Instituto Superior
de Engenharia
Politécnico de Coimbra

DEPARTMENT OF SYSTEMS AND COMPUTER ENGINEERING

Estimating Pointwise Reliability of Machine Learning Predictions in Clinical Context

Project Report to fulfill the Master's degree in Informatics Engineering

Specialization in Software Engineering

Author

Cláudio Diogo Carvalho Correia

Supervisor

Teresa Raquel Teixeira da Rocha

Co-Supervisor

Simão Pedro Mendes Cruz Reis Paredes

INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

Coimbra, October 2025

Abstract

As interest in machine learning expands into critical domains, such as clinical practice, the need for reliable predictions is critical, as decisions based on these outputs can have significant consequences. Traditional evaluation of machine learning models relies on global performance metrics, which provide aggregate information but does not indicate whether an individual prediction can be trusted. This limitation reinforces the importance of pointwise reliability methods that use measures to estimate the reliability of single predictions.

This work assesses pointwise reliability methods, with a particular focus on model agnostic approaches using the density principle and the local fit principle. These principles evaluate reliability by analyzing the similarity of new instances to the training data and by evaluating the performance of the model in local regions of the feature space. Three case studies were considered, covering two clinical classification problems related to treatment decisions and cardiovascular related mortality, as well as a regression task that involves the prediction of hospital admissions. Reliability was assessed using clustering based, distance based, and local fit methods. The performance of the methods was evaluated by analyzing the relationship between reliability intervals and the prediction errors.

The results demonstrate that methods combining density and local fit principles generally outperform those relying on a single principle, achieving lower error rates for predictions assigned high reliability. In classification tasks, the method combining density and local fit principles produced the most stable and interpretable results, offering a stronger basis for decision making. In addition to reliability analysis, the regression case study compared traditional statistical models with ML approaches in the forecasting of hospital admission. The findings suggest that the ML models achieved a greater correlation with the actual values and that, for the pointwise results, the methods combining both principles improved the alignment between absolute prediction errors and the assigned reliability scores.

Keywords: Density Principle, Local Fit Principle, Machine Learning, Pointwise Reliability

Resumo

À medida que o interesse por *machine learning* se expande para domínios críticos, como a prática clínica, a necessidade de medidas de confiança nas classificações é vital, dado que as decisões baseadas nesses resultados podem ter consequências significativas. A avaliação tradicional dos modelos de *machine learning* depende de métricas de desempenho globais, que fornecem informações agregadas, mas não indicam se uma classificação individual é confiável. Esta limitação reforça a importância dos métodos de confiabilidade pontual, que usam medidas para estimar a confiabilidade de classificações individuais de modelos de *machine learning*.

Este trabalho avalia métodos de confiabilidade pontual, com foco particular em abordagens agnósticas ao modelo e que utilizam o princípio da densidade e ajuste local. Estes princípios avaliam a confiabilidade analisando a semelhança de novas instâncias com os dados de treino e avaliando o desempenho do modelo em determinada região do espaço de dados. Neste trabalho, foram considerados três casos de estudo, abrangendo dois problemas de classificação clínica relacionados com decisões de internamento e mortalidade derivada de problemas cardiovasculares, e uma tarefa de regressão que envolve a previsão do numero de admissões hospitalares. A confiabilidade das classificações dos modelos foi avaliada utilizando abordagens baseadas em *clustering*, distância e ajuste local. O desempenho das abordagens foi avaliado através da análise da relação entre intervalos de confiabilidade e erros de classificação.

Os resultados demonstraram que os métodos que combinam o princípio de densidade e ajuste local geralmente apresentam melhores resultados do que os que se baseiam num único princípio, alcançando taxas de erro mais baixas para classificações avaliadas como altamente confiáveis. Em tarefas de classificação, o método que combina os princípios de densidade e ajuste local produziu os resultados mais estáveis, apresentando uma base mais sólida para a tomada de decisões críticas. Além da análise de confiabilidade, o caso de estudo com regressão comparou modelos estatísticos tradicionais com abordagens de *machine learning* na previsão de admissões hospitalares. Os resultados mostram que os modelos de *machine learning* alcançam uma maior correlação com os valores reais e que, para a análise de confiabilidade, os métodos que combinaram ambos os princípios melhoraram a correlação entre os erros absolutos das previsões e a confiabilidade atribuída.

Palavras-chave: *Density Principle*, *Local Fit Principle*, *Machine Learning*, *Pointwise Reliability*

Acknowledgments

I would like to express my sincere thanks to all those who, in some way, contributed to the completion of this project, namely:

- The PCDaS team, for providing the dataset and for the time devoted to clarifying the data and its context;
- Professor Teresa Rocha, Professor Simão Paredes, and Professor Jorge Henriques, for their guidance;
- To the faculty and staff of ISEC, whose dedication and shared knowledge shaped my academic journey;
- My family, especially my wife, for their unconditional support throughout my academic journey.

