the percentage was fixed to $p = 10\%$, and the weight in the balanced accuracy to $\alpha = 2/3$. The experiments showed that for $p < 10\%$ the changes in the model were not significant, whereas for higher values, the model suffers too many changes and becomes highly unstable. Regarding the parameter α, the value of $2/3$ showed a good trade-off to prioritize the improvement of the sensitivity performance without worsening the specificity.

The same parameters were used for the Occupancy dataset. Again, in this problem, the minority class is the positive one, and sensitivity is the main target, as the goal is to detect if there are people in the room or not.

3.4.3 Results

This section presents the results of the experimentation with the two datasets. We have tested the different configurations of the method, which include:

- The data used in the updating method: *NoError* (*N*), *ErrorLT* (*E*) or *All-Data*&*ErrorLT* (*A*);
- The voting method: *majority* (*M*) or *weighted* (*W*) voting;
- Whether the metrics are updated at each iteration (*Y*) or not (*N*).

This leads to 12 possible configurations. The obtained sensitivity and specificity results from the Diabetic Retinopathy dataset can be seen in tables 3.3 and 3.4 respectively. The results obtained from the Occupancy dataset can be seen in tables 3.5 and 3.6.

The columns Vx denote the different new validation sets of data received in each iteration. Note that the *NoError* method in the DR dataset has one iteration less than the other configurations. This is due to the fact that this method has fewer samples because its data is not expanded using the error samples from previous iterations.

Due to the amount of data available, only four updates can be done in the Diabetic Retinopathy dataset, while for the Occupancy case, we could perform up to six updates.

TABLE 3.3: Diabetic Retinopathy sensitivity results

Method	Vote	Update	V0	V1	V2	V3	V4
N	M	N	74.2	77.5	80.4	81.7	-
N	M	Y	74.2	77.5	79.2	80	-
N	W	N	74.6	77.5	81.7	82.1	-
N	W	Y	74.2	77.5	79.2	79.2	-
E	M	N	74.2	78.3	80.8	84.6	85.4
E	M	Y	74.2	78.3	81.2	86.7	87.1
E	W	N	74.6	77.1	81.7	84.6	87.5
E	W	Y	74.2	78.3	82.9	85.8	86.3
A	M	N	74.2	78.3	79.6	85.8	87.9
A	M	Y	74.2	78.3	81.7	87.5	90
A	W	N	74.6	77.1	81.7	83.8	88.7
A	W	Y	74.2	78.3	83.3	86.7	**91.7**

TABLE 3.4: Diabetic Retinopathy specificity results

Method	Vote	Update	V0	V1	V2	V3	V4
N	M	N	81.7	78.7	78.4	78.2	-
N	M	Y	81.7	78.7	78.1	77.5	-
N	W	N	81.5	78.6	79.4	78.6	-
N	W	Y	81.7	78.6	78	77.4	-
E	M	N	81.7	79.4	81.2	84.2	86.6
E	M	Y	81.7	79.4	81.2	83.7	87.7
E	W	N	81.5	79.1	80.9	83.9	86.1
E	W	Y	81.7	79.4	83	87.5	**91.5**
A	M	N	81.7	79.4	80.8	83	83.2
A	M	Y	81.7	79.4	80.4	83	83.5
A	W	N	81.5	79.1	81.1	82.9	84.2
A	W	Y	81.7	79.4	82.3	84.7	84.4

A first analysis can be performed on the overall results for the two datasets. The best sensitivity and specificity value in each table is marked in bold. *NoError* is the method obtaining the worst improvements in both sensitivity and specificity. The highest improvements in sensitivity and specificity in both datasets are obtained by the *AllData*&*ErrorLT* and *ErrorLT* methods, respectively.

It can be observed that by accumulating the data received in the previous iterations (in the *AllData*&*ErrorLT* method), we can further improve the sensitivity, keeping a reasonably good level of specificity. In contrast, the *ErrorLT* method does not improve the sensitivity as much, although it does improve the specificity. These conclusions are consistent in both case studies. For case A-W-Y, we have a remarkable improvement for the last version in comparison to the base model

TABLE 3.5: Occupancy sensitivity results

Method	Vote	Update	V0	V1	V2	V3	V4	V5	V6
N	M	N	74.7	74.7	75.9	76.2	76.5	76.5	76.5
N	M	Y	74.7	74.7	76.2	76.2	76.5	76.2	76.2
N	W	N	74.7	74.7	77.1	76.2	76.8	76.5	76.5
N	W	Y	74.7	74.7	76.8	76.2	76.5	76.2	76.2
E	M	N	74.7	75.9	75.9	78	78.9	81.5	88.4
E	M	Y	74.7	75.9	76.2	83.6	86.6	84.5	83.3
E	W	N	74.7	75.9	76.5	76.8	80.1	82.1	87.5
E	W	Y	74.7	75.9	76.5	82.7	86	87.2	85.4
A	M	N	74.7	75.9	76.8	79.2	83	88.7	82.7
A	M	Y	74.7	75.9	76.8	84.8	89.3	86.9	86.9
A	W	N	74.7	75.9	76.8	78.6	80.7	82.7	**89.6**
A	W	Y	74.7	75.9	77.7	85.1	85.7	92.9	86.9

TABLE 3.6: Occupancy specificity results

Method	Vote	Update	V0	V1	V2	V3	V4	V5	V6
N	M	N	87.6	85.5	84.6	84.4	85.5	83.9	83.8
N	M	Y	87.6	85.5	84.6	84.4	83.8	84.4	84.1
N	W	N	87.6	85.6	84.5	84.3	85.4	83.8	83.8
N	W	Y	87.6	85.6	84.5	84.3	85.4	84.3	84.4
E	M	N	87.6	85.4	84.8	87.5	90.4	88.9	83.6
E	M	Y	87.6	85.4	84.6	88.7	88.2	88	91.3
E	W	N	87.6	86.9	84.6	86.5	87.6	90.2	85.8
E	W	Y	87.6	87.1	87.4	86.2	91.2	83.8	**92.3**
A	M	N	87.6	85.4	86.7	85.8	81.1	78.9	82.1
A	M	Y	87.6	85.4	84.3	82.1	85.2	84.5	83.9
A	W	N	87.6	86.9	84.4	85.8	83	82.4	75.9
A	W	Y	87.6	87.1	86.7	88.4	89	84.1	84.6

(V0). In the DR classification, sensitivity increases from 74.2 to 91.7 and specificity from 81.7 to 84.4. In the case of Occupancy detection, a rise from 74.7 to 86.9 is seen in sensitivity, with only a small decrease in specificity, from 87.6 to 84.6.

To further analyse the results comparatively, they are summarised below in three different tables. Each one focuses on one of the features of the method: the update method (*NoError*, *ErrorLT* or *AllData*&*ErrorLT*), the voting method (*majority voting* or *weighted voting*), and whether metrics are updated or not. In each table, we study the difference between the initial sensitivity and specificity values and the final values in the last iteration of the method. There are two values in each cell. The left one is the difference between the worst result and the initial value. The right one is the difference between the best result and the initial value.

In both cases, the difference is a percentage, and it is computed flooring the results. This is performed because the weighted voting has slightly different initial results. This range of values represents the minimum and maximum change on the indicator with respect to the base original model. By analysing the range of variation obtained by each of the methods, an overall view of how each parameter affects the quality of the results can be seen.

Table 3.7 displays the results of the different updating versions. On the DR dataset, we can see that the *NoError* method improves the sensitivity values at the cost of slightly decreasing the specificity ones. The *ErrorLT* and *AllData*&*ErrorLT* methods, in contrast, increase both the sensitivity and specificity values. On the Occupancy dataset, we can see improvements in all methods in sensitivity but a decrease in the specificity values. Better results for sensitivity than specificity were expected because of the prioritization of sensitivity in the balanced accuracy calculation. In both datasets, the highest increase in the sensitivity values was achieved by the *AllData*&*ErrorLT* method. This is at the cost of having lower improvements on the specificity values when compared with the *ErrorLT* method. The lowest increase in sensitivity is in the *NoError* method. It also does not have better results in terms of specificity. The *ErrorLT* method has intermediate results. Sensitivity values are increased, and specificity increases and slightly decreases on the DR and OC datasets, respectively. We can conclude that the use of error samples in the method gives better results than not using them. Moreover, *AllData*&*ErrorLT* is able to give better sensitivity results than *ErrorLT* at the cost of a worse specificity.

TABLE 3.7: Method summarized improvement results

Dataset	Method	Sensitivity (%)	Specificity (%)
Diabetic Retinopathy	NoError	[6,8]	[-4,-3]
	ErrorLT	[11,13]	[5,10]
	AllData&ErrorLT	[13,17]	[2,3]
Occupancy	NoError	[2]	[-4,-3]
	ErrorLT	[9,14]	[-4,5]
	AllData&ErrorLT	[8,15]	[-12,-3]

The results in Table 3.8 summarise the results of applying either majority or weighted voting in the FRF. It can be seen that the use of weighted voting improves the best result in majority voting tests, but it also decreases the worst value in majority voting tests. After checking the results, the *NoError* method is the one that gets worse results using weighted voting, whereas *ErrorLT* and *AllData*&*ErrorLT*

get better results using it. According to these results, using the error data is also beneficial when computing the weights of each FDT.

TABLE 3.8: Vote method summarised improvement results

Dataset	Vote	Sensitivity (%)	Specificity (%)
Diabetic Retinopathy	Majority	[6,16]	[-4,6]
	Weighted	[5,17]	[-4,10]
Occupancy	Majority	[2,14]	[-5,4]
	Weighted	[2,15]	[-12,5]

Table 3.9 summarises the influence of updating the metrics in each iteration. Updating them keeps similar values for sensitivity, whereas it improves specificity. This update process seems to depend on the other parameters of the method. Observing all the obtained results, if a test configuration that does not update the metrics has good results, the update method is able to improve them even more. The opposite also occurs, so when the results are not that good, the updated metrics worsen them.

TABLE 3.9: Update metrics summarised improvement results

Dataset	Update	Sensitivity (%)	Specificity (%)
Diabetic Retinopathy	No	[7,14]	[-3,5]
	Yes	[5,17]	[-4,10]
Occupancy	No	[2,15]	[-12,-2]
	Yes	[2,12]	[-4,5]

Finally, on Table 3.10, we can see the difference in the results of the last iteration between using Balanced Accuracy (BA) or Weighted Balanced Accuracy (WBA) on all the tested configurations. The WBA is computed with a weighting factor $\alpha = 2/3$ as in the previous tests. They can be compared to the BA and WBA of the base model. In the Diabetic Retinopathy dataset, BA is around 78, and WBA is around 76.8, depending on the results of the weighted voting. In the Occupancy dataset, the BA is 81.15 and the WBA is 79.

As expected, *NoError* has the worst results, which are really similar to the base ones in both BA and WBA. When comparing *ErrorLT* and *AllData&ErrorLT*, the final results of both methods are quite similar in terms of the BA. But, as seen in the previous tables, they balance the sensitivity and the specificity differently. *AllData&ErrorLT* has a greater increase in sensitivity than *ErrorLT*, but the increase in specificity is not as high. This is also shown in the WBA, which is greater than its

corresponding BA in the *AllData*&*ErrorLT*, whereas in *ErrorLT* it maintains similar values.

TABLE 3.10: BA and WBA final results. Initial BA DR is 78 and WBA DR is 76.8; initial BA OC is 81.15 and WBA OC is 79

Method	Vote	Update	BA DR	WBA DR	BA OC	WBA OC
N	M	N	79.95	80.53	80.15	78.93
N	M	Y	78.75	79.17	80.15	78.83
N	W	N	80.35	80.93	80.15	78.93
N	W	Y	78.3	78.6	80.3	78.93
E	M	N	86	85.8	86	86.8
E	M	Y	87.4	87.3	87.3	85.97
E	W	N	86.8	87.03	86.65	86.93
E	W	Y	88.9	88.03	88.85	87.7
A	M	N	85.55	86.33	82.4	82.5
A	M	Y	86.75	87.83	85.4	85.9
A	W	N	86.45	87.2	82.75	85.03
A	W	Y	88.05	89.27	85.75	86.13

3.4.4 In-depth analysis of the updating components on the classification models obtained and their performance

To understand how the proposed method modifies the FRF model, a more detailed analysis of the trees has been performed. The selected configuration for this study is the *AllData*&*ErrorLT* method and *weighted* voting, because it is the configuration that makes more changes to the FRF during its execution. The analysis includes the cases of updating and not updating the metrics after each iteration because the update metrics setting produces even more changes to the FRF.

The first analysis, which we can see in Fig. 3.3 and Fig. 3.4, shows the percentage of trees generated in each iteration that belong to the updated FRF. It can be seen that updating the metrics results in replacing more FDTs from previous iterations than without updating. When there is no update, in the last iteration, a 50% and 60% of the trees are kept from the base model in the Diabetic Retinopathy and the Occupancy datasets, respectively. This percentage is lowered to 40% and 30% when metrics are updated at each iteration. As a consequence of this difference, more trees generated in subsequent iterations are present in the last iteration.

Despite this difference in both configurations, in general, it does not occur that trees are added in one iteration and removed in the following one. That means the

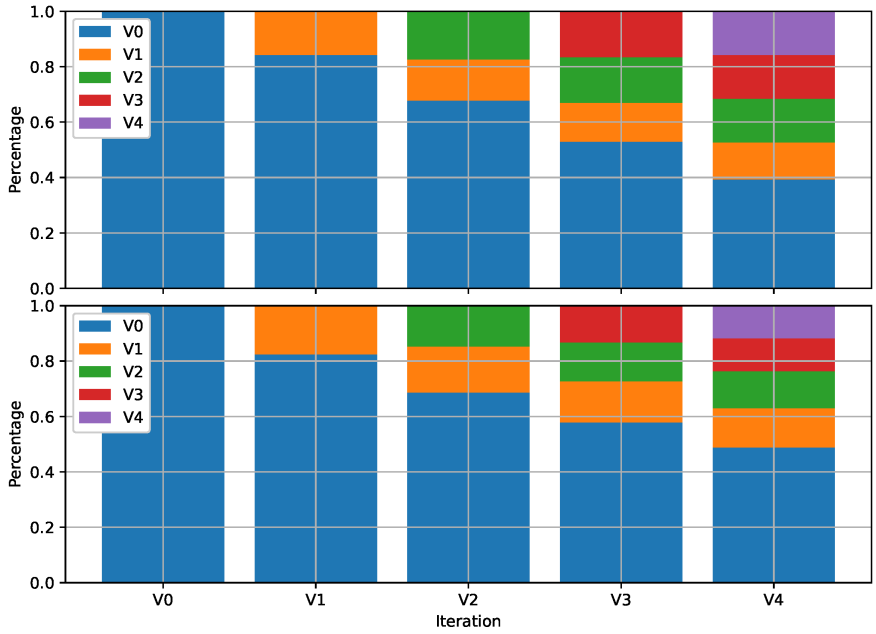


FIGURE 3.3: Percentage of trees created at each version for DR, when updating metrics (on top) and without updating (bottom)

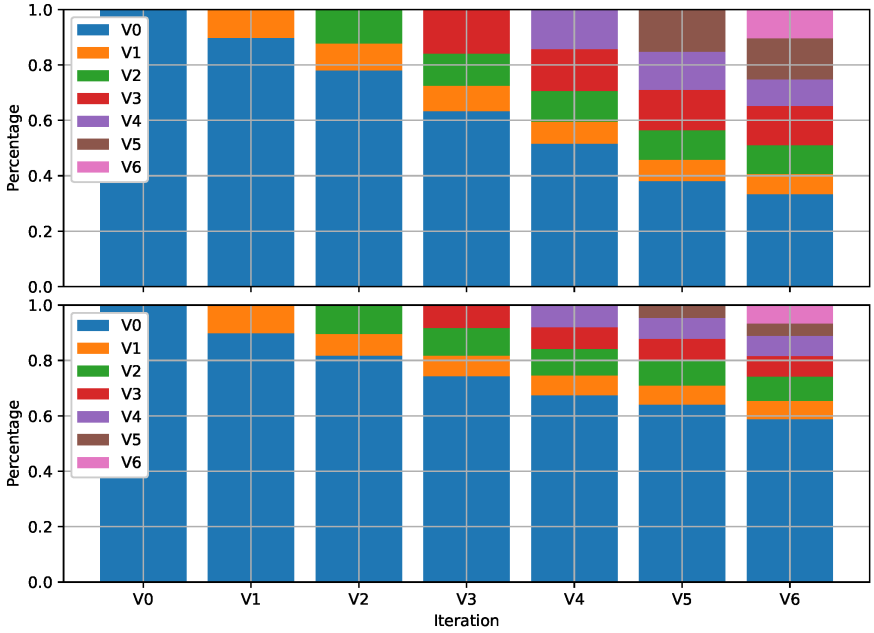


FIGURE 3.4: Percentage of trees created at each version for Occupancy, when updating metrics (on top) and without updating (bottom)

trees being incorporated into the FRF are, in fact, better trees than the ones being removed.

The second analysis consists of generating a histogram for each iteration. In each of them, we are plotting the balanced accuracy of each tree on the updated FRF. It can be seen in figures Fig. 3.5 and Fig. 3.6.

