



**RAQUEL FILIPA BIRRA SIMÃO**

Bachelor of Science in Biomedical Engineering

**UNCERTAINTY-AWARE AI FOR ECG  
ARRHYTHMIA MULTI-LABEL  
CLASSIFICATION**

MASTER IN BIOMEDICAL ENGINEERING

NOVA University Lisbon  
November, 2022



# UNCERTAINTY-AWARE AI FOR ECG ARRHYTHMIA MULTI-LABEL CLASSIFICATION

**RAQUEL FILIPA BIRRA SIMÃO**

Bachelor of Science in Biomedical Engineering

**Adviser:** Prof. Dr. Hugo Filipe Silveira Gamboa

*Associate Professor with Aggregation, NOVA University Lisbon*

## **Examination Committee**

**Chair:** Dra. Susana Isabel dos Santos Silva Sérgio Venceslau

*Assistant Professor, FCT-NOVA*

**Rapporteur:** Dr. Ricardo Nuno Pereira Verga e Afonso Vigário

*Associate Professor with Aggregation, FCT-NOVA*

**Adviser:** Dr. Hugo Filipe Silveira Gamboa

*Associate Professor with Aggregation, FCT-NOVA*

## **Uncertainty-Aware AI for ECG arrhythmia multi-label classification**

Copyright © Raquel Filipa Birra Simão, NOVA School of Science and Technology, NOVA University Lisbon.

The NOVA School of Science and Technology and the NOVA University Lisbon have the right, perpetual and without geographical boundaries, to file and publish this dissertation through printed copies reproduced on paper or on digital form, or by any other means known or that may be invented, and to disseminate through scientific repositories and admit its copying and distribution for non-commercial, educational or research purposes, as long as credit is given to the author and editor.

## ACKNOWLEDGEMENTS

I would like to acknowledge all the people who accompanied me during these 5 years of my academic journey and led to the writing of this dissertation.

Firstly, I would like to thank Professor Hugo Gamboa for believing in me and for the opportunity to embrace a project in data science and machine learning, which is a field I am passionate about.

I would like to thank the *Fraunhofer AICOS* and the Lisbon team for their kind welcome, for all of their assistance and guidance and for making me feel like part of the the group. I want to thank Marília Barandas in particular, for her availability and invaluable help. Without your knowledge and patience, I don't know how I would have got this far. I can not thank you enough.

Adri, Caril, Dani, Henrique, Isa, Luís, Marta, Matilde, Ortiz, Rita, Rui, Cunha, Madri, Meg and Maria : thank you so much for the laughs, the tears, the adventures and for all the friendship. You helped me become who I am today and you will always have a place in my heart.

Thank you, Bibs and Carlota, for always being there for me, for your unconditional support, for all the love and memories and for encouraging me to expand my horizons.

*Por fim, mas não menos importante, obrigada aos meus pais e irmão, pelo amor e apoio incondicional. Obrigada por estarem sempre presentes e por me mostrarem todos os dias que com vontade, coragem e perseverança conseguimos superar todos os nossos obstáculos e realizar grandes feitos. Espero ter-vos deixado orgulhosos tal como vocês me deixam todos os dias.*



## ABSTRACT

**Machine Learning (ML)** models are able to predict a variety of diseases, with performances that can be superior to those achieved by healthcare professionals. However, when implemented in clinical settings as decision support systems, their generalisation capabilities are often compromised, rendering healthcare professionals more susceptible into delivering erroneous diagnostics. This research focuses on uncertainty measures as a key method to abstain from classifying samples with high uncertainty as well as a selection criterion for active learning strategies.

For this purpose, it was employed four large public multi-label **Electrocardiogram (ECG)** databases for the classification of cardiac arrhythmias. Regarding the uncertainty measures, single distribution uncertainty and classical information-theoretic measures of entropy were tested and compared. Thus, three Deep Learning models were developed: a single convolutional neural network and two multiple-models using Monte-Carlo Dropout and Deep Ensemble techniques. When tested with samples from the same database used for training, all models achieved performances higher than 95% for F1-score. However, when tested on an external dataset, their performances dropped to approximately 70%, indicating a probable scenario of dataset shift. The Deep Ensemble model obtained the highest F1-score in both test sets with a maximum difference of 3% from the others. The classification with rejection option increased from a rejection of 10% to a range between 30% to 50% depending on the model or uncertainty measure, with the highest rejection rates being obtained on external data. This reveals that external dataset's classifications have higher uncertainty, also an indication of dataset shift. For the active learning approach, 10% of the highest uncertainty samples were used to retrain the models. The performances results increased by almost 5%, suggesting uncertainty as a good selection method.

Although there are still challenges to the implementation of **ML** models, the preliminary studies show that uncertainty quantification is a valuable method for classification with rejection option and active learning approaches under dataset shift conditions.

**Keywords:** Uncertainty Quantification, Monte Carlo Dropout, Deep Ensemble, Dataset shift, Active Learning

## RESUMO

Modelos de aprendizagem automática conseguem prever um leque de doenças, muitas vezes com desempenhos superiores aos obtidos pelos profissionais de saúde. Contudo, quando integrados em ambientes clínicos como sistemas de apoio à decisão, a generalização destes fica comprometida, o que leva a que profissionais de saúde fiquem mais suscetíveis de fornecer diagnósticos incorretos. Deste modo, este projeto foca-se no papel da incerteza na rejeição de classificações com elevada incerteza e na aprendizagem ativa.

Quatro bases de dados públicas de sinais *ECG multi-label* foram utilizadas na classificação de arritmias cardíacas. Relativamente à quantificação da incerteza, foram testadas e comparadas incertezas provenientes das distribuições e da teoria de informação clássica da entropia. Para tal, foram desenvolvidos três tipos de redes neurais convolucionais: um modelo único e dois modelos obtidos através das técnicas de *Monte-Carlo Dropout* e *Deep Ensemble*. Quando testados com dados da mesma base de dados de treino, os modelos alcançaram desempenhos superiores a 95% de F1-score. No entanto, quando testados com dados externos, os desempenhos desceram para cerca de 70%, revelando a possibilidade de *dataset shift*. O modelo *Deep Ensemble* obteve os melhores resultados em ambos os dados de teste, com uma diferença máxima de 3% em relação aos outros modelos. O threshold de rejeição de 10% em treino aumentou para valores entre 30% a 50%, dependendo do modelo e da medida de incerteza, sendo que as rejeições mais elevadas são obtidas nos dados externos. Isto revela que estes dados têm maior incerteza nas suas classificações, confirmando a presença de *dataset shift*. Para a abordagem de aprendizagem ativa, 10% de dados com elevada incerteza foram utilizados para retreinar os modelos. O desempenho destes aumentou quase 5%, sugerindo a incerteza como um bom critério de seleção.

Apesar de ainda existirem desafios na implementação de modelos de aprendizagem automática, os resultados preliminares revelam que a quantificação da incerteza é um método valioso na classificação com rejeição e na aprendizagem ativa, em condições de *dataset shift*.

**Palavras-chave:** Quantificação da Incerteza, *Monte-Carlo Dropout*, *Deep Ensemble*, *Dataset shift*, Aprendizagem ativa

# CONTENTS

|  |             |
|--|-------------|
| <b>List of Figures</b>                                 | <b>viii</b> |
| <b>List of Tables</b>                                  | <b>x</b>    |
| <b>Abbreviations</b>                                   | <b>xi</b>   |
| <b>1 Introduction</b>                                  | <b>1</b>    |
| 1.1 Context and Motivation . . . . .                   | 1           |
| 1.2 Goals . . . . .                                    | 2           |
| 1.3 Document Structure . . . . .                       | 3           |
| <b>2 Theoretical concepts</b>                          | <b>4</b>    |
| 2.1 Probability . . . . .                              | 4           |
| 2.1.1 Probability Theory . . . . .                     | 4           |
| 2.1.2 Decision Theory . . . . .                        | 5           |
| 2.1.3 Information Theory . . . . .                     | 6           |
| 2.2 Machine Learning . . . . .                         | 7           |
| 2.2.1 Traditional Machine Learning . . . . .           | 8           |
| 2.2.2 Deep Learning . . . . .                          | 8           |
| 2.2.3 Multi-label classification . . . . .             | 11          |
| 2.2.4 Performance Evaluation . . . . .                 | 12          |
| 2.3 Uncertainty in Machine Learning . . . . .          | 16          |
| 2.3.1 Uncertainty Quantification . . . . .             | 16          |
| 2.3.2 Classification with Rejection Option . . . . .   | 18          |
| 2.3.3 Uncertainty problems in medical domain . . . . . | 19          |
| <b>3 Literature Review</b>                             | <b>21</b>   |
| 3.1 Uncertainty Quantification . . . . .               | 21          |
| 3.2 Classification with Rejection Option . . . . .     | 23          |
| 3.3 Dataset Shift . . . . .                            | 24          |

|          |                                      |           |
|----------|--------------------------------------|-----------|
| 3.4      | Active learning                      | 25        |
| 3.5      | The role of uncertainty in ECG data  | 26        |
| <b>4</b> | <b>Electrocardiogram Datasets</b>    | <b>28</b> |
| 4.1      | Electrocardiography                  | 28        |
| 4.2      | Arrhythmia Classification            | 32        |
| 4.3      | Datasets                             | 36        |
| 4.3.1    | CPSC2018 dataset                     | 36        |
| 4.3.2    | PTB dataset                          | 37        |
| 4.3.3    | PTB-XL dataset                       | 37        |
| 4.3.4    | G12EC dataset                        | 37        |
| <b>5</b> | <b>Methodologies</b>                 | <b>38</b> |
| 5.1      | Overview                             | 38        |
| 5.2      | Deep Learning Model                  | 38        |
| 5.2.1    | ECG Data Preparation                 | 38        |
| 5.2.2    | Deep Learning Model Architecture     | 39        |
| 5.2.3    | Training and Testing                 | 41        |
| 5.2.4    | Threshold Optimization               | 42        |
| 5.3      | Uncertainty Quantification           | 43        |
| 5.4      | Classification with rejection option | 44        |
| 5.5      | Uncertainty in Active Learning       | 44        |
| <b>6</b> | <b>Results and Discussion</b>        | <b>46</b> |
| 6.1      | Performance Evaluation               | 46        |
| 6.2      | Classification with Rejection        | 52        |
| 6.3      | Active Learning                      | 59        |
| <b>7</b> | <b>Conclusions and Future Work</b>   | <b>64</b> |
| 7.1      | Conclusions                          | 64        |
| 7.2      | Future Work                          | 66        |
|          | <b>Bibliography</b>                  | <b>67</b> |
|          | <b>Appendices</b>                    |           |
| <b>A</b> | <b>Python Libraries and Modules</b>  | <b>78</b> |
| <b>B</b> | <b>Complementary Results</b>         | <b>80</b> |
|          | <b>Annexes</b>                       |           |
| <b>I</b> | <b>Publications</b>                  | <b>89</b> |

## LIST OF FIGURES

|     |  |    |
|-----|--|----|
| 2.1 | Traditional Machine Learning Components. . . . .   | 8  |
| 2.2 | Artificial single neuron representation. . . . .   | 9  |
| 2.3 | Deep Learning pipeline . . . . .   | 10 |
| 2.4 | An example of a ROC Curve . . . . .  | 15 |
| 2.5 | Aleatoric and knowledge uncertainty example for a classification problem. . . . .  | 17 |
| 2.6 | Active Learning visualization panel . . . . .  | 19 |
|     |  |    |
| 4.1 | Schematic representation of the heart, the ECG signal and action potentials obtained from different regions of the heart. . . . .  | 29 |
| 4.2 | Electrode positions for the Standard Limb Leads, Augmented Limb Leads and Precordial Leads . . . . .   | 31 |
| 4.3 | A standard 12-lead ECG. . . . .  | 32 |
| 4.4 | Representation of Sinus rhythm. . . . .  | 34 |
| 4.5 | Representation of Atrial Fibrillation. . . . .   | 34 |
| 4.6 | Representation of First-degree atrioventricular block. . . . .   | 35 |
| 4.7 | Representation of Left bundle branch block. . . . .  | 35 |
| 4.8 | Representation of Reft bundle branch block. . . . .  | 36 |
|     |  |    |
| 5.1 | Overview of this work methodology . . . . .  | 39 |
| 5.2 | Flow chart diagram of the preprocessing approach . . . . .   | 40 |
| 5.3 | The flowchart of the designed algorithm. . . . .   | 40 |
| 5.4 | Diagram of the treatment employed for an example of outputs obtained from Monte Carlo Dropout and Deep Ensemble method. . . . .  | 42 |
| 5.5 | Diagram of the pipeline use for active learning experiments . . . . .  | 45 |
|     |  |    |
| 6.1 | Micro average F1-score (up) and micro average Area Under a Receiver-Operator Curve (AUC-ROC) (down) metrics results in the Monte-Carlo (MC) Dropout and the Deep Ensemble (DE) models. . . . . | 47 |

|      |   |    |
|------|---|----|
| 6.2  | Micro average F1-score (up) and micro average AUC-ROC (down) metrics for the optimal and standard threshold in the MC Dropout and the Ensemble-1 models. . . . .                | 49 |
| 6.3  | Micro average F1-score (up) and micro average AUC-ROC (down) metrics results for the <b>test-in</b> and <b>test-out</b> sets when tested in the three developed models. . . . . | 51 |
| 6.4  | Uncertainty Quantification for both test sets in the single model. . . . .  | 53 |
| 6.5  | Uncertainty quantification for both test sets in the Ensemble-1 (up) and MC Dropout (down) models. . . . .  | 54 |
| 6.6  | Variation ratio for both test sets in the MC Dropout and Ensemble-1 models. . . . .   | 55 |
| 6.7  | F1-rejection curve for both test sets in the single model. . . . .  | 56 |
| 6.8  | F1-rejection curve for both test sets in the MC model. . . . .  | 57 |
| 6.9  | F1-rejection curve for both test sets in the Ensemble-1 model. . . . .  | 58 |
| 6.10 | F1-rejection curve per class for both test sets in the Ensemble-1 model. . . . .  | 60 |
| 6.11 | Micro average F1-score for the Active learning approach for the highest uncertainties (up) and for random samples (down). . . . .   | 61 |
| B.1  | F1-rejection curve per class for both test sets in the Monte Carlo Dropout model. . . . .   | 81 |
| B.2  | F1-rejection curve per class for both test sets in the single model. . . . .  | 82 |
| B.3  | AUC-ROC-rejection curve for both test sets in the single model. . . . .   | 83 |
| B.4  | AUC-ROC-rejection curve for both test sets in the Monte Carlo Dropout model. . . . .  | 84 |
| B.5  | AUC-ROC-rejection curve for both test sets in the Ensemble-1 model. . . . .   | 85 |
| B.6  | AUC-ROC-rejection curve per class for both test sets in the single model. . . . .   | 86 |
| B.7  | AUC-ROC-rejection curve per class for both test sets in the Monte Carlo Dropout model. . . . .  | 87 |
| B.8  | AUC-ROC-rejection curve per class for both test sets in the Ensemble-1 model. . . . .   | 88 |

## LIST OF TABLES

|     |  |    |
|-----|--|----|
| 4.1 | Location of Standard 12-Lead ECG electrodes. . . . .   | 33 |
| 4.2 | Classification classes with their abbreviation, Systematized Nomenclature of Medicine clinical term (SNOMED-CT) code, count and percentage in the four datasets. . . . . | 34 |
| 5.1 | Feature interpretation in the developed Convolutional neural network (CNN) model. Inspired by [112] . . . . .  | 41 |
| 6.1 | Precision, Recall, F1-Score and AUC-ROC metrics for the single models trained with the ECG leads aVR, V1 and V6. . . . .   | 48 |
| 6.2 | Micro average Precision and Micro average Recall of the single model tested on <b>test-in</b> set using various thresholds. . . . .                                      | 48 |
| 6.3 | Micro average F1-score per class for both test sets tested in the three developed model. . . . .   | 52 |
| 6.4 | Micro average F1-score per class for each step of the active learning approach in the single and Ensemble-1 models. . . . .  | 62 |
| A.1 | Python libraries employed in this dissertation . . . . .   | 78 |
| A.2 | Relevant modules used in this project. . . . .   | 79 |
| B.1 | AUC-ROC per class for both test sets tested in the three developed model. . . . .  | 80 |

## ABBREVIATIONS

|                 |  |
|-----------------|--|
| <b>AF</b>       | Atrial fibrillation ( <i>pp.</i> 34, 36, 37)   |
| <b>AI</b>       | Artificial Intelligence ( <i>pp.</i> 1, 4, 7)  |
| <b>ANN</b>      | Artificial Neural Network ( <i>pp.</i> 9, 10)  |
| <b>ARC</b>      | Accuracy-Rejection Curve ( <i>pp.</i> 18, 19, 23, 44)  |
| <b>AU</b>       | Aleatoric Uncertainty ( <i>pp.</i> 16, 18, 22, 23, 25, 26, 43, 52, 53, 57)   |
| <b>AUC-ROC</b>  | Area Under a Receiver-Operator Curve ( <i>pp.</i> <i>viii</i> – <i>x</i> , 2, 15, 24, 25, 27, 46–51, 59, 61, 64)               |
| <b>BNN</b>      | Bayesian Neural Network ( <i>pp.</i> 21, 22, 66)   |
| <b>BR</b>       | Binary Relevance ( <i>p.</i> 12)   |
| <b>CNN</b>      | Convolutional neural network ( <i>pp.</i> <i>x</i> , 2, 10, 11, 22, 24, 25, 27, 39, 41, 43, 64, 66)                            |
| <b>CPSC2018</b> | The China Physiological Signal Challenge 2018 ( <i>pp.</i> 27, 36, 41)   |
| <b>CQ</b>       | Classification quality ( <i>p.</i> 19)   |
| <b>DE</b>       | Deep Ensemble ( <i>pp.</i> <i>viii</i> , 2, 17, 22, 24, 26, 27, 41–43, 45–47, 64, 66)  |
| <b>DL</b>       | Deep Learning ( <i>pp.</i> 2, 9–11, 17–19, 22, 38, 55, 62, 64, 66)   |
| <b>DNN</b>      | Deep Neural Network ( <i>pp.</i> 10, 17, 21, 22, 24–27)  |
| <b>ECG</b>      | Electrocardiogram ( <i>pp.</i> <i>iv</i> , <i>v</i> , <i>viii</i> , <i>x</i> , 2, 3, 21, 26–30, 32, 33, 35–39, 44, 46, 48, 66) |
| <b>EU</b>       | Epistemic Uncertainty ( <i>pp.</i> 16, 18, 22, 23, 25–27, 43, 52, 53, 57, 59, 63)  |
| <b>FN</b>       | False Negative ( <i>pp.</i> 13, 14)  |
| <b>FP</b>       | False Positive ( <i>pp.</i> 13, 14)  |
| <b>FPR</b>      | False Positive Rate ( <i>pp.</i> 15, 42, 43)   |
| <b>G12EC</b>    | Georgia 12-lead ECG Challenge ( <i>pp.</i> 37, 41)   |
| <b>IAVB</b>     | First-degree atrioventricular block ( <i>pp.</i> 35–37)  |



|                  |   |
|------------------|---|
| <b>KU</b>        | Knowledge Uncertainty (pp. 16, 24)  |
| <b>KUE</b>       | Knowledge Uncertainty Estimation (pp. 43, 55, 65)   |
| <b>LBBB</b>      | Left bundle branch block (pp. 35–37)  |
| <b>LP</b>        | Label Power Set (p. 12)   |
| <b>MC</b>        | Monte-Carlo (pp. viii, ix, 2, 18, 21–27, 38, 41–43, 46, 47, 49, 50, 52–55, 57, 59, 64–66) |
| <b>ML</b>        | Machine Learning (pp. iv, 1, 2, 4, 7–9, 12, 16, 17, 19, 21, 23, 25, 26, 64–66)            |
| <b>MRI</b>       | Magnetic resonance imaging (p. 22)  |
| <b>NRA</b>       | Nonrejected accuracy (pp. 19, 23)   |
| <b>NSR</b>       | Sinus rhythm (pp. 32, 36, 37, 50)   |
| <b>PPV</b>       | Positive Predictive Value (p. 14)   |
| <b>PR</b>        | Precision-Recall (pp. 15, 16, 42, 43, 49)   |
| <b>PReLU</b>     | Parametric Rectified Linear Units (pp. 9, 10)   |
| <b>PTB</b>       | Physikalisch Technische Bundesanstalt (pp. 37, 41)  |
| <b>RBBB</b>      | Right bundle branch block (pp. 36, 37)  |
| <b>ROC</b>       | Receiver Operator Characteristic (pp. 15, 42, 43)   |
| <b>RQ</b>        | Rejection quality (p. 19)   |
| <b>SL</b>        | Supervised Learning (p. 7)  |
| <b>SNOMED-CT</b> | Systematized Nomenclature of Medicine clinical term (pp. x, 33, 34)                       |
| <b>TN</b>        | True Negative (p. 13)   |
| <b>TNR</b>       | True Negative Rate (p. 13)  |
| <b>TP</b>        | True Positive (pp. 13, 14)  |
| <b>TPR</b>       | True Positive Rate (pp. 13, 15, 42, 43)   |
| <b>UQ</b>        | Uncertainty Quantification (pp. 2, 3, 21, 22, 27, 38, 53, 64–66)                          |
| <b>VR</b>        | Variation Ratio (pp. 43, 53, 65)  |

# INTRODUCTION

This chapter presents the context and motivation of this dissertation, its main goals and the overall structure of this document.

## 1.1 Context and Motivation

Over the years, medical technology has been developed and improved in order to ensure the most effective healthcare to the general public. [Artificial Intelligence \(AI\)](#) is quickly evolving due to its potential to assist evidence-based clinical decision-making and achieve value-based care [2]. As a result, there has been a growing amount of scientific research regarding the use of [ML](#) algorithms in the medical domain. This is achievable because [ML](#) models are trained with patient data in order to identify patterns that would otherwise be undetected and, thereby, produce an estimate of a patient's current or future clinical state. [ML](#) models have progressed to the point that they can predict a variety of diseases, with performances that can be superior to those achieved by healthcare professionals.

However, these models, while showing promising results, still have some limitations for their deployment on a clinical setting. When they are implemented in the real world as decision support systems, their generalization capabilities are often compromised, resulting in lower performances and render healthcare professionals more susceptible into delivering erroneous diagnostics. As a result, it is critical that [ML](#) models include safety mechanisms to mitigate these situations and improve the trustworthiness of these models.

Quantifying the uncertainty of the models in their classifications is a method that has been explored in order to assess the model's confidence in their decisions. The development of uncertainty aware models will provide healthcare professionals with access to the model's confidence in its predictions but also refrain the model from delivering classifications with high uncertainty. Furthermore, since samples with high uncertainty have different characteristics and distributions than the ones learned by the model, these data can be used to retrain the [ML](#) model and improve its generalisation.

Therefore, the main motivation behind this dissertation is to explore the potential

of **Uncertainty Quantification (UQ)** on **ML** models, in particular **Deep Learning (DL)** models, in providing a more careful and safer application, as well as help to increase trust among healthcare professionals when using these models as decision support systems. This research will use as supporting application the classification of cardiac arrhythmias using **ECG** data, as this sort of data can vary significantly among patients and these heart disorders are relatively common diagnoses.

## 1.2 Goals

This dissertation aims to explore how to build and evaluate a **DL** model through a probabilistic approach. The primary focus will be on developing a classification approach with rejection option based on uncertainty measures and evaluate the uncertainty as a selection method for active learning. Although the main purpose is to develop an agnostic framework for the classification of cardiac arrhythmias, this work will concentrate on establishing the practical value of **UQ** applied in three types of **DL** models in different medical datasets and their role in the referred methods. This research aims at providing a better understanding of the capacity of the model's generalization through uncertainty estimation as well as demonstrate that uncertainty aware models are capable of containing safety mechanisms and, therefore, be considered trustworthy clinical decision support systems.

In summary, this dissertation will be divided into five main sequential goals:

1. Development of different types of **DL** models for the selected datasets;  
Three types of **CNN** models will be developed: a single model, a model obtained through **MC** Dropout and another through **DE**.
2. Distinction and quantification of the different sources of uncertainty on the developed algorithms;  
Shannon entropy and maximum probability will be calculated for the single model while the **DE** and **MC** Dropout models have their uncertainty separated into epistemic and aleatoric, with these two uncertainties and their combination (total uncertainty) being estimated.
3. Classification with rejection option using **UQ** measures;  
This method will be applied to the 3 models using all the calculated uncertainties. The performance of the models will be evaluated according to the rejection curve employing the F1-Score and the **AUC-ROC**. In addition, a possible optimal rejection threshold will be investigated.
4. Use of **UQ** as a sample selection criteria for active learning purposes;  
The samples with the highest level of uncertainty are chosen to retrain the models and determine whether the model's performance improves.

#### 5. Identification of dataset shift through UQ;

The difference in uncertainty behaviour between two test sets, one with data from the same training database and the other with data from an external database, will be evaluated through the employed approaches and compared.

### **1.3 Document Structure**

This document is organised into 7 chapters. The current chapter introduces the motivation behind the developed project and its main goals. The second Chapter presents the theoretical concepts required for a proper comprehension of the research conducted. The Chapter three provides a literature review on the topics of uncertainty estimation and related approaches, as well as their application in ECG data. The fourth Chapter describes the datasets and labels selected for this work in addition to the fundamental background associated with the ECG data. The methodologies employed in this work and the analyses of the obtained results are discussed in Chapters five and six, respectively. Lastly, the seventh chapter summarizes the main conclusions of the developed research as well as its limitations and recommendations for future work.

## THEORETICAL CONCEPTS

### 2.1 Probability

Probability is the field of mathematics that studies random phenomena and analyses the chance of a certain event occurring. If it is very likely to happen a certain event, the probability is considered high. If the possibility is low, so is the probability.

However, there are two distinct interpretations of probability. One is known as the Frequentist interpretation, in which probabilities describe the long-term frequency of events that can occur several times [3]. The other is called the Bayesian interpretation, where probability quantifies the degree of belief or uncertainty regarding an occurrence. As a result, it is intrinsically related to information rather than repetitive experiments [4, 5]. Bayesian interpretation is relevant in the uncertainty field since it can be used to represent the uncertainty of one-off events that do not have long term frequencies.

It should be emphasized that the rules of probability theory are the same regardless of the interpretation adopted [3].

Due to the evolution of computer technologies, the Bayesian approach has become of real practical use in machine learning, as many algorithms rely on probabilistic data [5, 6].

#### 2.1.1 Probability Theory

Probability theory provides a framework for manipulation and quantification of uncertainty [7]. In AI applications, the ML algorithms compute or approximate various expressions employing probability theory and the probability is used to theoretically analyse the behavior of proposed ML systems [8].

Probability theory is defined by three properties called **The Axioms of Probability** [5, 9]. Thus, for a sample space  $\Omega$  and the probability  $P$  of a certain event  $A$  with an event space  $F$ :

- $\Omega$  is a set of all the outcomes of a random experiment, where each outcome  $\omega \in \Omega$  can be seen as a complete description of the state of the real world on the experiment.

- A set of  $A \in F$  are subsets of  $\Omega$  (i.e.  $A \subseteq \Omega$  is a collection of possible outcomes of an experiment)
- Probability measure: A function  $P : F \rightarrow \mathbb{R}$  that satisfies the following properties :
  - $P(A) \geq 0, A \in F$ ;
  - $P(\Omega) = 1$ ;
  - If  $A_1, A_2, \dots$  are disjoint events (i.e.  $A_i \cap A_j = \emptyset$  whenever  $i \neq j$ ), then

$$P(\cup_i A_i) = \sum_i P(A_i) ;$$

One of the most important tools in probability theory is conditional probability. It is represented as  $P(Y|X)$  and measures the probability of an event  $Y$  happen when the event  $X$  is known. It is defined as:

$$P(Y|X) = \frac{P(Y, X)}{P(X)} \quad (2.1)$$

where  $P(Y, X)$  is the joint probability of  $X$  and  $Y$ . Note that this is not defined when  $P(X)$  is 0. Also, If the events  $X$  and  $Y$  are independent, that is, if the occurrence of one does not influence the probability of the other,  $P(X|Y)$  can be reduced to:

$$P(Y|X) = P(Y) \quad (2.2)$$

and

$$P(Y, X) = P(Y)P(X) \quad (2.3)$$

In the Bayesian approach, the conditional probability is known as Bayes rule:

$$P(H|D) = \frac{P(D|H)P(H)}{P(D)} \quad (2.4)$$

where  $D$  is the data and  $H$  the hypothesis. The term  $P(D|H)$  is called the *likelihood* and represents the probability of the observed data from the hypothesis. The term  $P(H)$  is known as the *prior*, as it reflects one's prior knowledge before the data. Furthermore, the term  $P(H|D)$  is named the *posterior* and is the probability of the hypothesis after consideration of the data [10].

The *evidence*,  $P(D)$ , is regarded irrelevant as it is considered a normalizing constant. By eliminating this scale factor, we reduce the Bayes theorem into:

$$posterior \propto likelihood \times prior \quad (2.5)$$

### 2.1.2 Decision Theory

Decision theory combined with probability theory enables the best decisions to be made in conditions of uncertainty, such as observed in pattern recognition [11]. In decision theory, the decision maker has a set of available actions,  $A$ , to select from. Each option has benefits and costs that are dependent on the  $\theta \in \Theta$  that parameterises the model. This

information can be translated into a loss function  $\ell(\theta, a)$ , that indicates the loss associated with  $a \in A$  [3, 12].

The posterior expected loss, i.e. the risk, given the known data  $\mathbf{x}$ , can be estimated by:

$$R(a|\mathbf{x}) = \sum_{\theta \in \Theta} \ell(\theta, a)p(\theta|\mathbf{x}) \quad (2.6)$$

The optimal policy, also known as Bayes estimator, determines what action to take for each potential observation so as to minimize the risk [3]:

$$\pi^*(\mathbf{x}) = \underset{a \in a}{\operatorname{argmin}}[R(a|\mathbf{x})] \quad (2.7)$$

Furthermore, the utility function  $U(\theta, a)$  estimates the desirability of each possible action in each possible state [3]. The lost function and the utility function are equivalent if we consider the utility to be merely the negative of the loss,  $U(\theta, a) = -\ell(\theta, a)$ . Thus, the optimal policy can be given by the maximum expected utility principle:

$$\pi^*(\mathbf{x}) = \underset{a \in a}{\operatorname{argmax}}[U(\theta, a)p(\theta|\mathbf{x})] \quad (2.8)$$

### 2.1.3 Information Theory

Information theory assesses a number of measures related to the transmission and processing of uncertainty and information, within the framework of probability theory [13].

If  $x$  is a random variable that can take the values  $x_1, x_2, \dots, x_N$  and  $p(x_k)$  is the probability that  $x$  takes on the value  $x_k$ , the self-information  $i(x_k)$  of the event  $x$  takes on  $x_k$  is defined by [14]:

$$i(x_k) = \log_2 \frac{1}{p(x_k)} = -\log_2 p(x_k) \quad (2.9)$$

Indubitably, if the probability of an event is high, there is very little information associated with its occurrence, while the occurrence of an event of low probability has more information associated with it [14]. The measure of information, therefore, depends on the probability distribution  $p(x_k)$ . The expected value of the self-information is known as Shannon entropy. The entropy  $H$  of a random variable  $x_k$  is given by [15]:

$$H(X) = - \sum_{x \in X} p(x_k) \log_2 p(x_k) \quad (2.10)$$

Furthermore, we can define a conditional entropy of  $X$  given  $W$  represented as  $H(X|W)$  [16]:

$$H(X|Y) = - \sum_{x,y} p(x, y) \log_2 p(x, y) \quad (2.11)$$

This metric can be described as the uncertainty that exists regarding the value  $X$  when  $W$  is known.

The difference between the uncertainty of  $X$ ,  $H(X)$ , and  $H(X|W)$  is the information regarding  $X$  contained in  $W$ . This value is known as the Mutual Information and is represented by  $I(X, W)$  [14, 16]:

$$I(X, W) = H(X) - H(X|W). \quad (2.12)$$

Some of the properties of Mutual Information are that it is always non-negative, is symmetric in  $X$  and  $W$  and is zero if  $X$  and  $W$  are independent [17].

## 2.2 Machine Learning

**ML** is a field of **AI** concerned with the development of computational methods capable of learning through known information and improving their own performance in order to execute a certain task.

A well-known definition of **ML** is given by Tom Mitchell that defined it as a computer program that can "learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ " [18].

These algorithms are capable of resolving a variety of learning problems through the aforementioned learning process. **ML** can be sub-divided into three main categories:

- **Supervised Learning**

**Supervised Learning (SL)** is the form of **ML** most widely used in practice [19]. Predictive models are learned from a large training dataset where each training sample corresponds to an event. A training example is a vector of inputs, mostly known as features, that describes the event and an output, i.e. a label attributing the class to which the training example belongs [20]. Classification problems are suitable to be solved by **SL** models, as they require the deduction of a mapping from the features to their respective labels [19].

- **Unsupervised Learning**

In Unsupervised Classification, the training examples are unlabeled and the objective of the model is to subdivide the dataset into clusters of similar examples or learn the entire probability distribution of the data. Self-**SL**, dimensionality reduction and clustering are some of the regularly used unsupervised learning methods [21].

- **Reinforcement Learning**

In Reinforcement learning, the agent (i.e. the model) learns through trial-and-error interactions with a dynamic environment [22]. On each step of interaction, it is received an input and the information about the current state of the environment. For every action chosen, the state of the environment changes and the value of this state transition is communicated through a scalar reinforcement signal, typically



between 0 and 1. The decision is the behavior that has the maximum value of the reinforcement signal.

### 2.2.1 Traditional Machine Learning

The development of a traditional ML model usually involves a set of core components, as represented in Figure 2.1.

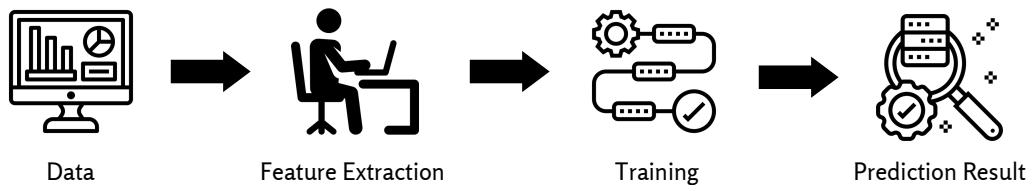


Figure 2.1: Traditional Machine Learning Components.

The initial step in traditional ML is data collection and data preparation. Data used as input in ML models is, in general, unstructured and incorporates a lot of noise. Therefore, data preprocessing is commonly applied. The next step is the feature extraction process that consists of transforming the data available in a way that highlights the relevant information contained in the dataset [23]. It results in a high number of features, some of which are not important to the learning process. Thus, a subset of features that are both relevant and nonredundant are selected, mitigating the negative impact of unnecessary features on the performance of the model.

In order to achieve the best results, it is important to select the best ML algorithm taking into account the data characteristics. The majority of ML algorithms need initial intervention by the user in order to choose the right values of various parameters for the given dataset [24]. Once all of the conditions are met, the model is trained using a portion of the chosen dataset as training data. Finally, once the model has learned from the training dataset, it can be used to classify test data.

Commonly used traditional ML models are Naive Bayes, Support Vector Machine, K-Nearest Neighbour, Decision Tree and Random Forest.

### 2.2.2 Deep Learning

Many traditional ML models have a simple two-layer architecture with the form of input  $x \rightarrow$  output  $y$ . Nevertheless, when we take into account the role of the brain in learning and decision-making, it can be observed that the brain has several levels of information processing. It is believed that in each level occurs the learning of features or representations

at increasing levels of abstraction [19]. This revelation prompted a new field in ML, named DL, which aims to reproduce this type of architecture..