Table of Contents

Index of Tables

Index of Figures

List of Abbreviations

ACF	Autocorrelation Function
ADF	Augmented Dickey-Fuller
CONFINE	Confidence estimation based on the Neighbours Errors
CONFIVE	Confidence estimation based on the Variance in the Environment
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
HCT	Hemocyte
ICM	Interpretable Confidence Measure
IEEE	Institute of Electrical and Electronics Engineers
IQR	Interquartile Range
ISEC	*Instituto Superior de Engenharia de Coimbra*
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MSE	Mean Squared Error
PACF	Partial Autocorrelation Function
PCDaS	*Plataforma de Ciência de Dados aplicada à Saúde*
R^2	Coefficient of Determination
RF	Random Forest
RMSE	Root Mean Squared Error
SARIMA	Seasonal Autoregressive Integrated Moving Average
t-SNE	t-distributed Stochastic Neighbor Embedding
XGB	eXtreme Gradient Boosting

List of Symbols

NO_2	Nitrogen Dioxide
O_3	Ozone
PM_{10}	Particulate Matter (diameter $\leq 10\,\mu\mathrm{m}$)
$PM_{2.5}$	Particulate Matter (diameter $\leq 2.5\,\mu\mathrm{m}$)
SO_2	Sulfur Dioxide

1 Introduction

Machine learning (ML) techniques have made significant advances in recent years in critical areas such as healthcare [1, 2, 3]. Predictive models are now capable of assisting clinicians in tasks such as disease diagnosis, treatment selection, risk stratification, and patient monitoring. However, despite significant advances in predictive accuracy, the deployment of ML models in real world clinical settings remains limited. A central barrier to their adoption lies not in the accuracy of the models, but in the absence of mechanisms to assess their reliability [4, 5].

Traditional evaluation metrics such as accuracy, precision, and mean squared error (MSE) are typically computed on test datasets and serve as global indicators of model performance. These aggregate statistics, while useful during model development, do not provide insight into how the model performs in individual cases. In deployment phase, the distribution of the input data may change, edge cases may arise, or a model may encounter data instances that are underrepresented in the training process [6, 7, 8]. In such cases, a single overall metric cannot reliably indicate the behavior of the model, and decision makers are left without information on whether a given prediction is reliable. This is particularly problematic in the clinical domain, where erroneous predictions can have serious consequences, an issue that remains largely unaddressed [9, 10].

Recent research has explored pointwise reliability [11, 12, 13], a quantification of how likely a given prediction will be correct. Assessing the pointwise reliability of an ML model aims to answer the question "Can I rely on the machine learning model to predict this current instance?". The response can be used to flag uncertain predictions for further review, guide clinicians toward more cautious decision making or trigger fallback procedures.

Two main principles guide current approaches to pointwise reliability [14]. The density principle states that the predictions are more likely to be correct when the input instance lies within a dense region of the training data. The local fit principle evaluates how well the model performed in regions of the feature space similar to the new instance.

Although various studies have proposed methods based on these principles, their evaluation has often been carried out independently, with different datasets, models, and validation criteria. Consequently, there is little direct comparison between pointwise reliability methods, and many methods are evaluated in domain specific environments.

To address this gap, this work implements and benchmarks a set of model agnostic pointwise reliability methods, using consistent datasets and models. This approach provides a way to directly compare these techniques and analyze their strengths and limitations under similar conditions.

The work focuses on three case studies. The first two are strictly clinical and involve classification tasks, one relating to treatment decisions and another focusing on mortality related to cardiovascular problems. The third case study shifts focus to the healthcare management domain, specifically a regression task for hospital admissions based on environmental and temporal variables. This third study is justified by its strong public health implications [15, 16] as well as the amount of data available. Furthermore, it introduces additional challenges that are not present in the other cases, such as data analysis and model selection validation.

By exploring and comparing the performance of multiple reliability estimation techniques in these diverse contexts, this work aims to provide information on when and how pointwise reliability can enhance trust in ML based systems in clinical contexts and its adjacent domains.

1.1 Main Objectives and Contribution

The main objective of this thesis is to study and implement pointwise reliability measures to evaluate the reliability of predictions of ML models. The work builds on the hypothesis that the reliability of a specific instance can be estimated by assessing its similarity to the training data and by evaluating whether the model performs well on training samples in the same local neighborhood. These reliability metrics are intended to provide users with guidance on the trustworthiness of predictions, especially in sensitive applications such as the clinical field.