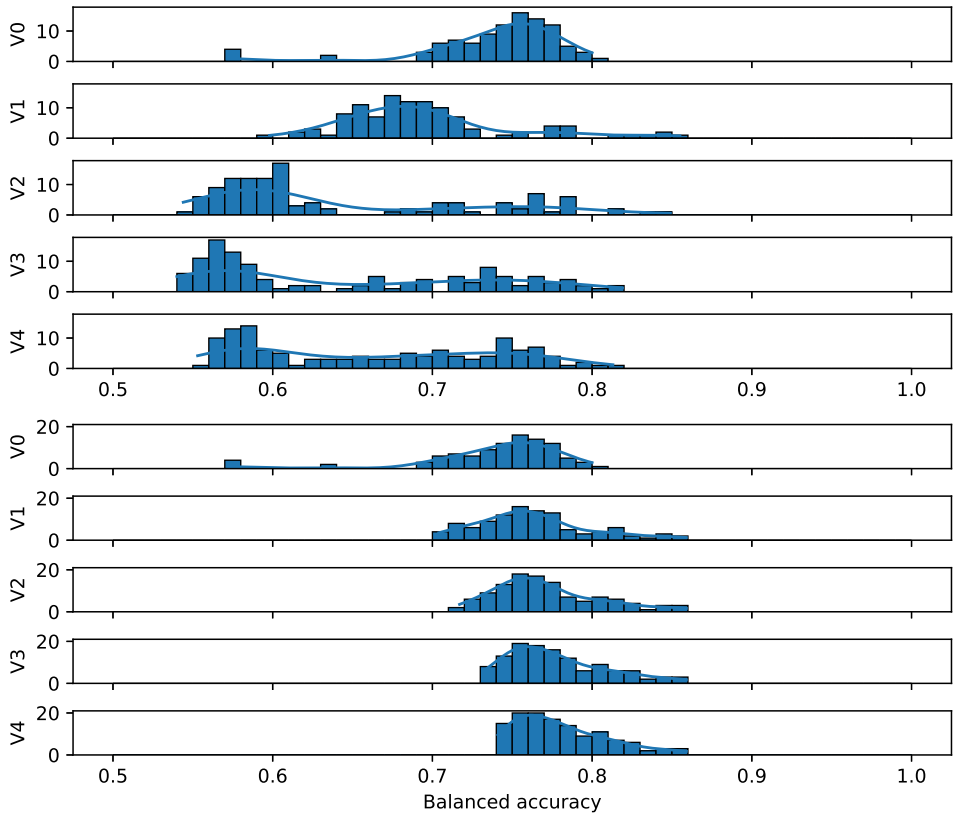


FIGURE 3.5: Histogram of the balanced accuracy of trees in the DR dataset. With updating metrics on top, no update below

When the balanced accuracy is not updated, the initial value computed using the out-of-bag samples is maintained during all iterations. This is why the values increase at each of them, because the worst trees with lower values are being removed and trees with higher values are being added. In contrast, when the weights are updated, all the values are recomputed at each iteration. At first sight, it would seem their results are worse because the values start decreasing. But considering they are recomputed using new data, the fact that they start increasing after some iterations proves they are better. Being able to increase the balanced

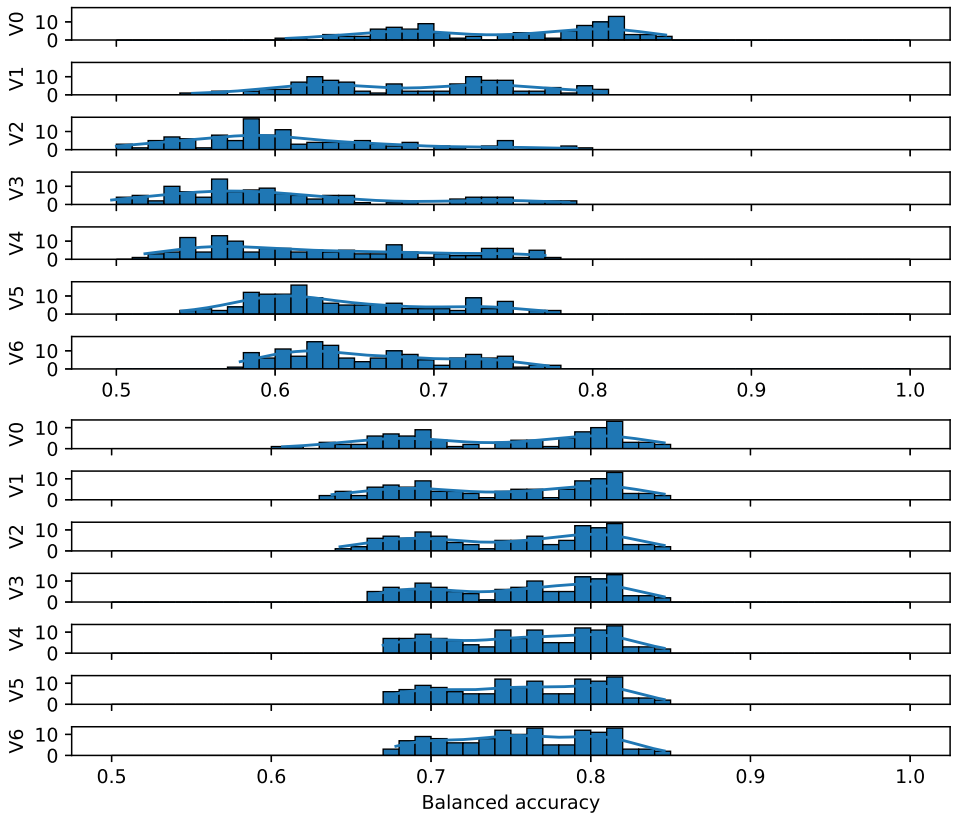


FIGURE 3.6: Histogram of the balanced accuracy of trees in the Occupancy dataset. With updating metrics on top, no update below

accuracy when using different data each iteration means the trees generalize better, which leads to better results.

The following analysis only includes the tests updating the metrics, as the final results are better than not updating them. In the third analysis, which can be seen in Fig. 3.7, there are the confusion matrix values obtained on the test set after each iteration. In the Diabetic Retinopathy dataset, the True Positives and True Negatives increase, and the False Positives and False Negatives decrease. Moreover, the changes are gradual during all the update iterations. In the Occupancy dataset, the results are similar, with the exception that the True Negatives decrease and the False Positives increase. Even though these results are not as good as the ones obtained on the Diabetic Retinopathy dataset, the detection of positive samples has improved, as desired.

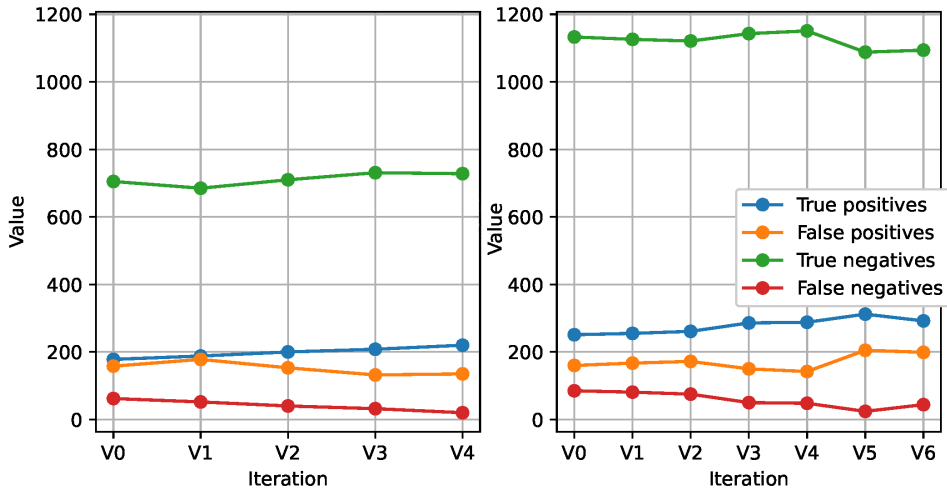


FIGURE 3.7: Confusion matrix values on the test set. Diabetic Retinopathy (left) and Occupancy (right)

The last analysis, illustrated in figure Fig. 3.8, shows the evolution of the accuracy, specificity, sensitivity, and precision of the updated model in the test set after each update iteration.

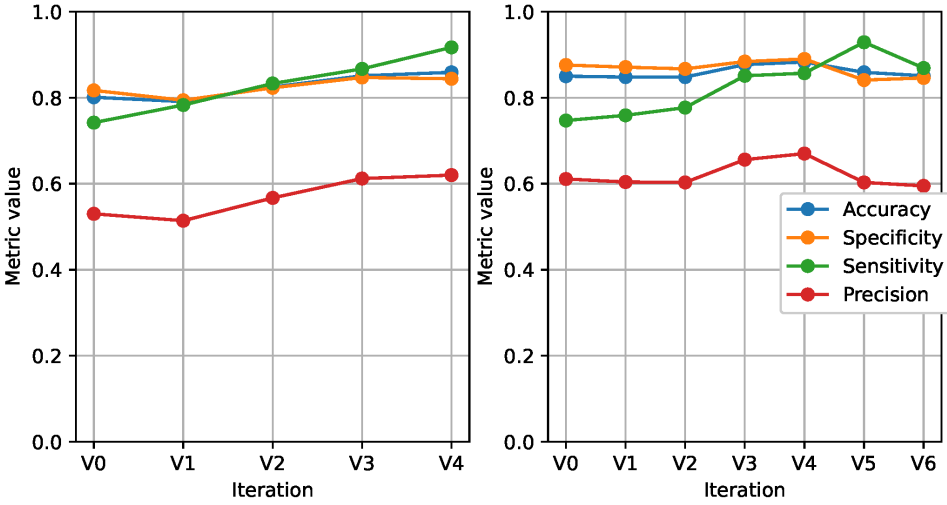


FIGURE 3.8: Quality metrics on the test set. Diabetic Retinopathy (left) and Occupancy (right)

As expected, looking at the confusion matrix results, the metrics on the Diabetic Retinopathy dataset gradually improved during all iterations. Moreover, the sensitivity is the metric that increases the most, as desired. The Occupancy results could also be expected. The sensitivity gradually improves, and the specificity ends up slightly decreasing. Even though it is not as desired as improving both metrics, it is still desired for our use case.

The final test shown in Table 3.11 compares three models. The initial one uses only the first dataset, called Base. A second dataset contains the base training data and all the validation datasets together. This extended dataset is used to train a unique model from scratch. The third one is the proposed iterative algorithm. To make a fair comparison with the iterative update method, weighted voting is also used. And because the last iteration of the update process has more than the 100 trees of the Base model, the Extended datasets are trained with the same number of FDTs: 127 in the Diabetic Retinopathy dataset and 136 in the Occupancy dataset.

TABLE 3.11: Results with original, iterative and extended datasets

Dataset	**Sensitivity**	**Specificity**	**F1-score**
Base DR	74.6	81.5	61.8
Extended DR	79.2	78.6	61.8
Iterative DR	91.7	84.4	74
Base OC	74.7	87.6	67.2
Extended OC	75	85.7	65.2
Iterative OC	86.9	84.6	70.6

The results in Table 3.11 show that the use of the extended training data gives a better sensitivity and slightly worse specificity than the base models. In contrast, the iterative update method increases in great measure the sensitivity values, whereas the specificity also increases in the Diabetic Retinopathy dataset and slightly decreases in the Occupancy dataset.

Even though the Extended tests use all the available data and additional FDTs compared to the Base tests, their results are similar, as their F1-scores show. In contrast, the F1-score of the proposed iterative method is greater. Because of the use of the same amount of data and FDTs as the Extended tests, we can say that the proposed method is able to improve the performance of the FRF model.

3.5 Conclusions

The method presented in this chapter is able to update a Fuzzy Random Forest in an iterative manner. It allows using newly incoming data to improve the Fuzzy Random Forest model without having to retrain it from scratch. Although it has been tested on a Fuzzy Random Forest, the same methods could be applied to regular Random Forests, since the fuzzy component is not involved in the proposed updating method.

It has been tested using data from two different domains: the assessment of the risk of developing Diabetic Retinopathy, and the Occupancy of an office room. Both of them are highly imbalanced towards the negative class, and the base results show a better detection of this class (higher specificity than sensitivity). Moreover, the Diabetic Retinopathy problem is also inherently ambiguous, given that very similar samples can belong to different classes.

When used in both datasets, the method has proven to improve the detection of the positive class. Moreover, the detection of the negative class has maintained similar accuracy values to the original model, or they have even improved. Therefore, the proposed method is a good solution to improve a Fuzzy Random Forest model when new data is available.

In the Diabetic Retinopathy case, it has been tested using real data from diabetic patients. The proposed method leads to improvements in the assessment of Diabetic Retinopathy risk, which can help to avoid unnecessary screenings of patients and reduce the workload of the ophthalmologists. The available resources can also be distributed among the patients, focusing on the ones that really need them.

Chapter 4

Adapting a Fuzzy Random Forest for ordinal multiclass classification

4.1 Introduction

In ordinal multiclass decision problems, we must assign a class to an instance from a set of k ordered possibilities, $C = \{Class_0, Class_1, ..., Class_{k-1}\}$, where $k > 2$. Depending on the problem, the order can be of increasing or decreasing preference, also called Gain or Cost. For example, in medical diagnosis, the usual order goes from the best to the worst medical conditions, so that $Class_0$ has the healthy people, and the greater the class index, the worse the disease level.

Multiclass decision problems in medical diagnosis have not been studied to a great extent, (Ahsan, Luna, et al., 2022). Having the ability to classify the disease level with greater precision allows for a better distribution of resources among the patients who need them the most. This allows for better treatments for the individuals that really need them and reduces unnecessary treatments for the healthiest individuals.

In this chapter, we adapt the 2-step binary classification process of FRF presented in Chapter 2 for the case of ordinal multiclass decision problems. In Section 4.2 the first classification step is adapted. Section 4.3 explains the modifications to the second classification step. Section 4.4 shows experimental results. Finally, Section 4.5 gives the conclusions and future work.

4.2 Fusion in a Fuzzy Decision Tree

The first modification must be introduced at the FDT level, when the predictions of the rules are aggregated to decide the output class given by the tree. The use of δ_1 is maintained in this proposal for multiclass classification, but because of the existence of multiple classes, the method must be adapted. In binary classification, δ_1 is just compared with the difference in support between the two classes. In the case of multiple classes, we propose the following strategy.

The result of the analysis of the j-th FDT will be a tuple with the predicted class and its support (P_j, S_j), calculated as described by Eq. 4.1. We first order decreasingly the set $C^* = \{C \bigcup Unknown\}$, according to the normalized decision support values D_i given by the FDT rules. Let us consider that C_a is the category with the highest decision support, and C_b is the second most supported category. Then, the outcome class is chosen between these two categories, taking into account the difference in their support. One of the strengths of an ensemble is the diversity of models that compose it. An unknown prediction is preferred when the model does not have a unique preferred class, which is better than making an incorrect prediction. The next section explains how the ensemble aggregates the predictions of the different trees to arrive at the final decision.

$$(P_j, S_j) = \begin{cases} (Unknown, 0), & \text{if } D_a - D_b < \delta_1 \\ (C_a, D_a), & \text{otherwise} \end{cases} \tag{4.1}$$

4.3 Fusion in a Fuzzy Random Forest

In the following subsections, we explain the method proposed to aggregate all the predictions of the ensemble for the multiclass case. Its main elements are a weighted voting, some heuristics for the final class assignment, and an OWA-based decision support score.

4.3.1 Weighted voting

A voting process is used to find the consensus class from the ensemble of different FDTs. We propose using weighted voting instead of the previous majority voting. Each FDT needs a weight assigned to it, which represents its prediction quality. It can be computed using the out-of-bag examples from the training phase. Each FDT is tested on their out-of-bag examples, and a quality metric is computed from

the obtained results. The quality metric has to be properly chosen to represent the overall quality of each FDT.

To aggregate all the predictions of a class through weighted voting, the weights of the trees that predicted it are summed, Eq. 4.2. As a result, each of the classes obtains a voting value v_i, which is used to decide the final prediction of the ensemble, as explained in the next subsection. Thus,

$$v_i = \sum_{j \in I} w_j \tag{4.2}$$

, where w_j is the weight assigned to the j-th tree of the set $I = \{t \mid P_t = Class_i\}$.

In the previous chapter, we tested several metrics for weighting the trees in the binary approach. An average accuracy balancing sensitivity (2/3) and specificity (1/3) was selected. This balanced accuracy is specially useful in domains such as the medical one, in which a good sensitivity is a priority in order to avoid false negatives.

For the case of multiclass problems, the most appropriate and usual quality measures are $F1$ (balancing precision and recall) and the Weighted Cohen's Kappa κ. If we take into account the order between the classes, then κ is the best evaluation index because it allows to define different penalisations for mistakes between classes depending on the distance between them (J. D. L. Torre, Puig, et al., 2017). For this reason, we propose to use κ in the weighting process of FRF on ordinal multiclass classification.

4.3.2 Final class assignment

To assign a final prediction with multiple classes, we will take C_a and C_b again as the first and second most voted classes, respectively, from the FRF. For the multiclass proposal, we will preserve the use of δ_2, but it is defined as a function depending on the difference between the two most voted classes, according to their position in the ordered set of possible categories C. Let us define Δv_{ab} as the normalized difference of votes between C_a and C_b, Eq. 4.3.

$$\Delta v_{ab} = \frac{v_a - v_b}{\sum_{i \in C^*} v_i} \tag{4.3}$$

With this normalization, the δ_2 threshold is now defined in two parts, Eq. 4.4. A first constant part $d \in [0, 0.5]$, which is the minimum difference in votes that permits to distinguish the support of the classes and make a class assignment. In

addition, the separation between the classes in the ordered scale C is also relevant to defining when a difference in votes is important or not. It is not the same choosing between consecutive classes as between extreme classes in C. For that reason, the second part of δ_2 is given by the square of the difference between the positions of the classes. The value is limited to 0.5 for all the cases where the most-voted class has more than half of the total votes.

Formally, let us define $index : class \rightarrow [0, k-1]$ as the function that returns the position of a given class in the ordered set C, and the distance between classes as $dist(C_a, C_b) = |index(C_a) - index(C_b)|$. Then, the definition of δ_2 is the following:

$$\delta_2 = \begin{cases} d, & \text{if } C_a = Unknown \text{ or } C_b = Unknown \\ \min(0.5, d + \frac{dist(C_a,C_b)^2}{100}), & \text{otherwise} \end{cases} \tag{4.4}$$

Using this new δ_2 definition in the same equation than for binary classifiers, Eq. 2.10, is a too conservative approach that generates many assignments to the *Unknown* category. To avoid the situation where the classifier does not provide an answer in too many cases, the following heuristics for the output class assignment A are proposed:

- **H1**: When one of the two most voted classes is the *Unknown* category and the difference in votes is small, then the model returns the class $C_n \neq Unknown$. However, if the difference is large enough, then the model returns the most-voted class. It may be *Unknown* or the other one.
- **H2**: If the two classes are not unknown and the difference in votes is large enough, the most voted class must be the output of the classification model.
- **H3**: In the cases where the two classes are not unknown and the number of votes is similar, $\Delta v_{ab} < \delta_2$, two options are considered, depending on the distance of the classes in the ordered set C. If there is a big distance between the positions of the classes in the ordered set C, the ensemble is considered not to be certain about the prediction, and the label *Unknown* is assigned. In the case of close classes in C, the selected class is the one with a higher index in this ordered set. The distance threshold is based on the number of classes k.