**Artificial Neural Network (ANN)** is at the very core of DL. The network is versatile, powerful, scalable and can extract high level features automatically, making it ideal to tackle large and highly complex DL tasks, such as classifying billions of data [25]. The behaviour of a single neuron is the key component for studying the non-linear characteristics of models such as the multi-layer neural networks [26]. An artificial single neuron is represented in Figure 2.2.

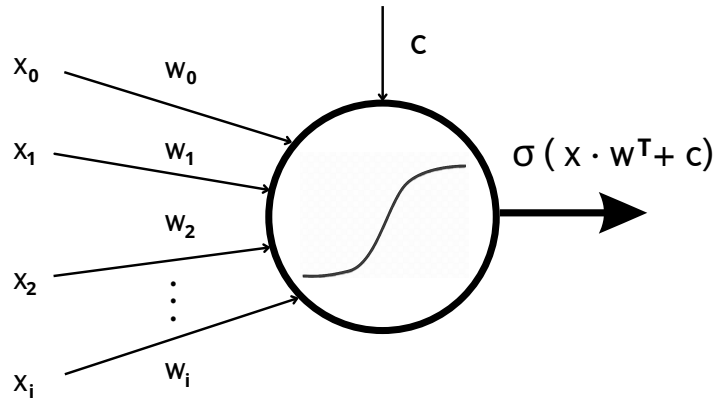


Figure 2.2: Artificial single neuron representation. Adapted from [27].

The input channels, represented by  $x_i$ , are the components that contain the data. The transformations applied to its input are parameterized by each weight,  $w_i$ , usually selected randomly [28]. The neuron sums the various weighted input signals and is passed through an activation function as soon as a threshold, in this case, the bias value  $c$ , is exceeded. The activation function adds non-linear and monotonic factors to the network and improves the model fitting [29].

The activation function of the neuron represented in Figure 2.2 is the sigmoid function that is defined by Equation 2.13.

$$f(x) = \frac{1}{1 + e^{-ax}} \quad (2.13)$$

This function converts the values to a range between 0 and 1 and its continuity allows the calculation of the derivative [27]. Although there are several activation functions, for this work development, besides sigmoid activation function, a **Parametric Rectified Linear Units (PReLU)** activation function was also used. PReLU is expressed by Equation 2.14.

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{if } x < 0 \end{cases} \quad (2.14)$$

In this function, if the sample in the neuron has a value greater than zero, the output is linearly related to the sample's value. When the value in the neuron is a fixed non-zero

lower number, the output is proportional to a learning parameter  $a$ , that is learned from the data. It was verified that the **PReLU** function has lower error rates than most activation functions and helps to overcome overfitting [30]. Overfitting is a problem that occurs when the model performs so well on training data that it is unable to generalize to new data [28].

**Deep Neural Network (DNN)** are more complex forms of **ANN**. These networks receive the inputs and map them via a sequence of layered transformations which are learned by exposure to the samples used to train the model [28]. This type of model consists of one input layer, one or more hidden layers and one final layer designed as the output layer. A representation of a common **DL** model pipeline is presented in Figure 2.3.

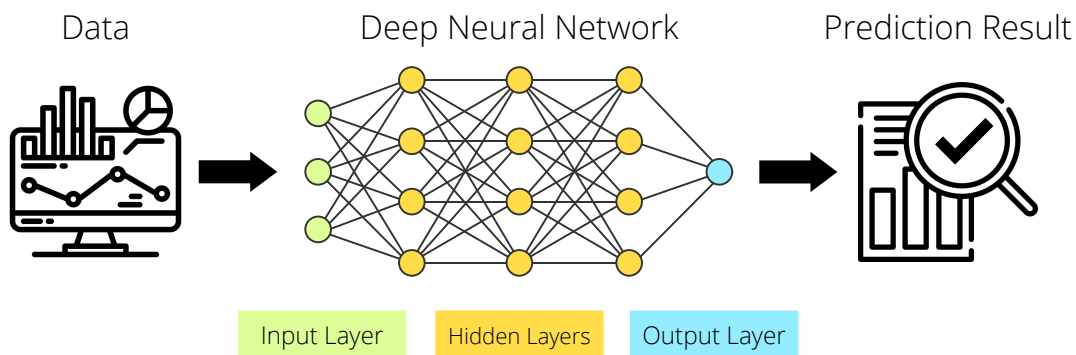


Figure 2.3: Deep Learning pipeline. In the Deep Neural Network, the green layer is the input layer, the yellow layers are the hidden layers and the blue layer is the output layer.

The input layer receives raw information and no operations are performed at this layer. The nodes simply relay information to the hidden layers. It is in the hidden layers that the low and high-level features necessary for classification are extracted, with higher-level learned features defined in terms of lower-level features [31, 32]. This final layer brings the information learned through the hidden layer and delivers the final value as a result.

Furthermore, learning can be defined as the process of finding the values of the network's layers in such a way that the input examples can be accurately assigned to their associated labels [28].

In order to evaluate the learning of the algorithm, it is necessary to quantify how far the calculated outputs of the network are from the true values. This is obtained by using a loss function. This result is then used as a feedback signal to adjust the weights and biases of the network, reducing the loss score for the current samples [28]. This process is known as back-propagation.

A network has learnt when the output values are as near as possible to the target values i.e. the loss function is at its minimum.

There are a wide range of **DL** architectures. The fully connected neural and **CNN** are two of the most used and almost all other **DL** neural networks stem from these.

### 2.2.2.1 Fully connected networks

A fully connected network, also known as dense network, is a combination of several simple neurons, where they are all connected with the preceding and subsequent layers. The layers transmit the transformations made by succeeding simple mathematical combinations and, subsequently, formulate complex non-linear calculus that ends in an activation function, as aforementioned. These networks have two main purposes: interconnecting layers with different dimensions and developing relations between the extracted features and the outputs [27].

### 2.2.2.2 Convolutional Neural Network

The CNN is a DL architecture which applies convolution operations between the kernels and a tensor. The kernels can be considered filters that detect features within local regions of the input data (called local receptive field), mapping it to a feature map [28]. The size of the receptive field is the same as the size of the filter.

The kernels are composed of weights that change as the network learns during the training process. In the forward propagation step, the kernels are activated by an activation function, as explained in Section 2.2.2. All the layers in a CNN are called convolutional layers and their output is the stacking of feature maps. After convolution, a pooling layer can be applied to decrease the dimensions of the feature maps and minimize the computational effort [27].

The final outputs are further flattened and submitted to one or more dense layers until the class with the highest probability is selected as the predicted class. The CNN can also include optional layers like batch normalization to improve the training time and Dropout layers to reduce the overfitting.

## 2.2.3 Multi-label classification

In single-label classification, the algorithm is learning from a set of examples that are categorized with a single label  $l$  from a set of disjoint labels  $L$ ,  $|L| > 1$ . The learning problem is referred to as a binary classification if  $|L| = 2$ , while if  $|L| > 2$ , then it is called a multi-class classification problem. In multi-label classification, the data is categorized with a set of labels  $Y \subseteq L$ .

The main difference between traditional and multi-label classification is in the label format. Where a traditional classifier returns only one value, a multi-label model produces a vector of output values. There are two main approaches in multi-label learning: data transformation and method adaptation. The first is based on transformation techniques that, when applied, are able to produce one or more binary or multi-class datasets [33]. The method adaptation focuses on adapting existing classification algorithms, so they are capable of dealing with multi-label data.

Two of the most popular transformations are the [Binary Relevance \(BR\)](#) and the [Label Power Set \(LP\)](#). The [BR](#) method converts a multi-label sample into several single-label samples [34]. After the multi-label data has been transformed, a set of binary classifiers are constructed for each class using the respective training dataset. In this technique each class is independent from each other, neglecting the relationship between different classes, which may have negative effects [35].

[LP](#) transforms the multi-label problem into one single-label multi-class classification, where each distinct labelset assignment is treated as a class, resulting in  $2^q$  transformed labels. Although this technique models labels together and achieve better predictive performances, many label sets only occur once or very rarely, resulting in an unbalanced problem that is difficult to learn from [36].

Furthermore, the Method Adaptation Approach focuses on classification models such as kNN classifiers, classification trees and neural networks that have been used to tackle both binary and multi-class classifications.

Multi-label classification can be supported directly by neural networks simply by selecting the number of target labels as the number of nodes in the output layer. As a result, the model will have one output node per class to address the multi-labelled data [37]. By applying a sigmoid function as activation function to each output node, we transform the algorithm into a binary classification for each class. The output probabilities will not sum 1 and the predictive possibility of each class is independent [35].

#### 2.2.4 Performance Evaluation

In [ML](#), datasets are usually divided into training, validation and testing set. Once the model has learned on a training dataset, it is expected to have a good performance on unseen data.

Validation is crucial since it allows us to identify if the classifiers suffer from underfitting or overfitting, both of which contribute to poor performance. Underfitting occurs when a model is incapable to understand the variability of the data, i.e. the model is too simple to describe the given set of data [38]. Overfitting, as explained previously, is associated with the model's inability to generalise to different data, This is one of the most prevalent problems in [ML](#) models.

Thus, before the system can be implemented, a validation technique must be selected to evaluate how much the model learned. The three most commonly used are:

- **k-fold Cross-Validation**

The dataset is distributed in  $k$  folds and, in each iteration, the classifier uses one fold for evaluating the model and the remaining  $k-1$  folds for training.

- **Leave one out**

For the total number of  $n$  samples on the dataset, in each iteration, a single sample is used for evaluating the model and  $n-1$  samples are used for training. This method is a specific case of k-fold Cross-Validation and is used for small datasets.

- **Bootstrapping**

Multiple bootstrap training sets with  $n$  samples are produced with uniform resampling. The performance of the model is calculated on the out-of-sample examples. Usually, resampling with replacement includes around 60% of the original samples in each bootstrap dataset, with the remaining used as out-of-sample test sets [39].

There are several evaluation metrics from which to choose when evaluating the performance and results of the model with the testing set. The commonly used are: Accuracy, Sensitivity, Specificity, Precision and F1-score [40]. The model's performance can also be evaluated using the Confusion Matrix [41].

The metrics presented below take into account when the sample is classified as the evaluated class correctly (**True Positive (TP)**), when the sample is classified as not being part of the evaluated class correctly (**True Negative (TN)**), when the sample is classified as the evaluated class incorrectly (**False Positive (FP)**) and when the sample is classified as not being part of the evaluated class incorrectly (**False Negative (FN)**).

#### 2.2.4.1 Accuracy

The model's accuracy is expressed as a percentage of the properly identified samples in all classes. It's represented by the total number of samples successfully classified divided by the total number of samples. For a binary classifier, it is defined by:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \quad (2.15)$$

#### 2.2.4.2 Sensitivity

Sensitivity, also known as recall, is the **True Positive Rate (TPR)**, i.e. it is the probability of identify correctly a positive sample. It is calculated by the number of correctly identified positive samples divided by the total number of positive samples.

$$TPR = \frac{TP}{TP + FN}. \quad (2.16)$$

#### 2.2.4.3 Specificity

Specificity, also known as the **True Negative Rate (TNR)**, is the probability of identify correctly a negative sample. It is represented by the number of correctly identified negative samples divided by the total number of negative samples. It is given by:

$$TNR = \frac{TN}{TN + FP}. \quad (2.17)$$

#### 2.2.4.4 Precision

Precision, also known as the **Positive Predictive Value (PPV)**, is defined by the percentage of **TP** samples among the samples that were classified as positive.

$$PPV = \frac{TP}{TP + FP}. \quad (2.18)$$

#### 2.2.4.5 F1-Score

The F1-score may be interpreted as the harmonic mean of the precision and recall. The metric's greatest score is 1 and its poorest is 0. Precision and recall both contribute equally to the F1-score. This metric is calculated as follows:

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}. \quad (2.19)$$

#### 2.2.4.6 Aggregate Metrics

Aggregate metrics such as macro, micro and weighted average provide a high-level view of how the model is performing. This is highly valuable to metrics such as precision, recall and F1-score. For imbalance datasets, F1-score is one of the most used metrics for performance evaluation, since it is a combination of both precision and recall. In the following points, macro, micro and weighted average F1-score are detailed.

- **Macro average F1-score**

This metric computes the arithmetic mean (unweighted mean) of the F1-score of all classes. It does not take label imbalance into account. In a multi-label classification, the macro-average for F1-score is as follow:

$$\text{macro avg F1-score} = \frac{\sum_{i=1}^N \text{F1-score}(l_i)}{N} \quad (2.20)$$

where  $l_i$  is a single label  $l$  from a set of disjoint labels  $L$  and  $N$  the number of labels.

- **Micro average F1-score**

Micro averaging computes a global average F1-score by counting the sums of the True Positives, False Negatives and False Positives. First, it is summed the respective **TP**, **FP**, and **FN** values across all classes and then it is calculated the F1-score.

$$\text{micro avg F1-score} = \frac{\sum_{i=1}^N TP}{\sum_{i=1}^N TP + \frac{1}{2} \sum_{i=1}^N (FP + FN)} \quad (2.21)$$

- **Weighted average F1-score**

The weighted-averaged F1-score is calculated by taking the weighted mean of the labels F1-score while considering each class's occurrences ( $support(l_i)$ ) in the dataset.

$$\text{Weighted avg F1-score} = \frac{\sum_{i=1}^N \text{F1-score}(l_i) * \text{support}(l_i)}{\sum_{i=1}^N \text{support}(l_i)} \quad (2.22)$$

### 2.2.4.7 Receiver Operator Characteristic Curve

**Receiver Operator Characteristic (ROC)** curves is a graphical representation of **TPR** as a function of **False Positive Rate (FPR)** (equals to  $1 - \text{TPR}$ ) of the test samples. This curve shows how the number of correctly classified positive examples varies with the number of incorrectly classified negative examples, illustrating the conditional probabilities of belonging to a particular predicted class given the true classification in a two-class classification [42]. However, **ROC** curves can present an overly optimistic view of an algorithm's performance if there is a large skew in the class distribution [43]. The **AUC-ROC** can also be calculated, and is a widely used metric to evaluate the performance of a model.

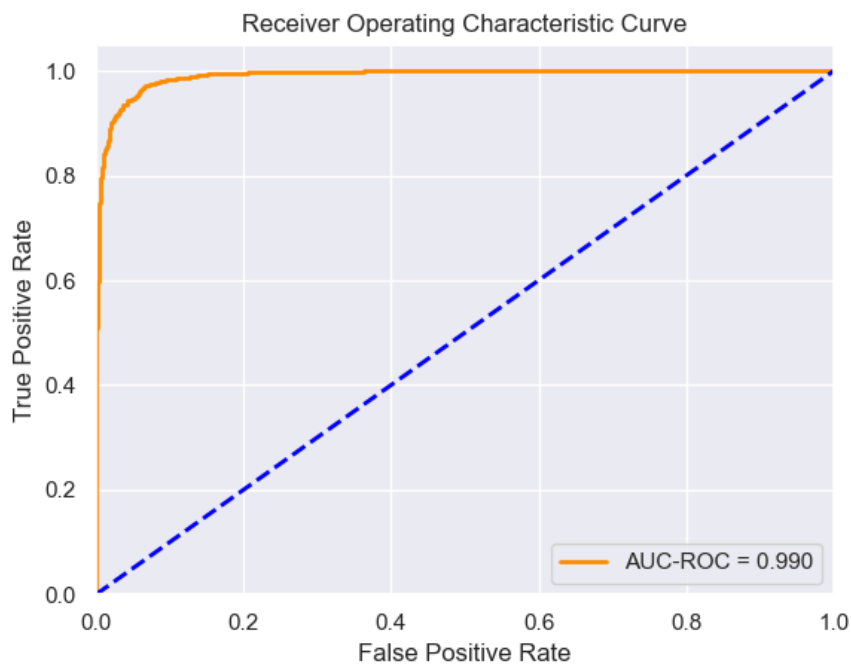


Figure 2.4: An example of a ROC Curve

### 2.2.4.8 Precision-Recall Curve

**Precision-Recall (PR)** curve is an evaluation tool for binary classification that allows the visualization of the trade-off between precision and recall for different thresholds. **PR** curves are increasingly used to evaluate performances, particularly for imbalanced datasets where one class is observed more frequently than the other [44]. A big area under the curve implies a low false positive rate, associated with a high precision, and a low false



negative rate which is related with a high recall. Besides the visual inspection of a **PR** curve, the Area under a **PR** Curve is used as a general measure of performance regardless of any specific threshold or operating point [44].

## 2.3 Uncertainty in Machine Learning

Uncertainty is ubiquitous and happens in every step of a **ML** pipeline. Thus, uncertainty managing in **ML** should be considered a key component of any **ML** system. In the general literature [45, 46], a distinction between two intrinsically different sources of uncertainty is done: aleatoric and epistemic.

**Aleatoric Uncertainty (AU)** is associated with the variability in the outcome of an experiment which is due to intrinsic randomness of the data generating process that cannot be explained away given more observations or data samples [45]. This uncertainty cannot be reduced even if more data is provided. Aleatoric uncertainty can be subdivided into two types: Homoscedastic and Heteroscedastic. In the first, uncertainty is assumed to be constant for all the inputs. Heteroscedastic uncertainty is relevant when modeling assumptions include variable noise on the input space [47].

**Epistemic Uncertainty (EU)** refers to the lack of knowledge of the model. This uncertainty can be decreased by increasing the training data, better modeling or better data analysis [46]. The **EU** can be further divided into model uncertainty and **Knowledge Uncertainty (KU)** [48]. Model uncertainty addresses the uncertainty in the adequacy and the parameters of the model. **KU** is caused by incomplete domain coverage, since unknown regions of the data space will always be presented. Furthermore, the presence of new classes that were not contemplated in the training of the model, constitutes an example of high **KU**.

An illustration of **AU** and **KU** in a classification problem can be seen in Figure 2.5. The **KU** is present in test samples located in regions without training samples and **AU** occurs in the overlapping region of the classes.

### 2.3.1 Uncertainty Quantification

In traditional probabilistic modeling and Bayesian inference, the uncertainty of a prediction is given by the posterior distribution [46]. As aforementioned, the posterior distribution can be obtained via the Bayes rule, mentioned in Section 2.1.1. The belief about the prediction  $y_i$  for an instance  $x$  is represented by the probability distribution of probability distributions [17]. Thus, a prediction is computed through averaging the predictions provided by different hypotheses  $h$  for a certain dataset  $D$ . The predicted posterior distribution is presented as follows:

$$p(y|x) = \int p(y|x, h)dP(h|D) \quad (2.23)$$

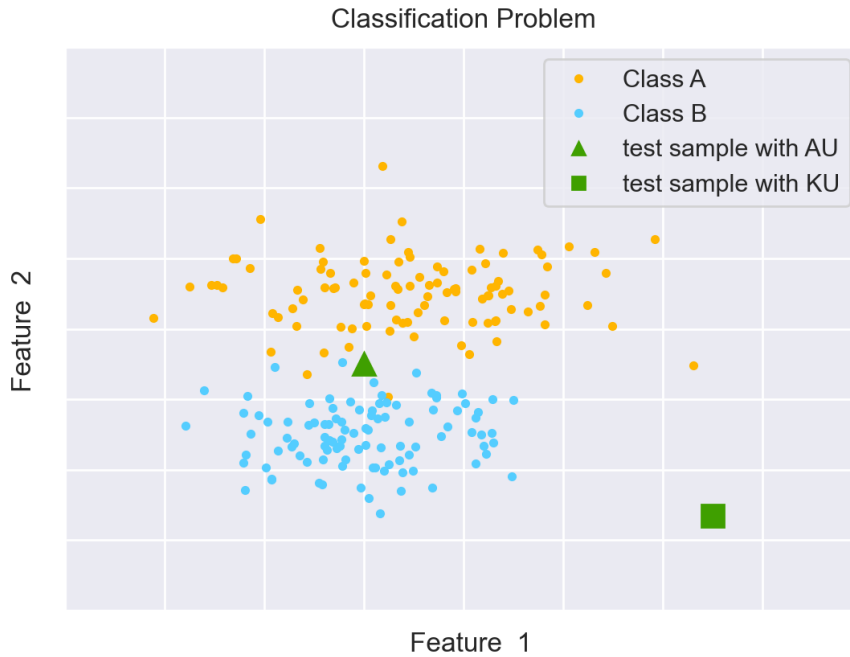


Figure 2.5: Aleatoric and knowledge uncertainty example for a classification problem. The blue and orange points indicate the train samples, the green triangle corresponds to aleatoric uncertainty and the green square corresponds to the knowledge uncertainty.

Since model averaging is computationally costly in **ML**, predictions are calculated considering a single probability distribution for each class. Therefore, the uncertainty of a single probability distribution can be calculated through the maximum probability of the predicted class that can be obtained by:

$$p(\hat{y}|x) = \max_k p(y_i|x, D) \quad (2.24)$$

, where  $y_i \in \{y_1, \dots, y_k\}$  consists of a finite set of  $k$  class labels in dataset  $D$ . This uncertainty measure combines both aleatoric and epistemic uncertainty.

Shannon’s entropy, described in Section 2.1.3, is the most well-known measure of uncertainty of a single probability distribution. This uncertainty measure captures the shape of the distribution and is mostly associated with the aleatoric part of the uncertainty [46].

In the context of **DL**, the randomness induced during training and inference can be used to obtain an uncertainty estimation [49]. Ensembling and Dropout are techniques commonly used for this quantification. Ensembling consists of training repeatedly different models or the same model with different parameters. When applied to the same **DNN**, the ensembling is called **DE**. In this method, due to the randomness in the initialization and training process, it is provided different samples of trained network parameters [50, 51]. Each model makes its own prediction independently of the other models in the ensemble. The final prediction is then derived from the composition of all models in the ensemble.

Dropout, in **DL**, is a method that omits a certain percentage of neurons at each layer of a neural network during training, with the missed neurons randomly selected for each iteration [52]. In **MC Dropout** the neural network is trained with Dropout at training time, and at test time the output is evaluated by dropping units randomly to generate samples from the predictive distribution [53]. Similarly to ensemble, the final prediction is obtained from the composition of all the predictions with distinct dropouts.

The total uncertainty can be computed as the entropy of the predictive posterior distribution  $H[p(y|x)]$  and the aleatoric uncertainty is measured in terms of the expectation of entropy with regard to the posterior probability  $E_{p(h|D)}H[p(y|x, h)]$  [54, 55].

Given the computational complexity of these measures, an approximation based on combinations of  $M$  hypotheses can be achieved. This approximation is shown as follow:

$$u_{total}(x) = H[E_{p(h,D)}p(y|x, h)] \approx H\left[\frac{1}{M} \sum_{i=1}^M P(y|x, h_i)\right] \quad (2.25)$$

$$u_{aleat}(x) = E_{p(h,D)}H[p(y|x, h)] \approx \frac{1}{M} \sum_{i=1}^M H[p(y|x, h_i)] \quad (2.26)$$

The **AU** can be measured since, by fixing a hypothesis  $h$ , the **EU** is essentially removed. Thus, the **EU** is measured in terms of the mutual information between hypotheses and outcomes,  $I(y, h|x, D)$  [46]. The expression of this uncertainty and its approximation is given as follow:

$$u_{epist}(x) = I(y, h|x, D) = H[E_{p(h,D)}p(y|x, h)] - E_{p(h,D)}H[p(y|x, h)] \quad (2.27)$$

### 2.3.2 Classification with Rejection Option

When a classifier is not sufficiently confident in the prediction, the model can abstain from producing an answer or discard a prediction if the uncertainty is sufficiently high. Therefore, a classifier with rejection can cope with unknown information, reducing the threat caused by the existence of unknown samples or mislabeled training samples that can compromise the performance of the model.

Frequently, rejected samples are divided into two distinct classes: confusion rejection and distance rejection [56, 57]. The first concerns the samples belonging to known classes and have associated aleatoric uncertainty. The distance rejection concerns samples that belong to unknown classes thus having high epistemic uncertainty. In classification with rejection that can distinguish between aleatoric and epistemic uncertainty, a confidence threshold value needs to be defined indicating the rejection point [58]. The evaluation of the performance of classifiers with rejection option typically uses standard metrics, such as accuracy, to obtain an **Accuracy-Rejection Curve (ARC)**. The **ARC** represents the accuracy of a classifier against its rejection rate, varying from 0 to 1 [59]. This curve plot has an accuracy of 100% for a rejection rate of 100%, i.e. corresponds to the point (1, 1).

It starts at a point  $(0, a)$ , where  $a$  is the accuracy percentage of the classifier when none of the events are rejected. The **ARC** is generated by adjusting the rejection threshold and, therefore, rejecting progressively the samples with the highest uncertainty values.

Some examples of performance measures for classification with rejection are **Nonrejected accuracy (NRA)**, **Rejection quality (RQ)** and **Classification quality (CQ)** [60]. The one applied in this work was the **NRA** that measures the ability of the classifier to accurately classify nonrejected samples.

Assuming  $A$  is a subset of accurately classified samples and  $N$  is a subset of nonrejected samples, this metric can be determined as:

$$NRA = \frac{|A \cap N|}{|N|} \quad (2.28)$$

### 2.3.3 Uncertainty problems in medical domain

**ML** models, mainly **DL** models, demand a vast labelled dataset to learn properly. The number of labelled data required grows with the complexity of the problem or the complexity of the input data.

This issue is particularly dominant in the medical field. For example, in order to automate the analysis of a given medical exam, it would be necessary an expert to annotate a large number of exams, labelling them to indicate if the patient has certain condition or not. However, obtaining the amount of the needed labelled data is time-consuming and expensive.

One possible solution to this problem is active learning. In this approach, the model chooses what unlabelled data is appropriate for training, and request an external “oracle”, for example a medical work, for the label of the selected data [61].

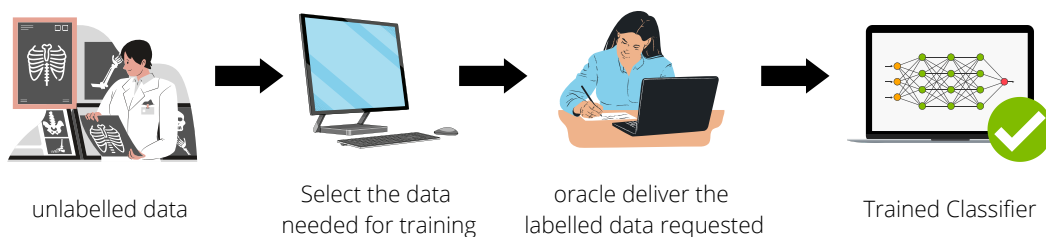


Figure 2.6: Active Learning visualization panel

The choice of the data to be labelled is selected by an acquisition function, which ranks points based on their potential informativeness [62]. There are a variety of acquisition functions and many of them rely on model uncertainty to evaluate the potential informativeness of the unlabelled data points. The more informative is the selected data, the fewer

labelled training examples are necessary to achieve a greater classifier accuracy. Therefore, the quantification of uncertainty plays a central role in active learning.

Another problem impacting the medical field is the shift of the dataset. In the real world, the conditions in which we use the medical systems diverge from the conditions in which these systems were created. Environments are nonstationary thus it is impractical and too expensive trying to match the development scenario to a particular environment [63]. This leads to mismatches between the training data and the data intended to be classified.

In general, the greater the degree of shift, the poorer is the model's performance [64, 65]. As a result, it's becoming increasingly vital to improve the model's robustness to distribution shifts and its estimates of predictive uncertainty [64], so that these distribution shifts can be detected and the model can be adjusted accordingly.

## LITERATURE REVIEW

Uncertainty is a key factor in the evaluation of the robustness of **ML** classifiers, particularly when applied in risk-sensitive domains like Healthcare. Nevertheless, most **ML** models either fail to measure uncertainty or require additional methods to do so. Furthermore, understanding and separating uncertainty makes it easier to comprehend and adapt the **ML** models to increase their reliability.

This chapter reviews relevant work regarding uncertainty in **ML** models. It is organized into 5 sections that comprise Uncertainty Quantification, Classification with Rejection Option, the problem of dataset shift, Active Learning and the role of uncertainty in **ECC** data.

### 3.1 Uncertainty Quantification

**DNN** have shown impressive state-of-the-art accuracy performances but poor uncertainty estimations. As a result, they are predisposed to produce overconfident predictions that, when incorrect, can be harmful. Therefore, it is essential to address **UQ** in real-world scenarios.

Recently, different approaches have been developed to address **UQ**. In several published papers [66, 67, 68], **Bayesian Neural Network (BNN)** have been employed to quantify uncertainty, where the authors demonstrated that aleatoric and epistemic uncertainties can be estimated by setting a distribution across the model's weight and model's output. These works showed that **UQ** plays a major role in improving models' performance and, consequently, their safety. However, due to the substantial modifications on the training procedure, **BNN** approaches are computationally more demanding and conceptually more complicated than non-Bayesian neural networks [69].

**MC Dropout** is a different technique to estimate uncertainty. The study conducted by Srivastava et al. [70] revealed that Dropout produces state of art results on a variety of benchmark datasets. It also showed that Dropout neural networks, in comparison with **BNN**, are much faster to train and operate at test time. Gal and Ghahramani established a new theoretical framework that incorporates a probabilistic interpretation of Dropout in

deep neural networks [62]. The results showed that this approach improves the problem of representing aleatoric and epistemic uncertainty without losing computational complexity and test accuracy.

Lakshminarayanan et al. [71] trained a combination of probabilistic neural network with DE and demonstrated that their method produced well-calibrated uncertainty as good as or better than a single BNN. They concluded that ensembles improve robustness because each model provides a functionally different explanation of the data. Malinin is known to adopt a Bayesian viewpoint on ensembles in DNN to estimate the total uncertainty of the model using the entropy of the predictive posterior [55]. The formulation of mutual information allows the total uncertainty to be decomposed into EU and AU. His recent work includes the distillation of an ensemble into a single model. Such approach obtained equivalent accuracy while reducing the computational costs.

CNN's have rarely provide uncertainty estimations [72], although having a state-of-the-art performance and being one of the most used DL models in the medical field. Wang et al. [72] analysed epistemic and aleatoric uncertainty using CNN in 2D and 3D fetal brain segmentation from Magnetic resonance imaging (MRI) slices and volumes respectively. The aleatoric uncertainty was estimated based on the entropy of the predicted posterior distribution of their formulated test-time augmentation. The entropy of the predictions calculated by MC Dropout was used to obtain the epistemic uncertainty. It was also demonstrated that a DE of networks can be used as an alternative for uncertainty estimation. Stoean et al. [73] presented a hybrid CNN to obtain information regarding the model uncertainty for Electrooculogram time-series data. MC Dropout was used to estimate the predicted uncertainty and was not only able to capture the thin delineation between the two similar classes but also discriminate between them.

Caldeira et al. [51], in their research, compared methods for UQ in DL algorithms such as BNN, Concrete Dropout and DE. Concrete Dropout usually offers better-calibrated uncertainty estimates than MC Dropout since it allows automatic tuning of Dropout rates using a principled optimization objective. For all the methods, fully-connected networks were trained and both predicted the same relative uncertainty regardless of the noise within the examples. The AU was well modelled but when the test set contained samples far from the training distribution, it was found that the methods failed to sufficiently increase the uncertainties associated with the predictions. This problem was particularly evident for Concrete Dropout. Overall, this study concluded that DE delivered the best results in all the performed tests.

Although several studies present promising results in the quantification of uncertainties, there are still several limitations in the separation of aleatoric and epistemic uncertainty, which remains an open research field. Furthermore, it would be valuable to expand these studies to different neural networks and larger datasets, to better reflect the UQ methods and model's potential.

## 3.2 Classification with Rejection Option

The standard approach for classification with rejection option, also known as Chow's theory [74], is the calculation of a rejection threshold that minimises the classification risk. This requires the estimation of the posterior probabilities as well as the employment of a cost function that quantifies the cost of both the misclassification and the rejection. However, using the class posterior probability ignores the possibility of having objects from unknown classes, hence typically rejecting samples with high AU and disregarding samples with high EU.

Mena et al. [75] proposed a black-box model with a rejection technique in order to increase the classifier's accuracy. By combining a Dirichlet output distribution with MC Dropout, it was possible to model the uncertainty while observing the black-box restrictions. The results showed that, in 34 different datasets, by measuring the sampling uncertainty and using it for rejection, the accuracy in all problems was improved 4% to 8% by rejecting only 10% of the samples. However, because it is not possible to neither alter the definition of a black-box model nor access to the internals, this work could not estimate and reject based on EU, only focusing on the AU.

Shaker and Hüllermeier [76] derived meaningful measures of aleatoric and epistemic uncertainty with an ensemble-based approach and analysed the corresponding measures in a classification with a reject option. The evaluation was made by producing a ARC. The experiments were realised in various well-known datasets from the UCI repository [77] and showed that the rejection performs well in general. The total uncertainty performed better than epistemic and aleatoric uncertainty, proving the benefit of combining both types of uncertainties compared to using either one of them.

The majority of the work in the field of rejection option use single thresholds and mostly with binary classification only. Pillai and Fumera, in [78], developed a specific framework that has precision and recall as rejection measures to attain a desired trade-off between classifier accuracy on non-rejected decisions. For different ML models, the use of a rejection option revealed an increase of the performance measures for increasing rejection rates, which was the desired behaviour.

Recently, Barandas et al. [46] studied uncertainty-based rejection for different ML tasks taking into account aleatoric, knowledge and model uncertainties. It was applied a rejection rule for each type of uncertainty, using rejection performance measures to define the confidence threshold. Using a Human Activity Recognition dataset, it was demonstrated that NRA was always higher than the baseline accuracy for all training sizes and classifiers being analysed. This indicates that the model uncertainty measure detected the regions in the feature space responsible for a high number of misclassifications and, by rejecting those samples, the accuracy of the model improved.

To summarise, the field of uncertainty in classifications with rejection option is mainly focused on rejection based on the difficulty in class distinction with no proper uncertainty estimation explicitly done. Therefore, the rejection taking into account the separation in



epistemic and aleatoric uncertainty needs to be further developed and expanded to more complex data, such as multi-class and multi-label data.

### 3.3 Dataset Shift

**DNN** have achieved state-of-art performance, allowing the models to fit to data with high precision. However, these networks make generalisation to unseen data a challenge. In general, the greater the degree of distributional shift of the data, the poorer is the model's performance.

Ovadia et al. [79] suggested that there is not a comprehensive evaluation of uncertainty estimation from different methods, such as **MC Dropout** and **DE**, under dataset shift. Thus, their work focuses on the effect of dataset shift on accuracy and calibration on **DNN**. For a diverse classification of benchmarks datasets, the results showed that the models with best accuracy and calibration do not usually translate to better results under dataset shift. The investigation also disclosed that the quality of uncertainty consistently deteriorates with increasing dataset shift regardless of method, i.e, the networks give wrong predictions with high confidence on out-of-distribution data. Furthermore, it was concluded **DE** seem to perform best and be more robust to dataset shift.

In the work of Malinin et al. [64], it was also proposed the evaluation of dataset shift through uncertainty estimation in order to evaluate the robustness of neural networks to distributional shift. Thus, a comparison was conducted of different uncertainty measures, obtained in an ensemble-based approach, such as Area Under the F1 curve and **AUC-ROC**. The dataset consists of data taken from large-scale industrial sources where distributional shift is present. According to the results, for out-of-distribution detection, measures that detect **KU** perform best, as suggested by the high **AUC-ROC** values. The measure of total uncertainty performs best for detecting misclassification errors since it is associated with a high Area Under the F1 curve.

In [80], Stacke et al. evaluated the generalisation performance to unseen data and presented a metric, called representation shift, to measure the statistical difference between source and target domains on histopathology data. The experiments, done using two datasets and training **CNN** for tumour classification, showed that a small difference in image characteristics, such as intensity augmentation, can result in completely separated distributions in the representation domain. Furthermore, the results showed that the out-of-distribution data can be measured with the representation shift metric, as classifications with a high decrease in accuracy have high values of representation shift.