During the course of the thesis, an additional dataset was made available in the healthcare management domain. This expanded the objectives to also include a regression problem, introducing exploratory analysis and assessment of model suitability before the application of reliability estimation methods.

Thus, the specific objectives of this project are: i) to identify and implement pointwise reliability methods based on the density and local fit principles; ii) to assess the performance of the selected methods; and iii) to extend the analysis to a healthcare management regression.

The key contribution is the practical implementation and comparative evaluation of model agnostic pointwise reliability methods. The work presents adaptations for classification and regression, validates the methods on real world datasets, and examines their applicability across different domains. This contribution is supported by three

case studies. Two address clinical classification tasks focused on treatment decisions and cardiovascular risk, while the other addresses a healthcare management task that predicts hospital admissions. Together, these studies show how pointwise reliability can strengthen trust in ML predictions in healthcare.

1.2 Project Planning

The main tasks implemented during the development of the project are as follow:

- **T1. State-of-the-art**: State-of-the-art of pointwise reliability of machine learning predictions, with emphasis on density and local fit principles;
- **T2. Reliability measures**: Research and implement model agnostic pointwise reliability measures for machine learning models;
- **T3. Use cases**: Validate the measures in representative clinical and healthcare management use cases;
- **T4. Final report**: Write the final report.

Table 1.1 shows the tasks implemented during the development of the project over time.

Table 1.1: Planned schedule for tasks.

Tasks	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
T1	X	X	X	X	X								
T2		X	X	X	X	X							
T3					X	X	X	X	X	X			
T4							X	X	X	X	X	X	X

Regarding the original plan, the state-of-the-art review was extended to cover the third case study, which prolonged its duration and consequently had an impact on the remaining tasks.

1.3 Document Structure

The remainder of this document is organized as follows:

- **Chapter 2** presents the concept of pointwise reliability, reviews the main methods, and introduces the benchmarking framework;
- **Chapter 3** reports on a classification case study involving patient treatment decisions, evaluating pointwise reliability methods;
- **Chapter 4** applies the methods to a second clinical classification task that focuses on predicting mortality related to cardiovascular problems;

- **Chapter 5** addresses a regression task on hospital admission forecasting. In addition to assess pointwise reliability, the chapter also discusses the preparation of the data and validation of the model for the case study;
- **Chapter 6** presents the main conclusions, discusses limitations, and future directions.

Additionally, the document includes the following appendices:

- **Appendix A - Patient Treatment**: Complementary results from the patient treatment classification case study (Chapter 3);
- **Appendix B - Hospital Admissions**: Additional experiments for the hospital admissions regression case study (Chapter 5), including models trained in different settings and the evaluation of further reliability metrics.

2 Pointwise Reliability

This chapter introduces the concept of pointwise reliability and presents the methods selected for evaluation in subsequent case studies. It also describes the benchmark used to assess their performance.

2.1 Definition of Pointwise Reliability

This section introduces the concept of pointwise reliability, focusing on the density and local fit principles that support methods that are agnostic to the Machine Learning (ML) model and applicable in the deployment stage.

Assessing the reliability of individual predictions aims to determine when the output of a model can be trusted and when it should be subjected to additional consideration or human review. Although the notion of reliability appears in multiple contexts [10, 17, 18, 19], in this work reliability is defined as the probability that a prediction of a model is correct for a given new instance, expressed as a continuous value within a defined range (*e.g.*, between 0 and 1). This discussion follows the taxonomy of Loureiro *et al.* [14] to present the principles and methods.

The **density principle** assumes that regions of the feature space with high data concentration are more likely to produce reliable predictions, whereas sparse regions are more uncertain. The main idea is that the reliability of each individual prediction can be assessed by evaluating how dense the neighborhood of a new instance is. In practice, the density principle is implemented through data driven methods that can be categorized as unsupervised or supervised:

- **Unsupervised:** These methods assess reliability based on the similarity or proximity of a new instance to other data points, often using distance metrics or clustering algorithms. For example, if a test sample is far from most of the training data or outside clusters, it is considered less reliable;
- **Supervised:** When label information is available, the reliability can also be assessed based on the consistency of the class. For example, if a new instance belongs to a region predominantly occupied by points of the same class, it may be considered more reliable.

The **local fit principle** assesses the reliability based on how well the ML model performs in the local neighborhood of a new instance. In essence, if the model performed

well in a region of the feature space similar to the model input, then its prediction is more likely to be reliable. Methods based on the local fit principle can be grouped according to the type of information they use:

- **Data driven:** These methods assess the variability or dispersion of the target variable in the neighborhood of the new instance. For example, in a regression, a high variance in the output labels of nearby training points suggests that the local region is less reliable;
- **Model driven:** These methods incorporate the predictions of the model to evaluate local error or uncertainty. Examples include calculating the mean squared error of neighboring points, assessing output variance from an ensemble of models or training auxiliary models to predict the error;
- **Complementary information:** These methods incorporate domain knowledge in the reliability assessment, such as logical or consistency rules, rather than relying on data or model predictions alone. A prediction is considered reliable if it satisfies some predefined rules. For example, in a clinical case, it should not be predicted that a patient with more severe symptoms is at lower risk than a patient with less severe symptoms.