These heuristics are formalised in the following equation:

$$A = \begin{cases} C_n, & \text{if } (C_a = Unk \text{ or } C_b = Unk) \text{ and } \Delta v_{ab} < \delta_2 \\ C_a, & \text{if } (C_a = Unk \text{ or } C_b = Unk) \text{ and } \Delta v_{ab} \geq \delta_2 \\ Unk, & \text{if } C_a \neq Unk \text{ and } C_b \neq Unk \text{ and } \Delta v_{ab} < \delta_2 \text{ and } dist(C_a, C_b) \geq \lfloor \frac{k}{2} \rfloor \\ C_m, & \text{if } C_a \neq Unk \text{ and } C_b \neq Unk \text{ and } \Delta v_{ab} < \delta_2 \text{ and } dist(C_a, C_b) < \lfloor \frac{k}{2} \rfloor \\ C_a, & \text{if } C_a \neq Unk \text{ and } C_b \neq Unk \text{ and } \Delta v_{ab} \geq \delta_2 \end{cases} \tag{4.5}$$

, where $C_n \neq Unknown, n \in \{a, b\}$, and $m = \max(index(C_a), index(C_b))$.

Notice that we assumed a minimization goal, where wrong classification into less severe classes is not desired. If the goal is maximization, then m should be the minimum.

4.3.3 Final decision support

We propose the Ordered Weighted Average (OWA) to perform the aggregation of the final decision support value of the prediction (Yager, 1988). OWA is a parameterized operator that permits conjunctive or disjunctive aggregation. The polarity of the operation is defined by a set of weights assigned to the input values according to their position after their reordering. Having a set of support values S_j obtained with Eq. 4.1 for each tree, and having a weight for each position $w_i, i = 1..n$. The result F is obtained with Eq. 4.6, where $S_{\sigma(i)} < S_{\sigma(i+1)}$.

In a FRF, the number of trees, n, is usually large (i.e., hundreds), but only a subset of the trees corresponds to the final class, A. Given the randomness in the selection of attributes, some of these trees may produce low support values. However, if at least a sufficient number of trees, $m <<< n$, are highly supporting the selected class, the confidence about this class should be high, which corresponds to a disjunctive policy of aggregation. Therefore, the weighting vector, where $\sum_{i=1}^{n} w_i = 1$, has been defined with weights that decrease, as established in Eq. 4.6.

$$F = \sum_{i=1}^{n} w_i S_{\sigma(i)}, \text{ where } w_i = \frac{i}{\sum_{j=n-m}^{n} j} \text{ for } i \in [n-m, n] \text{ , and } w_i = 0 \text{ otherwise} \tag{4.6}$$

4.4 Experiments

In this section, the obtained experimental results are analysed. The experiments will be done using two different datasets. The first one is the Diabetic Retinopathy risk detection problem, and the second one is the employee burnout prediction. To test the ordinal multiclass proposal, we conducted several tests to analyse the different modifications to the aggregation methods. The obtained results are shown in the following subsections.

4.4.1 Datasets

The first dataset we used is the DR dataset from the Hospital Sant Joan de Reus. It includes real data from 2084 diabetic patients. The data includes the same nine attributes: six numerical (Age, Evolution time of diabetes, HbA1c, CKDEPI, Microalbuminuria and Body Mass Index) and three categorical (Sex, Medical Treatment and Hypertension) as in the previous chapter's DR dataset. Because we are now dealing with a multiclass case, the ETDRS standard classification is considered for the target DR attribute (Wilkinson, Ferris, et al., 2003). It consists of four categories, which are ordered from the best to the worst medical condition: $C = \{NoDR, Mild, Moderate, Severe\}$.

The second dataset we used is the employees burning out dataset (HackerEarth, 2020). It predicts the burn rate of employees given a set of their job conditions. It consists of 8 attributes: 3 categorical (Gender, Company Type and WFH Setup Available) and 5 numerical (Designation, Resource Allocation, Mental Fatigue Score, Join Month, and Join Day). The task of this dataset is regression. The burn rate ranges between [0, 1]. We are working on classification problems, so it has been turned into a classification task. The burn rate has been categorized in 5 intervals of 0.2 ($[0, 0.2], (0.2, 0.4], ..., (0.8, 1]$) to create the classification task. Categories are ordered from lowest to highest burn rate, $C = \{Class_0, Class_1, Class_2, Class_3, Class_4\}$.

In both cases, the data has been split into two different datasets: training (80%) and testing (20%). Table 4.1 and Table 4.2 show the distribution of the data among the target attribute classes for both the DR and burnout datasets, respectively. In the DR case, there is a large imbalance towards the first class, $NoDR$. In contrast, the burnout dataset has a slight imbalance towards the third class, $Class_2$.

TABLE 4.1: Diabetic Retinopathy data distribution

	Training	Testing	Total
NoDR	1394 (83.6%)	349 (83.7%)	1743
Mild	191 (11.5%)	48 (11.5%)	239
Moderate	58 (3.5%)	14 (3.4%)	72
Severe	24 (1.4%)	6 (1.4%)	30
Total	1667	417	2084

TABLE 4.2: Burning Out data distribution

	Training	Testing	Total
Class0	1638 (11%)	409 (11%)	2047
Class1	4126 (27.7%)	1031 (27.7%)	5157
Class2	5802 (39%)	1451 (39%)	7253
Class3	2728 (18.3%)	682 (18.3%)	3410
Class4	578 (3.9%)	145 (3.9%)	723
Total	14872	3718	18590

4.4.2 Study of the weights of FDTs in the voting stage

From the different modifications proposed for the case of ordinal multiclass assignments with FRF, we start by testing the effect of using the κ index instead of accuracy to give a weight to each of the trees in Eq. 4.2. We compare three versions of the FRF classification algorithm:

1. **Base algorithm**: it does not consider the category *Unknown*, so that we always classify an instance to one of the output classes. Hence, $\delta_1 = 0$ and $\delta_2 = 0$.

2. **Base-δ algorithm**: it takes into account situations where two classes have similar conditions and then the answer is unknown, to try to avoid mistakes.

3. **New-δ algorithm**: it corresponds to the new procedure explained in this chapter.

We will denote as FN (False Negatives) the examples where the model predicts a class lower than the real (i.e., underestimation or type-II error). Similarly, we call FP (False Positives) when the predicted class is higher than the real one (i.e., overestimation or type-I error). FNs are a kind of error that is not desirable in medical diagnosis because the system does not detect the real risk to the person's health.

The confusion matrix in Table 4.3 shows the results of the Base version for the DR dataset. For example, in *Mild*, from the total of 48 patients, we have 15 classified to *NoDR* (FN=31%). Similarly, there is a 28% of FN in *Moderate* and 33% in *Severe*.

TABLE 4.3: DR base method confusion matrix

Real/Predicted	**NoDR**	**Mild**	**Moderate**	**Severe**
NoDR	278	30	21	20
Mild	15	13	8	12
Moderate	2	2	3	7
Severe	2	0	0	4

For the burnout dataset, the confusion matrix in Table 4.4 shows the results of the Base version. For example, in $Class_1$, from the total of 1031 patients, we have 146 classified to $Class_0$ (FN=14%). Similarly, there is a 15% of FN in $Class_2$, 12% in $Class_3$ and 10% in $Class_4$.

TABLE 4.4: Burnout base method confusion matrix

	Class0	**Class1**	**Class2**	**Class3**	**Class4**
Class0	359	50	0	0	0
Class1	146	699	186	0	0
Class2	0	214	939	295	3
Class3	0	0	85	458	139
Class4	0	0	0	13	132

We have defined the Base version as the model to improve. In the DR case, it makes too many mistakes, so we are expecting greater improvements on the base results.

Table 4.5 compares the results of using Accuracy, or κ as the quality metric in the weighted voting. The three versions of the algorithm are compared on Accuracy (Acc), Accuracy including unknowns as errors (Acc Unk), and Kappa index. The δ_1 and δ_2 thresholds were selected through empirical testing. For DR, the thresholds used are $\delta_1 = 0.1$ and $\delta_2 = 0.25$. For burnout, the thresholds used are $delta_1 = 0.15$ and $\delta_2 = 0.1$. An in-depth analysis of δ_2 is shown in subsection 4.4.3.

On the DR dataset, better quality values are obtained using κ as the weights of the trees. The accuracy index increases in all three versions. The Weighted Kappa index is maintained at a similar level. *Acc Unk* metric decreases a bit, meaning the κ Weight produces more unknown predictions than the Accuracy Weight. On

TABLE 4.5: Comparison of two weighted voting quality metrics

		Accuracy Weight			κ Weight		
		Acc (%)	*Acc Unk (%)*	*Kappa*	*Acc (%)*	*Acc Unk (%)*	*Kappa*
DR	**Base**	71	71	0.34	71.5	71.5	0.345
	Base-δ	90.8	25.9	0.529	92.3	11.5	0.509
	New-δ	70.2	60	0.312	73.4	56.1	0.318
Burnout	**Base**	69.4	69.4	0.864	69.6	69.6	0.865
	Base-δ	72.5	59.9	0.886	72.1	61.8	0.883
	New-δ	69.9	67.6	0.87	69.9	68.5	0.869

the burnout dataset, similar quality values are obtained using κ as the weights of the trees. The accuracy index remains almost the same for all three versions. The Weighted Kappa index is also maintained at a similar level. *Acc Unk* metric is greater using the κ index, meaning the Accuracy Weight produces more unknown predictions than κ.

To further analyse the weight selection in the voting stage, Figure 4.1 and Figure 4.2 show the distribution of correct, incorrect and unknown predictions in DR and burnout, respectively.

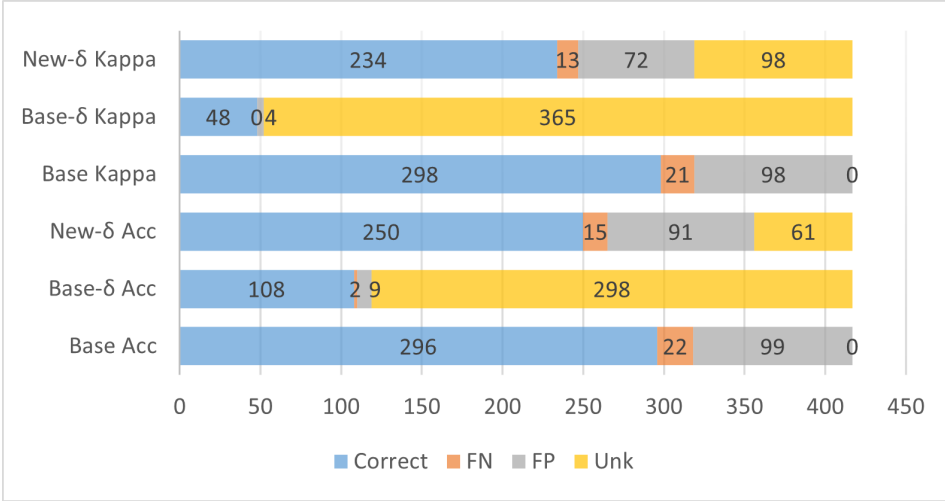


FIGURE 4.1: Distribution of correct, incorrect and unknown class assignments for different voting weights in DR

The Base-δ algorithm does not perform appropriately. For the DR, it has very few errors, but there are too many unknown predictions. For burnout, it has the

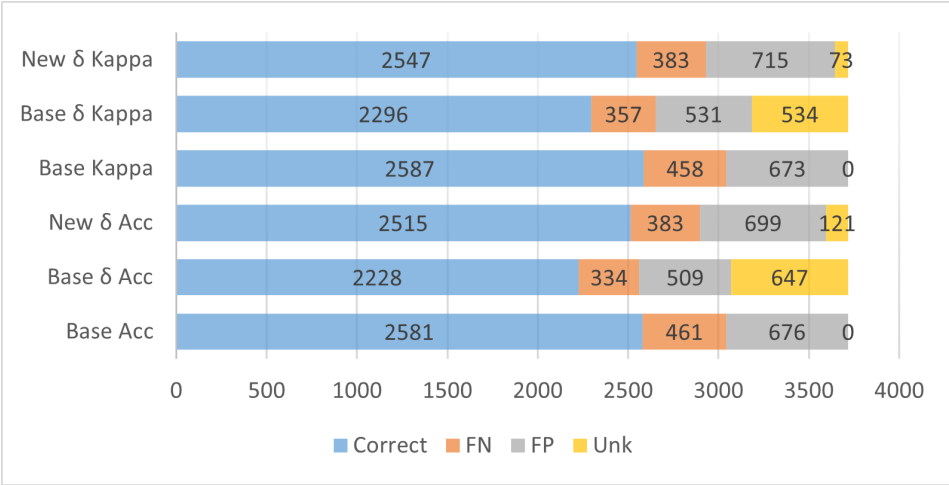


FIGURE 4.2: Distribution of correct, incorrect and unknown class assignments for different voting weights in burnout

worst results. In contrast, the New-δ algorithm is able to reduce the incorrect predictions by introducing a moderate amount of unknowns. Comparing Accuracy and κ in the New-δ algorithm, even though κ has fewer correct predictions in DR, the amount of incorrect predictions is also smaller. In burnout, the results are quite similar. We consider κ to have better results because of its more conservative results on unclear cases, which is the case with DR, whose base results are not good. Moreover, it obtains better global accuracy. It is also able to slightly improve base results regarding *Acc Unk* that are already good, which is the case with burnout.

4.4.3 Study of δ_2 for class assignment in ordinal FRF

We studied the effect of using different d values to compute δ_2 for the final class assignment. Figure 4.3 and Figure 4.4 show the effect of d on the distribution of predictions among correct, incorrect and unknown in DR and burnout, respectively. As expected, the higher d, the lower the number of unknown predictions. This is due mainly to the decreasing number of cases that enter the second condition in Eq. 4.5. Accordingly, correct and incorrect assignments increase as the amount of unknowns decreases. We can see that the d parameter allows modelling the trade-off between correct, incorrect and unknown predictions. For the DR risk assessment problem, $d = 0.25$ was chosen for its balance of reducing incorrect predictions while not increasing unknowns and reducing correct predictions in

excess. For the burning out rate problem, $d = 0.1$ was chosen for the same reasons. In other domains, δ_2 can be adapted according to the problem being solved and the implications of misclassifications.

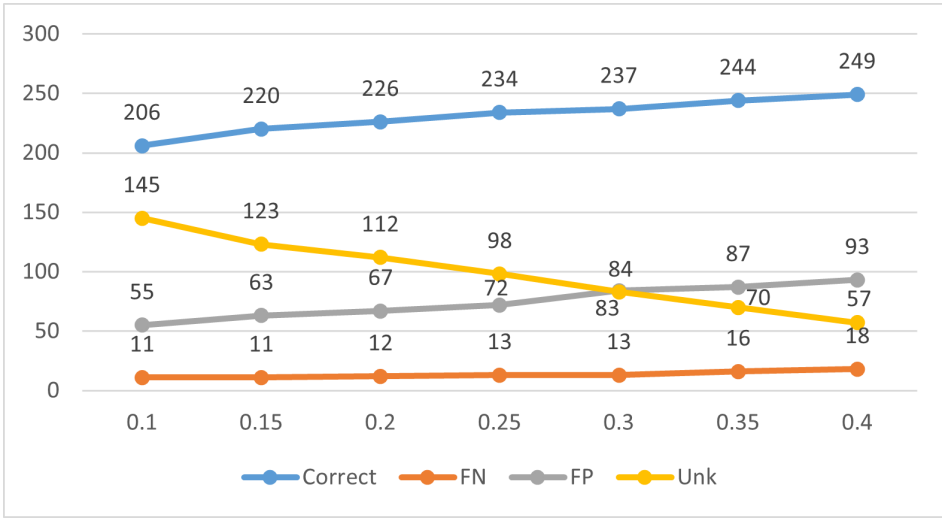


FIGURE 4.3: Distribution of correct, incorrect and unknown class assignments for different d values for DR

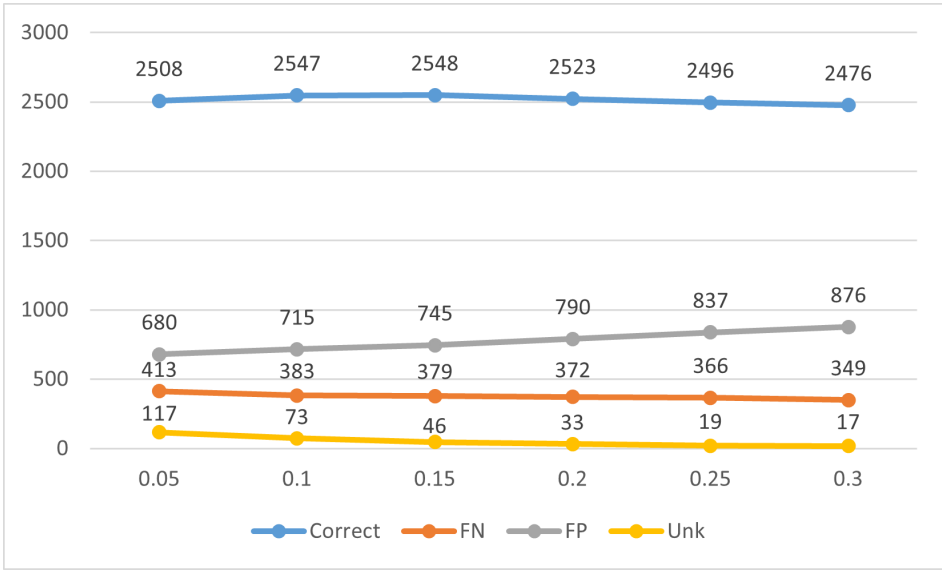


FIGURE 4.4: Distribution of correct, incorrect and unknown class assignments for different d values for Burnout

4.4.4 Study of the heuristics for class assignment in ordinal FRF

To study the effects of the proposed heuristics, the algorithm versions explained in 4.4.2 have been tested with two additional versions, Table 4.6. The first column gives the accuracy only for the objects classified, the second column gives the accuracy taking into account the unknowns and the last one gives the value of **Kappa** indicator for the classified objects. The rows in the table correspond to different versions. The additional versions differ in heuristic H3, which considers cases with the two most voted classes not being *Unknown* and a similar number of votes, $\Delta v_{ab} < \delta_2$. We eliminate the condition about the distance of the classes in the ordered set C, instead, a predetermined label is assigned. New-δ-Unk assigns *Unknown*, whereas New-δ-Max assigns the class with the higher index in C.