Also in the medical imaging field, Pooch et al. [81] analysed the extent of domain shift on four of the largest datasets of chest radiographs. To that end, it was used a **CNN** for multi-label classification at each of the four datasets and was evaluated their performance using the **AUC-ROC**. The results revealed that each dataset has the highest average **AUC-ROC** when tested with its own test set, but when tested in other datasets, it

displays a significant performance drop, as expected given the diversity of distributions among images based on the equipment that produces them.

Considering the current state of the field, ML models are not prepared to classify data with a wide range of possible distributional shifts, resulting in lower performances and wrong predictions with high confidence. It is therefore increasingly important to evaluate and improve uncertainty estimation, as well as develop strategies to increase both model's robustness to generalisation and distribution shift.

### 3.4 Active learning

Most DNN require large amounts of well-annotated training data to achieve state-of-the-art performance. However, obtaining labelled data can be a time-consuming and difficult procedure, specially in the medical field where annotations depend on the availability of a qualified expert whose time is expensive and scarce. Thus, only the most relevant samples should be delivered to the expert.

Gal et al. [82] introduced an active learning framework combined with a Bayesian CNN, achieving significant improvements in the field, notably on high dimensional data such as skin cancer diagnosis images. It was performed an approximate inference using MC Dropout in order to select the data to be labelled based on the uncertainty of the predictions. The training procedure, which began with 20 data samples, was repeated 100 times, each time acquiring 10 samples to the training set. The framework's performance was measured using AUC-ROC and the results indicated that, despite the longer run time associated, the strategy reduces the number of expert labels needed and the costs associated with such a system. Furthermore, it was studied the importance of model uncertainty in active learning by comparing the used model to a deterministic CNN. The Bayesian CNN achieved better performance results, demonstrating that the uncertainty measured throughout the Bayesian model represents more accurately the prediction confidence. This occurs since a deterministic model can only capture AU but not EU.

A few years later, Sadafi et al. [83] developed an active learning framework that extract the most relevant samples from a large set of unannotated data for expert annotation. A DNN was trained on images of red blood cells with seven subtypes and the confidence score was computed using Dropout variational inference. The selection of the data to be labelled takes into account the uncertainty of the detection of a single cell and the rarity of the classes presented in the image. This strategy was evaluated by comparing its performance with a baseline method in which the expert was asked to annotate randomly selected images. The results revealed that the performance increases by 5% as it was added 1000 newly annotated cells using active learning while with the same number of randomly annotated cells is around only 2%. Considering the prioritisation of rare classes in the data, the approach has a performance ranging from 15% to 50% for the same number of newly added, while the performance is unchanged in the random approach.

Nguyen et al. [84] investigated the usefulness of distinguishing different sources of uncertainty and to compare their performance in active learning. Their experiments were conducted on a binary classification using datasets from the UCI Repository [77] and, in each iteration, the data was evaluated and the samples with the highest degrees of uncertainty were selected. The procedure was repeated 1000 times and the results showed that the framework using EU outperformed the same framework using AU. This behaviour was expected since EU provides more useful information for the expert, whereas AU is unlikely to do so. Furthermore, it was suggested the potential of EU to serve as a stopping criterion for an active learning process. If the EU is low for the samples remaining in the pool set, this implies that additional sampling will bring little to no new information to the model. According to Nguyen et al., the difficulty with this strategy was setting an acceptable size of the training data set or a targeted performance level. The targeted performance level can be implemented by defining an uncertainty threshold and stopping the active learning process when the threshold is reached. This allows improving the time and cost of the training of the ML model.

Several active learning strategies have been proposed in the machine learning literature. However, active learning approaches that capture and distinguish the different types of uncertainty, particularly in multi-class and multi-label classification, are highly challenging and very sparse in the existing literature, leaving an open research field for future work.

### 3.5 The role of uncertainty in ECG data

There is a wide field of research in cardiology with ECG analysis, in which ML has become one of the most useful tools in a variety of medical problems, including the diagnosis of arrhythmias. Several studies [85, 86, 87, 88] demonstrated that multi-label ECG classification is effective, with the highest values of F1-score for each class above 80%. As for single label arrhythmia classification on DNNs, the accuracies range from 94% to 99% [89, 90, 91]. Despite these results, few of these models are ready to be implemented in clinical practice since limited attention has been devoted to whether such results can be trusted. Thus, quantifying predictions' uncertainty is critical to develop trust among healthcare workers and may even be more important than improving model's accuracy.

In [92], Vranken et al. studied uncertainty estimation methods applied to three 12-lead ECG datasets. The uncertainties were calculated using DE and the MC Dropout method. The results demonstrated that the combined uncertainty obtained better results overall. When using only one type of uncertainty, the results showed the largest dataset had a better performance when AU was applied, unlike the smaller datasets, which performed better when dealing with EU. The experience also highlighted that the regular DNN was 30% either over or underconfident, emphasising the need to incorporate uncertainty estimation in classification. Additionally, a rejection threshold was implemented to samples of each dataset test set. The accuracy of all models increased when the samples with the lowest

confidence were removed, showing that the samples with highest EU were rejected first, as expected.

Aseeri [93] introduced an uncertainty-aware DNN for cardiac arrhythmia classification using three benchmark medical datasets. The model's uncertainty was computed using MC Dropout and the uncertainty was evaluated using F1-score and AUC-ROC. The results revealed that the proposed framework outperforms existing approaches in multiclass classification. The average AUC-ROC's was 98.91% and the average of the macro F1-score was 98.10%. It was also applied DE method as a stress test to assess the model's performance and, when compared with the MC Dropout, the results showed the performances were similar, indicating the effectiveness of the proposed model. Aseer suggested, for future work, to combine DE with Dropout in order to obtain a strong method of rejecting wrong predictions with high uncertainty.

Recently, Zhang et al. [94] performed experiments on the multi-label 12-lead The China Physiological Signal Challenge 2018 (CPSC2018) dataset to classify ECG's with rejection based on uncertainty. Although the dataset contains multi-label samples, in order to facilitate the classification, it was only used the first labels. The aleatoric and epistemic uncertainties were computed using MC Dropout in a Bayesian CNN and then combined for each classification prediction. The results showed that the average F1-score of the nine classes was 66.35%. Furthermore, it was tested the predictions of the model with rejection under different thresholds, showing that the performance has better results when it is applied rejection to samples with the highest uncertainty. The highest F1-score was 86,88% with a threshold of 0.40, which is 21% higher than the F1-score without rejection. The study concluded the samples with less uncertainty are more likely to be classified correctly and the rejection can improve the model's performance, as was expected.

There has been relatively little research on the field of uncertainty in ECG classification. And even those works have several limitations. It is critical to try to correct the unmet needs of health professionals, such as addressing the presence of unknown medical conditions. A possible strategy to address these problems is the UQ and the rejection of samples with high uncertainty. A possible solution to handle inputs from unseen patients, rare diseases or difficult data to diagnose could be the use of active learning applied to rejected samples, avoiding delivering incorrect predictions to health workers. Furthermore, several diseases are frequently present within the same ECG. Therefore, it is important to investigate UQ for multi-label classification networks.

To the best of my knowledge, there are a few works that address UQ in a multi-label ECG classification. In particular, the only found was from Xie et al. [95] that presented a ECG classification framework applied to 5 datasets of cardiac arrhythmias and the uncertainty was only used to assess the robustness of the proposed method. As for the use of uncertainty for active learning in ECG data, no works were found.

## ELECTROCARDIOGRAM DATASETS

This Chapter introduces the theoretical concepts associated with the functioning of the heart and the procedures necessary to collect ECG recordings. Furthermore, the types of arrhythmia chosen as classes in the proposed model will be addressed, as well as a brief description of the databases used.

### 4.1 Electrocardiography

The mechanical cyclic behaviour of the heart, which pumps blood from the atria and the ventricles, is interrelated with an electrical stimulation, as can be seen in Figure 4.1.

The cardiac muscle has a resting potential, which is defined as the potential difference between the interior and the exterior of the cell, which is approximately  $-90$  mV [98]. Each heartbeat starts when the sinoatrial node produces the sinus rhythm action potential that travels through the right and left atria, reaching a potential of  $+20$  mV [98]. As a result, both atria contract almost simultaneously. Once the action potential reaches the atrioventricular node, there is a delay of almost 100ms that enables the blood to move into the ventricles before the impulse is transmitted to the bundle branches [99]. The impulse travels across the right and left bundle branches as well as to the Purkinje fibres, generating a contraction on both ventricles and pumping the rest of the blood to the respective arteries. The impulse fades after ventricular contraction, and the ventricles repolarise in preparation for the next heartbeat.

The different repolarisations and depolarisations produced in the different areas of the heart generate distinctive waves which when added together result in the characteristic ECG signal. The waves and the intervals between them are identified as follow [97]:

- **P Wave**

This wave represents the depolarisation of the atria and has a frequency interval between 5 and 30Hz. Atrial repolarisation is masked by the QRS complex and, thus, does not have a characteristic wave. It is during this phase that the blood travels from the atria to the ventricles.

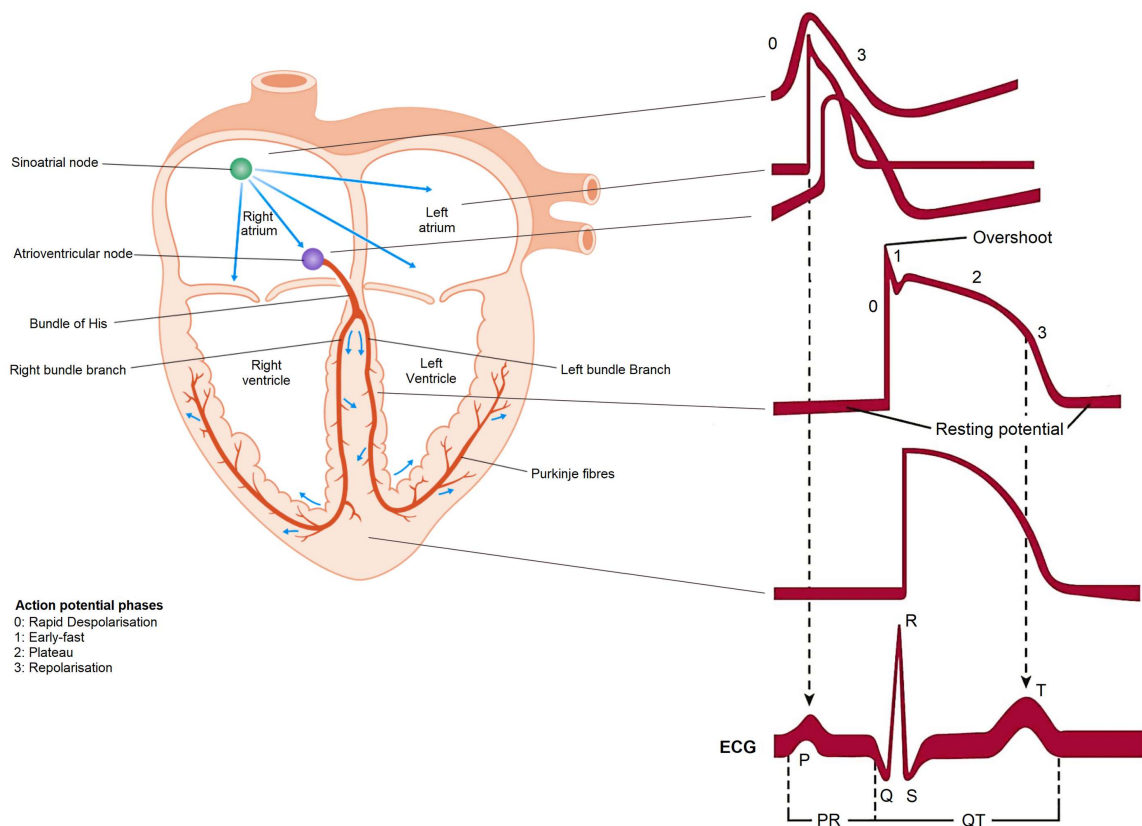


Figure 4.1: Schematic representation of the heart, the ECG signal and action potentials obtained from different regions of the heart. Adapted from [96, 97].

- **PR interval**

The PR interval is the period between the formation of the P wave and before the QRS complex. It includes the PR segment, which represents the conduction of the signal through the atrioventricular bundle.

- **QRS complex**

The QRS complex is composed of the waves Q, R, and S and, when combined, correspond to the depolarisation of the ventricles. The frequencies associated with this depolarisation are between 8 and 50Hz. The QRS complex has a duration similar to the P wave since the conduction in the ventricular conducting system is considerably faster than in the atrial system. It is during this phase that the ventricles are filled and inject blood to the respective arteries.

- **T wave**

The T wave represents the ventricular repolarisation and marks the beginning of ventricular relaxation, with frequencies between 0 and 10 Hz.

- **QT interval**

The QRS complex, ST segment and T wave are included in the QT interval. This interval represents the duration of the ventricular depolarisation and repolarisation. The ST segment is an isoelectric region that corresponds to the period in which the contraction of the ventricles is maintained to expel the blood from the heart.

Although it is not represented in Figure 4.1, there is a small deflection after the T wave, known as the U wave, that is not always visible. Its source is still undetermined although it is thought to be associated with the repolarization of the Purkinje fibres.

The ECG signal can be detected using electrodes that sense the potentials generated by the cardiac electrical behaviour [100]. These electrodes are placed in specific positions on the body and their measurement is influenced by the properties of the dermal and epidermal layers of the skin, the electrolytic gel that is applied to the skin and the contact between the electrode and the skin [101]. The electrical potentials obtained are then converted into leads, and each lead views the heart from a different perspective.

These leads are bipolar since are derived from the potential difference between two electrodes, one designed positive input and the other negative input [101]. The actual potential in each electrode is not known, just the difference between them. In most leads, the negative output is a combination of two or three electrodes electrically connected together. This arrangement of electrodes can also be referred to as a reference electrode.

There are three types of leads: Standard Limb Leads, Augmented Limb Leads and Precordial Leads. The placement of the electrodes and its respective leads can be seen in Figure 4.2

### Standard Limb Leads

The standard limb leads, also known as Willem Einthoven's original leads, are constituted by Lead I, II and III. As shown in Figure 4.2, Lead I results from the difference between the potential of the electrode on the left arm with the potential of the electrode on the right arm. Similarly, Lead II displays the potential difference between the left leg and right arm while Lead III shows the potential difference between the left leg and left arm. These leads represent the heart with an angle of 0°, 60° and 120° respectively.

The spatial organisation of these leads forms a triangle, known as Einthoven's triangle. In it, the potential in lead II equals the sum of potentials sensed in leads I and III, as shown by the Equation 4.1:

$$\text{Lead II} = \text{Lead I} + \text{Lead III} \quad (4.1)$$

Although not depicted in Figure 4.2, a right leg electrode can be used as an electrical reference, minimizing artifacts [101].



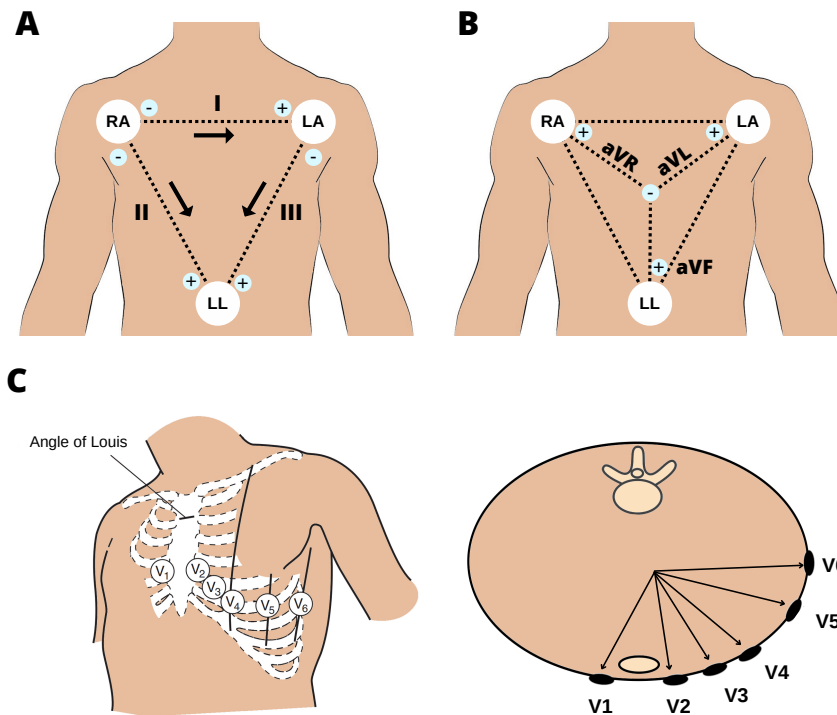


Figure 4.2: **A.** Electrode positions for the Standard Limb Leads I, II and III. RA, LA, and LL represent the locations of the electrodes on the right arm, left arm and left leg respectively. **B.** Electrode positions for recording the Augmented Limb Leads aVR, aVL and aVF. The location of the electrodes are the same as in A. **C.** Electrode positions for recording the Precordial Leads V1, V2, V3, V4, V5 and V6. The location of the electrodes are indicated with the leads name. Adapted from [98, 101, 102].

### Augmented Limb Leads

The three augmented limb leads are identified as aVR, aVL and aVF. For the lead aVR the positive input is the right arm electrode RA, for lead aVL is the left arm electrode LA and for lead aVF is the left leg electrode LL. The reference input is formed by averaging the potentials in the three limb LA, RA and LL. This combination of the three limb lead electrodes is known as Wilson's central terminal [103]. Due to this reference, the potential of these leads suffer "augmentation" i.e. it is increased the size of the deflections of the potential measured. As a result of the arrangement of the leads, aVR, aVL and aVF capture the electrical activity of the heart with angles of  $-150^\circ$ ,  $-30^\circ$  and  $90^\circ$  respectively on the frontal plane.

These leads are also known as unipolar limb leads since, unlike the standard limb leads, the reference input - Wilson's central terminal - potential is close to zero [104].

### Precordial Leads

The precordial leads detect the action potential at each of the six defined torso locations V1, V2, V3, V4, V5 and V6. The positive input of each lead is the electrode placed on



each precordial location [104]: The electrode V1 and V2 are located to the right and left of the sternum in the fourth intercostal space respectively; electrode V3 is located midway between leads V2 and V4; electrode V4 is placed in the mid-clavicular line in the fifth interspace and electrode V5 and V6 are located in the anterior axillary and mid-axillary line respectively, at the same level as lead V4.

The reference input is the Wilson central terminal mentioned previously. Thus, these leads are also considered unipolar. The leads V1, V2, V3, V4, V5 and V6 record the behaviour of the heart on the horizontal plane with angles of  $0^\circ$ ,  $30^\circ$ ,  $60^\circ$ ,  $75^\circ$ ,  $80^\circ$  and  $100^\circ$  respectively [105].

The standard 12-lead ECG, represented in Figure 4.3, is composed by the three standard bipolar limb leads, three augmented unipolar limb leads and by the six precordial leads. Together, these leads grant a three-dimensional representation of depolarization and repolarization of the atria and ventricles [104]. A summary of the position of the standard 12-lead ECG electrodes is given in Table 4.1.

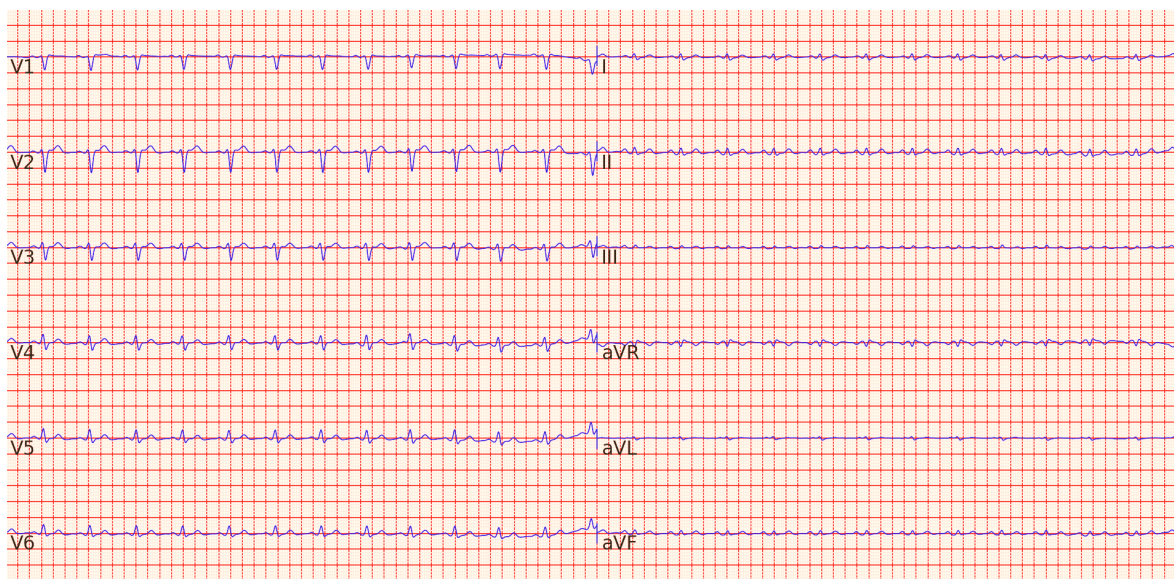


Figure 4.3: A standard 12-lead ECG.

## 4.2 Arrhythmia Classification

Automaticity refers to the capacity of cardiac cells to generate spontaneous action potentials. This property allows the mechanical and electrical connectivity of all the areas of the heart, ensuring appropriate cardiac function and rhythm. The normal heart rhythm is called **Sinus rhythm (NSR)** and has the following characteristics [106]:

- Rate of 60–100 beats/min;
- P wave precedes each QRS complex;

Table 4.1: Location of Standard 12-Lead ECG electrodes.

| Lead Type            | Positive Input                              | Negative Input            | Angle* |
|----------------------|---|---------------------------|--------|
| Standard Limb Leads  |   |                           |        |
| Lead I               | Left arm                                    | Right arm                 | 0°     |
| Lead II              | Left leg                                    | Right arm                 | 60°    |
| Lead III             | Left leg                                    | Left arm                  | 120°   |
| Augmented Limb Leads |   |                           |        |
| aVR                  | Right arm                                   | Wilson's central terminal | -150°  |
| aVL                  | Left arm                                    |                           | -30°   |
| aVF                  | Left leg                                    |                           | 90°    |
| Precordial Leads     |   |                           |        |
| V1                   | Right sternal margin; 4th intercostal space |                           | 100°   |
| V2                   | Left sternal margin; 4th intercostal space  |                           | 80°    |
| V3                   | Midway between V2 and V4                    | Wilson's central terminal | 75°    |
| V4                   | Mid-clavicular line; 5th interspace         |                           | 60°    |
| V5                   | Anterior axillary line; same level as V4    |                           | 30°    |
| V6                   | Mid-axillary line; same level as V5         |                           | 0°     |

\*The angles are on the frontal plane for the limb leads and on the horizontal plane for the chest leads.

- P wave is upright in leads III, aVF and inverted in lead aVR;
- PR interval 0.12–0.21 s;
- QRS duration  $\leq 0.10$  s;
- QTc  $\leq 0.44$  s ,

where QTc is known as the corrected QT interval and is determined by selecting the longest QT interval and dividing it by the square root of the cycle length [106]. The heartbeat of a healthy person is represented in Figure 4.4.

When the automaticity of the heart is disrupted, the heart has a condition called cardiac arrhythmia. Understanding the electrophysiological mechanism of an arrhythmia and conducting an accurate interpretation of an ECG can successfully diagnose and select the appropriate treatment for these conditions [107].

For the realisation of this work, four cardiac arrhythmia databases were used. Each dataset has at least one type of arrhythmia, which is identified by a numerical distinct code, known as SNOMED-CT. A subset of five codes have been selected as classes for classification. These classes were chosen since almost all of them are presented in each



Figure 4.4: Representation of Sinus rhythm.

dataset and are the most frequent classes overall. All other codes were ignored in this study. Table 4.2 presents a summary of the selected classes and their proportions in the datasets.

Table 4.2: Classification classes with their abbreviation, [SNOMED-CT](#) code, count and percentage in the four datasets.

| Class                               | Abbreviation | SNOMED-CT | Count (%)      |
|-------------------------------------|--------------|-----------|----------------|
| Atrial fibrillation                 | AF           | 164889003 | 3320 (11,12%)  |
| First-degree atrioventricular block | IAVB         | 270492004 | 2288 (7,66%)   |
| Left bundle branch block            | LBBB         | 164909002 | 1003 (3,36%)   |
| Sinus rhythm                        | NSR          | 426783006 | 20842 (69,82%) |
| Right bundle branch block           | RBBB         | 59118001  | 2399 (8,04%)   |

#### Atrial fibrillation (AF)

AF is the most common cardiac arrhythmia and is defined by a totally irregular ventricular rhythm and absence of P waves [106]. This pathology is represented in Figure 4.5.



Figure 4.5: Representation of Atrial Fibrillation.

During AF, the atria discharge at a rate of 350 to 600 beats per minute, which results in several wave of electrical activity with low amplitude haphazardly circulating in the atria.

The effective atrial contraction is reduced, leading to the absence of P waves. The AV node does not conduct every atrial impulse to the ventricles, blocking and delaying succeeding impulses. As a result, the ventricular rhythm becomes irregular. This is evident in the ECG by the appearance of irregular QRS complex, also described as variant RR.

#### First-degree atrioventricular block (IAVB)

IAVB is characterized by a PR interval longer than 200ms and is represented in Figure 4.6.

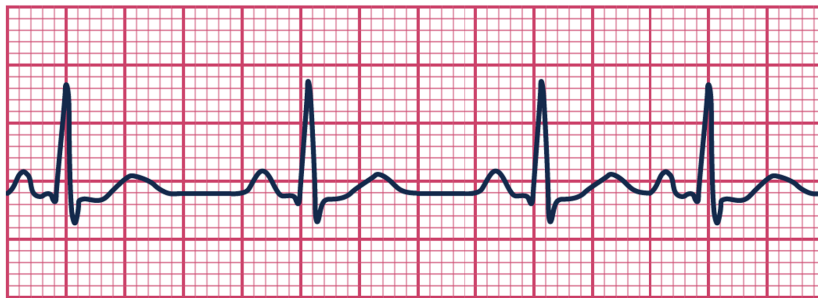


Figure 4.6: Representation of First-degree atrioventricular block.

This condition occurs when the atrial impulse is delayed and takes longer to reach the ventricles. As a result, there will be a greater distance between the P wave and the QRS complex and, thus, a prolongation of the PR interval.

#### Left bundle branch block (LBBB)

A LBBB is characterized by a QRS duration of more than 100 ms with a complex leftward skew in the second half of the QRS complex [106]. This pathology is represented in Figure 4.7.

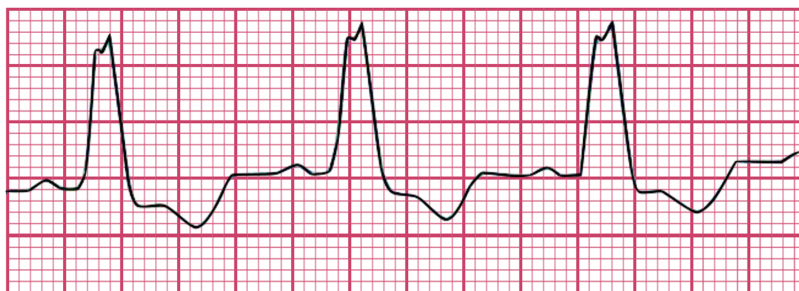


Figure 4.7: Representation of Left bundle branch block.

In LBBB, the impulses arise first from the right bundle branch. As a result, the activation of the left ventricle is delayed and occurs the prolongation of the duration of the QRS complex. The small negative Q waves seen in left ventricular leads (V5, V6, I and aVL) are replaced by a larger positive R wave. Deep S waves can also be seen in the leads V1, V2 and V3. LBBB can be intermittent.

### Right bundle branch block (RBBB)

RBBB is defined as having a QRS complex duration that surpasses 120 ms and the second half of the QRS complex is skewed rightward [106]. This pathology is represented in Figure 4.8.



Figure 4.8: Representation of Reft bundle branch block.

In this condition there is delay in the right ventricular activation, resulting in the right ventricle depolarizing after the left ventricle. This leads to an increase in duration of the QRS complex and a secondary R wave in leads V1 and V2, forming an M-shaped complex. It can also be found a wide S wave in the leads V5, V6, I and aVL.

## 4.3 Datasets

As previously mentioned, four public datasets with multi-label data from various countries were used in this work. These datasets were provided by the PhysioNet/Computing in Cardiology Challenge 2020, as proposed by Perez Alday et. al [108].

All records are 12-leads ECG and the data is provided in WFDB format. The length of the records and the number of arrhythmia classes vary for each dataset. Also, other details such as patient sex and age are also disclosed.

The data of each dataset was obtained in 16 bits with a 24 bit offset. The amplitude resolution is 1000 mV, the analog-to-digital converter resolution is 16 bits, and the baseline value corresponding to 0 physical units is 0.

### 4.3.1 CPSC2018 dataset

CPSC2018 dataset was provided in the 1st China Physiological Signal Challenge. The data was collected from 11 hospitals and contains 6877 records - 3178 female and 3699 male, with a mean age of 60.2 years [109]. The 12 leads ECG records have a length from 6 s to 60 s. The recordings were sampled at 500 Hz.

For the five classes selected in this work, the dataset has 4735 records, 1996 female and 2739 male. There is in total 1221 AF, 722 IAVB, 918 NSR, 236 LBBB and 1857 RBBB diagnoses.

### 4.3.2 PTB dataset

This database is named after and provided by the [Physikalisch Technische Bundesanstalt \(PTB\)](#) University and contains 516 records (male: 377, female: 139) with mean age of 56.3 years [108]. Each signal is digitized at 1000 samples per second and the records vary in length, having a mean duration of 110.8 seconds. This database has only 4 diagnoses as classes, in which one is [AF](#) and the other is [NSR](#). For these 2 classes, there is in total 95 records, 25 female and 70 male, with 15 [AF](#) and 80 [NSR](#) diagnoses.

### 4.3.3 PTB-XL dataset

The PTB-XL is a large dataset also provided by the [PTB](#) University and has 21837 clinical 12-lead [ECGs](#) (male: 11,379 and female: 10,458), with a mean age of 59.8 years [108]. The records are of 10 second length with a sampling frequency of 500 Hz. The labels assigned to each data sample were annotated by two cardiologists. This dataset does not have all the 5 chosen classes, only missing the class [RBBB](#). Furthermore, for the remaining 4 classes, the dataset has a total of 19814 records ( 9577 female and 10237 male), with 1514 [AF](#), 797 [IAVB](#), 18082 [NSR](#) and 536 [LBBB](#) diagnoses.

### 4.3.4 G12EC dataset

The [Georgia 12-lead ECG Challenge \(G12EC\)](#) database was collected in Georgia, having a distinct demography of the Southeastern United States. This database has 10344 12-lead [ECGs](#) records - 5551 male and 4793 female - with a mean age of 60.5 years [108]. The length of the [ECG](#) recordings is 10 seconds and they were sampled at 500 Hz.

For the five classes chosen, there are 3629 records in total (1622 female and 2007 male), with 570 [AF](#), 769 [IAVB](#), 542 [RBBB](#), 231 [LBBB](#) and 1752 [NSR](#) diagnoses.

## METHODOLOGIES

This Chapter provides a detailed description of methods deployed and developed during this dissertation, from the procedures involved in the **DL** model development to the **UQ** experiments for classification with rejection option and active learning. The most relevant Python Libraries and Modules used can be found in Table [A.1](#) and [A.2](#) of Appendix A.

### 5.1 Overview

This work methodology comprises several steps that address the development of a **DL** model, the **UQ** methods and their application for classification with rejection option and active learning. In more detail, the **DL** model was trained with two databases and tested using two test sets: 1) an independent test set originated from the same databases where the model was trained, and 2) a test set from two different databases. For the uncertainty estimation, three different approaches were employed for comparison purposes: 1) A single model using entropy-based measures; 2) A model that used **MC Dropout** as **UQ** method; and 3) A model developed using ensemble techniques. Finally, the **UQ** methods were applied to the classification with rejection option to investigate the occurrence of dataset shift and explore the role of uncertainty in active learning. An overview of this research is summarized in Figure [5.1](#)

### 5.2 Deep Learning Model

#### 5.2.1 ECG Data Preparation

As previously mentioned, four public available databases were used. The details of each database are described in Section [4.3](#).

To reduce the computational cost of the adopted approach, only one **ECG** lead was used. The lead chosen was aVR, since it was showed that, for the selected arrhythmia types, this lead produced the best results in the work of Chen et al. [[110](#)].

The data was downsampled to 125 Hz in order to reduce computational cost. For sample length, a 10 seconds window size was used. Thus, data below 10 seconds were



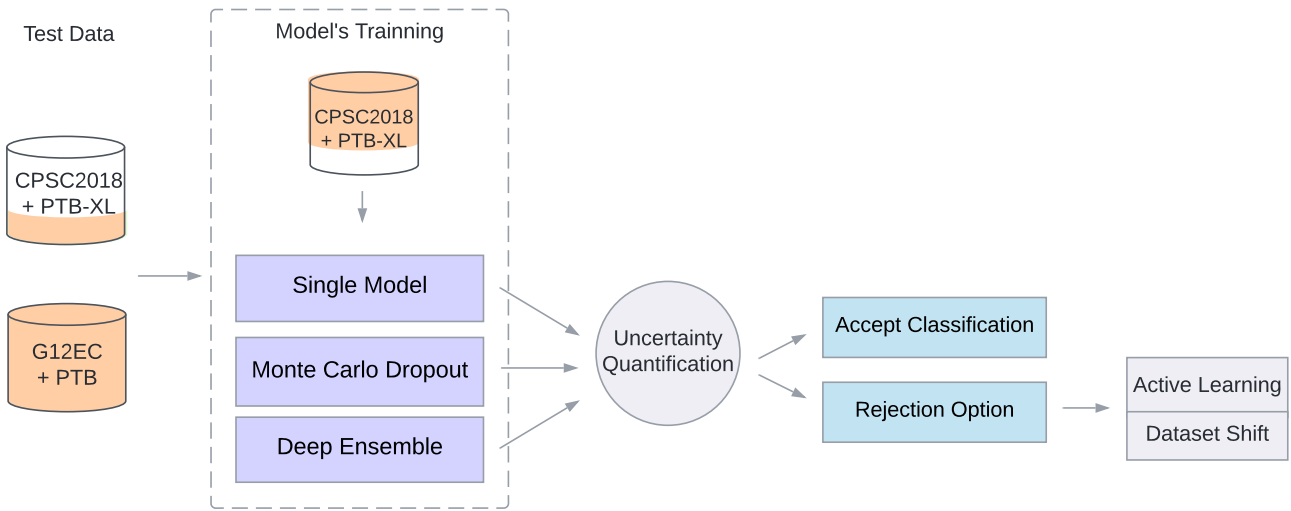


Figure 5.1: Overview of this work methodology

excluded and data above 10 seconds were truncated, so that all data has 1250 sample points.

The ECG signals were filtered using a 2nd order band-pass Butterworth filter between 1 and 40 Hz to remove high frequency noise such as Electromyogram noise, Additive white Gaussian noise and power line interference (50 Hz) as well as low frequency noises such as Baseline Wander [111]. It was also employed a smooth function using a window of 10 samples. This method is based on the convolution of the selected window to average the sample points with their neighbors. This filter also reduces high frequencies and enhances low frequencies in the signal.

Lastly, the data was standardised through a z-normalisation, where the data has the mean removed and is scaled to unit variance. It ought to be emphasised that the mean and standard deviation used in the normalisation are solely derived from training data. Figure 5.2 depicts a diagram of the preprocessing procedures adopted.