In the literature, a variety of methods have been proposed for assessing pointwise reliability that can be categorized into density, local fit, and a combination of both principles.

Methods based on the **density** principle assess how well supported a new instance is within the training data. Techniques including distance estimators such as nearest neighbor [20], which compute distances to evaluate the density of nearby points, clustering approaches such as DBSCAN [21], which identify dense regions that are likely to correspond to more reliable predictions, and fuzzy c-means [22], which iteratively finds cluster centers and assigns to each instance a degree of belonging to each cluster, are examples of methods that can be used to evaluate pointwise reliability following unsupervised methods.

For supervised methods, Nicora *et al.* [11, 23] propose a method that assesses reliability by evaluating the proximity of a new instance to the boundary instances in the training data for each feature, using the ordered projections algorithm [24] to define the borders. Considering both internal and external boundaries, the reliability is computed as the proportion of features of the new instance that fall outside the boundaries relative to the total number of features. Another approach based on geometric methods was assessed by Sousa *et al.* [25], who used convex hulls [26] to identify convex regions that contain data points of the same class. Given the complexity of the algorithm, the authors chose a small subset of the total features to compute the reliability of the predictions.

The method developed by Waa *et al.* [27], evaluates the similarity of new data points to previously observed data points using a supervised distance based approach. The method calculates a confidence score by weighting the influence of all data points based on their proximity to the new instance (all training data are considered). The same authors propose an alternative version of the method [28] in which new instances are compared with past cases using the nearest neighbors. The neighbors are divided into groups that either support or contradict the current prediction of the model, and a confidence score (reliability) is derived from the similarity of the supporting group. Experiments on various datasets have shown that the method provides reliable confidence scores.

Another supervised method was proposed by Kailkhura *et al.* [29] to address mixed data types (continuous and categorical features) for material discovery. The author used the Gower distance to measure the similarity between instances. Their reliability is assessed as the ratio of the average Gower distance of a test sample to samples in its own class to the total average distance to both its own class and other classes, assigning higher reliability to instances that are closer to their own class.

Local fit methods estimate reliability by analyzing the behavior of the model. Briesemeister *et al.* [30] proposed for regression tasks two methods: the Confidence estimation based on the Neighbours Errors (CONFINE) and Confidence estimation based on the Variance in the Environment (CONFIVE), which works using the characteristics of nearby data points of a new instance (using the training data). Using the output of the trained model, CONFINE estimates the mean squared error of the nearest neighbors of a new instance. CONFIVE, which can be categorized as a data driven method, evaluates the variance of labels of the nearest neighbors of a new instance, assuming that greater variance is correlated with lower reliability. Both methods have been validated on various datasets, proving their effectiveness and ML model agnostic characteristics.

A model driven approach using complementary models was introduced by Adomavicius and Wang [31], where a secondary model is trained to predict the absolute error of a primary model. This secondary model learns from a validation dataset and outputs a reliability classification for new instances. Trindade *et al.* [32] and Zhang *et al.* [33], who also train a secondary model to predict the error of the main regression model, proposed a similar strategy for regression tasks. The authors conclude that this approach performs better than relying on simple metrics such as local variance or average distance. Another approach using a complementary model was proposed by Myers [34], consisting of a separate risk model based on summary statistics of the training data. This auxiliary model estimates the likelihood of misclassifications, helping to assess the reliability of predictions. Although the results are encouraging, it is worth noting that using a secondary model may shift the reliability problem to that secondary model.

Methods based on complementary information incorporate expert knowledge to improve the reliability of predictions. In the works of Xie *et al.* [35] and Luu *et al.* [36], reliability is evaluated in nonclinical contexts by assessing whether classifiers produce consistent predictions when the input data are altered in ways that should not affect the outcome (such as adding irrelevant attributes or duplicating samples). This is done by defining expected behaviors under such transformations and comparing the output of the model before and after the modifications. Inconsistencies suggest implementation faults or unexpected behavior. Another method, applied in a clinical context, was proposed by Valente *et al.* [37]. The method combines an ensemble of decision rules based on clinical risk factors with a ML model for the prediction of mortality. Each rule contributes an acceptance score that indicates support for a specific outcome, and the reliability is computed as the difference between the mean acceptance of rules that support death and those supporting survival. Similarly, Zhao *et al.* [13] proposed a method in a clinical context where a set of rules (derived from ranges indicating when the model is more or less reliable) are formulated, and the reliability is then quantified as the proportion of these rules satisfied for a given prediction. Although these methods are designed for a specific domain, they demonstrate how domain knowledge can be embedded in the reliability assessment.