For DR the best method seems to be Base-δ, for its accuracy and *Kappa*, but it produces a lot of unknown answers (i.e., not classified patients) as can be seen in the low value of Acc Unk.

TABLE 4.6: Comparison of different versions of the method

	DR			Burnout		
	Accuracy (%)	*Acc Unk (%)*	*Kappa*	*Accuracy*	*Acc Unk*	*Kappa*
Base	71.5	71.5	0.345	69.6	69.6	0.865
Base-δ	92.3	11.5	0.509	72.1	61.8	0.883
New-δ-Unk	78.9	55.6	0.38	71.7	64.8	0.878
New-δ-Max	68.6	56.1	0.227	69.8	68.5	0.867
New-δ	73.4	56.1	0.318	69.9	68.5	0.869

The performance improvement can be more clearly seen in the distribution of correct, incorrect and unknown assignments in Figure 4.5 and Figure 4.6. Comparing New-δ-Unk and New-δ-Max with New-δ, we can conclude for both datasets that by taking into account the distance between the majority classes, we can balance the number of errors and unknown assignments. Even though New-δ-Unk is the version with fewer errors, it is not the preferred version, as it could lead to having too many predictions assigned to *Unknown*. In the case of New-δ-Max, we can see on the FP the effect of classifying to the class with the higher index when the FRF does not have a clear consensus towards one class, which is the case with DR. For the burnout dataset, both results are quite similar. This could be expected because of the greater performance obtained on this data. By merging both versions depending on the distance between classes, the amount of unknowns can be balanced while prioritising the classes with higher indexes. This behaviour is

desired in ordinal cases such as the DR risk assessment, where a FN would have much worse consequences than a FP.

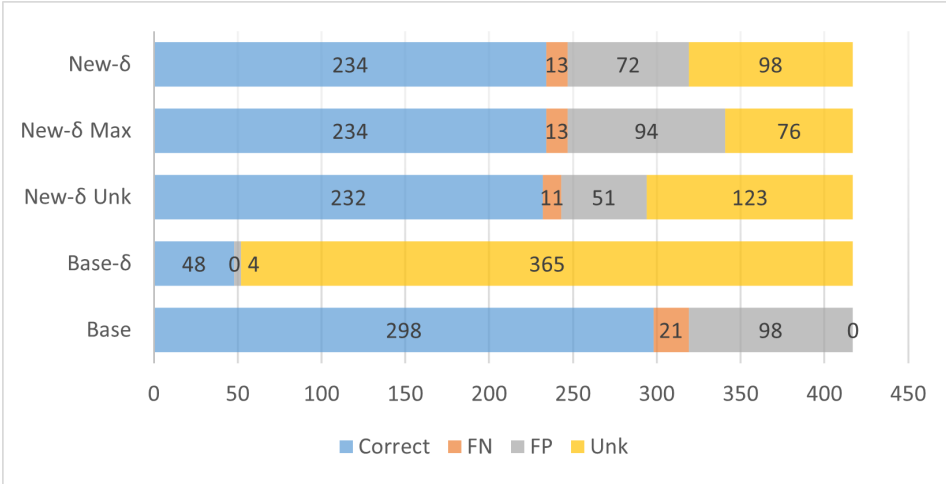


FIGURE 4.5: Distribution of correct, incorrect and unknown assignments in different versions of the FRF for DR

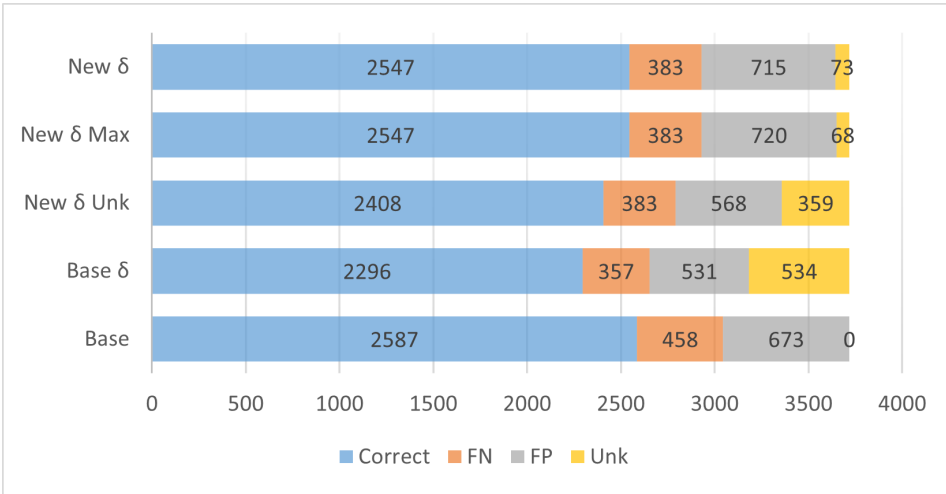


FIGURE 4.6: Distribution of correct, incorrect and unknown assignments in different versions of the FRF for burnout

4.4.5 Study of OWA for final decision support averaging

To study the effect of a disjunctive OWA in the aggregation of the decision support, it has been compared to an Arithmetic Average aggregation (AA), Table 4.7. Experiments have been performed with $n = 100$ trees and $m = \frac{n}{3}$ as the minimum number of trees supporting the selected class. The decision support values obtained from the test dataset, which can range in $[0, 1]$, have been split into three intervals to indicate three levels of confidence in the answer given to the user. For each of them, the number of correct predictions is counted. We consider that we should not have low decision support in cases where a sufficient number of trees are sure about the prediction. This is the result achieved by OWA in both datasets. The number of predictions in the higher intervals is greater than using an Arithmetic Average. As a consequence, the percentage of correct predictions in the higher interval is also greater. With AA, the user has more uncertain answers, which in medicine are cases that require additional attention by the doctors, requiring time and resources. So, the OWA operator is recommended.

TABLE 4.7: Decision support values with AA and disjunctive OWA

		Arithmetic Average			OWA		
		[0, 0.5]	*(0.5, 0.75]*	*(0.75, 1]*	*[0, 0.5]*	*(0.5, 0.75]*	*(0.75, 1]*
DR	**Total**	44	205	70	5	113	201
	# correct	37	149	48	5	80	149
	% correct	84 %	73 %	69 %	100 %	71 %	74 %
Burnout	**Total**	11	1158	2476	0	382	3263
	# correct	10	848	1689	0	294	2253
	% correct	91 %	73 %	68 %	0 %	77 %	69 %

4.5 Conclusions

In this chapter, we presented an adaptation of a binary FRF model for ordered multiclass classification. We have focused on the 2 steps of the prediction stage, and we have redefined the procedure to manage conflicting cases. The different contributions presented have been studied on two datasets. The DR and the burnout datasets. From these results, we can conclude that using the κ index as a weighting factor for trees performs better than accuracy in the ordinal multiclass case. The obtained results suggest that the proposed method can effectively model the trade-off between predictions and unknowns using the δ_2 parameter. The heuristics

employed within the model successfully balance the number of unknowns while prioritizing classes with higher indexes, which is particularly beneficial in medical applications where accurate classification is crucial. Furthermore, OWA provides an appropriate confidence value for the class assigned by the FRF, thus enhancing overall performance and decision-making capabilities. Even though accuracy and kappa metrics have been improved for the DR case, an accuracy of just 73.4%, with a 23.5% of the examples being Unknowns, and a kappa of 0.318 are obtained. This leaves margin for improvement for the multiclass classification case.

UNIVERSITAT ROVIRA I VIRGILI
FUZZY-BASED MACHINE LEARNING METHODS FOR CONTINUOUS DIAGNOSIS AND PROGNOSIS OF DIABETIC RETINOPATHY
Jordi Pascual Fontanilles

Chapter 5

Improving DR detection on long-term patients using temporal data

5.1 Introduction

In this chapter, we study the problem of ordinal classification using temporal series. To improve the performance of the DR grading classifier obtained in the previous chapter, we consider the exploitation of the knowledge provided by the history of previous visits to make the risk assessment instead of simply relying on the latest state of the patient. In particular, we propose a novel method for the classification of Diabetic Retinopathy into the different levels of severity of DR that takes advantage of the data stored in the EHR of the patient in his/her successive visits. The overall proposed method is illustrated in Fig. 5.1.

For this research, we have worked with a dataset with a total of 231,064 Type-2 diabetic patients from Catalonia (Spain), with medical records from 2010 to 2021. We observed that patients are usually visited by doctors every 6 to 24 months; therefore, a diagnosis must be made with rather short sequences of data. The characteristics of the available data in the EHR require designing appropriate pre-processing methods for the transformation of the patient's data into homogeneous time series. This is a crucial step in order to build proper classifiers. Moreover, the data will consist of multivariate time series, as each of the variables will be transformed into a different time series. Some of the challenges that need to be considered in this step are the different length of the sequences, the small number of visits per patient in the EHR, and the irregular time intervals between visits.

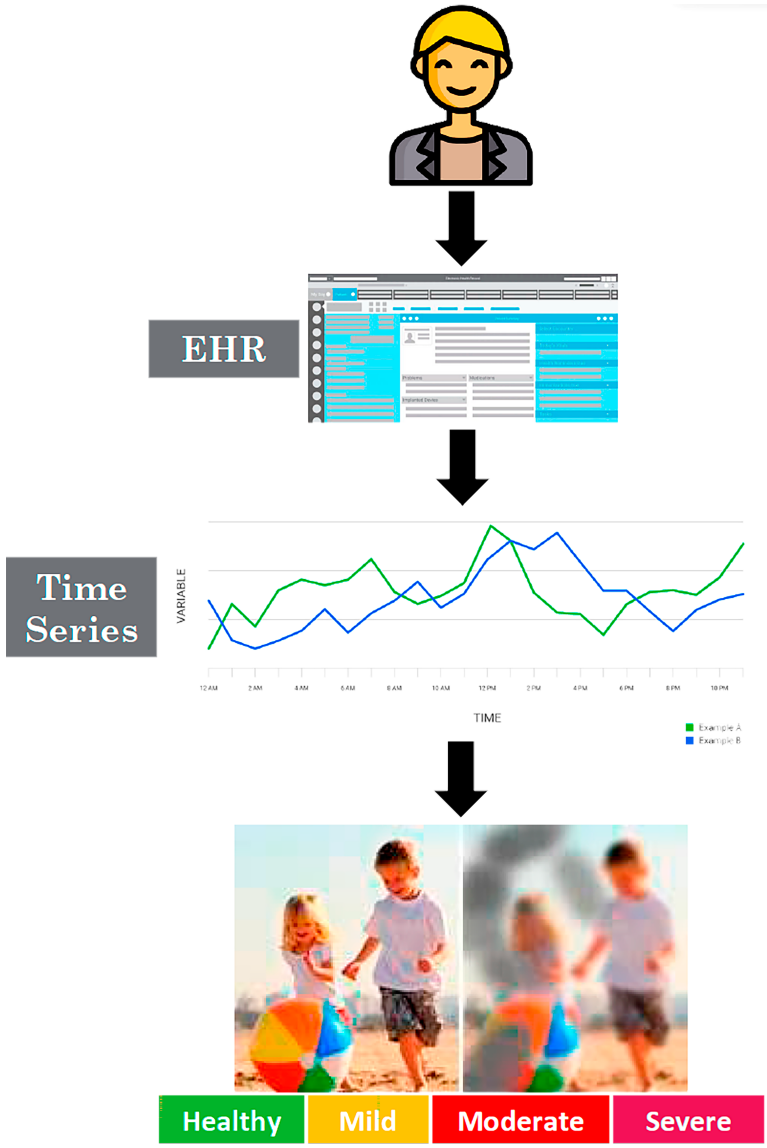


FIGURE 5.1: Proposed flow to improve DR detection by means of a time series classifier

As a first step, we define appropriate methods for solving these problems. In particular, due to the requirement of most state-of-the-art time series classifiers to have equal-length sequences, all patients with at least six records on different years in their EHR are turned into time series of length 10. The rest of the patient's

data (i.e., with less than six records) is discarded to avoid inferring too much data that could end up being incorrect.

After the data preprocessing, as a second step, we define a novel fuzzy-based method to compensate for class imbalance in DR time series data. The short time series that were discarded in the data transformation process into time series have been boosted to generate new positive examples. The proposed example generation method is applied to very short time series of real patients, boosting them with fictive data to obtain additional 10-length positive DR sequences.

A fuzzy approach has been used during the generation of new data values, because doctors reason qualitatively on the attribute values when assessing the patients' conditions (e.g., age: child/young/old; body mass: underweight/normal/overweight; hypertension: good control/bad control, etc.). For health treatment, a difference of one year in age or of one kilogram makes no difference in the diagnosis, as it is done at a more general level (with labels representing more general states). Fuzzy logic is a well-known paradigm for reasoning qualitatively (Zadeh, 1992). In the literature, several fuzzy-based clinical decision support systems can be found. Ahmadi et al. examined the use of fuzzy logic methods in disease diagnosis (Ahmadi, Gholamzadeh, et al., 2018). They found that fuzzy logic approaches were used to diagnose 38 different diseases, such as heart, kidney, thyroid or pulmonary diseases. They found that fuzzy logic had a positive effect in 91% of the analysed cases. Therefore, we can take advantage of the fuzzy linguistic model to generate different numerical values for fictive patients, which correspond to the same labels as real patients. We generate synthetic values by making use of fuzzy linguistic variables in order to introduce some degree of variability to the new examples without assigning unrealistic values. The numerical risk factors have been transformed into fuzzy linguistic variables with the knowledge of specialized ophthalmologists.

Once the multivariate time series dataset is created, we will train several time series classifiers to study their performance in solving the ordinal classification into the 4 categories of DR grading. A comparative study will be performed among some different types of time series classifiers to determine how they perform in the case of small-length series of diabetic patient's data.

Although the task we aim to solve could also be foreseen as forecasting, we did not consider it for the following reasons. Forecasting techniques would allow predicting the status of the patient at several points in the future, but our main objective is to predict the current DR status of a patient, given his or her historical

data. Moreover, because of the short historic periods we have available, we believe a classification task is more appropriate than forecasting.

The rest of the chapter is organized as follows. Section 5.2 presents the time series classification problem and briefly reviews how DR classification and the imbalance problem have been studied in the literature. In Section 5.3, we introduce the proposed method for pre-processing the EHR data into time series data. The short time series problem is also presented and undertaken in this section. Next, Section 5.4 presents the proposed approach for boosting short time series to compensate for class imbalance. Section 5.5 presents some state-of-the-art multivariate time series classifiers. In Section 5.6, the time series classifiers are compared and their obtained results are discussed. Finally, Section 5.7 presents the conclusions and the future work.

5.2 Related work

The Time Series Classification (TSC) problem is attracting a lot of interest since every day more time series data is being produced. Technology advances facilitate the collection and storing of data values over time. TSC can be used in many different areas and has a broad range of possible applications. One important field is biomedical and health care applications, such as the case of Diabetic Retinopathy classification.

Wang et al. reviewed TSC in the field of biomedical applications from 2016 to 2021 (W. K. Wang, Chen, et al., 2022). According to their study, the two main types of temporal data being used are electroencephalogram (EEG) and electrocardiogram (ECG), which are both signal series. The most common pre-processing method for EEG and ECG is filtering, which is used to remove artefacts and noise. Once the signal has been processed, some features are extracted, and the best ones are fed into a machine learning classifier. In this survey study, temporal data from EHR is in the fifth position, with half the articles compared to the former types. EHR might contain any kind of medical information, sometimes also including signals such as EEG or ECG. The authors also identified that the scarcity of data (a small dataset) is a common problem in biomedical applications. In this domain, a small dataset is considered when the amount of patients is low (e.g., less than 20). However, in many of those cases, the low number of patients is compensated by the availability of long data sequences, so the classifier's performance is good

because of the use of signal processing and feature engineering techniques on top of a classifier.

Pasos et al. reviewed how several algorithms performed on Multivariate Time Series Classification (MTSC) problems (Pasos-Ruiz, Flynn, et al., 2021). Dynamic Time Warping (DTW) was used as a baseline classifier, as it is still competitive in comparison to more recent proposed alternatives. According to their experiments, the ROCKET classifier (Dempster, Petitjean, et al., 2020) was the best performing overall in many applications. However, there is no classifier that outperforms the rest in all domains. Their suggestion is that ROCKET and DTW are good enough classifiers to use as an initial technique.

Temporal data has been used in other health care problems. However, the particular characteristics of the available EHR data lead to time series that do not have the common structure found in other time series data. Temporal data usually consists of a long sequence of equally-spaced signals, whereas we have series of EHR data coming from visits made at different time intervals (for each patient and for different patients). Some works in the literature have already addressed the problem of making predictions using the EHR data of patients as temporal data. Itzhak et al. predict acute hypertensive episodes by measuring four vital signs in patients in intensive care units (ICU) (Itzhak, Pessach, et al., 2023). They use temporal abstraction and mine time-interval-related patterns to extract features for a classifier. Sheikhalishahi et al. propose a novel ante-hoc interpretable neural network to provide a prediction of the onset of delirium to prioritise critically ill patients, which is common in the ICU (Sheikhalishahi, Bhattacharyya, et al., 2023). Regarding DR, Rabhi et al. modelled the evolution of HbA1c as an irregular variable-length sequence and tested several deep learning methods to predict whether Type-1 diabetic patients have DR using this single sequential variable (Rabhi, Blanchard, et al., 2022).