### 5.2.2 Deep Learning Model Architecture

The model developed is a one-dimensional CNN. The architecture consists of three convolutional blocks, each with a convolutional layer followed by a batch normalization layer, a PRelu activation function, a pooling layer and a Dropout layer with rate of 0.25. Each convolutional layer has the same kernel size (31x31) but different number of filters. The PRelu function has an initializer of 0.25. The pooling layer employed is a max pooling layer that consists of replacing consecutive patches of size  $n$  by their maximum value. Lastly, the algorithm has a flatten layer, to reduce the multidimensionality of the third block output. After the convolutional blocks, a flatten layer was applied, resulting in a Latent Vector. Three fully connected layers are added and the last one has a sigmoid activation function with the same number of neurons as classes. Figure 5.3 and Table 5.1



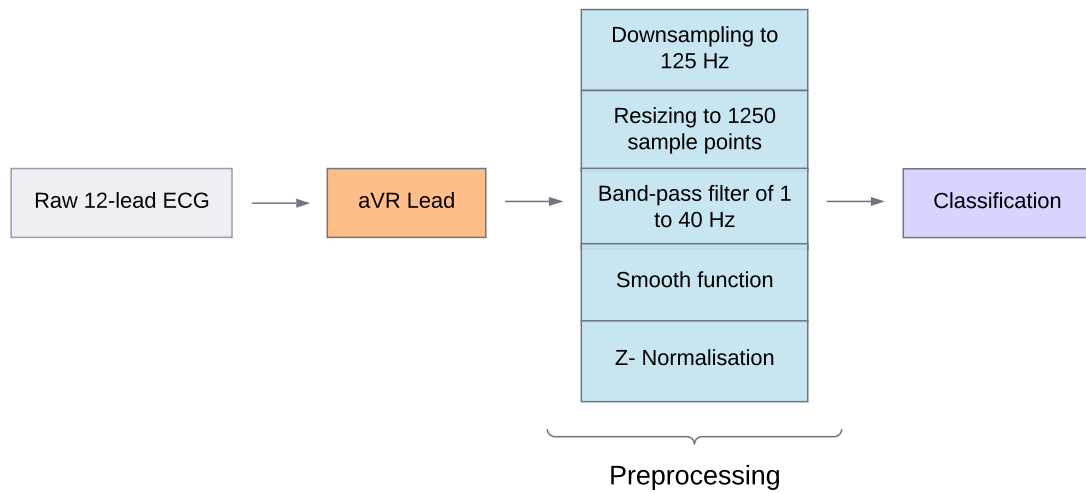


Figure 5.2: Flow chart diagram of the preprocessing approach

present the architecture and all the parameters used in the developed algorithm as well as the features map of each layer.

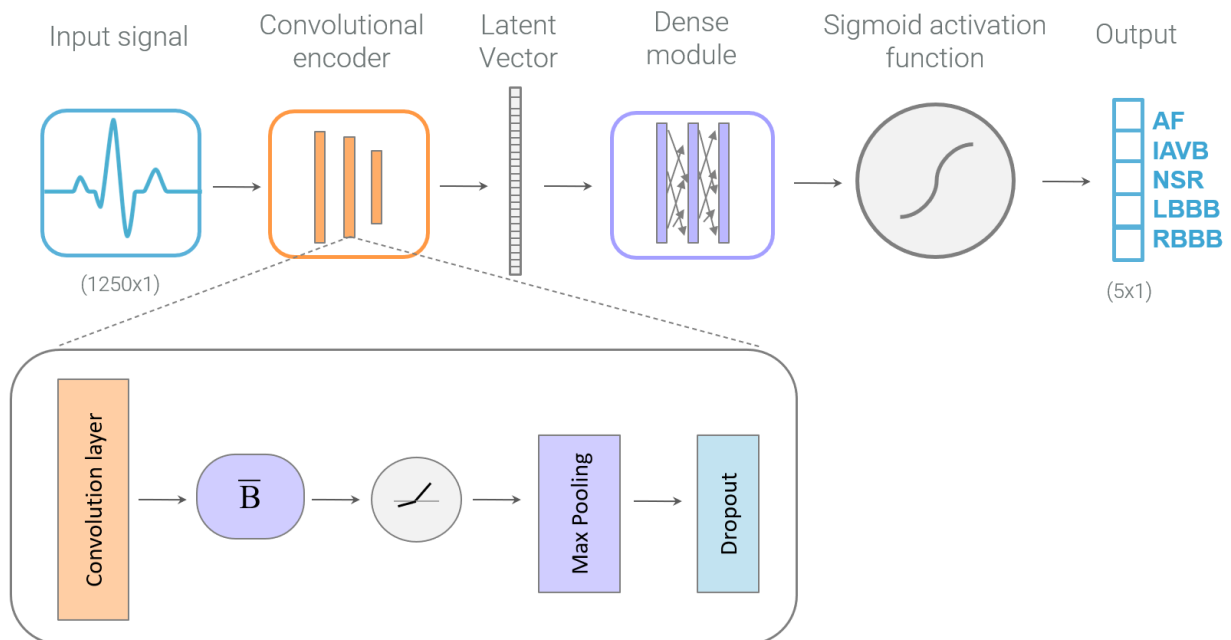


Figure 5.3: The flowchart of the designed algorithm.

The model was trained in 30 epochs with a batch size of 64. The loss function employed was the binary cross-entropy as well as an Adam optimizer with a learning rate of 0.1%.

Table 5.1: Feature interpretation in the developed CNN model. Inspired by [112]

| Layer             | Filters | Size  | Stride | Output    | Feature Interpretation               |
|-------------------|---------|-------|--------|-----------|--------------------------------------|
| Input             | -       | -     | -      | (1250,1)  | 1 lead with 1250 sample points       |
| Convolution 1     | 512     | 31x31 | 2      | (610,512) | 512 feature map                      |
| Max Pooling 1     | -       | 2x2   | 2      | (305,512) | Feature map reduction                |
| Convolution 2     | 256     | 31x31 | 2      | (153,256) | 256 feature map                      |
| Max Pooling 2     | -       | 2x2   | 2      | (77,256)  | Feature map reduction                |
| Convolution 3     | 128     | 31x31 | 2      | (39,128)  | 128 feature map                      |
| Max Pooling 3     | -       | 2x2   | 2      | (20,128)  | Feature map reduction                |
| Flatten 1         | -       | -     | -      | 2560      | Reduction of the multidimensionality |
| Fully Connected 1 | -       | -     | -      | 256       | Weight parameters                    |
| Fully Connected 2 | -       | -     | -      | 128       | Weight parameters                    |
| Fully Connected 3 | -       | -     | -      | 5         | Classes                              |

Since the model is trained with imbalanced datasets, it was added the class weight parameter that defines the weighting to adopt for each class when fitting the model.

### 5.2.3 Training and Testing

The data used to train the model was composed by two datasets, CPSC2018 and the PTB-XL. These datasets were equally split into 60% training, 20% validation and 20% testing. The test set from CPSC2018 and PTB-XL was used as an in-distribution set and will be referred as **test-in** from now on. Additionally, a test set composed from PTB and G12EC datasets were used and named as **test-out**, since the data come from different sources.

Firstly a single model was trained and its hyperparameters tuned. The final hyperparameters can be consulted in Section 5.2.2. Then, based on the obtained model, two different approaches were employed: the MC Dropout and the DE.

The MC Dropout method, as mentioned in Section 2.3.1, consists in removing neurons from the network during training and testing. Therefore, this method consists in testing a number of times in a given set but in each classification, different weights are removed. This approach was applied 30 times to both test sets.

Regarding the DE, also described in Section 2.3.1, is a method in which the results from every classification are obtained with models with the same structure but initialised with different data. Thus, DE was performed with 30 distinct initialisation resulting in 30

models used for each prediction. In addition, another DE model was created using the same data but different leads. This ensemble was then developed by training 3 models, each one with a different lead: lead aVR (the standard lead used in this work), lead V1 and lead V6. Thus, the result obtained is 3 different outputs for each sample of both test sets.

Lastly, to obtain the final prediction for both MC Dropout and DE approaches an aggregation mechanism was implemented. In order to compare the best aggregation mechanism, three aggregation methods were tested, namely the arithmetic mean, the mode per class and the mode per diagnosis (mode per vector). An example of the three aggregation methods is exemplified in Figure 5.4.

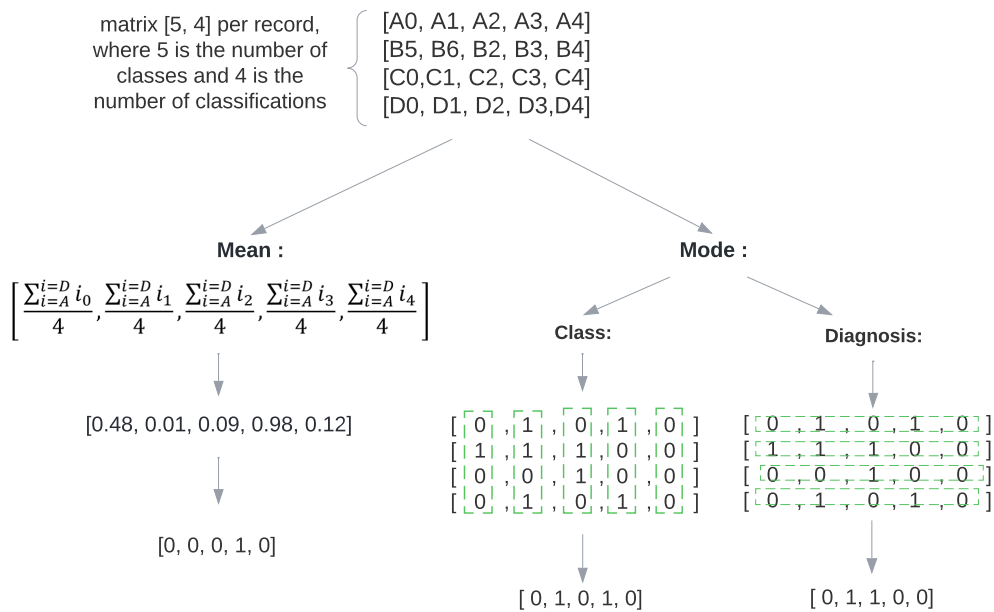


Figure 5.4: Diagram of the treatment employed for an example of outputs obtained from Monte Carlo Dropout and Deep Ensemble method.

### 5.2.4 Threshold Optimization

In a multi-label classification, the output is a vector with the same size as the number of classes. Each value of the vector corresponds to the probability of belonging to the given class. Usually, the probability threshold used to decide the limit in which the sample belongs to the class is 0.5. However, in imbalance datasets, the threshold of 0.5 may not produce the best prediction results.

Due to the class imbalance of the datasets used, a threshold optimization procedure was employed. The ROC and PR curves are commonly used for threshold optimization.

As explained in Section 2.2.4.7, the ROC curve is a graphical representation of TPR as a function of FPR. A possible optimized threshold is the G-mean value. This value is the geometric mean of recall and specificity (see Equation 5.1). Thus, for each class, it was

calculated the G-mean value for all the points in the ROC curve. The optimal threshold is the maximum value estimated, where the relationship between the TPR and FPR is optimized.

$$\text{G-mean} = \sqrt{\text{Recall} * \text{Specificity}} = \sqrt{\text{TPR} * (1 - \text{FPR})} \quad (5.1)$$

Another possible way to calculate an optimal threshold is through the Youden's index. This value is the sum of recall and specificity minus one, as shown in Equation 5.2. This index is defined for all points of an ROC curve and the optimal threshold value is the maximum difference between TPR - FPR. This optimal threshold was also calculated for each class.

$$\text{Youden's index} = \text{Recall} + \text{Specificity} - 1 = \text{TPR} - \text{FPR} \quad (5.2)$$

Lastly, the PR curve may also be used to determine the optimal threshold. As was explained in Section 2.2.4.8, the Precision-Recall curve is a graphical representation of the relation between precision and recall. The third possible optimal threshold for each class was calculated by determining the F1-score (see Section 2.2.4.5) for each point of the Precision-Recall curve and choosing the maximum value.

### 5.3 Uncertainty Quantification

For the comparison purposes of different uncertainty estimation methods, three different model techniques were trained, namely: 1) A single CNN model; 2) A model using MC Dropout method; and 3) two models developed using DE.

The applied uncertainty measures depend on the type of model being used. In the first model, a single CNN was trained, which means that a single probability distribution is returned. In this case, the maximum probability and the Shannon entropy of the predicted probabilities were used as uncertainty measures.

In the case of MC Dropout and DE, several models are used on the prediction phase, meaning that a probability distribution of probabilities distributions are used. In this cases, the decomposition of uncertainties using the classical information-theoretic measures were employed as described in Section 2.3.1. The decomposition results in the quantification of total uncertainty, EU and AU.

In addition, other metrics associated with uncertainty were applied, such as Variation Ratio (VR) [75], which reflects the statistical dispersion of the samples' distributions, and Knowledge Uncertainty Estimation (KUE) [46], that learns the estimation of feature density from the training data and determines the difference in feature density in the testing data.

Note that since the models were trained in a multi-label scenario, the outputted sigmoid values can not be directly interpreted as a probability to be used in the calculation of every uncertainty measures. For example, the entropy measure assumes the inputs as the

predictive probabilities of each class, which must sum 1. Since a prediction in a multi-label classification can return more than one class, the network sigmoid values do not sum 1. For this reason, in this multi-label scenario, each class was assumed as an independent binary case and the uncertainty calculated by each class. Besides the uncertainty by class, the summation of all class uncertainties was employed as the final prediction uncertainty.

## 5.4 Classification with rejection option

It is essential that the model rejects classifications when it is not confident in them. One approach to achieve this is through the uncertainty associated with every prediction.

The empirical evaluation of methods for quantifying uncertainty is a non-trivial problem, due to the lack of ground truth in uncertainty information. A common approach for indirectly evaluating the predicted uncertainty measures is using [ARC](#) (see Section 2.3.2). Due to the imbalance data, instead of using accuracy as a performance measure, the F1-score was used and the F1-Rejection curve was computed to evaluate the behaviour of the developed models. These curves were performed for the uncertainties referred in the previous section, with the rejection occurring from the sample with the highest uncertainty in its classification to the sample with the lowest uncertainty. This evaluation was performed considering the overall performance of model and the performance per class.

As explained in Section 5.3, since the data is multi-label, the uncertainty of an [ECG](#) sample is the sum of each class uncertainty. This results that each sample uncertainty is represented by a value between 0 and 5.

Furthermore, a optimal uncertainty limit for rejection was estimated for the overall model and for each class. This threshold is obtained through the Equation 5.3, introduced in [46]:

$$\tau_a = \operatorname{argmax}_{\theta} \left( |M \cap R_{\theta}| - \frac{c}{1-c} \cdot |A \cap R_{\theta}| \right) \quad (5.3)$$

where  $\theta$  is a threshold in the interval  $[0, 5]$  as previously explained and  $c$  is a rejection cost, set to 0.5. Considering  $A$  is the samples that were classified correctly,  $M$  the misclassified samples and  $R$  the rejected samples,  $|M \cap R_{\theta}|$  represents the true rejects and  $|A \cap R_{\theta}|$  represents the false rejects.

## 5.5 Uncertainty in Active Learning

In active learning, the model itself choose what unlabelled data would be most informative for it [62]. One approach is to use uncertainty estimates to select the samples with higher uncertainty, taking advantage of the separation between epistemic and aleatoric uncertainty, where the former is more relevant as a selection criterion [58]. In principle,

the potential informativeness of data points with high epistemic uncertainty is higher and can help the model generalise better.

Following this idea, the retraining process was performed for the single model and the DE model, where a new set was added to the previous training set for the retraining process. The data used were exclusively from the **test-out** set, since these data come from different origins than the data used for the initial training of the model, and, consequently, are more informative. Each model was retrained for more four epochs using the newly dataset and the same parameters previously used to train the initial models.

To validate if samples with high epistemic uncertainty are more informative to the DE model, three different sets composed by 10% of the **test-out** were defined to the retraining process, namely: 1) random samples; 2) samples with the highest epistemic uncertainty; 3) samples with the total uncertainty. For the single model, the retraining processed occurred with samples with the highest Shannon Entropy and for random samples as well.

Figure 5.5 represents the pipeline used to validate the information power of uncertainty estimates for active learning.

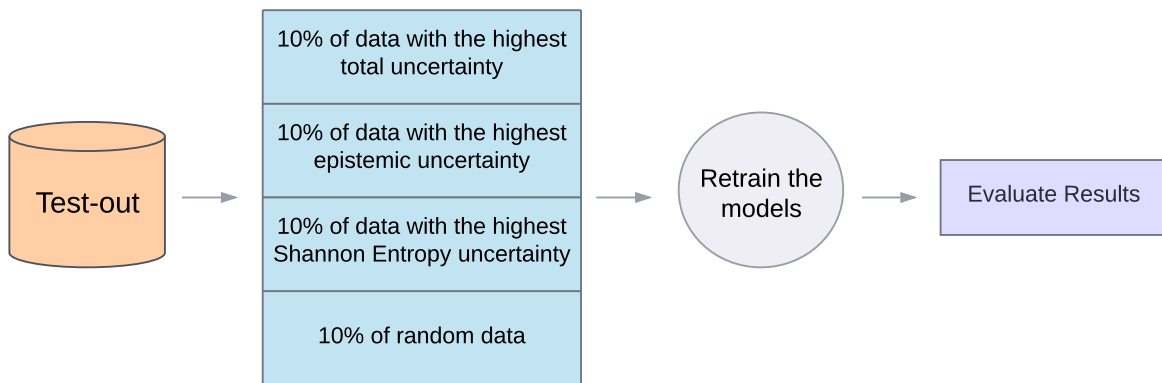


Figure 5.5: Diagram of the pipeline use for active learning experiments

## RESULTS AND DISCUSSION

This chapter presents the results obtained through the methodologies explained in Chapter 5. It is divided into three sections: Performance Evaluation, Classification with Rejection and Active Learning. This chapter also includes a throughout analysis and discussion of these results.

### 6.1 Performance Evaluation

This work comprises the development and evaluation of four models, namely: the single model, the model obtained using **MC** Dropout method and 2 models obtained using **DE** approaches: Ensemble-1 (trained with different training sets) and Ensemble-2 (trained with different **ECG** leads). For a detailed description of the developed models please see Section 5.2.3.

For the final performance evaluation, a preliminary analysis of different aggregation mechanisms for ensembles models, followed by the analysis of different threshold optimization methods was performed.

Regarding the aggregation methods for the **DE** and **MC** Dropout models, three approaches were tested, namely the mean, the mode per class and the mode per diagnosis (see Section 5.2.3). The performance of the three forms of aggregation in these methods were evaluated using the metrics of micro average F1-Score and micro average **AUC-ROC**. This evaluation was performed taking into account the results obtained with the **test-in** set.

Several conclusions can be drawn from Figure 6.1. Looking at the bar charts, it is possible to conclude that, apart from the mode per class in Ensemble-2, the performances using the three aggregations methods are quite similar, with the differences between them being negligible. Therefore, the mode per class was selected as aggregation method for the remainder of this work. The reasons for this decision are the following: 1) the mode per diagnosis aggregation has a high computational cost and do not produce higher performance measures; 2) The mean aggregation method depends on the probabilities obtained from the sigmoid function, which are not calibrated and can produce unreliable

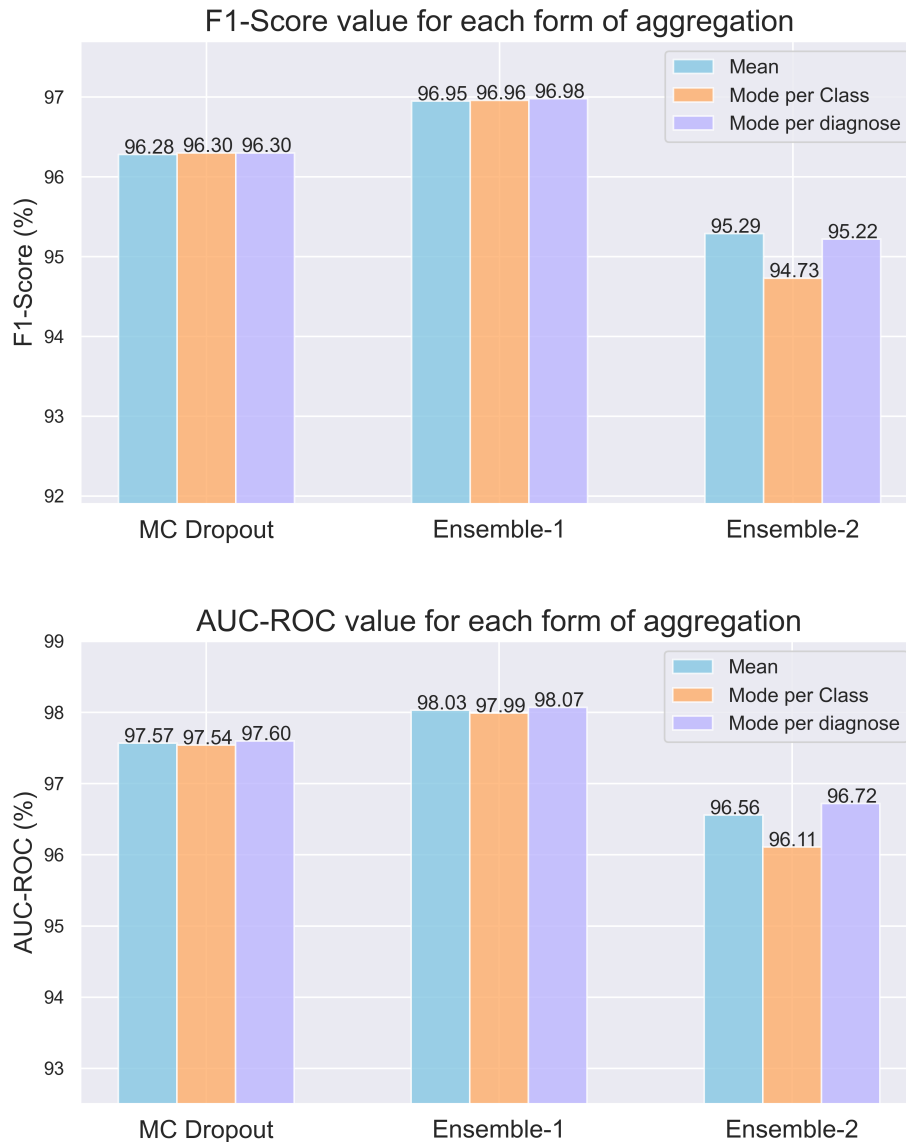


Figure 6.1: Micro average F1-score (up) and micro average [AUC-ROC](#) (down) metrics results in the [MC Dropout](#) and the [DE](#) models. The threshold used was the standard value and the results were obtained for the mean, mode per class and mode per diagnosis aggregation.

results between different models; 3) In the literature, the majority vote (mode per class) is the most common used for ensemble aggregation [113, 114, 115].

It is also possible to observe that the Ensemble-1 model is the one with the best performance results, followed by the model trained with the [MC Dropout](#) method. This result is in agreement with the literature. A possible reason for the Ensemble-2 model having the lowest performance of the three is the fact that this ensemble is performed with only 3 models, compared with the 30 models of Ensemble-1 and [MC Dropout](#). Besides the number of models, Ensemble-2 is a combination of three models trained with different



ECG leads. Thus, the information learnt is different and the performance between leads is also different. Table 6.1 shows the detailed performance measures for each single model of the Ensemble-2, where the model using aVR obtained the best performance measures.

A possible solution to improve the performance of Ensemble-2 could be to train more robust models with different leads. However, this would require training more single models, increasing the computational cost of the ensemble. In view of the results illustrated in Figure 6.1, the Ensemble-2 model will no longer be considered for further analysis since it obtained a considerable lower performance comparing with the other two models.

Table 6.1: Precision, Recall, F1-Score and AUC-ROC metrics for the single models trained with the ECG leads aVR, V1 and V6. This results were obtained using micro average and **test-in** set.

| Model                   | Micro avg Precision | Micro avg Recall | Micro avg F1-Score | Micro avg AUC-ROC |
|-------------------------|---------------------|------------------|--------------------|-------------------|
| Single model - lead aVR | 95,80%              | 95,66%           | 95,73%             | 94,97%            |
| Single model - lead V1  | 91,55%              | 88,51%           | 90,00%             | 85,10%            |
| Single model - lead V6  | 90,51%              | 87,62%           | 89,04%             | 83,51%            |

After algorithms' training and due to the imbalance datasets, a threshold optimization for each class was employed and its performance evaluated. As there are different methods for threshold optimization, a comparison between three methods and the standard threshold of 0.5 for binary classification was done. This comparison was done using the single model and the test set named **test-in**. Table 6.2 shows the results obtained using different thresholds for each class learned from the G-means, the Youden and the F1-Score method. Micro average precision and micro average recall were the performance metrics employed. Micro average F1-score and Micro average AUC-ROC were not included in the evaluation since they are biased toward the techniques used.

Table 6.2: Micro average Precision and Micro average Recall of the of the single model tested on **test-in** set using various thresholds. Besides the standard threshold, it was also applied different thresholds for each class obtained from the G-mean,the Youden and F1-score methods.

| Approach                | Micro avg Precision | Micro avg Recall |
|-------------------------|---------------------|------------------|
| Standard Threshold      | 95,80%              | 95,66%           |
| Maximum G-means value   | 91,75%              | 96,20%           |
| Maximum Younden's Index | 91,75%              | 96,20%           |
| Maximum F1-score value  | 96,64%              | 96,06%           |

From the analysis of Table 6.2 it is possible to conclude that F1-score approach obtained higher precision and recall compared with the standard threshold. The other two approaches (G-means and Younden's index) obtained equal performance measures between

each other, decreased approximately 4% in precision and increased less than 1% in recall when compared with standard threshold. For this reason, the maximum F1-score value was used as threshold optimization method for the rest of the analysis.

Having a form of aggregation and an optimal threshold method selected, it is necessary to investigate the performances of the single model, the MC Dropout model and Ensemble-1 in order to observe how they behave with an optimal threshold compared to using the standard threshold. It is worth mentioning that the optimal thresholds vary depending on the trained model. Thus, using the PR curve, it was calculated the optimal thresholds for the MC Dropout and the Ensemble-1 model.

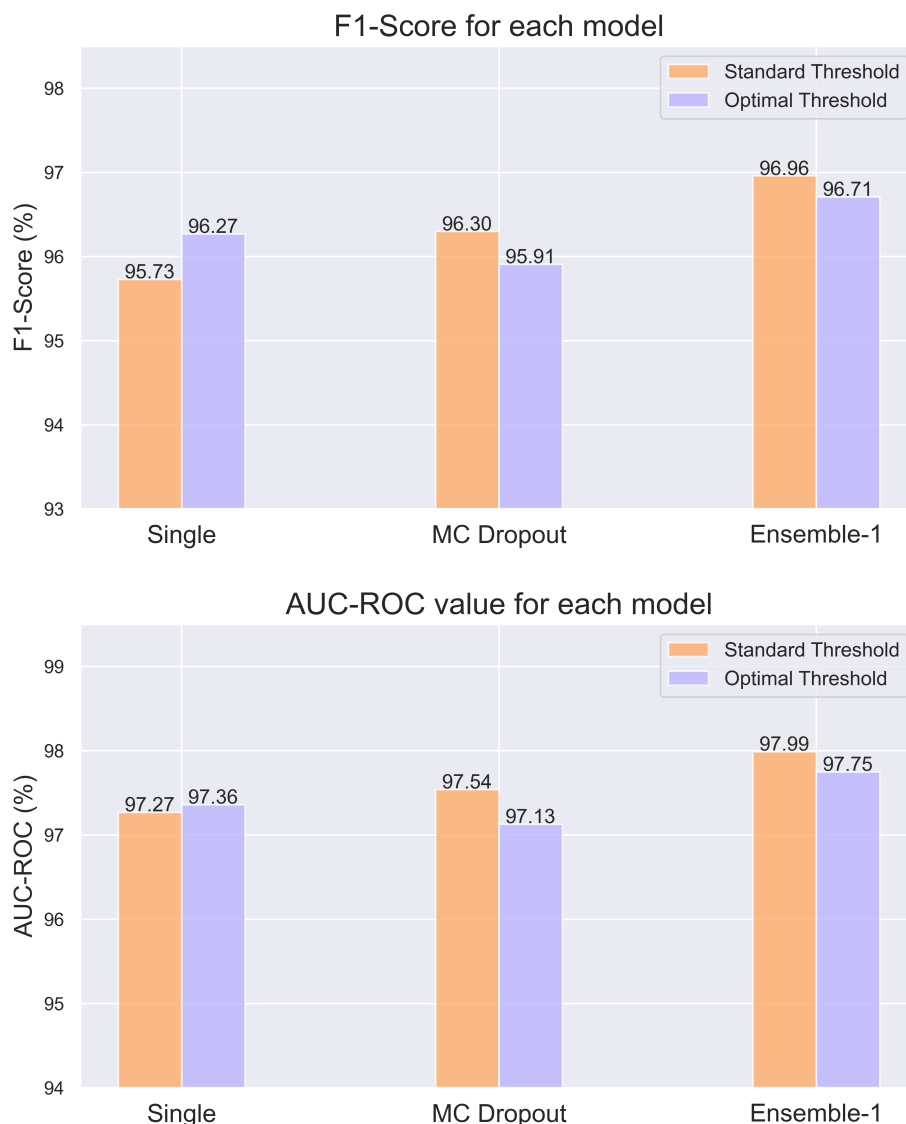


Figure 6.2: Micro average F1-score (up) and micro average AUC-ROC (down) metrics for the optimal and standard threshold in the MC Dropout and the Ensemble-1 models.

Figure 6.2 shows that the three models have similar performances. However, the

Ensemble-1 and MC Dropout methods outperform the single model, as expected since these models assist in reducing models' high confidence in incorrect classifications. A possible justification for the small differences in performance between these models is the fact that they have learned nearly the same information and, even though MC Dropout and Ensemble-1 obtained their outputs through different models, the structure of these models is the same, only the parameters vary. However, when classifying a large number of test samples, several discrepancies in the number of right classifications are required in order for the performance metrics to present high variations in their results. Thus, even the small differences presented in the results might be considered significant for model evaluation and comparison.

Interestingly, while the optimal thresholds for each class outperform the standard threshold in the single model, this is not the case in the MC Dropout and Ensemble-1 models. A possible justification for this difference can be related with the non calibrated probabilities. Since the Ensemble-1 and MC Dropout model are composed by 30 models, the global optimal threshold can negatively influence the predictions of some models, resulting in a lower performance. Additionally, adopting optimal thresholds that are different from the value 0.5 requires modifications on the calculation on the various uncertainties (presented in Section 2.3.1), which are not trivial. For these reasons, the remaining results will be presented using only the standard threshold.

The preliminary analysis was done using the **test-in** set for evaluation, which belongs to the same database as the training set, producing performance results that are comparable to the state of the art. Additionally, a test set from an external database, i.e. a database from a different domain than the train and **test-in** set, was used to validate the generalization capabilities of all models. Thus, the same models were tested with the **test-out**, and the obtained results are presented in in Figure 6.3.

Through the analysis of Figure 6.3, it can be observed that even when tested with the **test-out**, the Ensemble-1 model still has the best performance whereas the single model is the weakest of the three. However, and perhaps the most important conclusion to draw, is that the performance decreases significantly in all three models, going from micro-average F1-Score around 96% to 70%. The same is also true for the micro average AUC-ROC metric.

Table 6.3 presents the micro average F1-scores for each class of the three models. These results support the conclusions drawn since all class performances drop for the **test-out** set, with classes reaching approximately 50% of F1-score. The same behavior was seen for the AUC-ROC results, found in Table B.1 of Appendix B. However, it should be noted that, while the F1-score of the MC Dropout is higher than the single model in the **test-out** set, the F1-score of each class is slightly higher for single model, with the exception of the f1-score of the NSR class. This is due to the fact that, although the micro average F1-score takes unbalanced data into account, the performance of the classes with more data always have more weight in the overall performance than classes with less samples. Thus, for multi-class data, it is necessary to consider not only the model's performance but also the performance of each class.

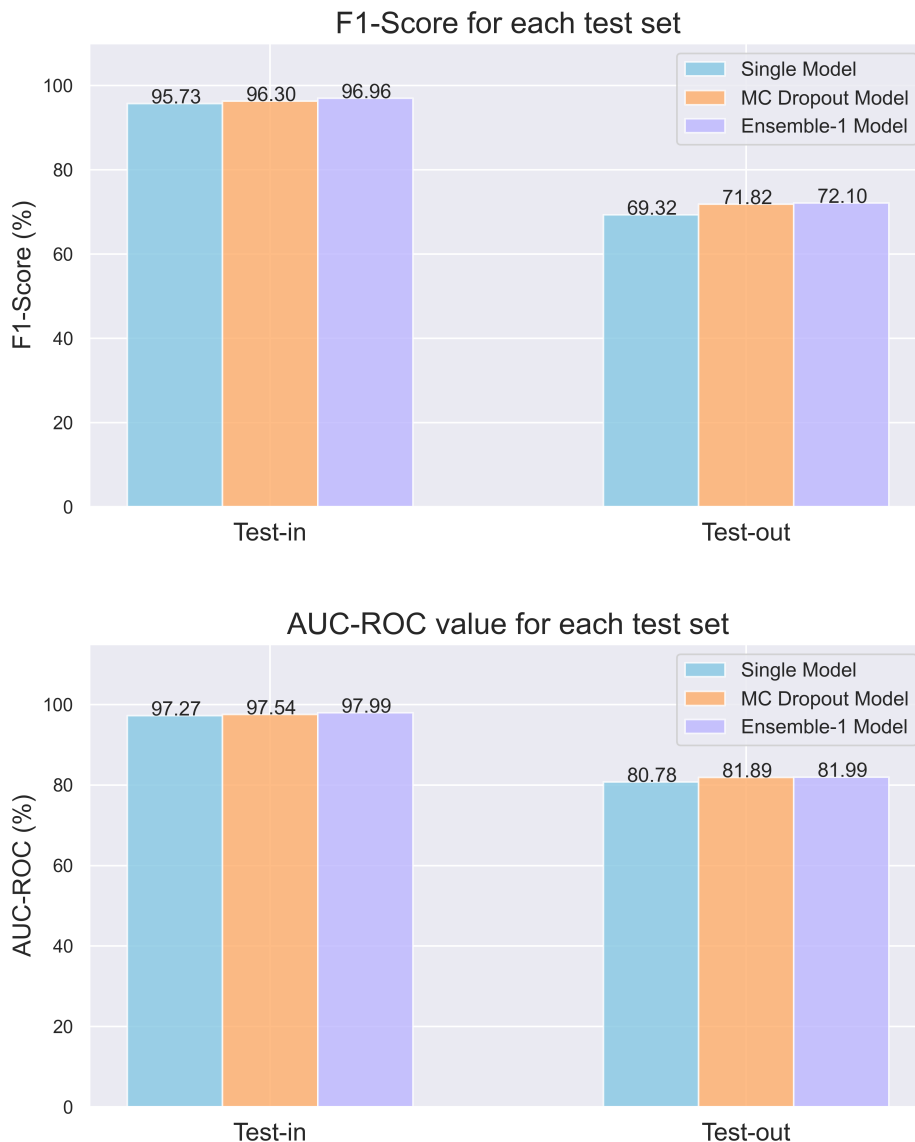


Figure 6.3: Micro average F1-score (up) and micro average [AUC-ROC](#) (down) metrics results for the **test-in** and **test-out** sets when tested in the three developed models.