There are also methods that combine the **density and local fit** principles, using both the data and the behavior of the model. An example is the method proposed by Henriques *et al.*[38] in the context of cardiovascular risk assessment. The reliability of each prediction is assessed using three components. The density component evaluates how close similar training samples are to a new instance, applying a threshold to determine whether the surrounding region can be considered "dense". The data agreement component measures whether the actual labels of nearby training samples align with the prediction of the model for the new instance. The third component, the ML agreement component, follows the same principle as data agreement but uses the predictions of the model instead of actual labels. This combined reliability assessment has been validated in cardiovascular risk prediction tasks and has shown potential in distinguishing between reliable and unreliable predictions.

2.2 Selected Pointwise Methods

This section presents the pointwise reliability methods selected for evaluation in the case studies presented in Chapters 3, 4, and 5. These methods are designed to evaluate the pointwise reliability independently of the underlying model, producing a normalized scalar or adjusted to be within the range $[0, 1]$.

The selected methods were implemented considering their data driven nature and model agnostic applicability, making them suitable for production stages. Clustering

methods were included due to their recent lack of attention in the literature, as well as distance based methods, such as the weighted distance approach [28] and a method that combines multiple metrics to integrate density and local fit [38]. For the regression case study presented in Chapter 5, additional local fit methods (specific for regression tasks) based on the work of [30] were explored. The methods were first introduced by Correia *et al.* [39].

2.2.1 Subtractive

The first method uses subtractive clustering [40, 41], where each data point is evaluated as a potential cluster center based on the density of nearby points within a specified influence range (r_a). The algorithm consists of three main iterative steps. In step 1, the potential (density - D_i) of each data point x_i is computed with respect to all n data points, as follows:

$$D_i = \sum_{j=1}^{n} exp\left(-\left(\frac{\| x_i - x_j \|^2}{\left(\frac{r_a}{2}\right)^2}\right)\right) \tag{2.1}$$

The second step selects the point with the highest potential as a cluster center and reduces nearby potentials. The data point with the highest potential (D_1) is chosen as the first cluster center (x_{c1}). Then, potentials of points close to this cluster center are reduced to avoid redundant centers, according to the following:

$$D_i = D_i - D_{c1}\, exp\left(-\left(\frac{\| x_i - x_j \|^2}{\left(\frac{r_b}{2}\right)^2}\right)\right) \tag{2.2}$$

where D_{c1} is the density of the selected cluster center x_{c1}, and $r_b = squash\_factor \times r_a$ (where the default squash factor is set to 1.25).

The third step evaluates and selects subsequent cluster centers iteratively. After potential reduction, the next candidate point with the highest remaining potential is selected, and the following acceptance criterion is applied (if the candidate potential D_{ck} satisfies the equation):

$$\frac{d_{min}}{r_a} + \frac{D_{ck}}{D_1} \geq 1 \tag{2.3}$$

The point is accepted as a new cluster center (where d_{min} is the shortest distance from the cluster centers previously identified). If not, it is rejected and the search continues. This process repeats until no points exceed a predefined rejection threshold (set by a rejection ratio, whose default is 0.15).

The workflow for this method includes the following:

- Parameter Optimization: A grid search was performed over the r_a parameter

(starting with the values 0.05 and ending with 0.5, with 18 additional values between) to maximize the number of valid clusters, ensuring that no cluster contained fewer than a predefined minimum number (3) of members.

- Clustering and Assignment: Using the optimal r_a, the algorithm assigned each data point to the nearest cluster if it was within the range; otherwise, the points were marked as unassigned.
- Reliability Computation: Reliability for each new instance is computed based on the following:

$$R_x = \min\left(\frac{|C|}{minimum\ cluster\ size}, 1\right) \tag{2.4}$$

where $|C|$ is the size of the cluster to which the instance belongs. Points outside any cluster received a reliability score of 0.

The "minimum cluster size" parameters were determined as described in Section 2.2.10.

2.2.2 DBSCAN

The second method uses the DBSCAN algorithm [21], which identifies clusters based on the density of points in a neighborhood defined by two parameters, the distance threshold (ε) and the minimum number of points (MinPts). A point is classified as a core point if at least MinPts other points are located within its ε. Clusters are formed by expanding from core points and adding all points that can be reached within ε.