In contrast to patients under constant monitoring at ICUs, a Type-2 diabetic patient is typically visited with an average frequency of once a year. Collecting a sufficient series of data takes at least 5-6 years, after the diagnosis of diabetes. Therefore, the classifier system we want to build must work with short sequences. In time series related works, small time series datasets usually have a low number of patients with long signal series (W. K. Wang, Chen, et al., 2022). In this work, the situation is the opposite, with a considerable number of patients, but the length of their time series is short. Moreover, we need to include several variables in the classifier, which, according to specialists, are required to properly diagnose this illness and to differentiate among the several risk levels of DR (i.e., multiclass). In

Section 5.3, a data pre-processing procedure for this kind of EHR time series is proposed.

Another problem we need to face is the inherent class imbalance on DR. Most state-of-the-art time series classifiers are not suited to solve problems with imbalanced class distributions; thus, methods to balance the class distribution at the data level are commonly used. Several approaches can be used to compensate time series for the imbalance among the minority classes:

1. **Sampling methods:** some techniques are applied to the data on the original dataset to oversample and/or undersample it. For instance, in random oversampling, examples of the positive (minority) classes are replicated to balance the class distribution. On the contrary, undersampling consists of randomly removing examples from the majority class.

2. **Synthetic data generation:** the examples introduced to compensate for the class imbalance are artificially generated from the existing data. The most common method is SMOTE (Synthetic Minority Over-Sampling Technique) (Chawla, Bowyer, et al., 2002). Synthetic data points are generated by taking one of the k-nearest neighbours of a sample and randomly choosing one point of the vector that unites the sample and the selected nearest neighbour. In the literature, several variations or methods based on the methodology of SMOTE can be found. For instance, T-SMOTE (P. Zhao, Luo, et al., 2022) is a variation for time series which takes into account the temporal characteristics of the data to select the nearest neighbours. T-SMOTE can be used on both univariate and multivariate time series.

3. **Data augmentation:** slightly modified copies of the data or synthetic examples created from the existing data are introduced to compensate for the class imbalance. Methods are highly dependent on the data types that have to be augmented. In the time series case, Iwana and Uchida (Iwana and Uchida, 2021) analysed over 50 data augmentation methods for time series, and they proposed a taxonomy with four families of methods: Random transformation methods apply a transformation function with some randomness to the time series; Pattern mixing combines patterns to generate new ones, which overcomes the assumption that all random transformations are possible on the data; Generative models use either statistical or neural network models to sample time series from feature distributions; Time series decomposition

uses feature extraction techniques to extract features or underlying patterns, which are then used to generate new examples.

In medical diagnosis, the patients' values of the different risk factors are not totally independent. Although doctors know that there are some underlying relations, they are not usually completely defined. For instance, doctors may know that some combinations of values are not possible. Consequently, it is important that the balancing method used does not generate examples that may not be real, as this may hamper the quality of the classifier built. This chapter proposes a new method that combines both synthetic data generation and random transformation data augmentation. On the one hand, short series are extended by synthetically generating the missing data. On the other hand, we are also conditioning the generated data to be similar to other existing examples (i.e., to a real patient), which is something that cannot be assured when using interpolation without introducing further pre-processing. In Section 5.4, the proposed method is explained.

5.3 Time series pre-processing for EHR data

Most state-of-the-art classification techniques require time series of equal length and with regular time intervals. That is, the difference between each of the two consecutive time points is always the same. When data comes from sensors, this requirement can be easily satisfied. On the contrary, when collecting data from patient's visits from the EHR, we face the problem of different time spans between consecutive visits, as it may depend on the patients' health state or the availability of doctors in certain periods. This is the case of Diabetic Retinopathy monitoring.

5.3.1 Diabetic Retinopathy series data

At each visit, ophthalmologists collect some clinical and analytical data about the diabetic patient, which is stored in his or her EHR.

After some years, each patient has a sequence of values for those variables, which are stored in his or her EHR, together with the DR diagnosis value assigned by the ophthalmologist in each visit ($DR = \{0, 1, 2, 3\}$). The DR diagnosis is included as a categorical variable in the training dataset, except for the last entry of the sequence, since this is the value the system must predict. This way, we can use the previous DR evolution to train the time series classifier. Moreover, we can use the last DR value as ground truth to validate the output value.

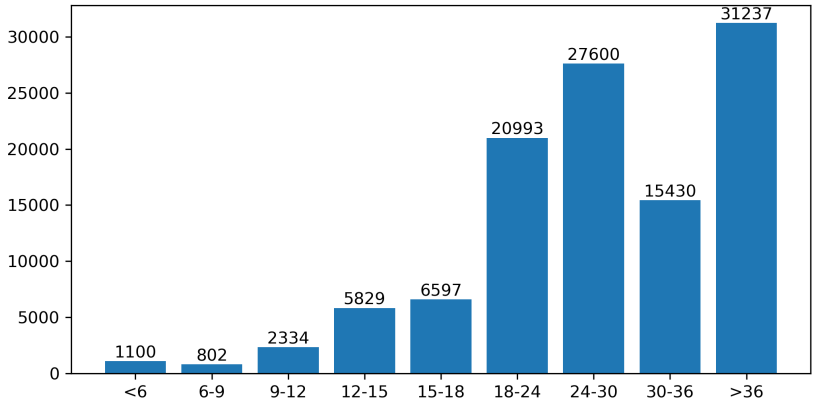


FIGURE 5.2: Mean frequency in months of patients visits to the ophthalmologists

From the dataset of series collected, we discard the ones with a single visit. The frequency of visits in months is deployed in Figure 5.2. It can be seen that the frequency of consecutive visits is not homogeneous. We also observe that most patients have a visit frequency of 18 months or higher, needing many years to collect a sufficiently long series of data.

After confirming the complexity of the Catalan diabetic population dataset in terms of both short length and irregular frequency, an appropriate pre-processing procedure is proposed in this chapter. It is explained in the following subsections.

5.3.2 Data binning

In order to determine the length of the patients' time series in the DR dataset, a study was conducted on the number of visits per patient. The counts are shown in Table 5.1. As can be observed, the majority of patients have a short number of visits. High-risk patients should be visited at most annually (some of them even in six months or less, Figure 5.2), and low-risk patients can be visited with a lower frequency (18 to 30 months).

According to the medical experts in DR, we decided to build series with yearly intervals. Thus, for patients with at least two visits, a binning has been applied with one year bins. In the cases where multiple visits have occurred in the same year, they are aggregated in a single visit that represents the status of the patient that year. To perform the aggregation, each variable has been handled differently.

TABLE 5.1: Number of visits per patient

Number of visits	Number of patients	Percentage (%)
1	119142	51.56
2	61465	26.6
3	30257	13.09
4	12961	5.61
5	4893	2.12
6	1564	0.68
7	551	0.24
8	171	0.074
9	47	0.02
10	11	0.0048
11	1	0.00043
12	1	0.00043

Categorical variables (TTM, HTAR and DR) take the maximum value. In all those cases, the higher the category, the worse is the health of the patient. By using the maximum, the worst status of the patient will be kept. The numerical variables (Age, EVOL and MA) have also been aggregated by using the maximum. So, Age and EVOL take the more up-to-date value for that year, while MA uses the maximum value, which is the worst one in the period. Finally, all the other numerical variables (HbA1c, CKDEPI and BMI) have been aggregated using the mean.

After binning the data, we have 228,956 patients with sequences of less than six entries (i.e., data from at least six different years). A total of 2108 patients have sequences between 6 and 12 years, whose frequency is shown in Figure 5.3. In the next subsection, the transformation process to obtain equal-length time series is explained.

5.3.3 Time series transformation

As indicated before, most TSC require that all the time series of a dataset are of the same length. Observing the DR binned data distribution, Figure 5.3, we selected the length of the time series to be of 10 years. According to the experts, this is a reasonable amount of historic data to be used to perform the DR risk assessment. In this subsection, we propose a procedure to obtain series of 10 years from the DR binned data.

After binning the data, the patients with fewer than six entries (i.e., data from at least six different years) in their time series have been excluded from the dataset. We considered this threshold to be the minimum amount of past information

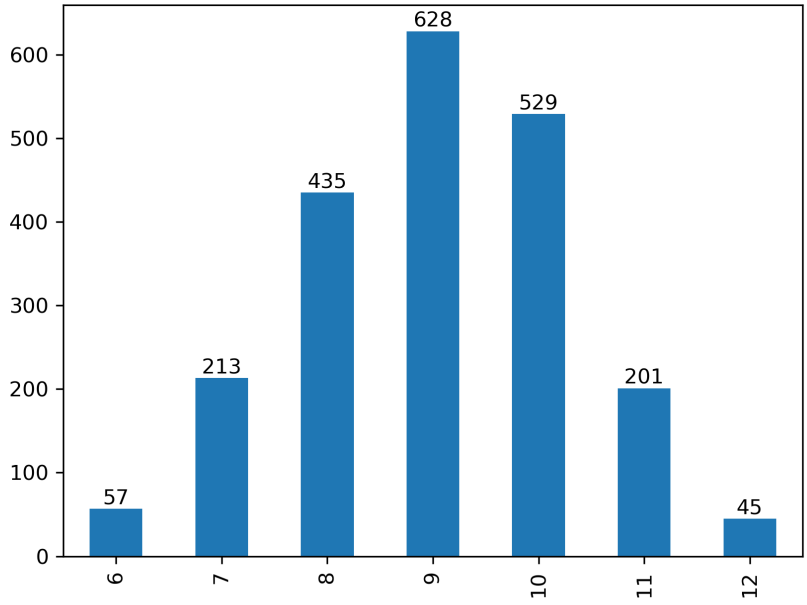


FIGURE 5.3: Frequency of the length of binned time series

needed to perform the transformation without inferring too much data, which might lead to incorrect data.

Three cases are considered according to the length l_i of a time series.

1. $l_i > 10$: the time series is truncated, preserving the 10 more recent entries.
2. $l_i = 10$: no transformation is applied to the time series.
3. $l_i < 10$: the time series is completed to a length of 10 by introducing fictive entries.

In the last case, as we have sequences of at least 6 entries, the maximum number of fictive ones is 4. One option is to use a linear interpolation to directly convert the time series into the proper length. With this approach, existing values are used to approximate some function $y = f(x)$, which is then used to find the values of the missing points between the extreme values. However, for this medical

problem, this method is too simple. Instead, we propose to use a double interpolation approach, which takes into account the specific characteristics of the patients' variables. It is composed of the following steps:

1. **Data initialization:** each binned data entry is assigned to a year. The years without available data are considered missing data. As an example, the first column on Figure 5.4 shows the six binned data values for a patient visited between 2012 and 2019 (8 years).

2. **First interpolation:** missing time-points on the initial data are filled according to each variable. Numerical variables are interpolated, taking into account the length of the intervals to be filled. The meaning and mathematical properties of the variables are taken into account; for example, age and EVOL must be monotonic non-decreasing, as they cannot decrease from one year to the next. For categorical variables, a backfill interpolation is performed. It uses the next existing value, so we use the latest available category. An example is shown in the second column of Figure 5.4, where two entries have been added in 2013 and 2017 for all variables.

3. **Second interpolation:** in the previous step, a complete time series has been obtained, but it still does not have the minimum required length. In this step, it will be linearly interpolated to the desired length, which is 10 for the DR case study. Additionally, for categorical variables, decimal values obtained at interpolation are rounded to avoid nonexistent categories. Because of the quality of the first interpolation step, we are not expecting the rounding to result in a significant change in the category assigned. Finally, time stamps are also replaced with ordered integer values $[0, 10)$, since it is not important the specific year, but their order. The third column in Figure 5.4 shows the final output of this step for this patient example.

The proposed time series interpolation method has been compared to a basic linear interpolation applied to the initial sequence of data. The comparison has been evaluated with the widely-used Dynamic Time Warping (DTW) as a distance measure between the sequences obtained with the two methods. The dependent version of DTW, DTW_D, has been selected based on the study made by Pasos et al. (Pasos-Ruiz, Flynn, et al., 2021). Results are shown as histograms in Figure 5.5.

It is clear from the distribution represented in the two histograms that the proposed interpolation method leads to equal-length time series that are more similar to the original short sequence than using a single linear interpolation. For

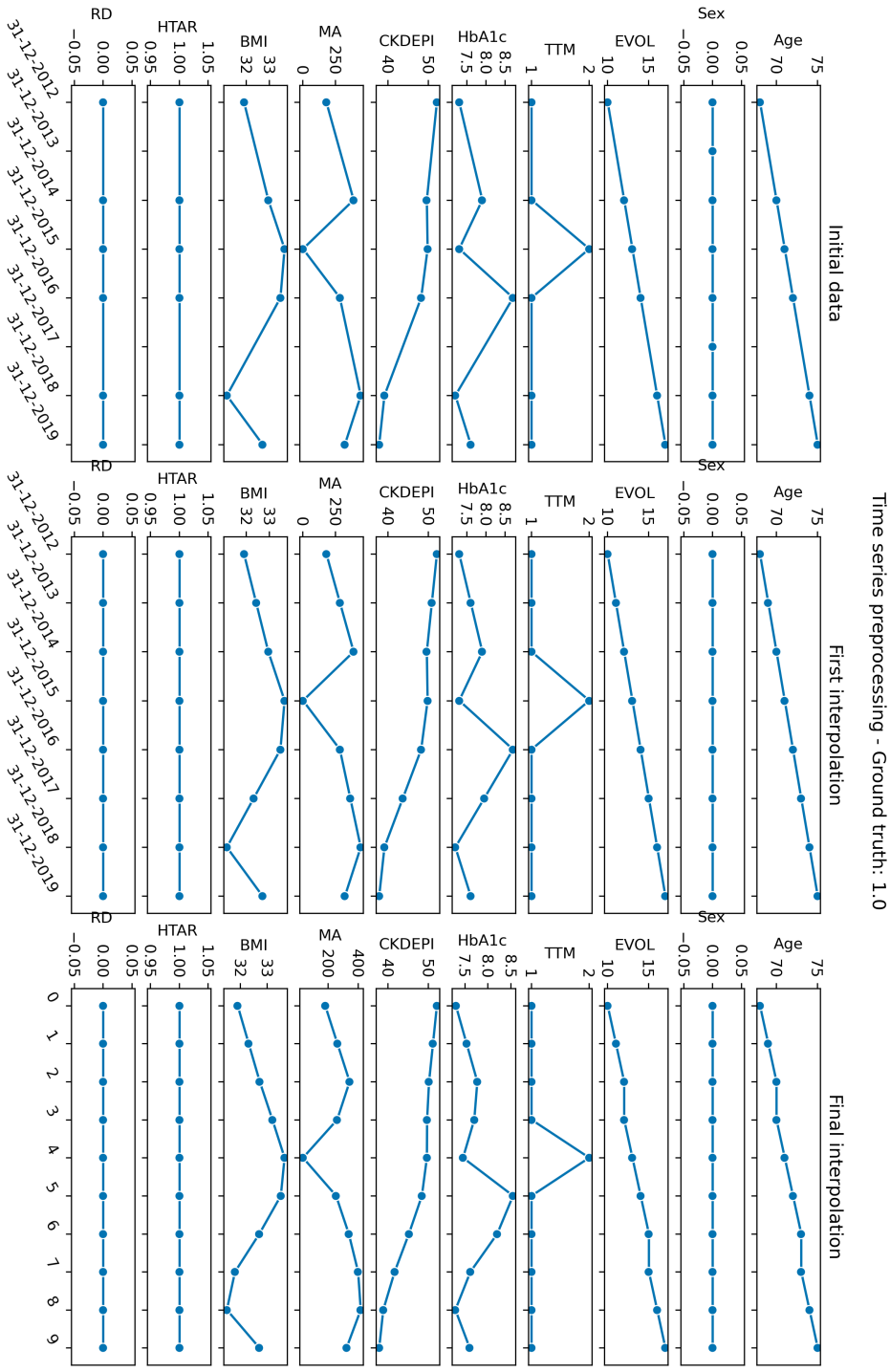


FIGURE 5.4: Double interpolation example for one patient

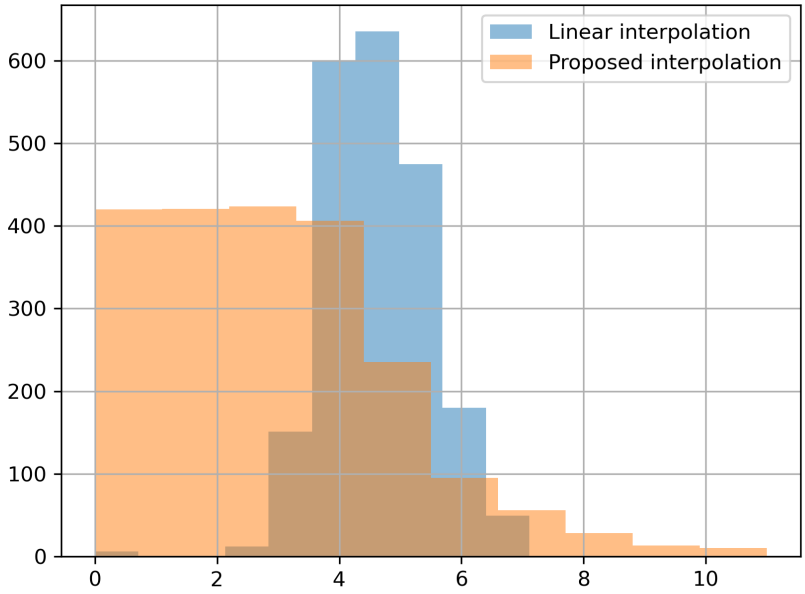


FIGURE 5.5: Histogram of DTW distances, comparing a linear interpolation with the proposed double interpolation

the double-interpolation method, the distance is below four in most of the series, with a few exceptions greater than six, while for the linear interpolation, most of the sequences have a distance between four and six, with almost no cases of very similar sequences (those at a distance below three).