The results presented in Figure 6.3 and Table 6.3 were expected and are a major indication of dataset shift. As mentioned in Section 2.3.3, this occurs when models are tested with data with different distributions from the ones used to train the models. As a result, these models become unable to classify the data with the same efficiency since they do not recognise the new distributions, as demonstrated through the use of the **test-out**. One way to try to mitigate this problem would be through data augmentation in the training set, in order to generate new information and therefore, turn the trained models more robust. The `tsaug` library [116] was employed to add noise and drift to the samples with the least common classifications. However, the strategy was abandoned since the performance of the single model tested with the **test-in** dropped considerably, leading to

Table 6.3: Micro average F1-score per class for both test sets tested in the three developed model.

| Model               | AF     | IABV   | LBBB   | NSR    | RBBB   |
|---------------------|--------|--------|--------|--------|--------|
| <b>test-in set</b>  |        |        |        |        |        |
| Single model        | 97.29% | 87.83% | 87.14% | 97.28% | 88.16% |
| MC Dropout          | 97.38% | 87.74% | 86.96% | 97.71% | 90.07% |
| Ensemble-1          | 98.00% | 89.13% | 89.44% | 98.05% | 92.99% |
| <b>test-out set</b> |        |        |        |        |        |
| Single model        | 72.59% | 64.41% | 61.62% | 76.97% | 51.31% |
| MC Dropout          | 71.50% | 59.13% | 60.16% | 80.80% | 52.67% |
| Ensemble-1          | 75.30% | 60.00% | 62.60% | 80.76% | 52.89% |

believe that the performance of the remaining models would not improve either.

Even if this approach succeeded, it would not be a viable solution since there are always data with new distributions. In the following sections, several techniques on how to identify and mitigate the dataset shift problem will be discussed.

## 6.2 Classification with Rejection

Although the classification with rejection option does not solve the problem of model’s generalization that leads to poor performance results under data shift, it can be a viable approach to abstain to predict a class under high uncertainty conditions.

Thus, the uncertainty estimation is calculated in order to identify possible misclassifications. In the single model, the uncertainties are calculated through maximum probability and Shannon entropy whereas in the MC Dropout and Ensemble-1 models the aleatoric, epistemic and total uncertainty are estimated using the classical information-theoretic measures. These uncertainties are calculated for the **test-in** and **test-out** sets and the results can be seen in Figures 6.4 and 6.5.

In both test sets for the single model, the behaviour of both measures are similar, increasing their value from **test-in** to **test-out**. The Shannon entropy captures higher uncertainty than the maximum probability, as it can be seen in Figure 6.4.

As for the results in Figure 6.5, for the **test-in** set, the Ensemble-1 and the MC Dropout estimate similar values of uncertainty, presenting the same median and the same range of total uncertainty. The MC Dropout presents a higher range of AU while the Ensemble-1 detects higher EU. As for the **test-out** set, all uncertainties tested suffer an increase when compared with the **test-in** set. Ensemble-1 captures higher total and epistemic uncertainty while the MC Dropout estimate higher aleatoric uncertainty.

As it can be observed in both Figures, while models have very good performance results (micro avg F1-score above 90%) for the **test-in** set, some samples were classified

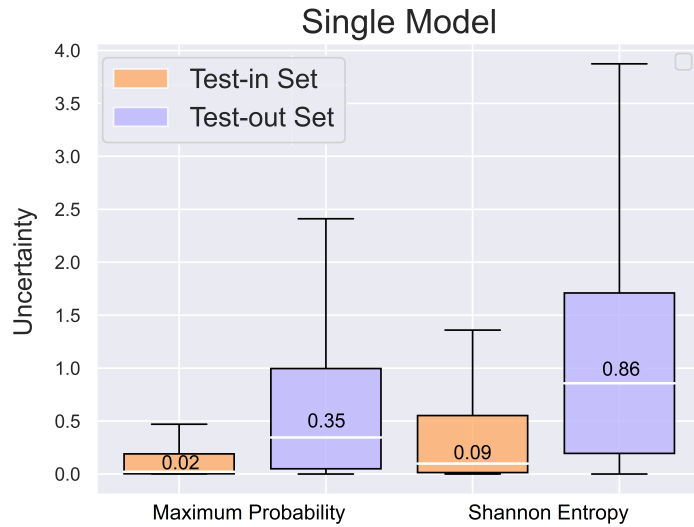


Figure 6.4: Uncertainty Quantification for both test sets in the single model.

with high uncertainty values. Also, the uncertainties calculated for the three models are much higher in the **test-out** than in the **test-in** set. This is an indication that **test-out** is from a different distribution, being the concept of dataset shift applied in this situation. Given the high level of uncertainties shown in the **test-out** results, this suggests that the model is quite indecisive of which cardiac arrhythmias are presented in the samples and, as a result, there is a possibility of misclassified samples. It is important to mention that it was expected that the **EU** would be higher than the **AU** in the **test-out** since the data have different distributions. This reveals that there are still challenges in capturing these two uncertainties correctly.

Additionally, the **VR** was applied to both test-sets to see if the results obtained through this method are consistent with the ones obtained using **EU**.

The Figure 6.6 shows that, as expected, the ratio is higher in the **test-out** than in the **test-in**. Moreover, the Ensemble-1 has higher values of **VR** than the **MC Dropout** model, which is consistent with the values of **EU** estimated in these two models.

Although the previous analysis gives an indication about the differences between **test-in** and **test-out** sets in terms of **UQ** measures, it is important to understand whereas the samples with high uncertainty represent the majority of misclassifications.

Since the empirical evaluation of methods for quantifying uncertainty is non-trivial, due to the lack of ground truth in uncertainty information, a common approach for indirectly evaluating the predicted uncertainty measures is using performance rejection curves.

Thus, to investigate the role of uncertainty in rejection, the F1-rejection curves were produced for the three models, rejecting the samples according to the highest calculated uncertainties.

As shown in the Figures 6.7, 6.8 and 6.9, we observe that, for the three models, the

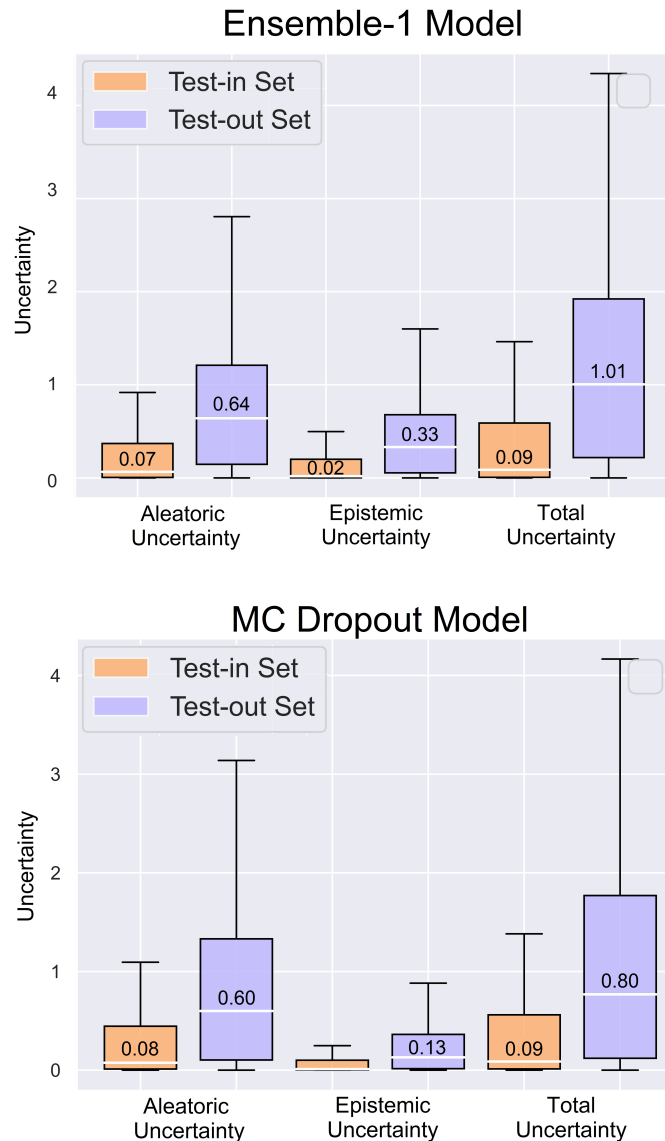


Figure 6.5: Uncertainty quantification for both test sets in the Ensemble-1 (up) and MC Dropout (down) models.

more samples rejected, the better is the models' performance. These results indicate that the higher the uncertainty in a given classification, the less confident the model is of it, implying that the model has misclassified the sample, as observed by the growing of the curve with the increasing rejection. To validate the rejection rate in both sets, a 10% rejection in the training set was applied and the uncertainty thresholds obtained. Using the same thresholds on **test-in** and **test-out**, the rejection rates increased to approximately 12% and 40%, respectively, using the single model for both probability and entropy measures. For the MC Dropout the rejection in **test-in** was 9% and vary between 31% and 34% for **test-out** depending on the uncertainty measure used. For the Ensemble-1 model, the rejections rates vary between the intervals [13%-16%] and [45%-51%] for **test-in** and

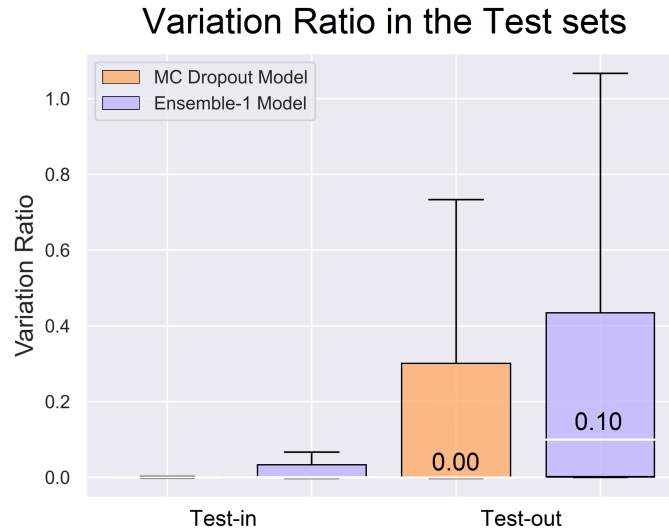


Figure 6.6: Variation ratio for both test sets in the MC Dropout and Ensemble-1 models. The variation ratio in the test-in set for the MC Dropout model and the Ensemble-1 model have a median of 0.0 and a range of uncertainty below 0.2. As a result, the median values for this two cases are not depicted in the figure.

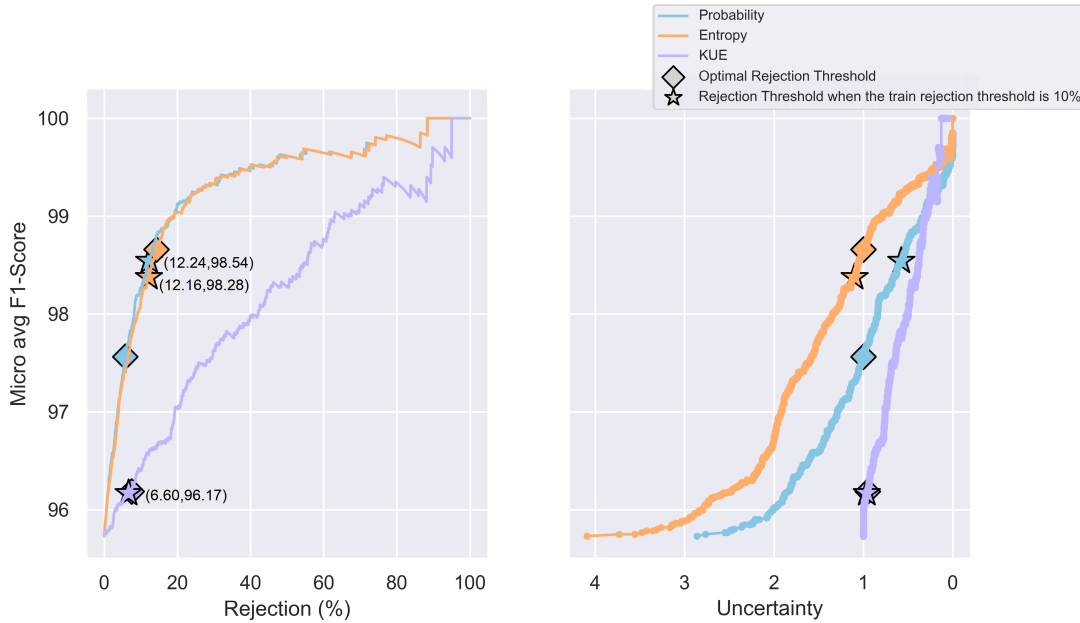
**test-out**, respectively. These rejection thresholds are identified by a star shape and their coordinates, which represent the rejection threshold and the micro average F1-score of the model when that threshold is applied. This rise in rejection threshold percentage indicates that the models are not as confident to classify the **test-in** and **test-out** data since these samples have higher uncertainties than the training data. This increase in the reject threshold percentages is quite substantial for the **test-out** data for all the models and applied uncertainties. This is another evidence of the dataset shift effect, revealing that the models are not as prepared to classify data with different distributions.

Looking at Figure 6.7, the curves obtained for the single model with the maximum probability and Shannon entropy are similar. KUE was also applied to this model by extracting deep features from the DL single model. KUE obtains lower results than Shannon entropy and maximum probability, which was expected since it only detects different distributions between classes. However, it was expected that KUE would grow faster at the beginning of the curve than the other uncertainties, especially in the **test-out** [46]. A possible explanation may be due to the fact that the measure was designed in a single-label setting and it cannot capture uncertainty as effectively as it would in multi-label data. For the multi-label setting, the independence between classes was applied and the independent KUE values by class were summed. This method must be further validate to understand if KUE is suitable to be applied in multi-label setting.

The F1-Rejection curves of the MC Dropout and Ensemble-1 models were examined and the Ensemble-1 model presents better micro average F1-Score results for the same rejection rate although the curves based on the different uncertainty methods are quite



F1-Rejection Curve for the test-in in the single model



F1-Rejection Curve for the test-out in the single model

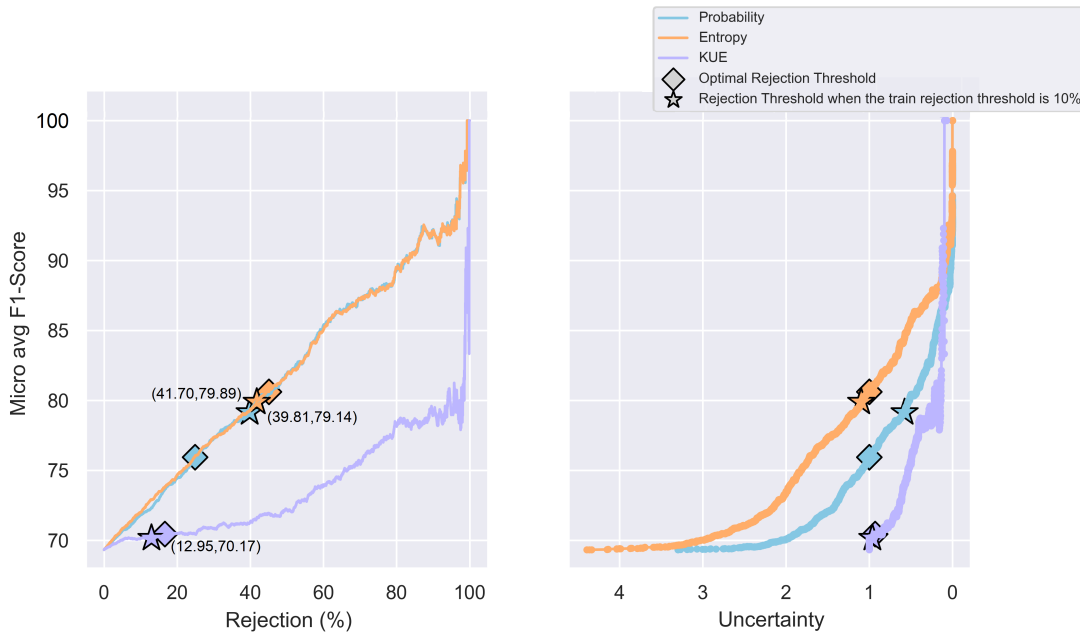
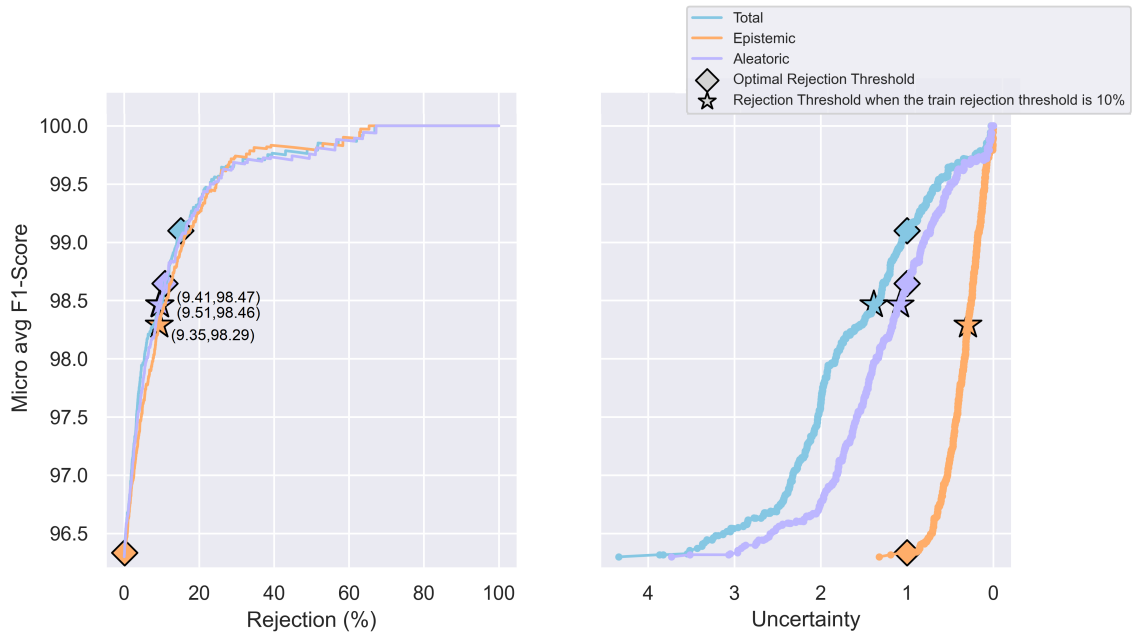


Figure 6.7: F1-rejection curve for both test sets in the single model. The star shaped points are the rejection threshold obtained from an initial rejection using the training data and the diamond shape points are the optimal rejection thresholds calculated.

similar for the two models. Even though the differences between rejection curves with the different uncertainties are minimal, the total uncertainty shows better results along the rejection curve than the epistemic and aleatoric uncertainty. This demonstrates the importance of combining these two uncertainties in the analysis of the model confidence in the classification.

F1-Rejection Curve for the test-in in the MC Dropout model



F1-Rejection Curve for the test-out in the MC Dropout model

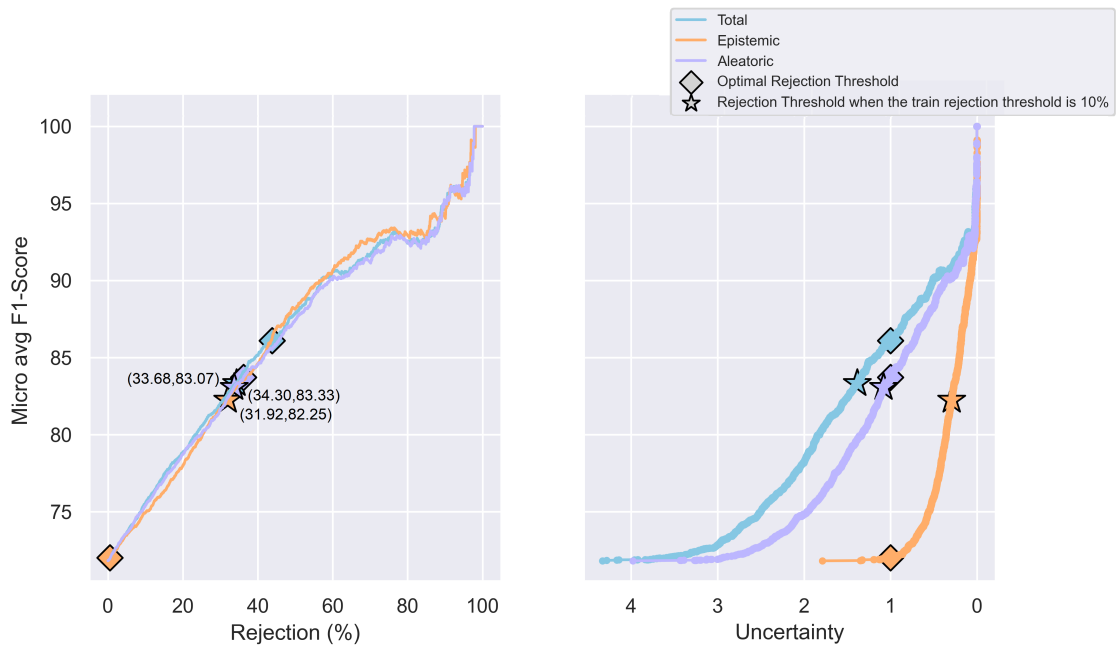
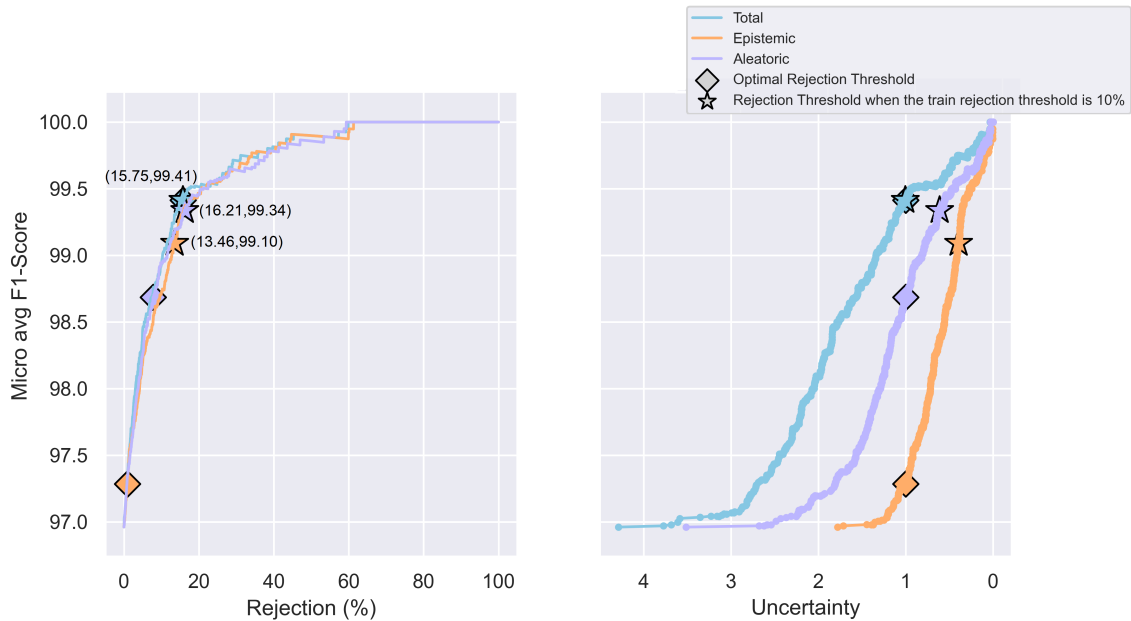


Figure 6.8: F1-rejection curve for both test sets in the MC model. The star shaped points are the rejection threshold obtained from an initial rejection using the training data and the diamond shape points are the optimal rejection thresholds calculated.

Figures 6.8 and 6.9 also show that AU grows faster than the EU at the beginning in both models and for both tests sets. However, the EU manages to overcome the AU in the second half of the curve, particularly in the test-out set. This fact reveals that the samples with higher uncertainty are dominated by AU. However, as noted above, it was expected that the test-out set would possess more EU than aleatoric since the major distinction

F1-Rejection Curve for the test-in in the Ensemble-1 model



F1-Rejection Curve for the test-out in the Ensemble-1 model

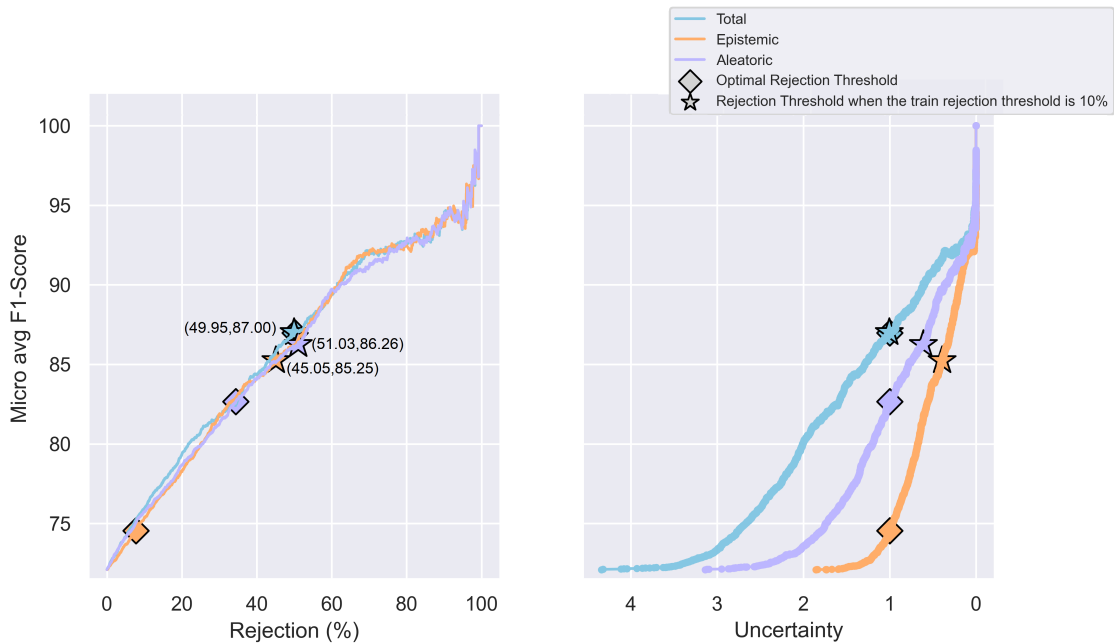


Figure 6.9: F1-rejection curve for both test sets in the Ensemble-1 model.. The star shaped points are the rejection threshold obtained from an initial rejection using the training data and the diamond shape points are the optimal rejection thresholds calculated.

between **test-in** and **test-out** data is the difference between distributions.

The optimal rejection threshold was estimated for each model and for each type of uncertainty method using Equation 5.3. These thresholds are represented in Figures 6.7, 6.8 and 6.9 by diamond-shaped points. This approach shows that the optimal rejection threshold, in all cases presented, rejects samples with uncertainty above 1. Since this

works deals with multi-label data, where the sum of uncertainty of each class varies from 0 to 5, this method provides very high rejection rates, especially in **test-out** data. This level of rejection is not viable to be implemented in the real world, since the models would reject more samples than classify them. The estimation of the optimal rejection threshold it is an important subject that should be address in future work.

Finally, the F1-rejection curves were evaluated for each class. Figure 6.9 presents these curves for the Ensemble-1 model for **test-in** and **test-out** sets. The uncertainty employed was the total uncertainty since, as mentioned previously, has the better performance of the three.

As observed in the curves produced for each model, the performance of each class's classification improves with rejection. However, for the **test-out** set, when the model rejects samples with low uncertainty, the performance drops significantly for almost all classes. This shows that even for the Ensemble-1, which presents the best performance of the three models, the model is quite confident in misclassification data, confirmed by the drop in performance for classifications with low uncertainty. This behaviour can be visualised in the other two models in Figures B.1 and B.2 found in Appendix B, where total uncertainty was used for the MC Dropout model and the Shannon entropy for the single model. These results prove once again that the models are under the dataset shift effect, presenting more uncertainty and false confidence in their classifications.

All the results presented in this section were also reproduced for the AUC-ROC metric, whose results drawn the same conclusions. These results can be consulted in Figures B.3, B.4, B.5, B.6, B.7 and B.8 in Appendix B.

### 6.3 Active Learning

Apart from employing the rejection option, a possible method to deal with dataset shift through uncertainty can be employing active learning.

Thus, 10% of the **test-out** samples (which corresponds to 370 samples) with the highest uncertainty were chosen to test this approach. Only the **test-out** set was used in this approach since it yields much lower results than those obtained with the **test-in** method, indicating that these data includes information that the models have yet to learn.

This was performed for the single model and for the Ensemble-1, as the latter showed better results in capturing uncertainty than MC Dropout. The total uncertainty and Shannon entropy were employed since presented the highest performance. It was also selected the EU in order to evaluate the performance of this uncertainty in this method. These 10% samples were then removed from the test set and the model was retrained with them.

Furthermore, to serve as control, this process was performed for 10% of random samples in order to observe the role of uncertainty in this approach. To obtain statistically significant results, this procedure was conducted 10 times.

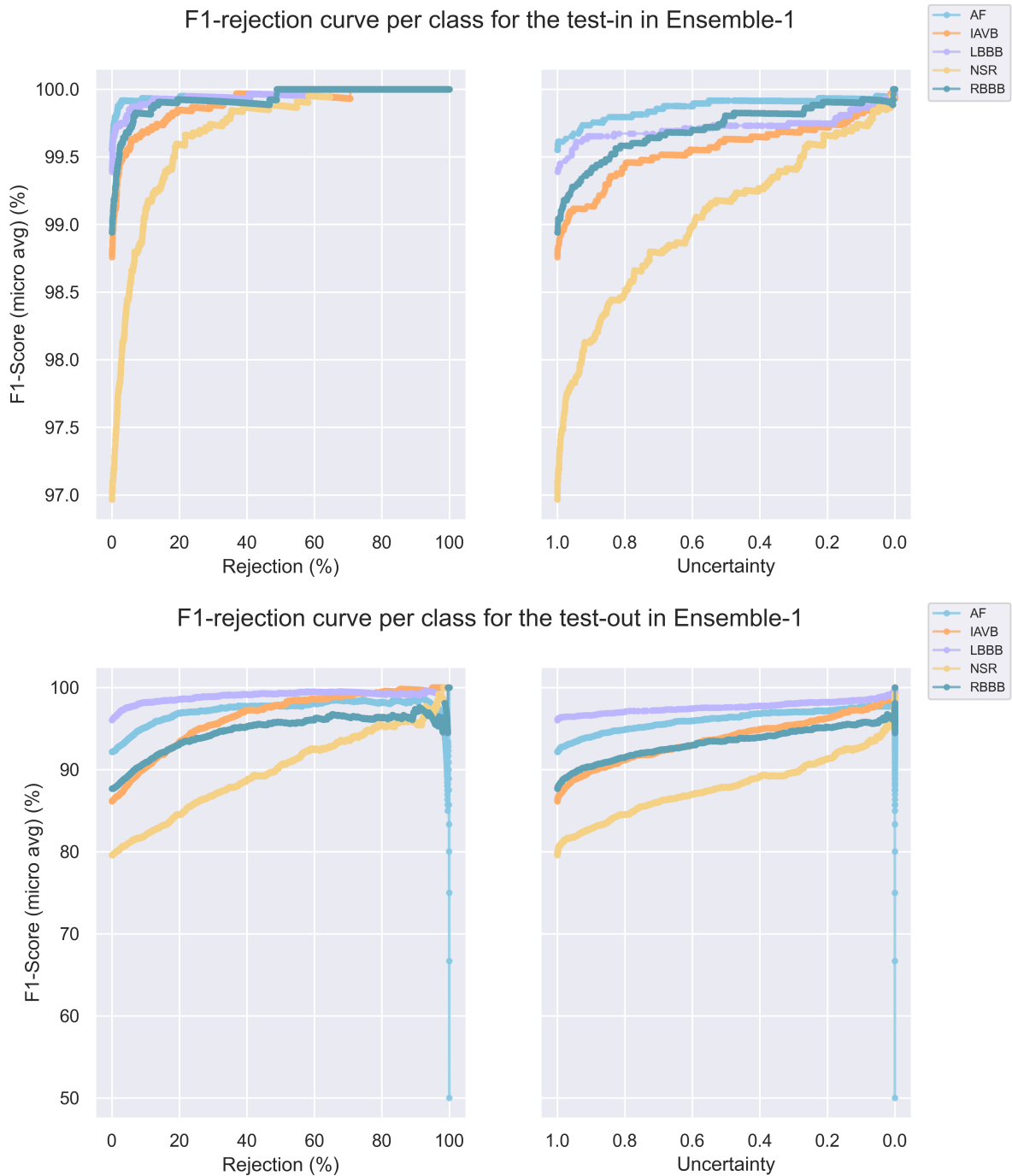


Figure 6.10: F1-rejection curve per class for both test sets in the Ensemble-1 model.

To evaluate the three methods, the retrained models were tested with the **test-out** set without the 10% samples to fairly compare the increase between the retrained model and the baseline model. Thus, the following nomenclature was used: 1) Previous trained model using the complete **test-out** set (Baseline - **test-out**); 2) Previous trained model tested only on 90% of **test-out**, i.e 10% of **test-out** was used to retrain the model (Baseline - **test-out-90**); 3) Retrained model using the selected 10% data and tested on the remaining

90% (Retrain - **test-out-90**). The results can be observed in Figure 6.11 and Table 6.4. Only the micro average F1-Score was used in this section since the results obtained using **AUC-ROC** have been consistent with the results obtained using the F1-Score measure.

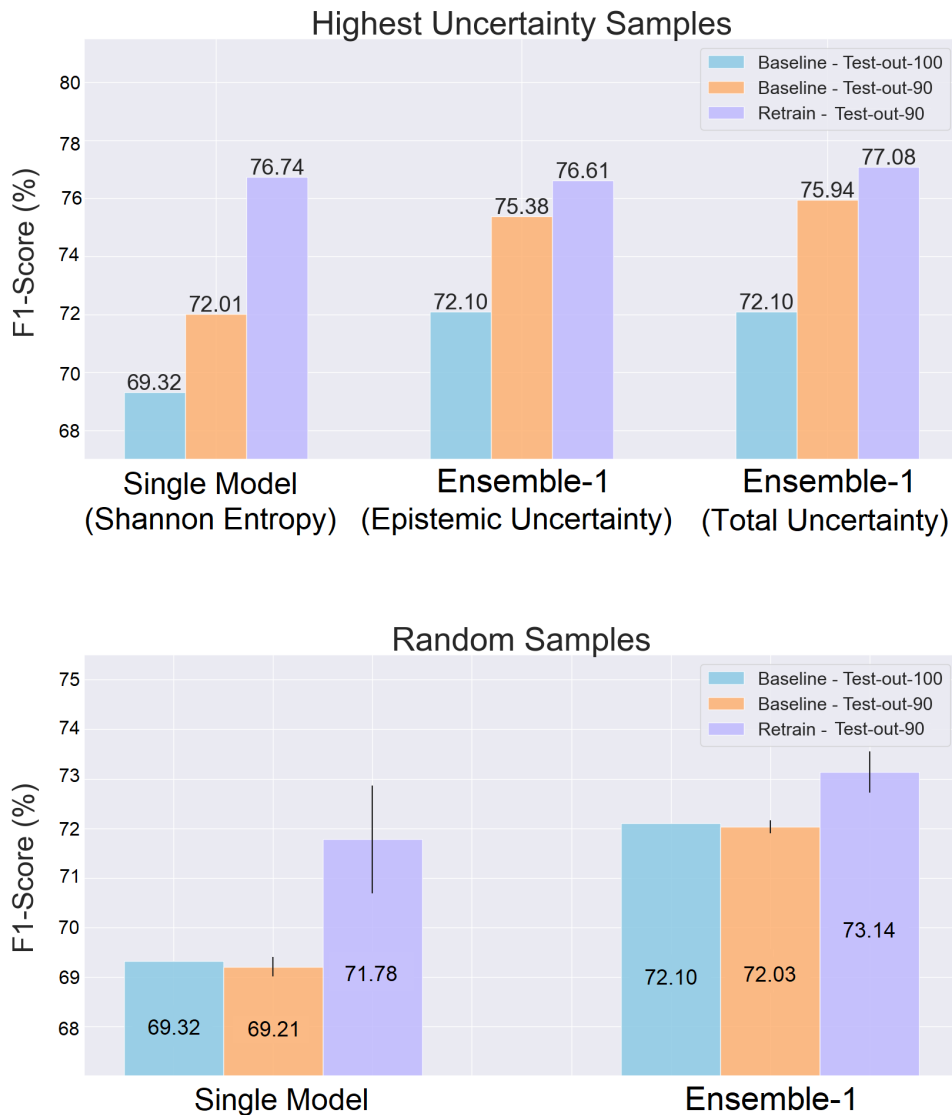


Figure 6.11: Micro average F1-score for the Active learning approach for the highest uncertainties (up) and for random samples (down).