The algorithm is iterative. Starting from an unvisited point, its neighborhood is examined. If the point is a core point, a new cluster is formed and all density reachable points are added to it. Border points, which are not core points themselves but are located within the ε neighborhood of a core point, are assigned to the corresponding cluster. Points that are not core or border are labeled as noise.

The workflow for this method includes the following:

- Parameter Optimization: A grid search was performed on the ε (10 values ranging from 0.01 to 0.5 in equal intervals) and MinPts (4 values, starting at 2 and increasing by 2 in each iteration) to maximize the number of clusters.
- Clustering and Assignment: Using optimal parameters, DBSCAN assigned points to clusters or labeled them as noise.

Reliability Computation: For each new instance, the nearest point (a dense point identified by DBSCAN) was identified using the Euclidean distance. If the nearest core point belonged to a valid cluster and the instance was within the defined distance threshold (ε), the instance was considered part of the cluster. Otherwise, it was treated as unassigned. Reliability was computed as the size of the cluster divided by a predefined

threshold, "minimum cluster size", with scores capped at 1. The calculation follows the same principles as previously described for the subtractive clustering method (Equation 2.4). The "minimum cluster size" parameters were determined as described in Section 2.2.10.

2.2.3 Distance

The Distance method computes the reliability by focusing on the density of neighbors within a given distance threshold. For each new instance, the Euclidean distances to all training points are computed, and neighbors within the distance threshold are identified.

The workflow for this method is as follows:

- Distance Calculation: The Euclidean distances from the new instance to all points in the training data are computed;
- Neighbor Identification: Points belonging to the same predicted class as the new instance are identified, and the number of these points within the defined distance threshold is computed;
- Reliability Computation: Reliability is computed as the total number of neighbors of the same class within the distance, divided by the "minimum cluster size" parameter. Reliability scores are capped at 1 for sufficiently dense regions.

The distance threshold and the "minimum cluster size" parameters were determined as described in Section 2.2.10.

2.2.4 ICM

The fourth method presented is an adaptation of the interpretable confidence measure (ICM) framework, based on the work of Waa *et al*. [28]. The method evaluates the reliability of a prediction by analyzing the support or opposition provided by the nearest neighbors of a new instance in the training data.

The method relies on two main steps: identifying the nearest neighbors of a new instance and using their distances to compute a weighted confidence score. Closer neighbors are given more influence, reflecting their proximity to the instance.

The workflow for this method is as follows:

- Neighbor Selection: The k-nearest neighbors of the new instance are identified from the training data using the Euclidean distance. These neighbors are then divided into two groups: S^{+}, representing neighbors that share the predicted label, and S^{-}, representing neighbors with a label different from the predicted one.

- Sigma Calculation: σ is computed as the mean squared distance between the new instance x and its neighbors as shown in

$$\sigma = \frac{1}{k} \sum_{x_i \in N(x)} \| x - x_i \|^2 , \tag{2.5}$$

  where $N(x)$ is the set of all k-nearest neighbors.

- Weight Calculation: Each neighbor is assigned a weight based on its distance to the new instance, with the closer neighbors having greater influence. The weight of a neighbor x_i is computed using:

$$W(x, x_i) = exp\left(- \left(\frac{\| x - x_i \|^2}{\sigma} \right)^2 \right) \tag{2.6}$$

- Weighted Contributions for Support and Opposition: The average weighted contributions of the supporting (S^+) and opposing (S^-) neighbors are computed as follows:

$$W^+ = \frac{1}{|S^+|} \sum_{x_i \in S^+} W(x, x_i), \;\; W^- = \frac{1}{|S^-|} \sum_{x_i \in S^-} W(x, x_i) \tag{2.7}$$

  where $|S^+|$ and $|S^-|$ denote the number of elements in S^+ and S^-, respectively.

- Reliability Computation: The confidence score, $C(x)$, is computed as the difference between the weighted contributions of the supporting and opposing neighbors:

$$C(X) = W^+ - W^- \tag{2.8}$$

- The confidence score is then rescaled to the range $[0, 1]$ using:

$$R = \begin{cases} 0, & if\ C(x) < 0 \\ C(x), & otherwise \end{cases} \tag{2.9}$$

The value of k used in this method was selected using the procedure described in Section 2.2.10.

2.2.5 Density and Local Fit

The Density and Local fit method is an implementation based on the work of Henriques *et al*. [38], which combines density measures with local fit principles to assess pointwise reliability. The method calculates the reliability for a new instance by integrating three components: density, data agreement, and machine learning agreement. These components evaluate the consistency of the new instance within the training data, the agreement with its neighbors, and the consistency of the model predictions.