After the interpolation stage, the last transformation applied to the series consists of a process of data standardization and encoding. Categorical variables have been encoded using one-hot encoding (OHE), whereas the numerical ones have been standardized by subtracting the mean and scaling to unit variance. This encoding is a requirement for the classifiers that we have used in this study.

5.4 Time series generation

After applying the previous data preparation steps, a multivariate multiclass time series dataset with 2108 patients is obtained. This quantity is quite low for training a classifier. It is due to the fact that many patients data were discarded after the

binning stage, specifically the ones with sequences of short length, as the interpolation method could not find appropriate values. In this section, we propose to take advantage of this patient's data to generate partially synthetic instances for the minority classes.

The proposed method for the generation of examples distinguishes two types of data sets, defined as follows:

- C_p are sets of complete time series for the minority classes, with length l_c. In the case of DR, it consists of three sets, one for each of the DR positive categories, $p \in \{1, 2, 3\}$.
- I is the set with incomplete time series, i_j, each one with a short length l_j. The length must be in the range $l_{min} \leq l_j < l_c$, where l_{min} must be determined by the characteristics of the data; in case of DR, $l_{min} = 5$.

The generation of new examples of length l_c will be done by means of extending (i.e., boosting) the information available in existing short series in I. For each minority class p, the additional entries added at the end of the existing sequence will take into account the information available in the set C_p. In that way, we introduce data values that are feasible, as some other patient has had similar values. In the following subsections, the method to boost the incomplete set I using the complete set C_p is explained. A different treatment is given depending on the nature of the variables. The method is applied to all examples of the incomplete set, $i_j \in I$, for each of the minority classes $p \in \{1, 2, 3\}$. Examples in I do not have a ground truth value about the DR diagnostic; hence, they can be completed using examples from different classes, generating different sequences for each class.

5.4.1 Demographic variables

First, we consider the demographic variables, whose progression is known in advance. In the DR case study, they are age, gender and EVOL (duration of diabetes). Age and EVOL are numerical, and they are measured in years, so at each time point in the series (yearly intervals), they increase by one unit. Gender is a categorical variable with a value fixed along the time, so the same category (woman or man) is maintained equal in all the new entries for each given time series i_j.

5.4.2 Medical variables

These are variables that store clinical and analytical information related to health. For the medical variables, we calculate the distance between a given incomplete series $i_j \in I$ (with length l_j) and a complete series $c_{p,k} \in C_p$ (with length l_c). As $l_j < l_c$, for the complete series, we only consider the first l_j entries for the distance calculation. Dynamic Time Warping (DTW) has been used as the distance measure for comparing the sequences.

This comparison is performed for each of the minority classes p. For each class, we find the example from the complete set with the minimum distance to i_j, i.e., the most similar in class p, denoted c_{p,sim_j}.

$$c_{p,sim_j} = argmin_k(DTW(i_j, c_{p,k}))\ \forall c_{p,k} \in C_p \tag{5.1}$$

Once we know the most similar complete series to an incomplete one, the procedure for assigning the following missing values to the sequence depends on being a numerical or categorical variable.

5.4.2.1 Categorical variables

We have three categorical variables: TTM, HTAR, and the class label DR. Their values in the incomplete entries of i_j are completed using the categorical values of the most similar series, c_{p,sim_j}. This corresponds to the missing time points $t \in (l_j, l_c]$. Regarding the class variable DR, which must be monotonic non-decreasing according to the medical specialists, a forward fill is applied in the time points where copying the value would produce a decrease from the previous DR level. This process mainly affects the first generated time points, where some discrepancies between the incomplete and complete sequences could be found.

5.4.2.2 Numerical variables

For the management of numerical values, we propose a procedure based on fuzzy sets. Doctors usually work with ranges of values with fuzzy boundaries rather than with precise numerical values. We consider that for each numerical variable $a \in A$, we can define a linguistic fuzzy variable f_a with a fixed set of ordered labels. Each label has a fuzzy set with its corresponding membership function, $\mu_{x \in f_a}$. In our case study, ophthalmologists provided appropriate linguistic labels and fuzzy sets for the numerical variables $A = \{CKDEPI, HbA1c, MA, BMI\}$.

For each variable ($a \in A$) and for all missing time points $t \in (l_j, l_c]$, the following procedure is proposed:

1. A forecasting method is used to predict the next numerical value for the incomplete time series, $i_j(a,t)$. Drift forecasting has been chosen because of its simplicity. Moreover, the amount of available past data is limited, so more complex forecasting techniques are not needed. It fits a line between the first and last points of the series and extrapolates them to the future.

2. The value of the same time point is obtained from the nearest complete time series, $c_{p,sim_j}(a,t)$.

3. The fuzzy sets of the variable f_a are then used to obtain the label with maximum activation for both the incomplete and complete time series values, x and y, respectively. If $x = y$, the forecasted value is stored in $i_j(a,t)$. Otherwise, a random value with maximum activation on the fuzzy term y is the one assigned to $i_j(a,t)$.

By forcing the forecasted value to be similar to one in the complete sequence, we can generate new examples that, although not having the same values, are similar. The use of fuzzy sets permits the assignment of values that are fuzzified with the same label, which means that they are falling in the same category according to the vocabulary given by the ophthalmologists.

In Section 5.6.3 random oversampling is compared to this new method for the DR dataset.

5.5 Multivariate multiclass time series classifiers

In this section, we introduce the four multivariate multiclass time series classifiers we have used. They are the best-performing classifiers according to the review of Pasos et al. (Pasos-Ruiz, Flynn, et al., 2021). Moreover, they are representatives of different kinds of approaches to classification.

5.5.1 K-nearest neighbours

K-nearest neighbours (KNN) is one of the most simple yet good-performing methods for classification problems. In MTSC problems, KNN is usually taken as the baseline result by using 1-nearest neighbour (i.e., $K = 1$). As it is a distance-based classifier, the selection of the distance measure is crucial. In this work, for time

series comparison, Dynamic Time Warping is used. Several strategies can be used to compute a DTW distance in the multivariate case:

1. Independent warping (DTW_I): the distance between time series is computed independently for each variable. The resulting DTW distance is the sum of the independent distances.

2. Dependent warping (DTW_D): a warping is assumed to be correct across all time series. The Euclidean distance is computed between the two vectors, representing all time series. Then DTW is applied to all time series simultaneously.

3. Adaptive warping (DTW_A): instead of selecting one of the previous two approaches, the choice of using DTW_I or DTW_D is made based on the characteristics of the data.

5.5.2 ROCKET

ROCKET (Random Convolutional Kernel Transform) combines numerous convolutional kernel transforms with a linear classifier (Dempster, Petitjean, et al., 2020). Kernels are randomly chosen with a variety of lengths, dilations, paddings, weights and biases. For the multivariate case, they are also randomly assigned a time series. A feature map is created by convolving a kernel, and it is aggregated to produce two features per kernel, which are the maximum value and the proportion of positive values (ppv). A linear classifier, such as a ridge regression classifier, is then trained on the extracted features.

5.5.3 TapNet

TapNet (Time series attentional prototype network) (X. Zhang, Gao, et al., 2020) defines an architecture to combine the strengths of both traditional and deep learning approaches to MTSC. It consists of three main components. First, a dimension permutation, which randomly combines the time series, is used to model the interactions between the different variables. Second, embeddings are learned using a Long Short-Term Memory structure and a 1-dimensional Convolutional Neural Network, in order to model the sequential information of the time series. Finally, Attentional Prototype Learning is applied using the learnt embeddings. It generates an embedding prototype of each class, and the classification is then based on the distance to these prototypes.

5.5.4 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are well-known neural networks in deep learning. They have been mainly used in image analysis problems, although Zhao et al. proposed a CNN for time series classification (B. Zhao, Lu, et al., 2017). They alternate the usage of convolutional and pooling operations to obtain deep features of the data. The final representation of the data is in a feature layer, which is created by connecting all the obtained feature maps. Finally, the feature layer is used to perform the classification using a multi-layer perceptron model.

5.6 Experimental results

This section presents the obtained experimental results. In subsection 5.6.1 the Diabetic Retinopathy dataset is presented. Subsection 5.6.2 compares the classification performance of non-temporal classifiers on long-term diabetic patients. Finally, subsection 5.6.3 discusses the obtained results on the temporal datasets, and a comparison between the tested classifiers is performed.

5.6.1 Dataset

The data used in this study comes from the diabetic population in our region, Catalonia, from period 2010 to 2021. It is a private dataset provided to us in the framework of a national research project. Even though it is an anonymised dataset, it contains sensible healthcare data from patients. Therefore, it is protected by the national privacy laws, and it cannot be disclosed. After performing the preprocessing steps explained in Section 5.3, we obtained a set with 2108 sequences of 10 entries. This dataset has been divided into two: training and testing, with 70% and 30% of the data, respectively (Table 5.2). It can be clearly seen the high imbalance towards the negative ($DR = 0$) class, which makes it more difficult to predict the classes with higher DR risk.

When balancing is needed, we either applied oversampling or the method proposed in Section 5.4 for completing short sequences for the three minority classes. In the latter, we use the incomplete set I. It is composed of previously discarded time series because of their short length. In this work, we just considered the incomplete series of the maximum length available, $l_j = 5$, which corresponds to

TABLE 5.2: Distribution of the Diabetic Retinopathy time series data in training/testing

Class/Dataset	**Training (70%)**	**Testing (30%)**	**Total**
DR = 0	1212 (82.2%)	518 (81.8%)	1730 (82.1%)
DR = 1	148 (10%)	61 (9.64%)	209 (9.9%)
DR = 2	92 (6.2%)	41 (6.5%)	133 (6.3%)
DR = 3	23 (1.6%)	13 (2.1%)	36 (1.7%)
Total	1475	633	2108

4547 patients. This is because the shorter the incomplete time series, the more fictive data has to be introduced, increasing the probability of introducing erroneous data.

5.6.2 Long-term DR patients classification

The first study consists of comparing the performance of the Retiprogram model explained in Chapter 4 on patients suffering from long-term Diabetic Retinopathy. Although the model we are using in this test is the same as in the previous chapter, we are using a different DR dataset. Retiprogram does not take into account the history and evolution of the patient since the first diagnosis of DR. On the contrary, it uses a single point state of the patient to diagnose the risk of DR. We want to prove that such a model is good for the initial diagnosis of patients who are starting DR (called new DR patients), but not for patients with an advanced progression of DR (called long-term DR patients).

The tests consists of comparing how Retiprogram classifies patients the first time they are diagnosed with diabetes and how it classifies them some years later. To perform this comparison, we have created two non-temporal DR datasets. One for new patients, which contains the first entry of the temporal series; another for long-term patients, which contains the latest entry instead.

All the available data of the 2108 patients with a complete series has been used, and Retiprogram was tested on both datasets. As the DR class in the ground truth is not the same in both datasets, the number of patients in each class is different in each dataset. The obtained confusion matrices for both tests are depicted in Figure 5.6, and their corresponding metrics are in Table 5.3.

All metrics about the classification of new patients are better than the ones for long-term patients. Having in mind that the ground truth is different in each dataset, it is still clear that the model has more difficulties correctly classifying the long-term patients. Most errors on long-term patients are due to misclassifications

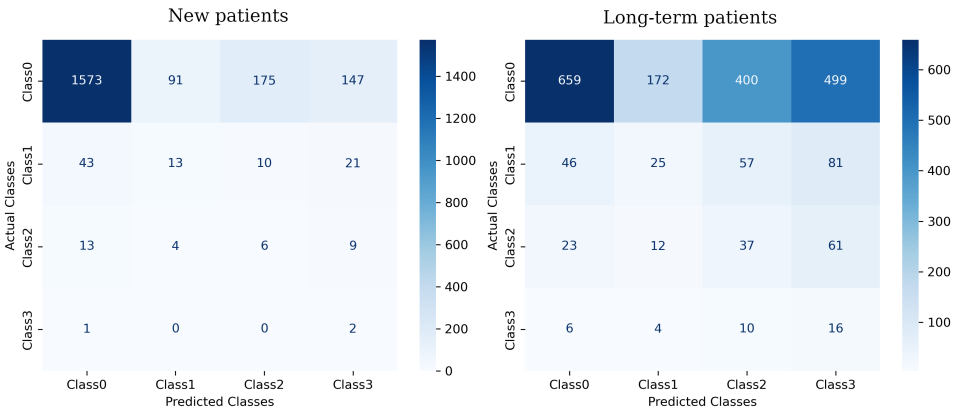


FIGURE 5.6: Comparison between new patients (left) and long-term patients (right)

TABLE 5.3: Performance indicators of new patients and long-term patients

Classifier/Metric (%)	New patients	Long-term patients
Accuracy	75.6	35
Kappa	0.095	0.068
Macro Precision	44.9	30.6
Weighted Precision	69.4	35
Macro Recall	28.2	27.8
Weighted Recall	75.6	35
Macro F1	27	20.4
Weighted F1	68.6	24

for patients that belong to $DR = 0$, as shown on its confusion matrix in Figure 5.6. Once a diabetic patient has been diagnosed with this disease, he or she starts taking some medications in order to have under control some dangerous values. These changes generates confusion for Retiprogram when evaluating the current patient's state, making it really hard for the model to predict the current status of DR. These results confirm the need for creating a new model that uses all the temporal information available for each patient, thus taking all states into account to perform a good prediction of the DR degree for those long-term patients.

5.6.3 Results and discussion

We tested the performance of the multivariate time series preparation and classification method explained in Section 5.5 on the DR dataset presented in Section 5.6.1.

To perform the tests, we have used the *sktime* toolkit (Löning, Király, et al., 2022). Tests have been performed both oversampling the data and using the proposed time series fuzzy generation method. Some of the default parameters of the classifiers have been used, but others have been adjusted, such as the ones related to the length of the series. Moreover, the epochs, activation and loss functions of the tested neural networks have also been experimentally tuned to fit with our ordinal multiclass problem. In the case of KNN, we found that oversampling requires a higher k value than the fuzzy generation method to avoid overfitting. The final configuration values for these parameters are given in Table 5.4.

TABLE 5.4: Parameters of the classifiers for time series

Classifier	**Parameters**
KNN Oversample	k: 9, distance: DTW_D
KNN Fuzzy Gen.	k: 3, distance: DTW_D
ROCKET	Number of kernels: 200
TapNet	Epochs: 20, batch: 16, activation: softmax, loss: categorical crossentropy, filter: (32, 32, 16), kernel: (8, 5, 3), layers: (50, 30)
CNN	Epochs: 20, batch: 16, activation: softmax, loss: categorical crossentropy, kernel: 5, avg pool size: 2

A 10-fold cross-validation has been chosen to validate the performance of the parameters, choosing the values with the best performance. The whole training set shown in Table 5.2 has been used to perform the validation. First, the data is split into 10 folds. One of them is selected as the test set. The rest is combined to form a training set for that fold. The training data is then balanced either by means of oversampling or by employing our proposed fuzzy sample generation method. Finally, the model is trained using the balanced training data and then tested using the test fold. The process is repeated using all folds as test data, and the obtained performance metrics in all folds are averaged.

The results are shown in Table 5.5. Columns contain each of the tested classifiers, either balancing the data using oversampling (O), or the proposed fuzzy sample generation (FG). Several standard multi-classification metrics have been used to measure the performance of the different classifiers. In particular, we consider accuracy, precision, recall, F1-Score and the quadratic weighted kappa. For precision, recall and F1-Score, we have taken both macro and weighted averages. Because of the class imbalance, we are expecting the macro average to be lower than the weighted average. Quadratic weighted kappa is a relevant metric for

this ordinal multiclass problem because it penalizes the mistakes according to the distance between the ground truth and the predicted class. In medical decision support, a short difference between the correct class and the predicted one is crucial in order to not affect the health of the patient. Hence, we aim to minimise it as much as possible.

TABLE 5.5: 10-fold cross-validation performance indicators of different DR series classifiers

	CNN		**KNN**		**ROCKET**		**TapNet**	
	FG	**O**	**FG**	**O**	**FG**	**O**	**FG**	**O**
Accuracy	94	90.6	91.2	89.6	91.1	93.2	**94.1**	86.4
M. Precision	80.4	74.7	71.3	68.8	86.7	**91.1**	82.8	76.7
M. Recall	75.9	83.1	60.5	61.1	64.9	75.4	75.4	**90**
M. F1	77.1	77.4	63.8	63	71.3	**80.3**	77.1	79.9
W. Precision	93.7	92.4	90.9	89.6	90.7	93.3	**93.9**	93.3
W. Recall	94	90.6	91.2	89.6	91.1	93.2	**94.1**	86.4
W. F1	**93.6**	91.3	90.6	89.1	89.7	92.7	**93.6**	88.1
Kappa	0.856	0.764	0.640	0.670	0.736	0.805	**0.868**	0.738

Comparing the results obtained by using oversampling to fuzzy sample generation, overall, fuzzy sample generation (FG) obtains better accuracy, kappa, and weighted metrics. In contrast, oversampling has better or similar results on the macro recall and macro F1-Score. The only exception is the ROCKET classifier, where oversampling obtains better results in all cases. The best result for each of the metrics is marked in bold. TapNet gets the best overall results, with the best accuracy, weighted metrics and quadratic weighted kappa. CNN and ROCKET also have good overall results. CNN draws with TapNet on the best weighted F1-Score, and ROCKET has the best macro precision and macro F1-Score. KNN, on the other hand, has the worst results among the tested classifiers.

A second evaluation has been done with the traditional training/testing division using a percentage split evaluation. Thus, using the best configuration parameters, the models were trained again on the whole training set (70% of the data) and tested on the test set (30% of the data). The training data was also balanced using O and FG techniques. The obtained performance indicators for the different tested classifiers and balancing techniques are shown in Table 5.6.