Figure 6.11 exhibits that, when the samples with the highest uncertainty are removed from the **test-out**, the model performance increases slightly. This indicates that these samples were misclassified, as concluded previously. After retraining the model with these samples and evaluating it without them, a slight increase in performance is observed above compared to the baseline models that are tested without these samples. This fact, as expected, indicates that samples with high uncertainty have information that the model has not learned and cannot handle. By retraining the model with the chosen samples,

Table 6.4: Micro average F1-score per class for each step of the active learning approach in the single and Ensemble-1 models.

| Model                                 | AF     | IABV   | LBBB   | NSR    | RBBB   |
|---------------------------------------|--------|--------|--------|--------|--------|
| <b>Baseline - test-out-100</b>        |        |        |        |        |        |
| Single model                          | 72.59% | 64.42% | 61.62% | 76.97% | 51.31% |
| Ensemble-1                            | 75.30% | 60.02% | 62.60% | 80.76% | 52.89% |
| <b>Baseline - test-out-90</b>         |        |        |        |        |        |
| Single model<br>(Shannon Entropy)     | 77.15% | 66.91% | 62.03% | 79.09% | 52.61% |
| Ensemble-1<br>(Epistemic uncertainty) | 79.46% | 64.46% | 67.27% | 82.64% | 52.93% |
| Ensemble-1<br>(Total uncertainty)     | 80.08% | 65.38% | 68.39% | 82.74% | 55.64% |
| <b>Retrain - test-out-90</b>          |        |        |        |        |        |
| Single model<br>(Shannon Entropy)     | 78.47% | 62.08% | 63.58% | 85.57% | 55.20% |
| Ensemble-1<br>(Epistemic uncertainty) | 80.20% | 65.25% | 67.29% | 84.06% | 54.45% |
| Ensemble-1<br>(Total uncertainty)     | 79.46% | 63.66% | 69.08% | 84.49% | 59.38% |

which are quite a few considering the number of samples needed for a DL model to learn, allows the model to acquire that information, improving its ability to appropriately classify other samples. Furthermore, these results demonstrate that a small number of samples with relevant information can have an impact on model performance, suggesting the possibility of training models with less data as long as they provide valuable insights. One possible way to identify such data, as observed, is through the uncertainty values associated with its classification. The method with the highest performance improvement was the Shannon entropy in the single model. This is due to the fact that this model is less robust than Ensemble-1, so any new valuable information learned has a greater impact on the performance of the single model.

These conclusions are supported through the results served as a control, where the samples selected are random. Observing the second graphic of the Figure 6.11, the retrained models have similar performance as the original models. The small increase in the performance for the retrained models may be due to the fact that even though the samples are randomly selected, they still learn from data that has information that can help improve the models. The maximum difference between the Baseline-test-out-90 and the retrained model is only 2%, less than half of the improvement observed in the

retrained models based on uncertainty.

Lastly, Table 6.4, which displays the micro average F1-score values for each class, reiterates the conclusions drawn. However, it is possible to observe that there is a trade-off in the performance of the classes: while some classes have their performance improved, others have it decreased. This does not occur when applied EU in the active learning approach, where all classes have their performance improved. This confirms the importance of EU as a method to identify samples with relevant information for model improvement.



## CONCLUSIONS AND FUTURE WORK

This chapter concludes this dissertation by providing a summary of the developed methods and achievements of this research. Furthermore, future work is suggested for the proposed strategies.

### 7.1 Conclusions

To make the decision support systems as trustworthy as possible, it is critical to assess the confidence that ML models have in their classifications. UQ has shown to be one of the most effective techniques for that purpose. This work studied these concepts using four large public ECG databases for the classification of cardiac arrhythmias. As multiple cardiac arrhythmias can be presented within the same recording, a multi-label classification setting was adopted for the development of DL models. Regarding the UQ measures, single distribution uncertainty measures and the decomposition of uncertainty using classical information-theoretic measures of entropy by means of ensemble approaches were tested and compared. Thus, three types of DL models were developed, including a single CNN-based model, a model obtained using MC Dropout techniques, and a DE model, known in this work as Ensemble-1.

The performance of these models was assessed for two test sets, where the **test-in** has data from the same database as the training and the **test-out** presents data from a different database. Although these models produced similar performance results for the same test set, the Ensemble-1 model revealed to have a better performance than the other models, which is consistent with the literature. When tested with the **test-out** set, the performance of all the three models decreases significantly, confirmed by the decrease of F1-Score from around 95% to 70% and an AUC-ROC from 97% to 81%. These results indicate the presence of dataset shift since the data from **test-out** has different characteristics and distributions than the data used for training.

Regarding UQ, the Shannon entropy and maximum probability were estimated for the single model as were the aleatoric, epistemic, and total uncertainty for the MC Dropout and Ensemble-1 models. For the single model, Shannon entropy was the method that captured

the highest range of uncertainty in the samples, while for the **MC Dropout** and **Ensemble-1** model was the total uncertainty. This suggests the benefit of estimating uncertainty using the combination of epistemic and aleatoric uncertainty. The **VR** was also applied for the **MC Dropout** and **Ensemble-1** model, having a behavior similar to the epistemic uncertainty in these models, as expected. Additionally, all the uncertainties computed for the **test-out** were significantly higher than for the **test-in** set. This demonstrates that the models are not as confident in the **test-out** classifications as they were for the **test-in** set hence an indication of dataset shift.

In order to improve the trustworthiness of the models, the classification with rejection option was applied, where models can abstain from providing a prediction when there is a large amount of uncertainty for a given sample. For both test sets, the models performance increased with rejection, revealing that the higher the uncertainty in a given classification, higher is the probability of the models to misclassify the samples. Additionally, the uncertainty threshold, selected from the training data, increased from 10 % to a range between 30% to 50% depending on the model or uncertainty measure employed. The increase in rejection rate confirms that high uncertainty was presented in the classification and the uncertainty is higher in the **test-out** set. The **KUE** was applied for the single model by extracting deep features. The results obtained were lower than expected but since this method was designed for single-label, it is necessary to validate the suitability of this method in multi-label data in future work. Lastly, a method for the calculation of an optimal rejection threshold was applied. However, due to the multi-label setting, the obtained thresholds rejected all the samples with an uncertainty higher than 1, rejecting more samples than classifying them. Therefore, further analysis in a multi-label setting should be carry out in future work.

Another alternative to improve the models' performance and reliability is through the active learning approach. This was applied only for the **test-out** and the 10% of samples with the highest uncertainty were selected. The total and epistemic uncertainty were used in **Ensemble-1** and the Shannon entropy for the single model. This strategy was also applied to 10% of random samples of the **test-out** to serve as baseline. The results showed that the models improved their performance by almost 5% when using uncertainty versus 2% when using a random selection. These results demonstrate that data with high uncertainty has information that the model has not yet learned and hence the models benefit from the retraining with this selection method.

To conclude, the results of this work suggest that **UQ** should be considered a key feature of any **ML** model as a safety mechanism. It is also possible to infer the role of uncertainty as a valuable method under dataset shift conditions and in strategies such classification with rejection option and active learning approaches. Since data with different characteristics and distributions from those learnt by the **ML** models will always exist, predictions with the employed methods produce safer models to implement as a decision support system in clinical settings.

## 7.2 Future Work

Although the preliminary work revealed promising results, this dissertation possesses a few limitations which should be address in future research.

Firstly, there is a lack of available literature devoted to the estimation of uncertainty in a multi-label setting. However, in a ECG recording more than one cardiac arrythmia can be presented and, thus, it is critical to conduct a more in-depth investigation on this topic for a trustworthy representation of uncertainty in ML models.

Model calibration was not addressed in this work. However, the probabilities obtained through DL models are usually not calibrated, which results in the probability values being either too low or too high for each class. Therefore, it is essential to employ calibration methods in future work, which will allow the probabilities to be better distributed so that the models are not overly confident in its classifications, and, consequently, the results obtained in UQ approaches more reliable.

The CNN's employed for the MC Dropout and DE models have their classifications' uncertainties calculated using classical information-theoric measures. However, several works in the literature have shown the potential of BNN as a source of uncertainty since aleatoric and epistemic uncertainties can be estimated by setting a distribution across the BNN's weights and outputs. Thus, it would be interesting to analyse and validate the behaviour of this model in the classification with rejection option and in the active learning method approached.

The role of uncertainty as a safety mechanism has been widely explored in DL models, however, such research has not been applied to traditional ML models. It would be relevant to apply the same approaches presented in this work to traditional models and compare the results, including the generalization ability when submitted to dataset shift.

Finally, despite the promising results obtained when applied the rejection and active learning approaches, there are still no viable methods for obtaining optimal uncertainty thresholds. It is essential, therefore, the development of methods to obtain these thresholds, especially considering multi-label data for that purpose.

## BIBLIOGRAPHY

- [1] J. M. Lourenço, *The NOVAthesis L<sup>A</sup>T<sub>E</sub>X Template User's Manual*, NOVA University Lisbon, 2021. [Online]. Available: <https://github.com/joaomlourenco/novathesis/raw/master/template.pdf> (cit. on p. ii).
- [2] M. Chen and M. Decary, "Artificial intelligence in healthcare: An essential guide for health leaders", *Healthcare Management Forum*, vol. 33, no. 1, pp. 10–18, 2020, PMID: 31550922. DOI: [10.1177/0840470419873123](https://doi.org/10.1177/0840470419873123) (cit. on p. 1).
- [3] K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. [Online]. Available: [probml.ai](http://probml.ai) (cit. on pp. 4, 6).
- [4] E. T. Jaynes, *Probability theory: The logic of science*. Cambridge university press, 2003, ISBN: 978-0521592710 (cit. on p. 4).
- [5] H. Kobayashi, B. L. Mark, and W. Turin, *Probability, random processes, and statistical analysis: applications to communications, signal processing, queueing theory and mathematical finance*. Cambridge University Press, 2011, ISBN: 978-0521895446 (cit. on p. 4).
- [6] J. D. Malley, J. Kruppa, A. Dasgupta, K. G. Malley, and A. Ziegler, "Probability machines", *Methods of information in medicine*, vol. 51, no. 01, 2012. DOI: [10.3414/ME00-01-0052](https://doi.org/10.3414/ME00-01-0052) (cit. on p. 4).
- [7] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006, ISBN: 978-0387-31073-2 (cit. on p. 4).
- [8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org> (cit. on p. 4).
- [9] A. Maleki and T. Do, "Review of probability theory", *CS*, vol. 229, no. 2, 2000 (cit. on p. 4).
- [10] B. A. Olshausen, "Bayesian probability theory", *The Redwood Center for Theoretical Neuroscience, Helen Wills Neuroscience Institute at the University of California at Berkeley, Berkeley, CA*, 2004 (cit. on p. 5).

- [11] Encyclopaedia Britannica, Inc, Ed., *The New Encyclopaedia Britannica*, 15th ed. Chicago: Encyclopaedia Britannica, 2007, ISBN: 9781593392925 (cit. on p. 5).
- [12] M. Al-Saleh and F. A. Masoud, "A note on the posterior expected loss as a measure of accuracy in bayesian methods", *Applied Mathematics and Computation*, vol. 134, no. 2, pp. 507–514, 2003, ISSN: 0096-3003. DOI: [https://doi.org/10.1016/S0096-3003\(01\)00298-3](https://doi.org/10.1016/S0096-3003(01)00298-3) (cit. on p. 6).
- [13] S. Weijs, G. v. Schoups, and N. Van De Giesen, "Why hydrological predictions should be evaluated using information theory", *Hydrology and Earth System Sciences*, vol. 14, no. 12, pp. 2545–2558, 2010. DOI: <https://doi.org/10.5194/hess-14-2545-2010> (cit. on p. 6).
- [14] K. Sayood, "Information theory and cognition: A review", *Entropy*, vol. 20, no. 9, p. 706, 2018. DOI: <https://doi.org/10.3390/e20090706> (cit. on pp. 6, 7).
- [15] A. Mohammad-Djafari, "Entropy, information theory, information geometry and bayesian inference in data, signal and image processing and inverse problems", *Entropy*, vol. 17, no. 6, pp. 3989–4027, 2015. DOI: <https://doi.org/10.3390/e17063989> (cit. on p. 6).
- [16] S. Keshmiri, "Conditional entropy: A potential digital marker for stress", *Entropy*, vol. 23, no. 3, p. 286, 2021. DOI: <https://doi.org/10.3390/e23030286> (cit. on pp. 6, 7).
- [17] M. Thomas and A. T. Joy, *Elements of information theory*. Wiley-Interscience, 2006, ISBN: 978-0471241959 (cit. on pp. 7, 16).
- [18] T. Mitchell, "Machine learning", 1997 (cit. on p. 7).
- [19] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012, ISBN: 978-0262018029 (cit. on pp. 7, 9).
- [20] Z.-H. Zhou, "A brief introduction to weakly supervised learning", *National science review*, vol. 5, no. 1, pp. 44–53, 2018. DOI: <https://doi.org/10.1093/nsr/nwx106> (cit. on p. 7).
- [21] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 132–149. DOI: <https://doi.org/10.48550/arXiv.1807.05520> (cit. on p. 7).
- [22] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey", *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996. DOI: <https://doi.org/10.1613/jair.301> (cit. on p. 7).
- [23] N. Chumerin and M. M. Van Hulle, "Comparison of two feature extraction methods based on maximization of mutual information", in *2006 16th IEEE signal processing society workshop on machine learning for signal processing*, IEEE, 2006, pp. 343–348. DOI: [10.1109/MLSP.2006.275572](https://doi.org/10.1109/MLSP.2006.275572) (cit. on p. 8).

- [24] J. Alzubi, A. Nayyar, and A. Kumar, "Machine learning from theory to algorithms: An overview", in *Journal of physics: conference series*, IOP Publishing, vol. 1142, 2018, p. 012012. DOI: <https://doi.org/10.1088/1742-6596/1142/1/012012> (cit. on p. 8).
- [25] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2019, ISBN: 9781492032595 (cit. on p. 9).
- [26] Š. Raudys, "Evolution and generalization of a single neurone: I. single-layer perceptron as seven statistical classifiers", *Neural Networks*, vol. 11, no. 2, pp. 283–296, 1998. DOI: [https://doi.org/10.1016/S0893-6080\(97\)00135-4](https://doi.org/10.1016/S0893-6080(97)00135-4) (cit. on p. 9).
- [27] D. J. d. S. A. Belo, "Learning biosignals with deep learning", 2021. [Online]. Available: <http://hdl.handle.net/10362/126518> (cit. on pp. 9, 11).
- [28] M. A. Wani, F. A. Bhat, S. Afzal, and A. I. Khan, *Advances in deep learning*. Springer, 2020, ISBN: 978-981-13-6794-6 (cit. on pp. 9–11).
- [29] W. Qingjie and W. WenBin, "Research on image retrieval using deep convolutional neural network combining l1 regularization and prelu activation function", in *IOP Conference Series: Earth and Environmental Science*, IOP Publishing, vol. 69, 2017. DOI: <https://doi.org/10.1088/1755-1315/69/1/012156> (cit. on p. 9).
- [30] B. Ding, H. Qian, and J. Zhou, "Activation functions and their characteristics in deep neural networks", in *2018 Chinese Control And Decision Conference (CCDC)*, 2018, pp. 1836–1841. DOI: [10.1109/CCDC.2018.8407425](https://doi.org/10.1109/CCDC.2018.8407425) (cit. on p. 10).
- [31] F. Q. Lauzon, "An introduction to deep learning", in *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, IEEE, 2012, pp. 1438–1439. DOI: [10.1109/ISSPA.2012.6310529](https://doi.org/10.1109/ISSPA.2012.6310529) (cit. on p. 10).
- [32] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", *nature*, vol. 521, no. 7553, pp. 436–444, 2015. DOI: <https://doi.org/10.1038/nature14539> (cit. on p. 10).
- [33] F. Herrera, F. Charte, A. J. Rivera, and M. J. d. Jesus, "Multilabel classification", in *Multilabel Classification*, Springer, 2016, pp. 17–31. DOI: [10.1007/978-3-319-41111-8\\_2](https://doi.org/10.1007/978-3-319-41111-8_2) (cit. on p. 11).
- [34] E. A. Cherman, M. C. Monard, and J. Metz, "Multi-label Problem Transformation Methods: a Case Study", *CLEI Electronic Journal*, vol. 14, pp. 4–4, 2011-04, ISSN: 0717-5000 (cit. on p. 12).
- [35] S. Hong, W. Zhang, C. Sun, Y. Zhou, and H. Li, "Practical lessons on 12-lead ecg classification: Meta-analysis of methods from physionet/computing in cardiology challenge 2020", *Frontiers in Physiology*, p. 2505, 2022. DOI: <https://doi.org/10.3389/fphys.2021.811661> (cit. on p. 12).

- [36] J. Read, A. Puurula, and A. Bifet, "Multi-label classification with meta-labels", in *2014 IEEE International Conference on Data Mining*, 2014, pp. 941–946. DOI: [10.1109/ICDM.2014.38](https://doi.org/10.1109/ICDM.2014.38) (cit. on p. 12).
- [37] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification", *Pattern recognition*, vol. 37, no. 9, pp. 1757–1771, 2004. DOI: <https://doi.org/10.1016/j.patcog.2004.03.009> (cit. on p. 12).
- [38] H. Jabbar and R. Z. Khan, "Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study)", *Computer Science, Communication and Instrumentation Devices*, vol. 70, 2015 (cit. on p. 12).
- [39] I. Tsamardinos, E. Greasidou, and G. Borboudakis, "Bootstrapping the out-of-sample predictions for efficient and accurate cross-validation", *Machine Learning*, vol. 107, no. 12, pp. 1895–1922, 2018. DOI: <https://doi.org/10.1007/s10994-018-5714-4> (cit. on p. 13).
- [40] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations", *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015. DOI: [10.5121/ijdkp.2015.5201](https://doi.org/10.5121/ijdkp.2015.5201) (cit. on p. 13).
- [41] A. Luque, A. Carrasco, A. Martín, and A. de Las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix", *Pattern Recognition*, vol. 91, pp. 216–231, 2019. DOI: <https://doi.org/10.1016/j.patcog.2019.02.023> (cit. on p. 13).
- [42] L. Gonçalves, A. Subtil, M. R. Oliveira, and P. de Zea Bermudez, "ROC Curve Estimation: An Overview", *REVSTAT-Statistical Journal*, vol. 12, no. 1, pp. 1–20, 2014-04. DOI: [10.57805/revstat.v12i1.141](https://doi.org/10.57805/revstat.v12i1.141). [Online]. Available: <https://revstat.ine.pt/index.php/REVSTAT/article/view/141> (cit. on p. 15).
- [43] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves", in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240. DOI: <https://doi.org/10.1145/1143844.1143874> (cit. on p. 15).
- [44] K. Boyd, K. H. Eng, and C. D. Page, "Area under the precision-recall curve: Point estimates and confidence intervals", in *Joint European conference on machine learning and knowledge discovery in databases*, Springer, 2013, pp. 451–466. DOI: [https://doi.org/10.1007/978-3-642-40994-3\\_29](https://doi.org/10.1007/978-3-642-40994-3_29) (cit. on pp. 15, 16).
- [45] M. H. Shaker and E. Hüllermeier, "Aleatoric and epistemic uncertainty with random forests", in *International Symposium on Intelligent Data Analysis*, Springer, 2020, pp. 444–456. DOI: [https://doi.org/10.1007/978-3-030-44584-3\\_35](https://doi.org/10.1007/978-3-030-44584-3_35) (cit. on p. 16).



- [46] M. Barandas, D. Folgado, R. Santos, R. Simão, and H. Gamboa, “Uncertainty-based rejection in machine learning: Implications for model development and interpretability”, *Electronics*, vol. 11, no. 3, 2022, ISSN: 2079-9292. DOI: [10.3390/electronics11030396](https://doi.org/10.3390/electronics11030396). [Online]. Available: <https://www.mdpi.com/2079-9292/11/3/396> (cit. on pp. 16–18, 23, 43, 44, 55).
- [47] M. S. Ayhan and P. Berens, “Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks”, in *Medical Imaging with Deep Learning*, 2018. [Online]. Available: <https://openreview.net/forum?id=rJZz-knjz> (cit. on p. 16).
- [48] B. Merz and A. H. Thielen, “Separating natural and epistemic uncertainty in flood frequency analysis”, *Journal of Hydrology*, vol. 309, no. 1-4, pp. 114–132, 2005. DOI: <https://doi.org/10.1016/j.jhydrol.2004.11.015> (cit. on p. 16).
- [49] L. Mi, H. Wang, Y. Tian, and N. Shavit, “Training-free uncertainty estimation for neural networks”, 2019-09 (cit. on p. 17).
- [50] N. Ståhl, G. Falkman, A. Karlsson, and G. Mathiason, “Evaluation of uncertainty quantification in deep learning”, pp. 556–568, 2020. DOI: [https://doi.org/10.1007/978-3-030-50146-4\\_41](https://doi.org/10.1007/978-3-030-50146-4_41) (cit. on p. 17).
- [51] J. Caldeira and B. Nord, “Deeply uncertain: Comparing methods of uncertainty quantification in deep learning algorithms”, *Machine Learning: Science and Technology*, vol. 2, no. 1, p. 015002, 2020. DOI: <https://doi.org/10.1088/2632-2153/aba6f3> (cit. on pp. 17, 22).
- [52] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors”, *arXiv preprint arXiv:1207.0580*, 2012. DOI: <https://doi.org/10.48550/arXiv.1207.0580> (cit. on p. 18).
- [53] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”, in *international conference on machine learning*, PMLR, 2016, pp. 1050–1059. [Online]. Available: <https://proceedings.mlr.press/v48/gal16.html> (cit. on p. 18).
- [54] S. Depeweg, J.-M. Hernandez-Lobato, F. Doshi-Velez, and S. Udluft, “Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning”, in *Proceedings of the 35th International Conference on Machine Learning*, J. Dy and A. Krause, Eds., ser. Proceedings of Machine Learning Research, vol. 80, PMLR, 2018-07, pp. 1184–1193. [Online]. Available: <https://proceedings.mlr.press/v80/depeweg18a.html> (cit. on p. 18).
- [55] A. Malinin, B. Mlodozieniec, and M. Gales, “Ensemble distribution distillation”, *arXiv preprint arXiv:1905.00076*, 2019. DOI: <https://doi.org/10.17863/CAM.49348> (cit. on pp. 18, 22).



- [56] B. Hanczar and M. Sebag, “Combination of one-class support vector machines for classification with reject option”, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2014, pp. 547–562. DOI: [https://doi.org/10.1007/978-3-662-44848-9\\_35](https://doi.org/10.1007/978-3-662-44848-9_35) (cit. on p. 18).
- [57] H. Mouchere and E. Anquetil, “A unified strategy to deal with different natures of reject”, in *18th International Conference on Pattern Recognition (ICPR’06)*, IEEE, vol. 2, 2006, pp. 792–795. DOI: [10.1109/ICPR.2006.193](https://doi.org/10.1109/ICPR.2006.193) (cit. on p. 18).
- [58] E. Hüllermeier and W. Waegeman, “Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods”, *Machine Learning*, vol. 110, no. 3, pp. 457–506, 2021. DOI: <https://doi.org/10.1007/s10994-021-05946-3> (cit. on pp. 18, 44).
- [59] M. S. A. Nadeem, J.-D. Zucker, and B. Hanczar, “Accuracy-rejection curves (arcs) for comparing classification methods with a reject option”, in *Proceedings of the third International Workshop on Machine Learning in Systems Biology*, S. Džeroski, P. Guerts, and J. Rousu, Eds., ser. Proceedings of Machine Learning Research, vol. 8, Ljubljana, Slovenia: PMLR, 2009-09, pp. 65–81. [Online]. Available: <https://proceedings.mlr.press/v8/nadeem10a.html> (cit. on p. 18).
- [60] F. Condessa, J. Bioucas-Dias, and J. Kovačević, “Performance measures for classification systems with rejection”, *Pattern Recognition*, vol. 63, pp. 437–450, 2017, ISSN: 0031-3203. DOI: <https://doi.org/10.1016/j.patcog.2016.10.011> (cit. on p. 19).
- [61] B. Settles, “Active learning literature survey”, University of Wisconsin–Madison, Computer Sciences Technical Report 1648, 2009. [Online]. Available: <http://digital.library.wisc.edu/1793/60660> (cit. on p. 19).
- [62] Y. Gal *et al.*, “Uncertainty in deep learning”, 2016 (cit. on pp. 19, 22, 44).
- [63] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset shift in machine learning* (Neural Information Processing series). Mit Press, 2008, ISBN: 9780262170055 (cit. on p. 20).
- [64] A. Malinin *et al.*, “Shifts: A dataset of real distributional shift across multiple large-scale tasks”, *arXiv preprint arXiv:2107.07455*, 2021. DOI: <https://doi.org/10.48550/arXiv.2107.07455> (cit. on pp. 20, 24).
- [65] P. W. Koh *et al.*, “Wilds: A benchmark of in-the-wild distribution shifts”, *Proceedings of Machine Learning Research*, vol. 139, M. Meila and T. Zhang, Eds., pp. 5637–5664, 2021-07. [Online]. Available: <https://proceedings.mlr.press/v139/koh21a.html> (cit. on p. 20).
- [66] R. Michelmore, M. Kwiatkowska, and Y. Gal, “Evaluating uncertainty quantification in end-to-end autonomous driving control”, *arXiv preprint arXiv:1811.06817*, 2018. DOI: <https://doi.org/10.48550/arXiv.1811.06817> (cit. on p. 21).

- [67] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?", vol. 30, I. Guyon *et al.*, Eds., 2017. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/file/2650d6089a6d640c5e85b2b88265dc2b-Paper.pdf> (cit. on p. 21).
- [68] G. Li, L. Yang, C.-G. Lee, X. Wang, and M. Rong, "A bayesian deep learning rul framework integrating epistemic and aleatoric uncertainties", *IEEE Transactions on Industrial Electronics*, vol. 68, no. 9, pp. 8829–8841, 2021. DOI: [10.1109/TIE.2020.3009593](https://doi.org/10.1109/TIE.2020.3009593) (cit. on p. 21).
- [69] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks", vol. 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018. [Online]. Available: <https://proceedings.neurips.cc/paper/2018/file/3ea2db50e62ceefceaf70a9d9a56a6f4-Paper.pdf> (cit. on p. 21).
- [70] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting", *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014 (cit. on p. 21).
- [71] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles", vol. 30, I. Guyon *et al.*, Eds., 2017. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/file/9ef2ed4b7fd2c810847ffa5fa85bce38-Paper.pdf> (cit. on p. 22).
- [72] G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, and T. Vercauteren, "Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks", *Neurocomputing*, vol. 338, pp. 34–45, 2019. DOI: <https://doi.org/10.1016/j.neucom.2019.01.103> (cit. on p. 22).
- [73] R. Stoean, C. Stoean, M. Atencia, R. Rodríguez-Labrada, and G. Joya, "Ranking information extracted from uncertainty quantification of the prediction of a deep learning model on medical time series data", *Mathematics*, vol. 8, no. 7, p. 1078, 2020. DOI: <https://doi.org/10.3390/math8071078> (cit. on p. 22).
- [74] C. Chow, "On optimum recognition error and reject tradeoff", *IEEE Transactions on Information Theory*, vol. 16, no. 1, pp. 41–46, 1970. DOI: [10.1109/TIT.1970.1054406](https://doi.org/10.1109/TIT.1970.1054406) (cit. on p. 23).
- [75] J. Mena, O. Pujol, and J. Vitrià, "Uncertainty-based rejection wrappers for black-box classifiers", *IEEE Access*, vol. 8, pp. 101 721–101 746, 2020. DOI: [10.1109/ACCESS.2020.2996495](https://doi.org/10.1109/ACCESS.2020.2996495) (cit. on pp. 23, 43).
- [76] M. H. Shaker and E. Hüllermeier, "Ensemble-based uncertainty quantification: Bayesian versus credal inference", in *PROCEEDINGS 31. WORKSHOP COMPUTATIONAL INTELLIGENCE*, vol. 25, 2021, p. 63, ISBN: 978-3-7315-1131-1. DOI: [10.5445/KSP/1000138532](https://doi.org/10.5445/KSP/1000138532) (cit. on p. 23).

- [77] D. Dua and C. Graff, *UCI machine learning repository*, 2017. [Online]. Available: <http://archive.ics.uci.edu/ml> (cit. on pp. 23, 26).
- [78] I. Pillai, G. Fumera, and F. Roli, “Multi-label classification with a reject option”, *Pattern Recognition*, vol. 46, no. 8, pp. 2256–2266, 2013. DOI: <https://doi.org/10.1016/j.patcog.2013.01.035> (cit. on p. 23).
- [79] Y. Ovadia *et al.*, “Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift”, *Advances in neural information processing systems*, vol. 32, 2019 (cit. on p. 24).
- [80] K. Stacke, G. Eilertsen, J. Unger, and C. Lundström, “Measuring domain shift for deep learning in histopathology”, *IEEE journal of biomedical and health informatics*, vol. 25, no. 2, pp. 325–336, 2020. DOI: [10.1109/JBHI.2020.3032060](https://doi.org/10.1109/JBHI.2020.3032060) (cit. on p. 24).
- [81] E. H. Pooch, P. Ballester, and R. C. Barros, “Can we trust deep learning based diagnosis? the impact of domain shift in chest radiograph classification”, in *International Workshop on Thoracic Image Analysis*, Springer, 2020, pp. 74–83. DOI: <https://doi.org/10.48550/arXiv.1909.01940> (cit. on p. 24).
- [82] Y. Gal, R. Islam, and Z. Ghahramani, “Deep Bayesian active learning with image data”, in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., ser. Proceedings of Machine Learning Research, vol. 70, PMLR, 2017-08, pp. 1183–1192. [Online]. Available: <https://proceedings.mlr.press/v70/gal17a.html> (cit. on p. 25).
- [83] A. Sadafi *et al.*, “Multiclass deep active learning for detecting red blood cell subtypes in brightfield microscopy”, in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2019, pp. 685–693. DOI: [https://doi.org/10.1007/978-3-030-32239-7\\_76](https://doi.org/10.1007/978-3-030-32239-7_76) (cit. on p. 25).
- [84] V.-L. Nguyen, M. H. Shaker, and E. Hüllermeier, “How to measure uncertainty in uncertainty sampling for active learning”, *Machine Learning*, vol. 111, no. 1, pp. 89–122, 2022. DOI: <https://doi.org/10.1007/s10994-021-06003-9> (cit. on p. 26).
- [85] Z. Sun, C. Wang, Y. Zhao, and C. Yan, “Multi-label ecg signal classification based on ensemble classifier”, *IEEE Access*, vol. 8, pp. 117 986–117 996, 2020. DOI: [10.1109/ACCESS.2020.3004908](https://doi.org/10.1109/ACCESS.2020.3004908) (cit. on p. 26).
- [86] A. H. Ribeiro *et al.*, “Automatic diagnosis of the 12-lead ecg using a deep neural network”, *Nature communications*, vol. 11, no. 1, pp. 1–9, 2020. DOI: <https://doi.org/10.1038/s41467-020-15432-4> (cit. on p. 26).
- [87] Y. Li, Z. Zhang, F. Zhou, Y. Xing, J. Li, and C. Liu, “Multi-label classification of arrhythmia for long-term electrocardiogram signals with feature learning”, *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021. DOI: [10.1109/TIM.2021.3077667](https://doi.org/10.1109/TIM.2021.3077667) (cit. on p. 26).

- [88] H. Zhu *et al.*, “Automatic multilabel electrocardiogram diagnosis of heart rhythm or conduction abnormalities with deep learning: A cohort study”, *The Lancet Digital Health*, vol. 2, no. 7, e348–e357, 2020. DOI: [https://doi.org/10.1016/S2589-7500\(20\)30107-2](https://doi.org/10.1016/S2589-7500(20)30107-2) (cit. on p. 26).
- [89] U. R. Acharya *et al.*, “A deep convolutional neural network model to classify heartbeats”, *Computers in Biology and Medicine*, vol. 89, pp. 389–396, 2017, ISSN: 0010-4825. DOI: <https://doi.org/10.1016/j.combiomed.2017.08.022> (cit. on p. 26).
- [90] Z. He, X. Zhang, Y. Cao, Z. Liu, B. Zhang, and X. Wang, “Litenet: Lightweight neural network for detecting arrhythmias at resource-constrained mobile devices”, *Sensors*, vol. 18, no. 4, 2018, ISSN: 1424-8220. DOI: [10.3390/s18041229](https://doi.org/10.3390/s18041229). [Online]. Available: <https://www.mdpi.com/1424-8220/18/4/1229> (cit. on p. 26).
- [91] T. J. Jun, H. M. Nguyen, D. Kang, D. Kim, D. Kim, and Y.-H. Kim, “Ecg arrhythmia classification using a 2-d convolutional neural network”, *arXiv preprint arXiv:1804.06812*, 2018. DOI: <https://doi.org/10.48550/arXiv.1804.06812> (cit. on p. 26).
- [92] J. F. Vranken *et al.*, “Uncertainty estimation for deep learning-based automated analysis of 12-lead electrocardiograms”, *European Heart Journal-Digital Health*, vol. 2, no. 3, pp. 401–415, 2021. DOI: <https://doi.org/10.1093/ehjdh/ztab045> (cit. on p. 26).
- [93] A. O. Aseeri, “Uncertainty-aware deep learning-based cardiac arrhythmias classification model of electrocardiogram signals”, *Computers*, vol. 10, no. 6, p. 82, 2021. DOI: <https://doi.org/10.3390/computers10060082> (cit. on p. 27).
- [94] W. Zhang, X. Di, G. Wei, S. Geng, Z. Fu, and S. Hong, “A deep bayesian neural network for cardiac arrhythmia classification with rejection from ecg recordings”, *arXiv preprint arXiv:2203.00512*, 2022. DOI: <https://doi.org/10.48550/arXiv.2203.00512> (cit. on p. 27).
- [95] X. Xie, H. Liu, D. Chen, M. Shu, and Y. Wang, “Multilabel 12-lead ecg classification based on leadwise grouping multibranch network”, *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022. DOI: [10.1109/TIM.2022.3164141](https://doi.org/10.1109/TIM.2022.3164141) (cit. on p. 27).
- [96] B. G. Katzung and A. J. Trevor, “Basic & clinical pharmacology”, 2012 (cit. on p. 29).
- [97] L. S. Costanzo, *Physiology, E-Book*. Elsevier Health Sciences, 2013 (cit. on pp. 28, 29).
- [98] A. Luthra, *ECG made easy*. Jaypee Brothers Medical Publishers, 2019 (cit. on pp. 28, 31).
- [99] J. G. Betts *et al.*, *Anatomy and physiology*. 2013 (cit. on p. 28).