The workflow implemented for this method is as follows:

- Density Component: The density is computed by counting how many training data points fall within a predefined (euclidean) distance threshold around the new instance. The count is limited to a maximum number of neighbors ("minimum cluster units"), ensuring that the density score is normalized to a range between 0 and 1;
- Data Agreement Component: The data agreement is computed as the proportion of neighbors that share the same label as the predicted label of the new instance based on the distance threshold;
- ML Agreement Component: The ML agreement measures the consistency between the predictions of the model for the neighbors and the predicted label of the new instance (based on neighbors with consistent predictions divided by total neighbors within the threshold);
- Reliability Computation: The final reliability score is computed by multiplying the three components. This formulation ensures that all three aspects contribute to reliability.

The predefined distance threshold and "minimum cluster units" were selected based on the strategy described in Section 2.2.10.

2.2.6 CONFIVE

The CONFIVE method was proposed by Briesemeister *et al.* [30] as a model agnostic pointwise reliability estimator for regression problems. It assesses reliability by evaluating the variance in the target values of the k nearest neighbors of a given new instance.

Since variance can take any positive value, the authors propose a normalization strategy to map the result to the $[0, 1]$ interval. Specifically, the normalized score of a new instance is computed as the fraction of training instances with lower (less reliable) CONFIVE scores than the new instance.

The workflow for this method is as follows:

- Neighbor Identification: The k nearest neighbors of a new instance are identified using the Euclidean distance;
- Score Computation: For the new instance, the unbiased sample variance of the target values of its k neighbors is computed as:

$$\text{CONFIVE} = 1 - \frac{1}{k-1}\sum_{i=1}^{k}(\bar{y} - y_i)^2 \tag{2.10}$$

  where $\bar{y}$ is the mean of the target values of the k neighbors, and y_i is the target

value of the i-th neighbor;

- Normalization: The final reliability score is determined by computing the proportion of training instances whose CONFIVE scores are smaller than the new instance.

The parameter k (number of nearest neighbors) was selected using the strategy described in Section 2.2.10.

2.2.7 CONFINE

The CONFINE method, also proposed by Briesemeister *et al.* [30] as a pointwise metric for regression tasks, differs from CONFIVE by estimating reliability based on the local prediction error rather than on the variance of target values. The method evaluates the mean squared error (MSE) between the predictions of the model and the actual labels of the k nearest neighbors of a new instance.

The same normalization strategy described in Section 2.2.6 is applied (using CONFINE), mapping the raw scores to the $[0, 1]$ interval.

The workflow implemented for this method is as follows:

- Neighbor Identification: The k nearest neighbors of the new instance are identified using Euclidean distance;
- Score Computation: For each neighbor, the model prediction is obtained, and the mean squared error (MSE) is computed between the predicted and actual label values:

$$\text{CONFINE} = 1 - \frac{1}{k}\sum_{i=1}^{k}(y_i - \hat{y}_i)^2 \tag{2.11}$$

  where y_i denotes the true label and $\hat{y}_i$ represents the model prediction for the i-th neighbor;

- Normalization: The final reliability score is obtained by computing the proportion of training instances whose CONFINE scores are smaller than the new instance.

The parameter k (number of nearest neighbors) was selected using the procedure described in Section 2.2.10.

2.2.8 denCONFIVE

The denCONFIVE method combines the CONFIVE method with distance based density component. In practice, the method combines the assessment of the local variance (as in the CONFIVE method) with a simple density verification.

The workflow for this method is:

- Neighbor Identification: The k nearest neighbors of the new instance are identified using Euclidean distance;
- CONFIVE Score Computation: The CONFIVE score is computed as the unbiased sample variance of the target values of the k nearest neighbors, following the full procedure described in Section 2.2.6, including the normalization step;
- Density Component: The density is computed by counting the number of training instances within a predefined Euclidean distance threshold from the new instance. The count is capped at the same k value used in the CONFIVE computation;
- Reliability Computation: The reliability is computed by multiplying the two components.

Parameters k (number of neighbors) and the "distance threshold" were selected using the procedure described in Section 2.2.10.

2.2.9 iqrDenCONFIVE

The last method follows the same procedure as denCONFIVE, described in Section 2.2.8, where the CONFIVE score is normalized using a different approach.

Instead of computing the normalized score as the proportion of training instances with lower CONFIVE values than a new instance, iqrDenCONFIVE scales the raw variance directly using a Min-Max scaling strategy (scaling the values to the interval $[0, 1]$). This alternative addresses a limitation of the original normalization strategy, which may assign identical reliability scores to instances with substantially different raw variance values, reducing the ability of the method to discriminate between levels of reliability.

The scaling is performed by identifying, for each training instance, the k nearest neighbors (using the same k as in CONFIVE) and computing the unbiased sample variance of their target values.