In this evaluation, it can also be observed how fuzzy sample generation again obtains better overall results than oversampling. The classifiers that make the greatest improvements when using the proposed balancing technique are deep learning classifiers (i.e., CNN and TapNet). The classifier performing worst among

TABLE 5.6: Performance indicators of different DR series classifiers in testing stage

	CNN		KNN		ROCKET		TapNet	
	FG	**O**	**FG**	**O**	**FG**	**O**	**FG**	**O**
Accuracy	91.63	81.36	85.15	84.04	90.68	89.26	**93.68**	91.94
M. Precision	78.83	52.17	55.81	70.63	**89.71**	84.96	84.26	77.41
M. Recall	76.49	63.87	50.15	45.97	64.42	66.22	**79.39**	70.25
M. F1	76.89	56.11	51.91	48.32	73.47	73.36	**81.17**	72.58
W. Precision	91.78	86.14	84.28	83.53	90.20	88.60	**93.41**	91.55
W. Recall	91.63	81.36	85.15	84.04	90.68	89.26	**93.68**	91.94
W. F1	91.52	83.21	84.42	82.92	89.43	88.38	**93.36**	91.25
Kappa	0.840	0.650	0.551	0.623	0.734	0.729	**0.868**	0.856

the tested ones is KNN, for which the obtained metrics are lower than the rest of the classifiers.

In contrast, TapNet using fuzzy sample generation has achieved the best results for all metrics except the precision. It specially stands out on the quadratic weighted kappa, with a value of 0.868, which is a remarkably high score for this measure. These results confirm the ones obtained with cross-validation. We also see that testing results are quite close to the ones of 10-fold cross-validation, confirming the lack of overfitting.

The other two classifiers, ROCKET and CNN, are similar in terms of the metrics that take into account the class imbalance, such as accuracy, weighted recall, and weighted F1-Score. That is because they are able to properly classify most of the negative examples.

Metrics that do not compensate for class imbalance, macro recall and macro F1, are more interesting for our use case because they will show which classifiers can better identify the positive patients. Here, KNN is again the worst. ROCKET and CNN have similar results, with CNN performing slightly better. Tapnet shows the best macro recall and the second best in macro precision. To further analyse the results obtained by TapNet, the confusion matrix for this test is depicted in Figure 5.7.

Tapnet is excellent in classifying the minority classes (categories 2 and 3), which is important in the medical domain. Consequently, the study concludes that TapNet is the classifier that exhibits superior performance among the tested classifiers for short time series in the Diabetic Retinopathy problem. Its high quadratic weighted kappa demonstrates its ability to properly classify positive examples in the ordinal case. Additionally, when an error is made, the class assigned is close

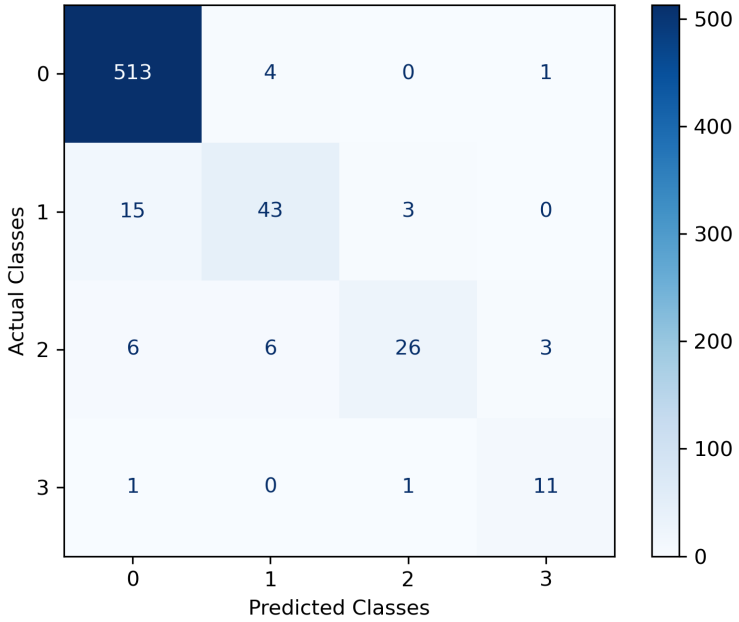


FIGURE 5.7: Confusion matrix in testing with TapNet and fuzzy sample generation balancing

to the correct one.

Finally, the same test set has been used to evaluate the performance of Retiprogram for the last time point of the DR series. Its results are shown in Figure 5.8. We observe much confusion in the class assignments made for patients in all rows and large errors in the case of patients with $DR = 0$.

From the obtained results, it can be concluded that using temporal information on long-term diabetic patients is beneficial in order to correctly classify them into their current DR degree.

5.7 Conclusions

In this chapter, we presented a novel approach for estimating the different levels of Diabetic Retinopathy using only retrospective data from the EHR of the patient. Motivated by the discussed special characteristics of such medical series data, a technique for pre-processing EHR data has been presented. As a first contribution,

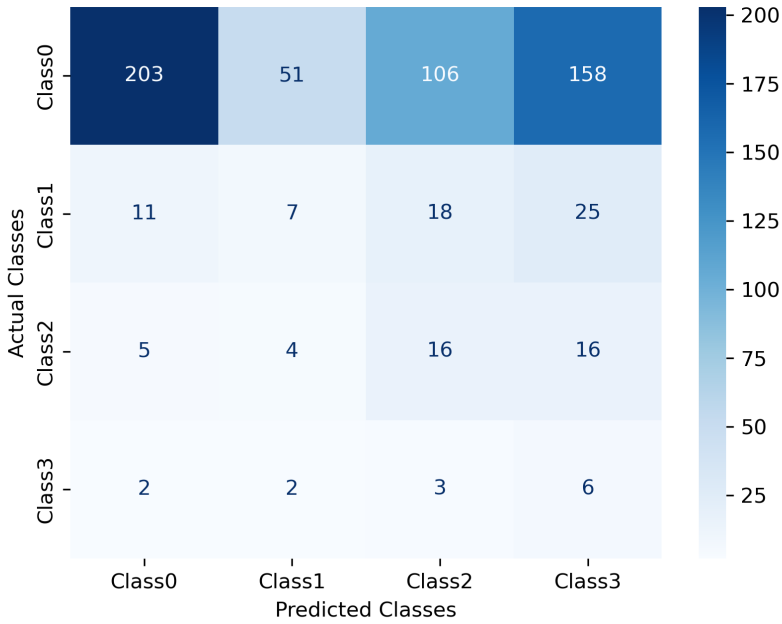


FIGURE 5.8: Retiprogram results on the test set

to construct multivariate time series of the same length for all the patients, missing entries have been completed with a double interpolation technique. Results show a much closer similarity with the original sequence when using the proposed method, compared with a classic linear interpolation. As a second contribution, we presented a new fuzzy-based approach to boost short time series to generate new examples that may alleviate the problem of class imbalance in health care data. It consists of completing short sequences by using information from similar completed ones. Three types of variables have been distinguished when completing their values. For medical numerical values, a method based on the use of linguistic fuzzy variables has been proposed. By using the membership functions, we can find new input values that generate series similar enough to real examples.

Lastly, several multivariate time series classifiers have been compared using the DR EHR data time series prepared with the previous techniques. From the results, we conclude that TapNet is the best classifier for this problem. TapNet is able to appropriately learn the underlying patterns of the minority DR examples, as metrics depict. It has achieved a kappa value of 0.868 and an accuracy of 93.7%,

which are outstanding results for such an imbalanced health care problem.

Chapter 6

Conclusions and future work

Clinical Decision Support Systems have been extensively studied to assist clinicians in several tasks. Recent advancements in AI-based techniques make it possible to evolve these kinds of systems to improve patients' diagnosis precision and users' satisfaction, achieving also a cost and time reduction for health centres. For diseases such as Diabetic Retinopathy, they can be really helpful, as detecting the illness in its early stages can avoid severe consequences on vision loss as well as avoiding unnecessary retina screening. Instead of following the usual approach based on eye-fundus image analysis, this thesis has focused on the use of clinical data available in the Electronic Health Records. The Retiprogram model based on Fuzzy Random Forest has been the starting point for this doctoral work. In this chapter, we conclude the PhD thesis by summarising the contributions made and proposing some avenues for future research.

6.1 Summary and discussion of the thesis contributions

This dissertation continues the line of research of the ITAKA research group on CDSS systems for the diagnosis of DR using the Electronic Health Records of DM patients. The initial binary classifier known as Retiprogram is based on a Fuzzy Random Forest, and it is able to assess the risk of developing DR using clinical and analytical data. In this work, we have proposed several methods to improve the classification performance of Retiprogram, by proposing new methods that exploit new data collected in a dynamic environment and in the presence of historical information obtained from EHR.

The first contribution of this dissertation is a proposal to leverage the data that is gathered from the use of a CDSS, in our case, Retiprogram. As the system is in use, the newly obtained data can be used to improve the existing Fuzzy Random Forest model without retraining it from scratch. Besides the reduced time compared to training the model from scratch, as the model retains most of the original model, the validation process of the updated model is quicker. Moreover, the updated models offer improved performance and a focus on the detection of positive patients. In the Diabetic Retinopathy case, all metrics improved with the use of the proposed dynamic updating method. Sensitivity increased from 74.6% to 91.7%, specificity from 81.5% to 84.4% and F1-score from 61.8% to 74%. On the Occupancy dataset, sensitivity and F1-score improve, although specificity slightly decreases. Sensitivity increased from 74.7% to 86.9%, F1-score from 67.2% to 70.6% and specificity decreased from 87.6% to 84.6%. In both cases, the method has proven to improve the detection of the positive class. Moreover, the detection of the negative class has maintained similar accuracy values, or they have even been improved. In the Diabetic Retinopathy case, it has been tested using real data from diabetic patients. The proposed method leads to improvements in the assessment of Diabetic Retinopathy risk, which can help to avoid unnecessary screenings on patients and to reduce the workload of the ophthalmologists. In that way, the scarcely available resources can be better distributed among the patients, focusing on the ones that really need them.

Another contribution of this thesis is the redefinition of the two steps of the prediction stage of a binary Fuzzy Random Forest to classify examples of the ordinal multiclass case. The procedure to manage conflicting cases has also been redefined. By allowing the classification into more accurate categories, patients can get more accurate predictions of their risk. Because of the added heuristics, higher classes are also prioritised, which is desired in the medical use case. For both datasets, Diabetic Retinopathy and burnout, we can see that using a Kappa κ validation index works better than accuracy for weighting the trees. In both cases, by using the proposed heuristics, the accuracy is improved by introducing a moderate amount of unknown predictions in the conflicting cases. Diabetic Retinopathy improves from 71.5% to 73.4%, and burnout from 69.6% to 69.9%. The usage of a dynamic way of computing δ_2 using the d parameter also allows for modelling the trade-off between predictions and unknowns, which can be adapted according to the specific data that is being used. Finally, the use of OWA for the calculation of the final decision support is more appropriate than the arithmetic average to give the FRF confidence value for the final class. The number of predictions in the higher

intervals is greater, and as a consequence, the percentage of correct predictions in the higher intervals is also greater. In that way, medical doctors can spend less time and resources managing uncertain cases.

The last contribution of this work is a novel method for estimating the different levels of DR using only retrospective data obtained from the EHR of the patient. Motivated by the special characteristics of this kind of medical series data, we first presented a technique for pre-processing EHR data. To construct an equal-length multivariate time series for all the patients, missing entries have been completed with a double interpolation technique. Quality improvement is seen when computing the DTW_D distance to the original series. With the new algorithm, most of the sequences are at a distance below four, whereas using a linear interpolation obtains most distances between four and six. The proposal also includes a novel technique to generate new positive examples to compensate for the class imbalance in the data, which is quite common in health care problems. The method consists of completing short sequences by using information from similar completed ones. For medical numerical values, a method based on the use of linguistic fuzzy variables has been proposed. By using the membership functions, we can find new input values that generate series similar enough to real examples. Overall, the proposed fuzzy-based method for new samples generation shows improved results over balancing the data using oversampling. The experiments also show how long-term DM patients are harder to classify by common machine learning models. When tested using Retiprogram, new diabetic patients are correctly classified with an accuracy of 75.6%, whereas long-term diabetic patients just obtain a 35% accuracy. By using the historical data from patients, we can overcome the difficulties, obtaining more accurate results. From the tested classifiers, we conclude that TapNet is the best classifier for this problem. It is able to appropriately learn the underlying patterns of the minority DR examples, as metrics depict. It has achieved a kappa value of 0.868 and an accuracy of 93.7%, which are outstanding results for such an imbalanced health-care problem.

With these contributions, all the objectives of this doctoral research have been satisfactory achieved. New techniques and tools are now available not only for the research community, but also for the medical experts of the hospital that supported this work.

6.2 Future work

The advances and outcomes of cutting-edge research work always bring new ideas. Several interesting research lines could be pursued to continue the work presented in this PhD thesis. They are the following:

- The developed methods were mainly focused on Diabetic Retinopathy risk assessment, but they could also be applicable to other domains. The methods could be applied to any use case where a classification with ordered unbalanced classes is required. Some of them have already been tested with other datasets, but a further extensive validation should be made to confirm the possible usage of the methods in other application domains.

- The proposed methods in Chapter 3 have been studied for Fuzzy Random Forests, but they are also applicable to regular Random Forests. The quality of the classification results of the proposed algorithms should also be validated on their non-fuzzy counterparts.

- It would be interesting to study how different metrics affect the method proposed in Chapter 3. Different functions could be considered in the dynamic update method that selects the Fuzzy Decision Trees that are removed or introduced into the updated Fuzzy Random Forest. The metric choice could be affected by the specific data configuration, such as the usage of balanced data instead of imbalanced data or a non-fuzzy dataset instead of a fuzzy one.

- The possibility of including some control mechanisms in the update process of Chapter 3 could also be studied. If a specific update is not improving the Fuzzy Random Forest as much as expected, it could be discarded or tuned in order to obtain even better results. The number of examples added in the *AllData*&*ErrorLT* method could also be controlled to avoid accumulating too much data in use cases with high numbers of update iterations. A way of finding a suitable method to discard the less useful examples could be studied. Moreover, the convergence of the method after some update iterations should also be analysed.

- The ordered multiclass classification method for Fuzzy Random Forests proposed in Chapter 4 could also incorporate the dynamic update method, similar to the one proposed in Chapter 3. However, the dynamic update learning

method should be adapted to the ordered multiclass classification case by selecting an appropriate metric for the update process, and the validity of combining both proposals should be tested.

- The possibility of adapting the proposed time series procedure for incomplete time series of other lengths (shorter or larger) could also be studied in order to evaluate its validity in different data characteristics. A study of different ways of encoding the numerical variables would also be interesting, since they are quite common and relevant in Clinical Decision Support Systems. Taking advantage of fuzzy variables may also bring some contributions in making a high-level linguistic encoding of the data, which could be suitable for this kind of application.

- A final interesting line of work is the combination of the current CDSS-based methods that make use of EHR data with the computer vision models that analyse the eye-fundus images. The current diagnosis workflow for DR explained in Chapter 1 could be completed with a system that is able to combine information from both EHR and image data. Indicators such as the number of lesions, their location, and their type could be valuable information for the classifiers presented in this thesis. The automatisation of the extraction of these indicators from images is currently under study in our research group, however, it is still an open problem as the existing models do not have enough precision and recall.

The open research lines presented in this PhD thesis have demonstrated the potential for developing high-quality clinical decision support systems that can significantly improve medical diagnoses and ultimately lead to an enhanced overall quality of life.

UNIVERSITAT ROVIRA I VIRGILI
FUZZY-BASED MACHINE LEARNING METHODS FOR CONTINUOUS DIAGNOSIS AND PROGNOSIS OF DIABETIC RETINOPATHY
Jordi Pascual Fontanilles

Appendix A

Awards

A.1 Awards

In this appendix, we present the awards obtained by the works of this PhD thesis:

- In the 23rd International Conference of the Catalan Association for Artificial Intelligence (CCIA 2021) that took place on October 20-22, 2021 in University of Lleida, we received the best poster award (ex aequo) in the conference. The work that won the award is entitled "Iterative update of a Random Forest classifier for Diabetic Retinopathy".

CCIA 2021

23rd International Conference of the Catalan Association for Artificial Intelligence

October 20-22, 2021 / Campus de Cap Pont / University of Lleida

BEST POSTER AWARD

The Scientific Committee of the

23rd International Conference of the Catalan Association for Artificial Intelligence

Universitat de Lleida, 20-22 October 2021

certifies that the best paper award (*ex aequo*) goes to

Jordi Pascual Fontanilles, Aida Valls,

Antonio Moreno and Pedro Romero Aroca

for the poster

"Iterative update of a Random Forest classifier for Diabetic Retinopathy"

Cèsar Fernández Camón
Congress Local Chair

Organized by:

ACIA · GREIA research group · Universitat de Lleida · Universitat Rovira i Virgili · Universitat de Girona

Bibliography

Ahmadi, Hossein, Marsa Gholamzadeh, et al. (July 2018). "Diseases diagnosis using fuzzy logic methods: A systematic and meta-analysis review". In: *Computer Methods and Programs in Biomedicine* 161, pp. 145–172. ISSN: 0169-2607. DOI: 10.1016/J.CMPB.2018.04.013.

Ahsan, Md Manjurul, Shahana Akter Luna, et al. (Mar. 2022). "Machine-Learning-Based Disease Diagnosis: A Comprehensive Review". In: *Healthcare 2022, Vol. 10, Page 541* 10 (3), p. 541. DOI: 10.3390/HEALTHCARE10030541.

Atwany, Mohammad Z, Abdulwahab H Sahyoun, et al. (2022). "Deep learning techniques for diabetic retinopathy classification: A survey". In: *IEEE Access* 10, pp. 28642–28655.

Candanedo, Luis M. and Véronique Feldheim (Jan. 2016). "Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models". In: *Energy and Buildings* 112, pp. 28–39. ISSN: 0378-7788. DOI: 10.1016/J.ENBUILD.2015.11.071.

Chawla, Nitesh V, Kevin W Bowyer, et al. (2002). *SMOTE: Synthetic Minority Over-sampling Technique*.