## BIBLIOGRAPHY

---

- [100] M. R. Neuman and J. G. Webster, "Chapter 6: Biopotential amplifiers", *Medical Instrumentation Application and Design 4th Edition*, Wiley, pp. 341–391, 2009 (cit. on p. 30).
- [101] D. M. Mirvis and A. L. Goldberger, "Electrocardiography", *Heart disease*, vol. 1, pp. 82–128, 2001 (cit. on pp. 30, 31).
- [102] J. Loscalzo, *Harrison's cardiovascular medicine 2/E*. McGraw-Hill Education, 2013 (cit. on p. 31).
- [103] M. AlGhatrif and J. Lindsay, "A brief review: History to understand fundamentals of electrocardiography", *Journal of Community Hospital Internal Medicine Perspectives*, vol. 2, no. 1, p. 14383, 2012. DOI: [10.3402/jchimp.v2i1.14383](https://doi.org/10.3402/jchimp.v2i1.14383) (cit. on p. 31).
- [104] A. Goldberger, *Goldberger's Clinical Electrocardiography*. Elsevier, 2018 (cit. on pp. 31, 32).
- [105] A. Pyysing, "Movement artifacts in electrocardiography", English, pp. 74+7, 2018. [Online]. Available: <http://urn.fi/URN:NBN:fi:aalto-201802231638> (cit. on p. 32).
- [106] D. H. Bennett, *Bennett's Cardiac Arrhythmias: Practical notes on interpretation and treatment*. John Wiley & Sons, 2012 (cit. on pp. 32–36).
- [107] A. L. Wit, P. A. Boyden, M. E. Josephson, and H. J. Wellens, *Electrophysiological Foundations of Cardiac Arrhythmias: A Bridge Between Basic Mechanisms and Clinical Electrophysiology*. Cardiotext Publishing, 2020 (cit. on p. 33).
- [108] E. A. P. Alday *et al.*, "Classification of 12-lead ecgs: The physionet/computing in cardiology challenge 2020", *Physiological measurement*, vol. 41, no. 12, p. 124003, 2020. DOI: <https://doi.org/10.1088/1361-6579/abc960> (cit. on pp. 36, 37).
- [109] F. Liu *et al.*, "An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection", *Journal of Medical Imaging and Health Informatics*, vol. 8, no. 7, pp. 1368–1373, 2018. DOI: <https://doi.org/10.1166/jmih.2018.2442> (cit. on p. 36).
- [110] T.-M. Chen, C.-H. Huang, E. S. Shih, Y.-F. Hu, and M.-J. Hwang, "Detection and classification of cardiac arrhythmias by a challenge-best deep learning neural network model", *Iscience*, vol. 23, no. 3, p. 100886, 2020. DOI: <https://doi.org/10.1016/j.isci.2020.100886> (cit. on p. 38).
- [111] A. Velayudhan and S. Peter, "Noise analysis and different denoising techniques of ecg signal-a survey", *IOSR journal of electronics and communication engineering*, vol. 1, no. 1, pp. 40–44, 2016 (cit. on p. 39).
- [112] B. Tutuko *et al.*, "Afibnet: An implementation of atrial fibrillation detection with convolutional neural network", *BMC Medical Informatics and Decision Making*, vol. 21, no. 1, pp. 1–17, 2021. DOI: <https://doi.org/10.1186/s12911-021-01571-1> (cit. on p. 41).

- [113] R. Atallah and A. Al-Mousa, "Heart disease detection using machine learning majority voting ensemble method", in *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, 2019, pp. 1–6. DOI: [10.1109/ICTCS.2019.8923053](https://doi.org/10.1109/ICTCS.2019.8923053) (cit. on p. 47).
- [114] "Chapter 8 - improving the prediction accuracy of heart disease with ensemble learning and majority voting rule", in *U-Healthcare Monitoring Systems*, ser. Advances in Ubiquitous Sensing Applications for Healthcare, N. Dey, A. S. Ashour, S. J. Fong, and S. Borra, Eds., Academic Press, 2019, pp. 179–196. DOI: <https://doi.org/10.1016/B978-0-12-815370-3.00008-6> (cit. on p. 47).
- [115] A. Dogan and D. Birant, "A weighted majority voting ensemble approach for classification", in *2019 4th International Conference on Computer Science and Engineering (UBMK)*, 2019, pp. 1–6. DOI: [10.1109/UBMK.2019.8907028](https://doi.org/10.1109/UBMK.2019.8907028) (cit. on p. 47).
- [116] *Tsaug*. [Online]. Available: <https://tsaug.readthedocs.io/en/stable/> (visited on 2022-08-16) (cit. on p. 51).

## PYTHON LIBRARIES AND MODULES

This Appendix is composed by two tables. The first one (Table A.1) lists and briefly describes the Python libraries employed in this project. Table A.2 provides a brief an overview of some important modules used in different stages of this work.

Table A.1: Python libraries employed in this dissertation

| Library           | Description   |
|-------------------|---|
| Matplotlib        | A collection of Python visualisation tools.   |
| NumPy             | Library of various mathematical functions to operate upon arrays and matrices.  |
| Pandas            | Data analysis and manipulation methods for machine learning algorithms.   |
| Scikit-learn      | Library that provides tools for predictive data analysis such as model fitting, data pre-processing and model evaluation.         |
| Scikit-multilearn | Library that provides methods for multi-label classification.   |
| SciPy             | Collection of mathematical operations for optimization, interpolation, statistics and algebraic equations, among other functions. |
| Seaborn           | Interface that produces informative statistical graphics.   |
| TensorFlow        | High-level library for data processing and the development of machine learning models.  |



Table A.2: Relevant modules used in this project.

| Library                         | Module  | Description  |
|---------------------------------|---|--|
| Scikit-learn                    | preprocessing.StandardScaler.   | Z-score normalization of data.   |
|                                 | metrics.classification_report   | Calculates evaluation metrics over a set of predictions.                         |
|                                 | metrics.f1_score  | F1-Score value over a set of predictions.  |
|                                 | metrics.precision_recall_curve  | Computes the precision-recall curve  |
|                                 | metrics.roc_auc_score   | AUC-ROC value over a set of predictions.   |
|                                 | metrics.roc_curve   | Computes the ROC curve.  |
| Scikit-multilearn               | iterative_train_test_split  | Encoding methods for categorical multi-label data.                               |
| TensorFlow                      | keras.layers.Conv1D   | Computes a convolution kernel over a single temporal dimension.                  |
|                                 | keras.layers.MaxPooling1D   | Downsamples by taking the maximum value over a spatial window.                   |
|                                 | keras.layers.BatchNormalization.  | Normalization to maintain mean close to 0 and the standard deviation close to 1. |
|                                 | keras.layers.Dropout  | Randomly sets units to 0 with a certain frequency during training time.          |
|                                 | keras.layers.Activation   | Applies the rectified linear unit activation function.                           |
|                                 | keras.layers.Flatten  | Removes a dimension.   |
|                                 | keras.layers.Dense  | Applies the operation of a fully-connected neural layer.                         |
| keras.losses.BinaryCrossentropy | Computes the cross-entropy loss between true labels and predicted labels. |  |



## COMPLEMENTARY RESULTS

Table B.1: AUC-ROC per class for both test sets tested in the three developed model.

| Model        | AF     | IABV   | LBAB   | NSR    | RBBB   |
|--------------|--------|--------|--------|--------|--------|
| Test-in set  |        |        |        |        |        |
| Single model | 98.94% | 93.51% | 89.57% | 95.13% | 97.69% |
| MC Dropout   | 98.97% | 87.31% | 87.98% | 93.72% | 88.78% |
| Ensemble-1   | 99.14% | 86.99% | 90.90% | 94.42% | 94.41% |
| Test-out set |        |        |        |        |        |
| Single model | 84.53% | 76.26% | 75.79% | 76.79% | 73.64% |
| MC Dropout   | 82.36% | 72.76% | 73.64% | 78.65% | 70.24% |
| Ensemble-1   | 85.52% | 72.88% | 76.06% | 79.71% | 70.98% |

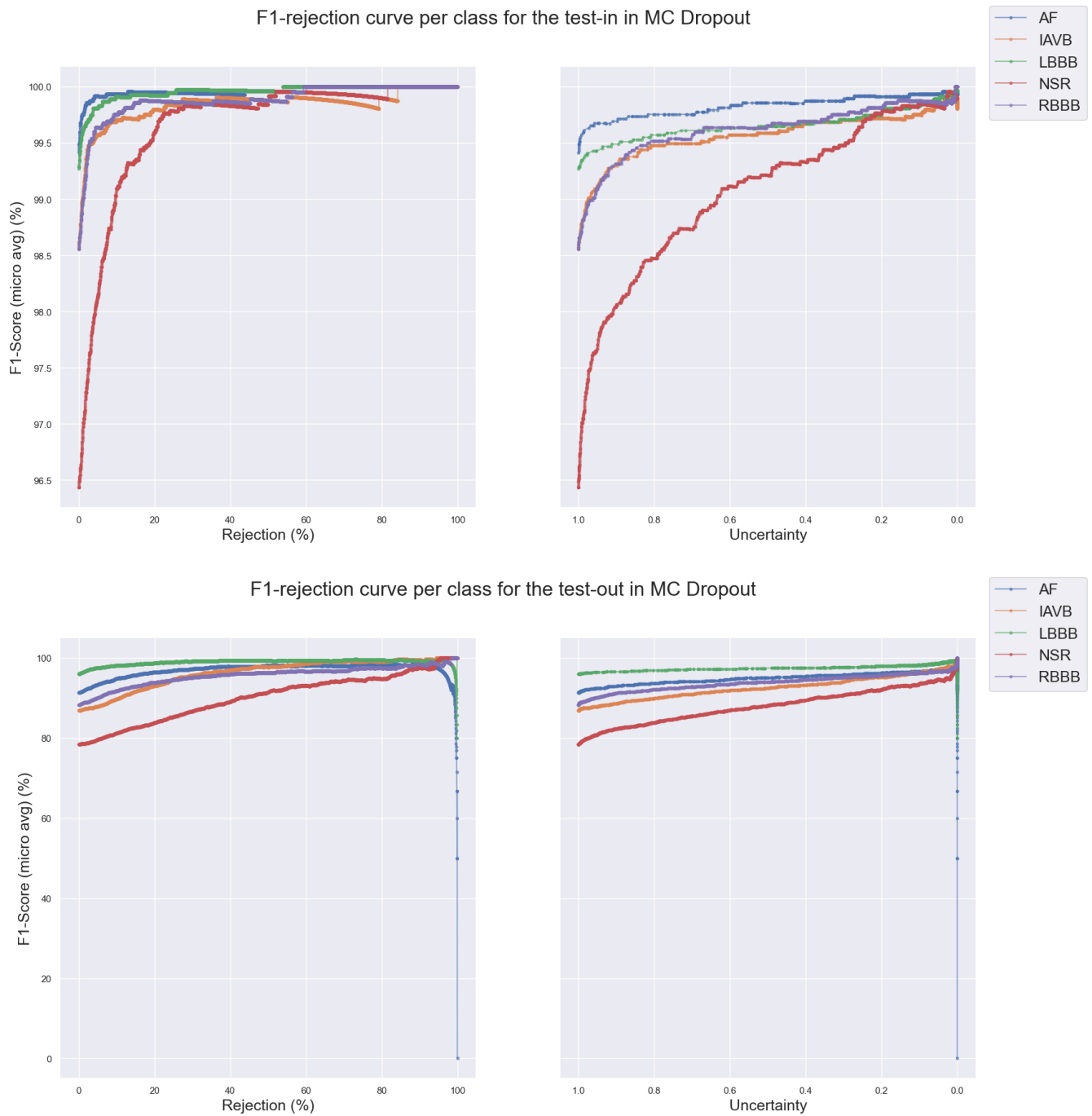


Figure B.1: F1-rejection curve per class for both test sets in the Monte Carlo Dropout model.

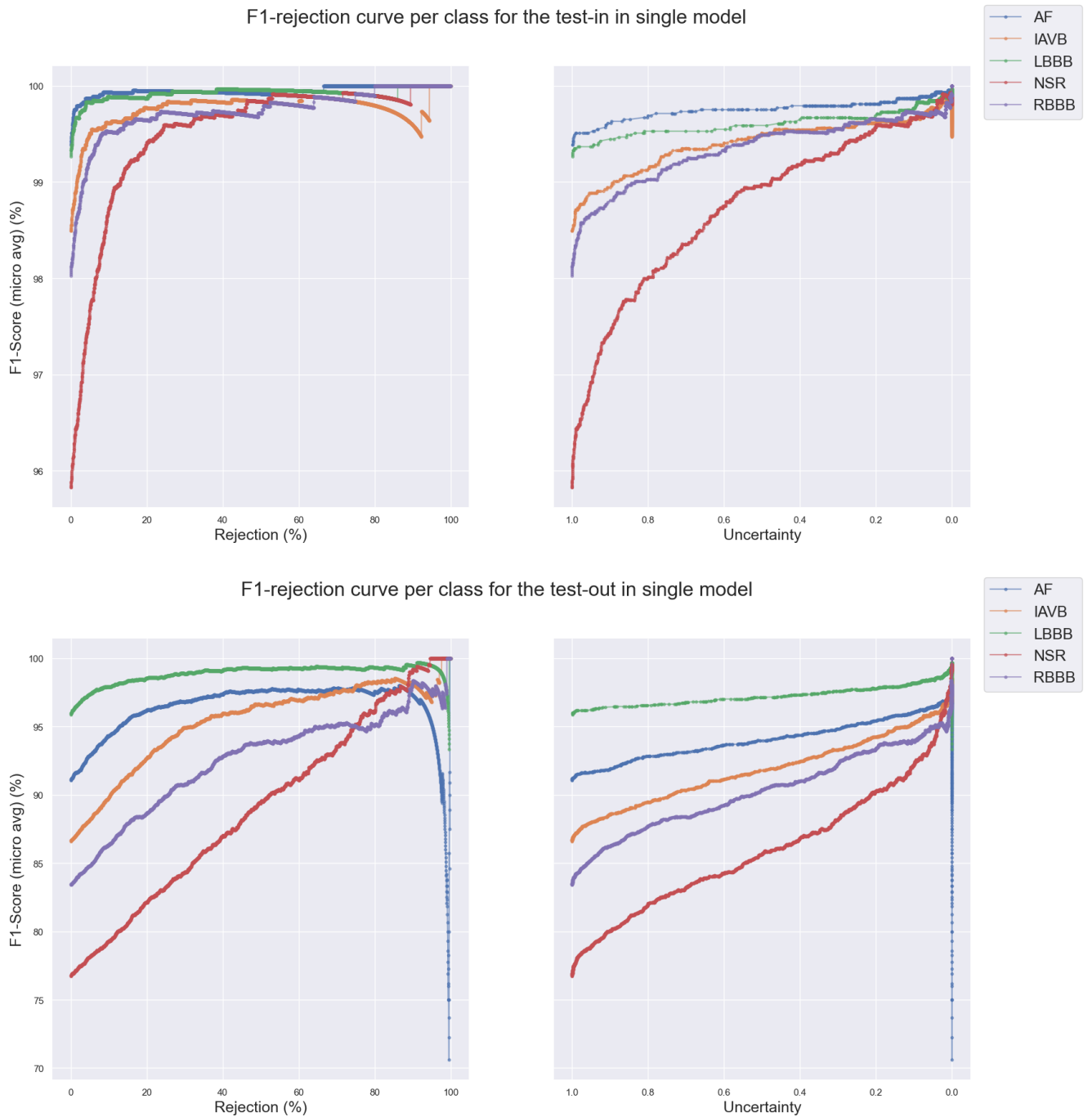


Figure B.2: F1-rejection curve per class for both test sets in the single model.

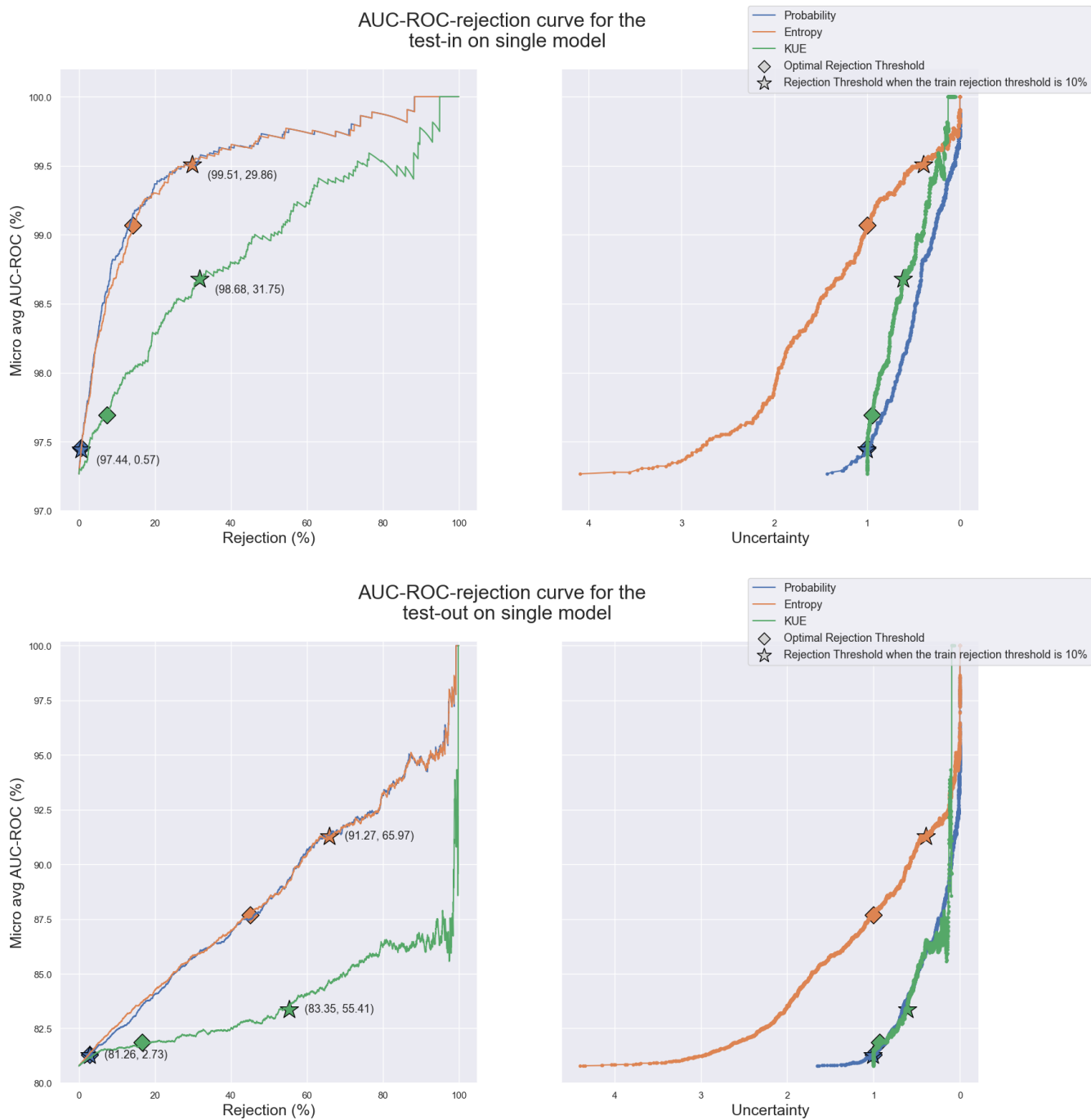


Figure B.3: AUC-ROC-rejection curve for both test sets in the single model.

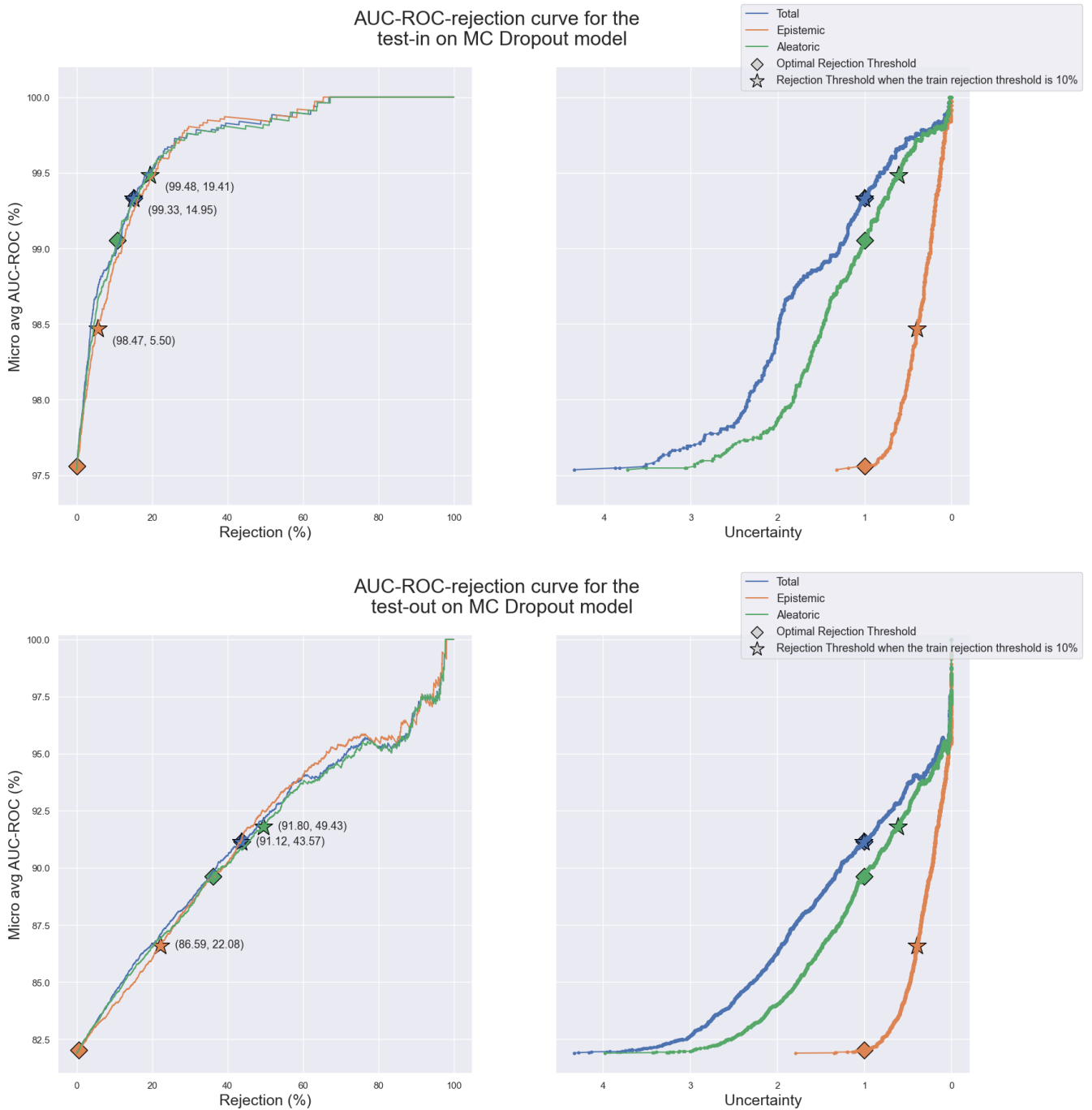


Figure B.4: AUC-ROC-rejection curve for both test sets in the Monte Carlo Dropout model.

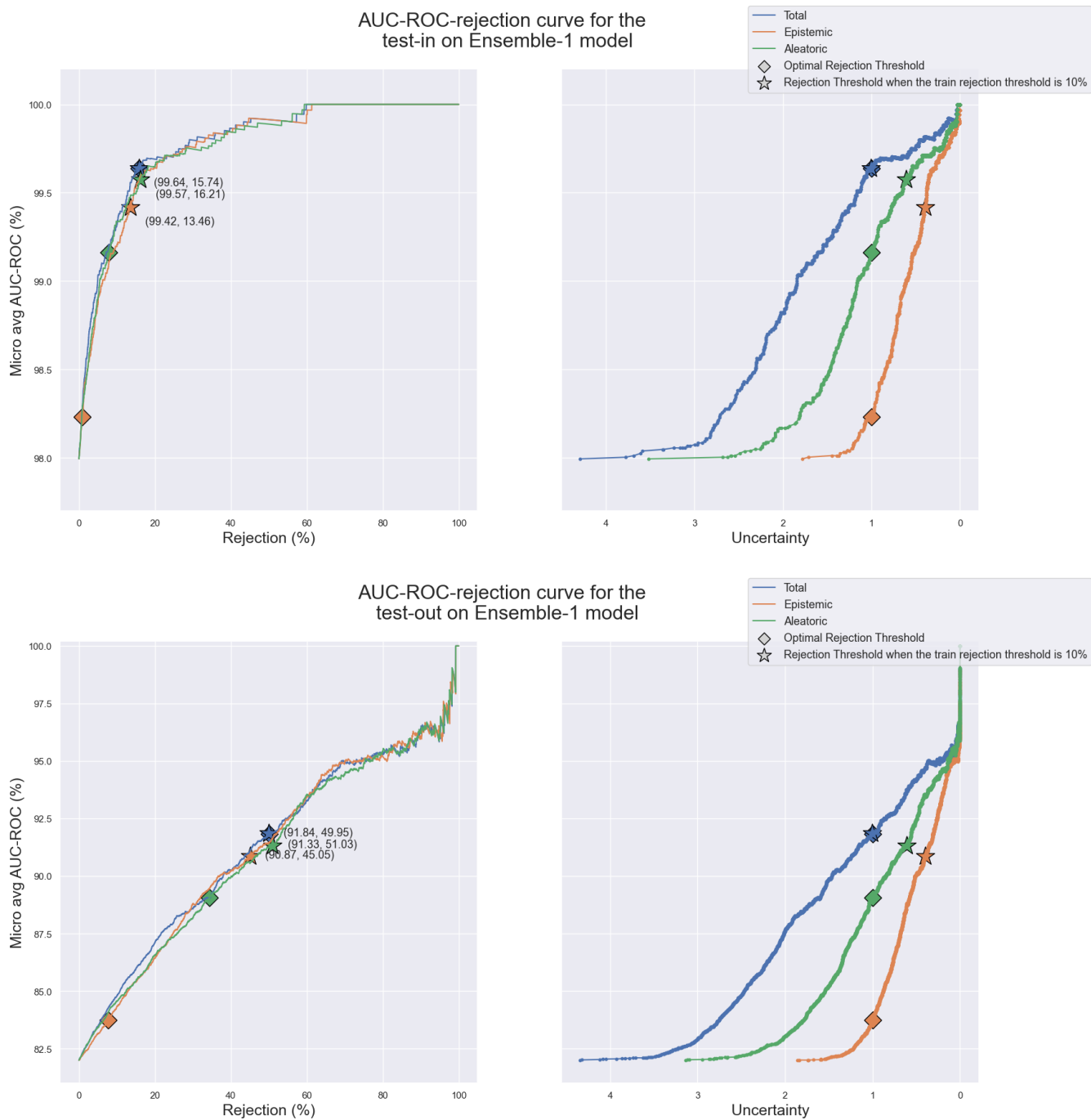


Figure B.5: AUC-ROC-rejection curve for both test sets in the Ensemble-1 model.

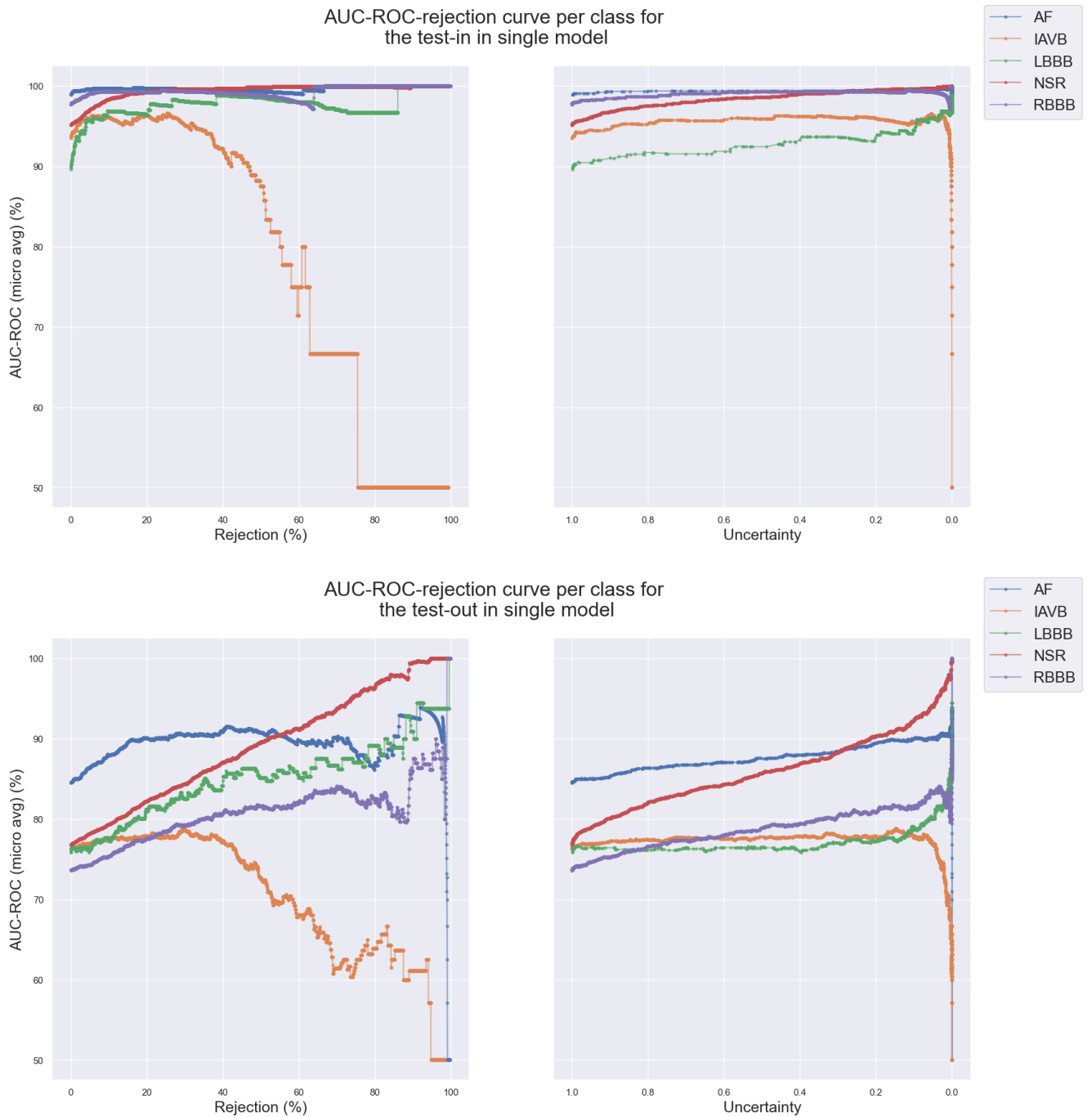


Figure B.6: AUC-ROC-rejection curve per class for both test sets in the single model.

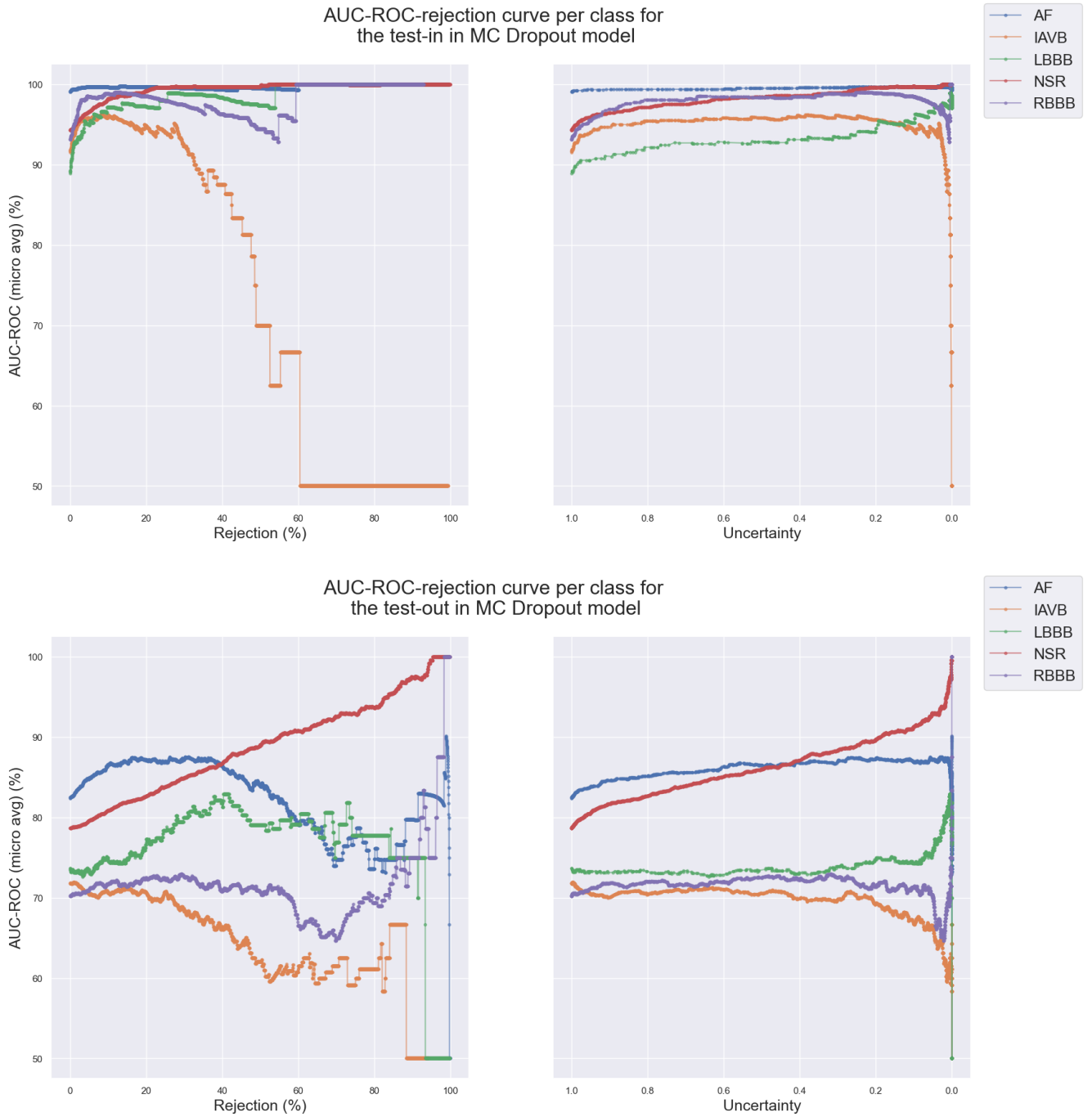


Figure B.7: AUC-ROC-rejection curve per class for both test sets in the Monte Carlo Dropout model.