Dempster, Angus, François Petitjean, et al. (2020). "ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels". In: *Data Mining and Knowledge Discovery* 34, pp. 1454–1495. DOI: 10.1007/s10618-020-00701-z.

Dogan, Alican and Derya Birant (Sept. 2019). "A Weighted Majority Voting Ensemble Approach for Classification". In: *UBMK 2019 - Proceedings, 4th International Conference on Computer Science and Engineering*. IEEE, pp. 366–371. ISBN: 9781728139647. DOI: 10.1109/UBMK.2019.8907028.

Dubey, Shradha and Manish Dixit (Sept. 2022). "Recent developments on computer aided systems for diagnosis of diabetic retinopathy: a review". In: *Multimedia Tools and Applications 2022 82:10* 82 (10), pp. 14471–14525. DOI: 10.1007/S11042-022-13841-9.

El Habib Daho, Mostafa, Nesma Settouti, et al. (Sept. 2014). "Weighted vote for trees aggregation in Random Forest". In: *International Conference on Multimedia Computing and Systems -Proceedings*. IEEE, pp. 438–443. ISBN: 9781479938247. DOI: 10.1109/ICMCS.2014.6911187.

Escorcia-Gutierrez, José, Jose Cuello, et al. (2023). "Grading Diabetic Retinopathy Using Transfer Learning-Based Convolutional Neural Networks". In: *"Computer Information Systems and Industrial Management"*. Springer, Cham, pp. 240–252.

Fernández-Delgado, Manuel, Eva Cernadas, et al. (2014). "Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?" In: *Journal of Machine Learning Research* 15, pp. 3133–3181.

Gomes, Heitor M., Albert Bifet, et al. (Oct. 2017). "Adaptive random forests for evolving data stream classification". In: *Machine Learning* 106 (9-10), pp. 1469–1495. DOI: 10.1007/s10994-017-5642-8.

Gomes, Heitor Murilo, Jean Paul Barddal, et al. (2017). "A survey on ensemble learning for data stream classification". In: *ACM Comput. Surv* 50 (23). DOI: 10.1145/3054925.

Guetova, Marina, Steffen Hölldobler, et al. (2002). "Incremental Fuzzy Decision Trees". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 2479, pp. 67–81. DOI: 10.1007/3-540-45751-8_5.

HackerEarth (2020). *Are Your Employees Burning Out?* DOI: 10.34740/KAGGLE/DS/949779.

Ichihashi, H, T Shirai, et al. (1996). "Neuro-fuzzy ID3: a method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning of E". In: *Fuzzy Sets and Systems* 81, pp. 157–167.

Isazadeh, Ayaz, Farnaz Mahan, et al. (Sept. 2016). "MFlexDT: multi flexible fuzzy decision tree for data stream classification". In: *Soft Computing* 20.9, pp. 3719–3733. DOI: 10.1007/S00500-015-1733-2/FIGURES/12.

Itzhak, Nevo, Itai M. Pessach, et al. (May 2023). "Prediction of acute hypertensive episodes in critically ill patients". In: *Artificial Intelligence in Medicine* 139, p. 102525. DOI: 10.1016/J.ARTMED.2023.102525.

Iwana, Brian Kenji and Seiichi Uchida (July 2021). "An empirical survey of data augmentation for time series classification with neural networks". In: *PLOS ONE* 16 (7), e0254841. DOI: 10.1371/JOURNAL.PONE.0254841.

Kalles, Dimitrios and Tim Morris (Sept. 1996). "Efficient incremental induction of decision trees". In: *Machine Learning* 24 (3), pp. 231–242. ISSN: 0885-6125. DOI: 10.1007/bf00058613.

Khalid, Saif, Hatem A. Rashwan, et al. (Mar. 2024). "FGR-Net: Interpretable fundus image gradeability classification based on deep reconstruction learning". In: *Expert Systems with Applications* 238, p. 121644. ISSN: 0957-4174. DOI: 10.1016/J.ESWA.2023.121644.

Khan, Umer, Hyunjung Shin, et al. (2008). "WFDT: weighted fuzzy decision trees for prognosis of breast cancer survivability". In: *Proceedings of the 7th Australasian Data Mining Conference-Volume 87*. Citeseer, pp. 141–152.

Kim, Hyunjoong, Hyeuk Kim, et al. (Dec. 2011). "A weight-adjusted voting algorithm for ensembles of classifiers". In: *Journal of the Korean Statistical Society* 40.4, pp. 437–449. ISSN: 12263192. DOI: 10.1016/j.jkss.2011.03.002.

Lakshminarayanan, Balaji, Daniel M Roy, et al. (2014). "Mondrian Forests: Efficient Online Random Forests". In: *Advances in Neural Information Processing Systems* 27, pp. 3140–3148.

Landis, J Richard and Gary G Koch (1977). "The Measurement of Observer Agreement for Categorical Data". In: *Biometrics* 33 (1), pp. 159–174. DOI: 10.2307/2529310.

Li, Hong Bo, Wei Wang, et al. (2010). "Trees Weighting Random Forest method for classifying high-dimensional noisy data". In: *Proceedings - IEEE International Conference on E-Business Engineering, ICEBE 2010*, pp. 160–163. ISBN: 9780769542270. DOI: 10.1109/ICEBE.2010.99.

Livieris, Ioannis E., Andreas Kanavos, et al. (Mar. 2019). "A weighted voting ensemble self-labeled algorithm for the detection of lung abnormalities from X-rays". In: *Algorithms* 12.3, p. 64. DOI: 10.3390/A12030064.

Löning, Markus, Franz Király, et al. (Sept. 2022). *sktime/sktime: v0.13.4*. Version v0.13.4. DOI: 10.5281/zenodo.7117735. URL: https://doi.org/10.5281/zenodo.7117735.

Ogunyemi, Omolola and Dulcie Kermah (2015). "Machine Learning Approaches for Detecting Diabetic Retinopathy from Clinical and Public Health Records". In: *AMIA Annual Symposium Proceedings* 2015, p. 983.

Ogunyemi, Omolola I., Meghal Gandhi, et al. (2019). "Predictive Models for Diabetic Retinopathy from Non-Image Teleretinal Screening Data". In: *AMIA Summits on Translational Science Proceedings* 2019, p. 472.

Pascual-Fontanilles, Jordi (Mar. 2022). "Dynamic update of Fuzzy Random Forests to improve classification of Diabetic Retinopathy". In: *7th URV Doctoral Workshop In Computer Science And Mathematics*. ISBN: 9788413650333.

Pascual-Fontanilles, Jordi, Lenka Lhotska, et al. (Oct. 2022). "Adapting a Fuzzy Random Forest for Ordinal Multi-Class Classification". In: *Artificial Intelligence Research and Development*. Vol. 356. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 181–190. DOI: 10.3233/FAIA220336.

Pascual-Fontanilles, Jordi, Aida Valls, et al. (Oct. 2021). "Iterative Update of a Random Forest Classifier for Diabetic Retinopathy". In: *Artificial Intelligence Research and Development*. Vol. 339. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 207–216. DOI: 10.3233/FAIA210136.

— (Dec. 2022). "Continuous Dynamic Update of Fuzzy Random Forests". In: *International Journal of Computational Intelligence Systems* 15 (1), pp. 1–16. DOI: 10.1007/S44196-022-00134-0.

Pascual-Fontanilles, Jordi, Aida Valls, et al. (June 2023a). "A fuzzy-based method to boost short time-series to solve class imbalance in health care data". Paper presented at The 20th International Conference on Modeling Decisions for Artificial Intelligence.

Pascual-Fontanilles, Jordi, Aida Valls, et al. (Oct. 2023b). "Challenges in the Exploitation of Historical Clinical Data for the Classification of Diabetic Retinopathy Patients". In: *Artificial Intelligence Research and Development*. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 204–207. DOI: 10.3233/FAIA230683.

Pascual-Fontanilles, Jordi, Aida Valls, et al. (2023c). "Multivariate data binning and examples generation to build a Diabetic Retinopathy classifier based on temporal clinical and analytical risk factors". In: *Artificial Intelligence in Medicine*. Submitted.

Pasos-Ruiz, Alejandro, Michael Flynn, et al. (2021). "The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances". In: *Data Mining and Knowledge Discovery* 35, pp. 401–449. DOI: 10.1007/s10618-020-00727-3.

Pecori, Riccardo, Pietro Ducange, et al. (Aug. 2020). "Incremental learning of fuzzy decision trees for streaming data classification". In: pp. 748–755. ISBN: 9789462527706. DOI: 10.2991/eusflat-19.2019.102.

Rabhi, Sara, Frédéric Blanchard, et al. (Nov. 2022). "Temporal deep learning framework for retinopathy prediction in patients with type 1 diabetes". In:

Artificial Intelligence in Medicine 133, p. 102408. DOI: 10.1016/J.ARTMED.2022.102408.

Roglic, Gojka (2016). "WHO Global report on diabetes: A summary". In: *International Journal of Noncommunicable Diseases* 1 (1), p. 3. DOI: 10.4103/2468-8827.184853.

Romero-Aroca, Pedro, Marc Baget, et al. (Nov. 2023). "Prediction and progression of diabetic retinopathy". In: *INTELIGENCIA ARTIFICIAL Y OFTALMOLOGÍA: ESTADO ACTUAL EN CATALUÑA*. Vol. 31. 4. Òrgan de la Societat Catalana d'Oftalmologia, pp. 256–263. ISBN: 978-84-19264-38-1.

Romero-Aroca, Pedro, Sofia De La Riva-Fernandez, et al. (Oct. 2016). "Changes observed in diabetic retinopathy: eight-year follow-up of a Spanish population". In: *British Journal of Ophthalmology* 100 (10), pp. 1366–1371. DOI: 10.1136/BJOPHTHALMOL-2015-307689.

Romero-Aroca, Pedro, Aida Valls, et al. (2019). "A Clinical Decision Support System for Diabetic Retinopathy Screening: Creating a Clinical Support Application". In: *Telemedicine and e-Health* 25 (1), pp. 31–40. DOI: 10.1089/tmj.2017.0282.

Romero-Aroca, Pedro, Raquel Verges, et al. (Oct. 2023). "Effect of Lipids on Diabetic Retinopathy in a Large Cohort of Diabetic Patients after 10 Years of Follow-Up". In: *Journal of Clinical Medicine 2023, Vol. 12, Page 6674* 12 (20), p. 6674. ISSN: 2077-0383. DOI: 10.3390/JCM12206674.

Romero-Aroca, Pedro, Raquel Verges-Puig, et al. (Aug. 2020). "Validation of a Deep Learning Algorithm for Diabetic Retinopathy". In: *Telemedicine and e-Health* 26.8, pp. 1001–1009. DOI: 10.1089/tmj.2019.0137.

Saffari, Amir, Christian Leistner, et al. (2009). "On-line random forests". In: pp. 1393–1400. ISBN: 9781424444427. DOI: 10.1109/ICCVW.2009.5457447.

Saleh, Emran, Jerzy Błaszczyński, et al. (2018). "Learning ensemble classifiers for diabetic retinopathy assessment". In: *Artificial Intelligence in Medicine* 85, pp. 50–63. DOI: 10.1016/j.artmed.2017.09.006.

Saleh, Emran, Aida Valls, et al. (2016). "Diabetic Retinopathy Risk Estimation Using Fuzzy Rules on Electronic Health Record Data". In: *Modeling Decisions for Artificial Intelligence*. Ed. by Vicenç Torra, Yasuo Narukawa, et al. Springer International Publishing, pp. 263–274. ISBN: 978-3-319-45656-0.

Saleh, Emran, Aida Valls, et al. (Nov. 2019). "A Hierarchically -Decomposable Fuzzy Measure-Based Approach for Fuzzy Rules Aggregation". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 27 (Suppl.1), pp. 59–76. ISSN: 02184885. DOI: 10.1142/S0218488519400038.

Sheikhalishahi, Seyedmostafa, Anirban Bhattacharyya, et al. (Oct. 2023). "An interpretable deep learning model for time-series electronic health records: Case study of delirium prediction in critical care". In: *Artificial Intelligence in Medicine* 144, p. 102659. DOI: 10.1016/J.ARTMED.2023.102659.

Sun, Yunlei and Dalin Zhang (2019). "Diagnosis and Analysis of Diabetic Retinopathy Based on Electronic Health Records". In: *IEEE Access* 7, pp. 86115–86120. DOI: 10.1109/ACCESS.2019.2918625.

Teo, Zhen Ling, Yih Chung Tham, et al. (Nov. 2021). "Global Prevalence of Diabetic Retinopathy and Projection of Burden through 2045: Systematic Review and Meta-analysis". In: *Ophthalmology* 128 (11), pp. 1580–1591. ISSN: 15494713. DOI: 10.1016/j.ophtha.2021.04.027.

Torre, Jordi de la, Aida Valls, et al. (July 2020). "A deep learning interpretable classifier for diabetic retinopathy disease grading". In: *Neurocomputing* 396, pp. 465–476. ISSN: 0925-2312. DOI: 10.1016/J.NEUCOM.2018.07.102.

Torre, Jordi De La, Domenec Puig, et al. (2017). "Weighted kappa loss function for multi-class classification of ordinal data in deep learning". In: *Pattern Recognition Letters* 105, pp. 144–154. DOI: 10.1016/j.patrec.2017.05.018.

Tsao, Hsin Yi, Pei Ying Chan, et al. (Aug. 2018). "Predicting diabetic retinopathy and identifying interpretable biomedical features using machine learning algorithms". In: *BMC Bioinformatics* 19 (9), pp. 111–121. DOI: 10.1186/s12859-018-2277-0.

Utgoff, Paul E., Neil C. Berkman, et al. (1997). "Decision Tree Induction Based on Efficient Tree Restructuring". In: *Machine Learning* 29 (1), pp. 5–44. DOI: 10.1023/A:1007413323501.

Utkin, Lev V., Maxim S. Kovalev, et al. (June 2019). "A deep forest classifier with weights of class probability distribution subsets". In: *Knowledge-Based Systems* 173, pp. 15–27. DOI: 10.1016/j.knosys.2019.02.022.

Valls, Aida, Antonio Moreno, et al. (Nov. 2023). "RETIPROGRAM and MIRA software". In: *INTELIGENCIA ARTIFICIAL Y OFTALMOLOGÍA: ESTADO ACTUAL EN CATALUÑA*. Vol. 31. 4. Òrgan de la Societat Catalana d'Oftalmologia, pp. 206–213. ISBN: 978-84-19264-38-1.

Wang, Will Ke, Ina Chen, et al. (2022). "A Systematic Review of Time Series Classification Techniques Used in Biomedical Applications". In: *Sensors* 22, p. 8016. DOI: 10.3390/S22208016.

Wilkinson, C P, Frederick L Ferris, et al. (2003). "Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales". In: *Ophthalmology* 110 (9), pp. 1677–1682. DOI: 10.1016/S0161-6420(03)00475-5.

Winham, Stacey J., Robert R. Freimuth, et al. (Dec. 2013). "A weighted random forests approach to improve predictive performance". In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 6.6, pp. 496–505. ISSN: 19321864. DOI: 10.1002/sam.11196.

Yager, Ronald R. (1988). "On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking". In: *IEEE Transactions on Systems, Man and Cybernetics* 18 (1), pp. 183–190. DOI: 10.1109/21.87068.

Yang, Chun and Xu Cheng Yin (Oct. 2019). "Diversity-Based Random Forests with Sample Weight Learning". In: *Cognitive Computation* 11.5, pp. 685–696. ISSN: 18669964. DOI: 10.1007/s12559-019-09652-0.

Yuan, Yufei and Michael J Shaw (1995). "Induction of fuzzy decision trees". In: *Fuzzy Sets and Systems* 69, pp. 125–139.

Zadeh, Lotfi A (1992). "Knowledge representation in fuzzy logic". In: *An introduction to fuzzy logic applications in intelligent systems*. Springer, pp. 1–25.

Zhang, Xuchao, Yifeng Gao, et al. (Apr. 2020). "TapNet: Multivariate Time Series Classification with Attentional Prototypical Network". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (04), pp. 6845–6852. DOI: 10.1609/AAAI.V34I04.6165.

Zhao, Bendong, Huanzhang Lu, et al. (Feb. 2017). "Convolutional neural networks for time series classification". In: *Journal of Systems Engineering and Electronics* 28 (1), pp. 162–169. DOI: 10.21629/JSEE.2017.01.18.

Zhao, Pu, Chuan Luo, et al. (2022). "T-SMOTE: Temporal-oriented Synthetic Minority Oversampling Technique for Imbalanced Time Series Classification". In: DOI: 10.1613/jair.953.

Zhao, Yuedong, Xinyu Li, et al. (May 2022). "Using Machine Learning Techniques to Develop Risk Prediction Models for the Risk of Incident Diabetic Retinopathy Among Patients With Type 2 Diabetes Mellitus: A Cohort Study". In: *Frontiers in Endocrinology* 13, p. 885. DOI: 10.3389/FENDO.2022.876559.

Zhong, Yuan, Hongyu Yang, et al. (Aug. 2020). "Online random forests regression with memories". In: *Knowledge-Based Systems* 201-202, p. 106058. ISSN: 09507051. DOI: 10.1016/j.knosys.2020.106058.

Zhu, Min, Jing Xia, et al. (Jan. 2018). "Class weights random forest algorithm for processing class imbalanced medical data". In: *IEEE Access* 6, pp. 4641–4652. ISSN: 21693536. DOI: 10.1109/ACCESS.2018.2789428.

Zhukov, Aleksei V., Denis N. Sidorov, et al. (2017). "Random forest based approach for concept drift handling". In: *Communications in Computer and Information*

Science. Vol. 661. Springer Verlag, pp. 69–77. DOI: 10.1007/978-3-319-52920-2_7.

UNIVERSITAT ROVIRA I VIRGILI
FUZZY-BASED MACHINE LEARNING METHODS FOR CONTINUOUS DIAGNOSIS AND PROGNOSIS OF DIABETIC RETINOPATHY
Jordi Pascual Fontanilles

UNIVERSITAT
ROVIRA i VIRGILI