APPENDIX B. COMPLEMENTARY RESULTS

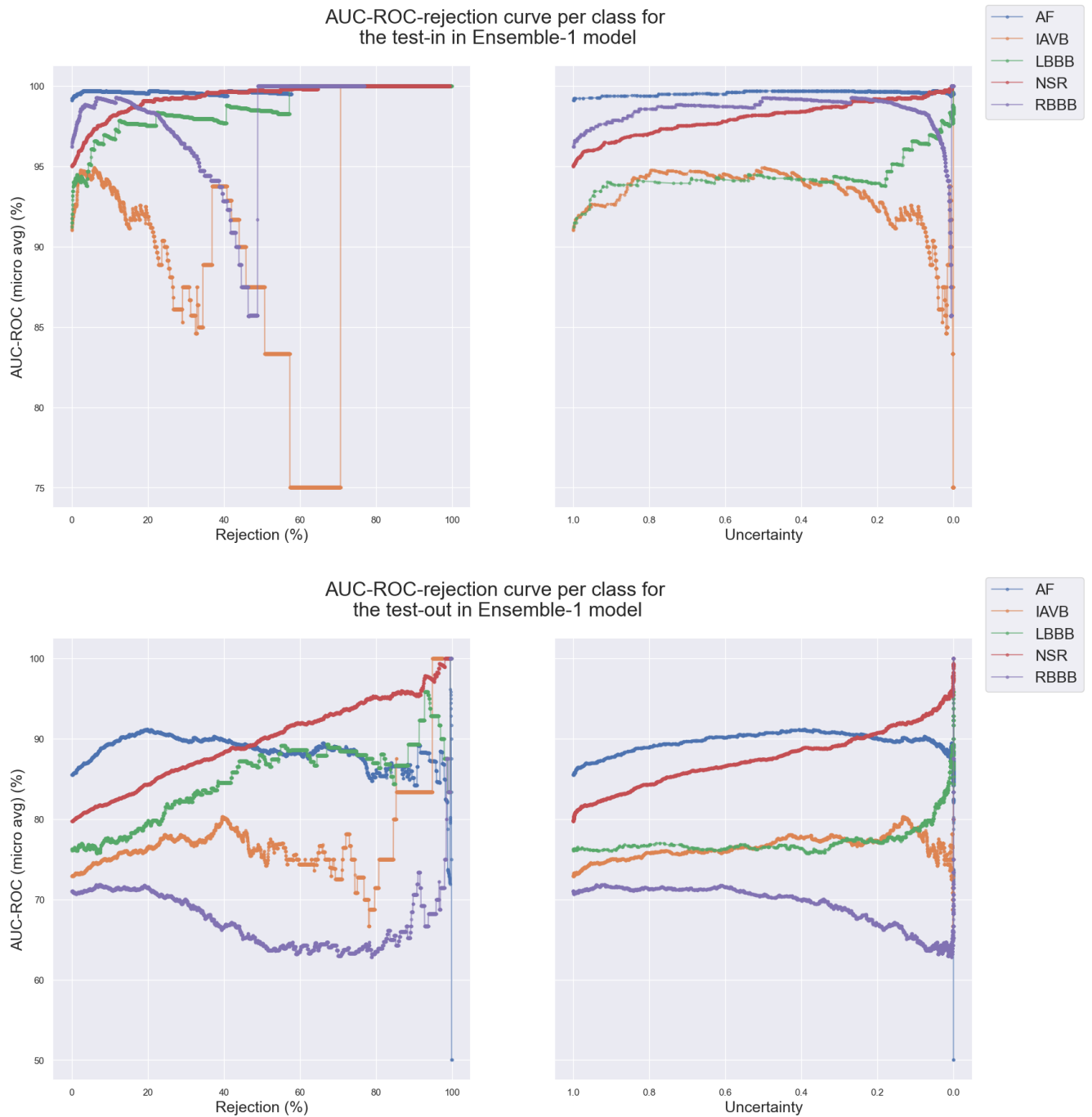


Figure B.8: AUC-ROC-rejection curve per class for both test sets in the Ensemble-1 model.

## PUBLICATIONS

In this Annex is included the scientific paper reporting some of the results obtained in this research, which is entitled "Study of uncertainty quantification using multi-label ECG in deep learning models". The paper was submitted and accepted at the conference BIOSIGNALS 2023 of the "16th International Joint Conference on Biomedical Engineering Systems and Technologies" (BIOSTEC 2023), that will take place on the 16th, 17th and 18th of February of 2023.

# Study of uncertainty quantification using multi-label ECG in deep learning models

Raquel Simão<sup>12\*</sup><sup>a</sup>, Marília Barandas<sup>12\*</sup><sup>b</sup>, David Belo<sup>2</sup><sup>c</sup> and Hugo Gamboa<sup>12</sup><sup>d</sup>

<sup>1</sup> LIBPhys (Laboratory for Instrumentation, Biomedical Engineering and Radiation Physics), NOVA School of Science and Technology, Campus da Caparica, 2829-516, Portugal

<sup>2</sup> Associação Fraunhofer Portugal Research, Rua Alfredo Allen 455/461, 4200-135 Porto, Portugal

\*These authors contributed equally to this work.

**Keywords:** Uncertainty Quantification, Monte Carlo Dropout, Deep Ensemble, Dataset shift, Active Learning

**Abstract:** Machine Learning (ML) models can predict diseases with noteworthy results. However, when implemented, their generalization are compromised, resulting in lower performances and render healthcare professionals more susceptible into delivering erroneous diagnostics. This study focuses on the use of uncertainty measures to abstain from classifying samples and use the rejected samples as a selection criterion for active learning. For the multi-label classification of cardiac arrhythmias different methods for uncertainty quantification were compared using three Deep Learning (DL) models: a single model and two pseudoensemble models using Monte-Carlo (MC) Dropout and Deep Ensemble (DE) techniques. When tested with an external dataset, the models' performances dropped from a F1-Score of 96% to 70%, indicating the possibility of dataset shift. The uncertainty measures for classification with rejection resulted in an increase of the rejection rate from 10% in the training set to a range between 30% to 50% on the external dataset. For the active learning approach, 10% of the highest uncertainty samples were used to retrain the models and their performance increased by almost 5%. Although there are still challenges to the implementation of ML models, the results show that uncertainty quantification is a valuable method to employ in safety mechanisms under dataset shift conditions.


## 1 INTRODUCTION


Over the years, medical technology has been developed and improved in order to ensure the most effective healthcare to the general public. Artificial Intelligence (AI) is quickly evolving due to its potential to assist evidence-based clinical decision-making and achieve value-based care (Chen and Decary, 2020). As a result, there has been a growing amount of scientific research regarding the use of ML algorithms in the medical domain. ML models have progressed to the point that they can predict a variety of diseases, with performances that can be superior to those achieved by healthcare professionals. This is achievable because ML models are trained with patient data in order to identify patterns that would otherwise be undetected and, thereby, produce an estimate of a patient's current or future clinical state.


However, while showing promising results, these


models still have some limitations for their deployment on clinical settings since their generalization capabilities are often compromised, resulting in lower performances and rendering healthcare professionals more susceptible into delivering erroneous diagnostics. This occurs since conditions in which we use the medical systems diverge from the conditions in which these systems were created, leading to mismatches between the training data and the data intended to be classified. This problem is called dataset shift and, in general, the greater the degree of shift, the poorer is the model's performance (Malinin et al., 2021). This is one of many problems that contribute to the limited number of models implemented in real life setting, with only 64 AI/ML medical systems approved by the FDA up until 2020 (Benjamins et al., 2020). As a result, it is critical that ML models include safety mechanisms to mitigate the dataset shift problem and improve the trustworthiness of these models. If AI/ML models fail to possess these mechanisms, they will be unable to be effectively implemented with FDA approval, leading AI/ML models to oblivion as decision support models.

Quantifying the uncertainty of models' predic-

<sup>a</sup>  <https://orcid.org/0000-0002-1678-5709>

<sup>b</sup>  <https://orcid.org/0000-0002-9445-4809>

<sup>c</sup>  <https://orcid.org/0000-0002-5337-0430>

<sup>d</sup>  <https://orcid.org/0000-0002-4022-7424>

tions is a key method to assess the model’s confidence in their decisions. Although uncertainty quantification has already demonstrated promising results in different fields, the literature on ECG classification is scarce. The works of (Vranken et al., 2021) and (Aseeri, 2021) are relevant works under this topic, however a single-label classification is applied, even though multi-label datasets are used.

In this paper, we develop a classification approach with rejection option based on uncertainty measures and evaluate the uncertainty as a selection method for active learning. Although the main purpose is to develop an agnostic framework for the classification of cardiac arrhythmias, this work will concentrate on establishing the practical value of the uncertainty quantification applied in three types of DL models in different medical datasets and their role in the referred methods. This research aims at providing a better understanding of the capacity of the model’s generalization through uncertainty estimation as well as demonstrate that uncertainty aware models are capable of containing safety mechanisms and, therefore, be considered trustworthy systems to be implemented in clinical settings.

## 2 RELATED WORK

### 2.1 Uncertainty Estimation Measures

In the general literature (Shaker and Hüllermeier, 2020; Barandas et al., 2022), a distinction between two intrinsically different sources of uncertainty is done: aleatoric and epistemic. Aleatoric Uncertainty (AU) is associated with the variability in the outcome of an experiment which is due to intrinsic randomness of the data generating process that cannot be explained away given more observations or data samples (Shaker and Hüllermeier, 2020). Epistemic Uncertainty (EU) refers to the lack of knowledge of the model and usually is caused by incomplete domain coverage since unknown regions of the data space will always be presented. The presence of new classes that were not contemplated in the training of the model, are an example of high EU. This uncertainty can be reduced by increasing the training data, better modeling or better data analysis (Barandas et al., 2022).

In traditional probabilistic modeling and Bayesian inference, the uncertainty of a prediction is given by the posterior distribution. Considering a finite dataset  $D$  composed of instances  $x$  and labels  $y$ , where  $y_k \in \{y_1, \dots, y_K\}$  is a set of  $K$  class labels, an hypothesis  $h$  maps the instances  $x$  to the outcomes  $y$ . The posterior  $P(h|D)$  can be obtained via the Bayes rule:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \quad (1)$$

where  $P(D|h)$  is the probability of data given  $h$  and  $P(h)$  is a prior distribution. For a single probability distribution, an uncertainty measure that combines both aleatoric and epistemic uncertainty can be calculated through the probability of the predicted class, given by:

$$p(\hat{y}|x) = \max_k p(y_k|x, D) \quad (2)$$

The entropy of the predictive posterior modeled by Shannon’s entropy is also an uncertainty measure for single probability distribution defined by:

$$H[p(y|x)] = - \sum_{k=1}^K p(y_k|x) \log_2 p(y_k|x) \quad (3)$$

In DL the randomness induced during training and inference can be used to obtain an uncertainty estimation (Mi et al., 2019). DE and MC Dropout are techniques commonly used for this quantification. DE consists of training repeatedly the same neural network with different parameters due to the randomness in the initialization and training process (Ståhl et al., 2020). Each model makes its own prediction and the final prediction is derived from the composition of all models in the ensemble. MC Dropout is a method that omits a certain percentage of neurons at each layer of a neural network during training and testing, with the missed neurons randomly selected for each iteration and each test time (Gal et al., 2016). The final prediction is obtained from the composition of all the predictions with distinct dropouts.

For these methods, the approximation proposed by Depeweg et al (Depeweg et al., 2018) can be used to obtain a measure of total, aleatoric and epistemic uncertainty:

$$u_{total}(x) := H\left[\frac{1}{M} \sum_{i=1}^M p(y|x, h_i)\right] \quad (4)$$

$$u_{aleat}(x) := \frac{1}{M} \sum_{i=1}^M H[p(y|x, h_i)] \quad (5)$$

$$u_{epist}(x) := u_{total}(x) - u_{aleat}(x) \quad (6)$$

### 2.2 Classification with Rejection Option

When a classifier is not sufficiently confident in the prediction, the model can abstain from producing an answer or discard a prediction if the uncertainty is sufficiently high. Therefore, a classifier with rejection can cope with unknown information, reducing

the threat caused by the existence of unknown samples or mislabeled training samples that can compromise the performance of the model. The standard approach for classification with rejection option, also known as Chow’s theory (Chow, 1970), is the calculation of a rejection threshold that minimises the classification risk. One approach to achieve this is through the uncertainty associated with every prediction. The empirical evaluation of methods for quantifying uncertainty is a non-trivial problem, due to the lack of ground truth uncertainty information. A common approach for indirectly evaluating the predicted uncertainty measures is using Accuracy-Rejection Curve (ARC). The ARC represents the accuracy of a classifier against its rejection rate, varying from 0 to 1 (Nadeem et al., 2009).

### 2.3 Active Learning

ML models, particularly DL models, demand a vast labelled dataset to learn properly. The number of labelled data required grows with the complexity of the problem or the complexity of the input data. This issue is particularly dominant in the medical field. In order to automate the analysis of a given medical exam, it would be necessary an expert to annotate a large number of exams, labelling them to indicate if the patient has certain condition or not. However, obtaining the amount of the needed labelled data is time-consuming and expensive. One possible solution to this problem is active learning. In this approach, the model chooses what unlabelled data is appropriate for training, and request an external “oracle”, for example a medical work, for the label of the selected data (Settles, 2009). The choice of the data to be labelled is selected by an acquisition function, which ranks points based on their potential informativeness (Gal et al., 2016). There are a variety of acquisition functions and many of them rely on model uncertainty to evaluate the potential informativeness of the unlabelled data points. The more informative is the selected data, the fewer labelled training examples are necessary to achieve a greater classifier accuracy. Therefore, the quantification of uncertainty plays a central role in active learning and can be valuable to improve the model’s performance when implemented in clinical settings.

## 3 METHODOLOGIES

### 3.1 Databases

Four public multi-label cardiac arrhythmia datasets from various countries were employed, having been provided by the PhysioNet/Computing in Cardiology Challenge 2020, as proposed by Perez Alday et. al (Alday et al., 2020). A subset of five classes were selected for classification: Atrial fibrillation (AF), First-degree atrioventricular block (IAVB), Left bundle branch block (LBBB), Right bundle branch block (RBBB) and Sinus rhythm (NSR). These classes were chosen since almost all of them are presented in each dataset and are the most frequent classes overall. The training database is composed of the CPSC2018 and PTB-XL dataset. The PTB and G12EC databases are used as external data in this research.

### 3.2 Data Preparation

To reduce the computational costs, only the ECG aVR lead was used since this lead produced the best results in the work of Chen et al. (Chen et al., 2020). The data was downsampled to 125 Hz and a 10 seconds window size was used. Data with length below that value were excluded and data above 10 seconds were truncated, so that all the samples have 1250 sample data points. The ECG signals were filtered using a 2nd order band-pass Butterworth filter between 1 and 40 Hz and it was also employed a smooth function using a window of 10 samples. Lastly, the data was normalised through a z-normalisation.

### 3.3 Proposed Algorithm

The model developed is a one-dimensional CNN. The architecture consists of three convolutional blocks, each with a convolutional layer followed by a batch normalization layer, a PRelu activation function with an initializer of 0.25, a max pooling layer and a dropout layer with rate of 0.25. Each convolutional layer has the same kernel size (31x31) but different number of filters (the first has 512 filters, the second has 256 and the last one has 128 filters). After the convolutional blocks, a flatten layer was applied, resulting in a Latent Vector. Three fully connected layers are added and the last one has a sigmoid activation function with the same number of neurons as classes. The flowchart of the proposed algorithm is shown in Figure 1.

The model was trained in 30 epochs with a batch size of 64. The loss function employed was the binary cross-entropy and an Adam optimizer with a learning

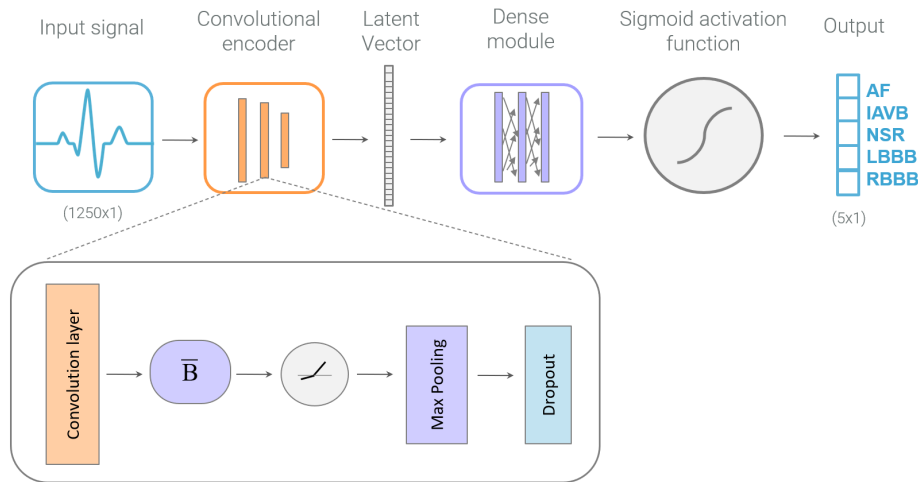


Figure 1: The flowchart of the designed algorithm. The algorithm architecture consists of three convolutional blocks, each with a convolutional layer followed by a batch normalization layer ( $\bar{B}$ ), a PRelu activation function with an initializer of 0.25, a max pooling layer and a dropout layer with rate of 0.25. A flatten layer was applied, resulting in a Latent Vector. Three fully connected layers are added and the last one has a sigmoid activation function with the same number of neurons as classes.

rate of 0.1. Since the model is trained with imbalanced datasets, it was added the class weight parameter that defines the weighting to adopt for each class when fitting the model.

### 3.4 Training and Testing

The data from CPSC2018 and PTB-XL database was split into 60% training, 20% validation and 20% testing. The test set from this database was used as an in-distribution set and will be referred as **test-in** from now on. The test set composed from all the samples in the PTB and G12EC datasets is named **test-out**. Two approaches were employed: the MC dropout and the DE. Both approaches were applied 30 times to both test sets, resulting in 30 models for each. To obtain the final prediction with both MC Dropout and DE approach, it was applied the majority vote for each class.

### 3.5 Uncertainty Approaches

For the single CNN, the predicted posterior probability, also known as maximum probability, and the Shannon entropy of the predicted probabilities were used as uncertainty measures. In the case of MC Dropout and DE, the total uncertainty, EU and AU measures were estimated. Since a prediction in a multi-label classification can return more than one class, the network sigmoid values do not sum 1. For this reason, in this multi-label scenario, each class was assumed as an independent binary case and the uncertainty calculated by each class. Besides the un-

certainty by class, an aggregation mechanism based on the sum of all class uncertainties was employed as the final prediction uncertainty.

Regarding the uncertainty evaluation, a common approach for evaluating the predicted uncertainty is by using ARC. However, due to the imbalance data, instead of using accuracy as a performance measure, the F1-score was used and the F1-Rejection curve was computed to evaluate the behaviour of the developed models. These curves were performed for the uncertainty measures mentioned previously with the rejection occurring from the sample with the highest uncertainty in its classification to the sample with the lowest uncertainty. This evaluation was performed considering the overall performance. Since the data is multi-label, the uncertainty of an ECG sample is the sum of each class uncertainty and, therefore, each sample uncertainty is represented by a value between 0 and 5.

### 3.6 Active Learning

Uncertainty estimation can be used to select the samples with higher uncertainty, taking advantage of the separation between epistemic and aleatoric uncertainty, where the former is more relevant as a selection criterion (Hüllermeier and Waegeman, 2021). Following this idea, the retraining process was performed for the single model and the DE model, where a new set was added to the previous training set for the retraining process. Each model was retrained for four more epochs using the newly dataset and the same parameters previously used to train the initial models.

To validate if samples with high epistemic uncertainty are more informative to the DE model, three different sets composed by 10% of the **test-out** were defined to the retraining process, namely: 1) random samples; 2) samples with the highest epistemic uncertainty; 3) samples with the highest total uncertainty. For the single model, the retraining was done with samples with the highest Shannon Entropy and for random samples as well.

## 4 RESULTS

In order to access the models' generalization capacities, it was compared the performance of the single, MC Dropout and DE models tested with **test-in** and tested with **test-out**.

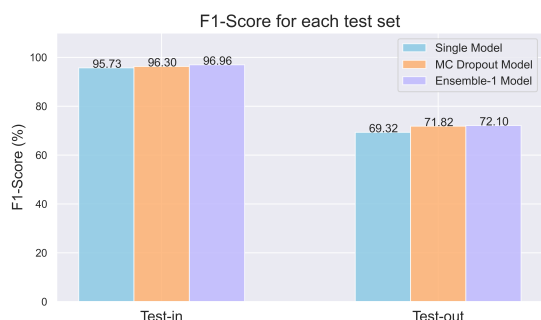


Figure 2: Micro average F1-score results for the three developed models tested in test-in and test-out sets.

As it can be seen in Figure 2, the three models, when tested with the **test-in** set, have similar performances, with micro average F1-score around 96%-97%, being comparable to the state of the art results. However, when the models are tested with the **test-out** set, their performances decrease significantly in all three models, having a micro-average F1-Score of approximately 70%. The DE model obtained the highest F1-score in both test sets with a maximum difference of 3% from the other models.

Regarding the classification with rejection option, even though this method does not solve the problem of model's generalization that leads to poor performance results under data shift, it can be a viable approach to abstain to predict a class under high uncertainty conditions. For each model, the uncertainties measures presented in Section 2.1 were calculated for the **test-in** and **test-out** sets and the results can be seen in Figures 3 and 4. For the single model, the behaviour of both uncertainties measures in **test-in** and **test-out** are similar. However, both uncertainty measures obtain higher uncertainty in the **test-out** set.

As for the results in Figure 4, for the **test-in** set,

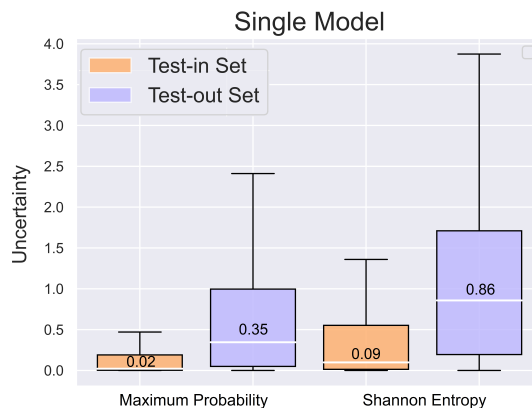


Figure 3: Uncertainty Estimation for both test sets in the single model.

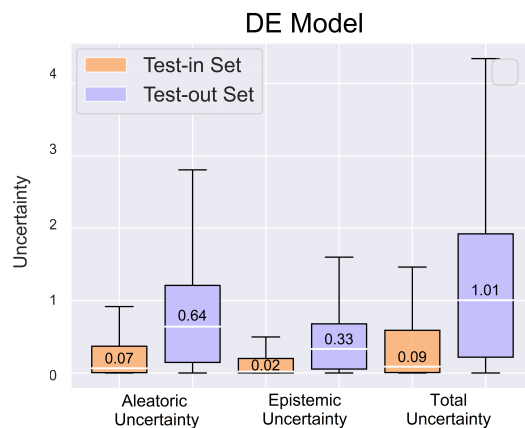
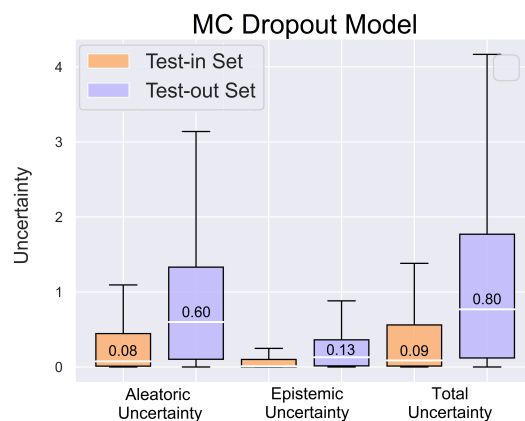


Figure 4: Uncertainty Estimation for both test sets in the MC Dropout(up) and DE(down) models.

the MC Dropout and DE models estimate similar values of uncertainty, presenting the same median and the same range of total uncertainty. The MC Dropout presents a higher range of AU while the DE detects higher EU. As for the **test-out** set, both models cap-

ture higher uncertainty than for the **test-in** set in all the three types of uncertainty measures.

To investigate the role of uncertainty in rejection, the F1-rejection curve was produced for the three models, rejecting the samples according to the highest calculated uncertainties. To validate the rejection rate in both sets, a 10% rejection in the training set was applied and the uncertainty thresholds obtained. Using the same thresholds on **test-in** and **test-out**, the rejection rates increased to approximately 12% and 40%, respectively, using the single model for both maximum probability and entropy measures. For the MC Dropout the rejection in **test-in** was 9% and vary between 31% and 34% for **test-out** depending on the uncertainty measure used. The DE model vary the rejections rates between the intervals 13%-16% and 45%-51% for **test-in** and **test-out**, respectively. Furthermore, as it can be deduced for the micro average F1-Scores presented in the Table 1, for all the three models and for all uncertainty measures, the more samples rejected, the better is the models' performance. Even though the curves based on the different uncertainty methods are quite similar, throughout the rejection, the DE model presents better micro average F1-Score results for the same rejection rate.

Apart from employing the rejection option, a possible method to deal with dataset shift is by retraining the model with samples that have crucial information to help improve its performance. A potential solution is the active learning approach, in which the samples used to retrain the model contain the highest uncertainty associated with their classifications. To evaluate the three uncertainties in this approach, the retrained models were tested with the **test-out** set without the 10% samples to fairly compare the increase between the retrained model and the baseline model. Thus, the following nomenclature was used: 1) Previous trained model using the complete test-out set (Baseline - test-out-100); 2) Previous trained model tested only on 90% of **test-out**, i.e 10% of **test-out** was used to retrain the model (Baseline - **test-out**-90); 3) Retrained model using the selected 10% data and tested on the remaining 90% (Retrain - **test-out**-90). Furthermore, to serve as control, this process was performed for 10% of random samples in order to observe the role of uncertainty in this approach. This procedure was conducted 10 times and the mean and standard deviation of the results are represented in Figure 5.

As it can be observed in Figure 5, when the samples with the highest uncertainty are removed from the test-out, the model performance increases slightly, from 2%-4%. After retraining the two models with these samples and evaluating it without them, a max-

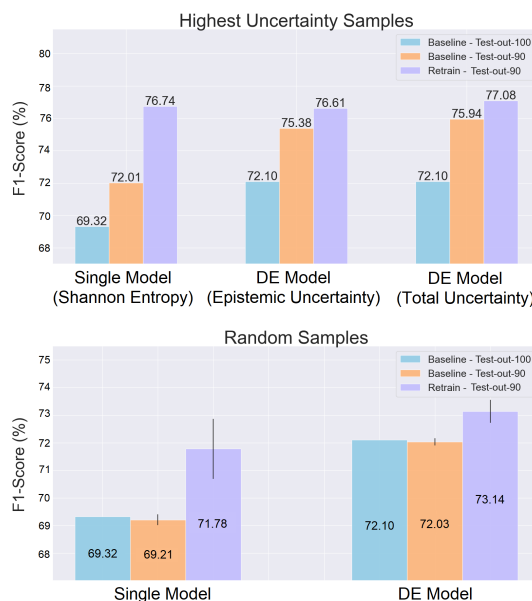


Figure 5: Micro average F1-score for the Active learning approach for the highest uncertainties and for random samples.

imum increase of almost 5% is observed when compared to the baseline models that are tested with all the samples of **test-out**. These conclusions are supported through the results served as a control, where the samples selected are random and the trained models have similar performance as the original models.

## 5 DISCUSSION

To make the decision support systems as trustworthy as possible, it is critical to access the confidence that ML models have in their classifications. This work studied these concepts using four large public ECG databases for the classification of cardiac arrhythmias. As multiple cardiac arrhythmias can be presented within the same recording, a multi-label classification setting was adopted for the development of DL models.

The performance of the three models developed were assessed for two test sets, where the **test-in** has data from the same database as the training and the **test-out** presents data from a different database. Although these models produced similar performance results for the same test set, the DE and MC Dropout outperform the single model, as expected since these models assist in reducing models' high confidence in incorrect classifications. The DE model revealed has the better performance in both test sets, which it is consistent with the literature. When tested with the



Table 1: Rejection rate results and the respective F1-Score values for each uncertainty.

| Model      | Uncertainty         | Test-in   |          | Test-out  |          |
|------------|---------------------|-----------|----------|-----------|----------|
|            |                     | Rejection | F1-Score | Rejection | F1-Score |
| Single CNN | Maximum Probability | 12.24%    | 98.54%   | 39.81%    | 79.14%   |
|            | Shannon Entropy     | 12.16%    | 98.38%   | 41.70%    | 79.89%   |
| MC Dropout | Aleatoric           | 9.51%     | 98.46%   | 31.92%    | 82.25%   |
|            | Epistemic           | 9.35%     | 98.29%   | 33.68%    | 83.07%   |
|            | Total               | 9.41%     | 98.47%   | 34.30%    | 83.33%   |
| DE         | Aleatoric           | 16.21%    | 99.34%   | 51.03%    | 86.26%   |
|            | Epistemic           | 13.46%    | 99.10%   | 45.05%    | 85.25%   |
|            | Total               | 15.75%    | 99.41%   | 49.95%    | 87.00%   |

**test-out** set, the performance of all the three models drops significantly, confirmed by the decrease of F1-Score from around 96% to 70%. These results indicate the possible presence of dataset shift since the data from **test-out** has different characteristics and distributions than the data used for training.

Regarding the uncertainty estimations, the Shannon entropy and maximum probability were estimated for the single model and the aleatoric, epistemic, and total uncertainty for the MC Dropout and DE models. For the single model, both maximum probability and entropy obtained similar results, while for the MC Dropout and DE the total uncertainty presented slightly better result. This suggests the benefit of estimating uncertainty using the combination of epistemic and aleatoric uncertainty. Additionally, all uncertainties computed for the **test-out** were significantly higher than for the **test-in** set. This shows that the model is less confident on the classification of cardiac arrhythmias and as result there is higher probability of misclassified samples. This is an indication of dataset shift and the main reason of models' performance drop in **test-out** set. Furthermore, it is important to mention that it was expected that the EU would be higher than the AU in **test-out** since the data comes from a different source and might be a different distribution. This reveals that there are still challenges in capturing these two uncertainties correctly.

In order to improve the trustworthiness of the models, the classification with rejection option was applied. For both test sets, the models performance increased with rejection, revealing that the higher the uncertainty in a given classification, the higher is the probability of the models to misclassify the samples. Additionally, the uncertainty threshold, selected from the training data, increased from 10% to a range be-

tween 30% to 50% depending on the model or uncertainty measure employed. The increase in rejection rate confirms that high uncertainty is presented in the classifications and the uncertainty is higher in the **test-out** set. This is another evidence of the dataset shift effect and that the models are not as prepared to classify data with different distributions.

Another alternative to improve the models' performance and reliability is through the retraining of models with unseen data. In this manner, it was employed an active learning approach, using 10% of the samples with the highest uncertainty in the **test-out** set. The results showed that the models improved their performance by a maximum of almost 5% when using uncertainty versus 2% when using a random selection. These results demonstrate that data with high uncertainty has information that the model has not yet learned and hence the models benefit from the retraining with this selection method. Moreover, when removing the 10% of the samples with the highest uncertainty and test 90% of the **test-out** in the baseline models, the performance improved, showing that the samples with highest uncertainty are misclassified. This underlines the importance of the uncertainty quantification in detecting incorrect classifications.

## 6 CONCLUSIONS

The evaluation and comparison of uncertainty measures has proven to be essential in an in-depth analysis of ML models, allowing us to understand their limitations. Furthermore, the preliminary results reveal that the quantification of uncertainty should be considered a key feature of any ML model as a safety

mechanism.

Although there are still no ground truth for the estimation of uncertainty, all the metrics used were capable to detect uncertainty in multi-label data. Nevertheless, there are still challenges in capturing the uncertainty through the employed measures, specially in the separation of epistemic and aleatoric uncertainty. It is also possible to infer the role of uncertainty as a valuable method under dataset shift conditions and in strategies such classification with rejection option and active learning approaches.

Thus, the development of uncertainty aware models will provide healthcare professionals with access to the model's confidence in its predictions but also refrain the model from delivering classifications with high uncertainty. Furthermore, samples that have different characteristics and distributions than the ones learned by the models have higher uncertainty associated with their classifications and, therefore, can be used to retrain the ML models and improve its generalization and robustness. The active learning approach is a reliable method for this purpose, demonstrating that it is a technique capable to self-regulate the learning of the models in a real life setting, with a reduction in computational cost as well as in the cost of labelling the data usually required. Despite the encouraging results, much more research is needed in the area of clinical data uncertainty, particularly in multi-label data.

To conclude, data with different characteristics and distributions from those learnt by the ML models will always exist, so it is imperative that AI systems possess uncertainty associated methods as safety mechanisms to produce reliable models to implement as a decision support system in clinical settings.

## REFERENCES

- Alday, E. A. P., Gu, A., Shah, A. J., Robichaux, C., Wong, A.-K. I., Liu, C., Liu, F., Rad, A. B., Elola, A., Seyedi, S., et al. (2020). Classification of 12-lead ecgs: the physionet/computing in cardiology challenge 2020. *Physiological measurement*, 41(12):124003.
- Aseeri, A. O. (2021). Uncertainty-aware deep learning-based cardiac arrhythmias classification model of electrocardiogram signals. *Computers*, 10(6):82.
- Barandas, M., Folgado, D., Santos, R., Simão, R., and Gamboa, H. (2022). Uncertainty-based rejection in machine learning: Implications for model development and interpretability. *Electronics*, 11(3).
- Benjamins, S., Dhunnoo, P., and Meskó, B. (2020). The state of artificial intelligence-based fda-approved medical devices and algorithms: an online database. *NPJ digital medicine*, 3(1):1–8.
- Chen, M. and Decary, M. (2020). Artificial intelligence in healthcare: An essential guide for health leaders. *Healthcare Management Forum*, 33(1):10–18. PMID: 31550922.
- Chen, T.-M., Huang, C.-H., Shih, E. S., Hu, Y.-F., and Hwang, M.-J. (2020). Detection and classification of cardiac arrhythmias by a challenge-best deep learning neural network model. *Iscience*, 23(3):100886.
- Chow, C. (1970). On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46.
- Depeweg, S., Hernandez-Lobato, J.-M., Doshi-Velez, F., and Udluft, S. (2018). Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1184–1193. PMLR.
- Gal, Y. et al. (2016). Uncertainty in deep learning.
- Hüllermeier, E. and Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506.
- Malinin, A., Band, N., Chesnokov, G., Gal, Y., Gales, M. J., Noskov, A., Ploskonosov, A., Prokhorenkova, L., Provilkov, I., Raina, V., et al. (2021). Shifts: A dataset of real distributional shift across multiple large-scale tasks. *arXiv preprint arXiv:2107.07455*.
- Mi, L., Wang, H., Tian, Y., and Shavit, N. (2019). Training-free uncertainty estimation for neural networks.
- Nadeem, M. S. A., Zucker, J.-D., and Hanczar, B. (2009). Accuracy-rejection curves (arcs) for comparing classification methods with a reject option. In Džeroski, S., Guerts, P., and Rousu, J., editors, *Proceedings of the third International Workshop on Machine Learning in Systems Biology*, volume 8 of *Proceedings of Machine Learning Research*, pages 65–81, Ljubljana, Slovenia. PMLR.
- Settles, B. (2009). Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.
- Shaker, M. H. and Hüllermeier, E. (2020). Aleatoric and epistemic uncertainty with random forests. In *International Symposium on Intelligent Data Analysis*, pages 444–456. Springer.
- Ståhl, N., Falkman, G., Karlsson, A., and Mathiason, G. (2020). Evaluation of uncertainty quantification in deep learning. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 556–568. Springer.
- Vranken, J. F., van de Leur, R. R., Gupta, D. K., Juarez Orozco, L. E., Hassink, R. J., van der Harst, P., Doevendans, P. A., Gulshad, S., and van Es, R. (2021). Uncertainty estimation for deep learning-based automated analysis of 12-lead electrocardiograms. *European Heart Journal-Digital Health*, 2(3):401–415.



# Algebraic Geometry

Algebraic geometry is a branch of mathematics that studies the geometric properties of solutions to systems of polynomial equations. It combines techniques from algebra and geometry to analyze the structure of algebraic varieties.

The central objects of study in algebraic geometry are algebraic varieties, which are sets of points in a projective space defined by the common zeros of a set of homogeneous polynomials. The theory of varieties is closely related to the theory of rings and modules.

Key concepts in algebraic geometry include the dimension of a variety, the degree of a curve, and the intersection theory of divisors. The theory of curves and surfaces is a fundamental part of the subject, and it has many applications in physics and engineering.

Algebraic geometry is a rich and active area of research, with many open problems and new discoveries. It is a beautiful and challenging field that offers a deep understanding of the geometry of algebraic structures.