To reduce the influence of extreme outliers in the variance distribution, which could bias the scaling process, an interquartile range (IQR) based filtering strategy is applied. All variance values above $Q3 + 1.5 \times$ IQR are excluded, and a Min-Max scaler is then fitted using the remaining values to normalize the local variance for each new instance. As shown in Figure 2.1, for the problem described in Chapter 5, the distribution of local variances exhibits a long right tail, indicating the presence of high variance outliers and supporting the use of IQR based filtering prior to normalization.

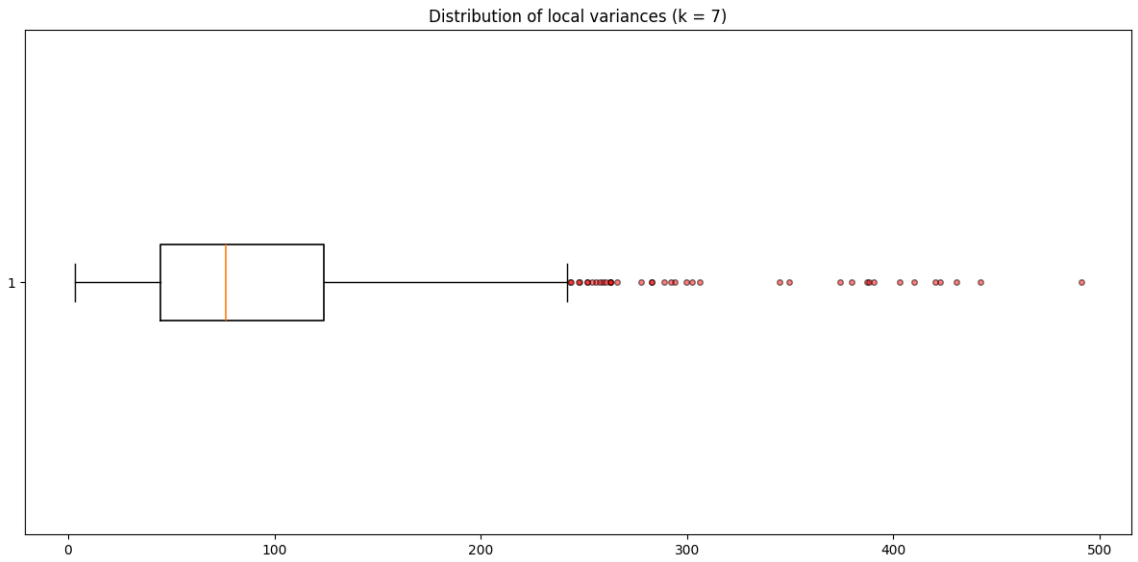


Figure 2.1: Boxplot of the distribution of local variances for all training instances using 7 neighbors.

The reliability score is then computed as:

$$C(x) = 1 - \text{scaled\_variance} \tag{2.12}$$

The final score is adjusted using:

$$R(x) = \begin{cases} 0, & \text{if } C(x) < 0 \\ 1, & \text{if } C(x) > 1 \\ C(x), & \text{otherwise} \end{cases} \tag{2.13}$$

This formulation ensures that the final reliability score $R(x) \in [0, 1]$, where the higher reliability corresponds to a lower local variance. Additionally, it penalizes instances whose variance falls outside the trimmed upper range observed in the training data.

For the final reliability score, this method retains the same structure as denCONFIVE combining the normalized CONFIVE score with a density component.

The parameters k (number of neighbors) and the "distance threshold" were selected using the procedure described in Section 2.2.10.

2.2.10 Parameter Selection

The performance of pointwise reliability methods often depends on parameters such as the distance threshold, the minimum cluster unit, or the number of neighbors.

Some studies have proposed strategies to determine these parameters [14]. For example, Sahigara *et al.* [42] explored a range of global thresholding techniques based

on distances between training instances, including the use of the maximum observed distance, fixed multiples of the average distance, and percentile based thresholds. To select the number of neighbors (k), Briesemeister *et al.* [30] proposed a cross-validation approach with two folds, selecting the value of k that produces the best results for the method.

A more recent strategy, used by Henriques *et al.* [38], is based on the distribution of pairwise distances between all training instances. In this approach, a global distance threshold is defined using a selected percentile of the distance distribution, and the "minimum cluster unit" is computed as the average number of neighbors found within this threshold.

In the present work, a similar percentile based strategy [38] is adopted. For each case study (Chapters 3, 4, and 5), pairwise distances between training instances are computed and a relatively low percentile is selected to define the distance threshold. This selection aims to ensure that reliability assessments focus on dense regions, while still preserving enough neighbors to support the evaluations. The specific threshold values vary depending on the characteristics of the dataset.

Figure 2.2 illustrates this process for the case study presented in Chapter 4. The selected threshold percentile corresponds to a small pairwise distance between instances.

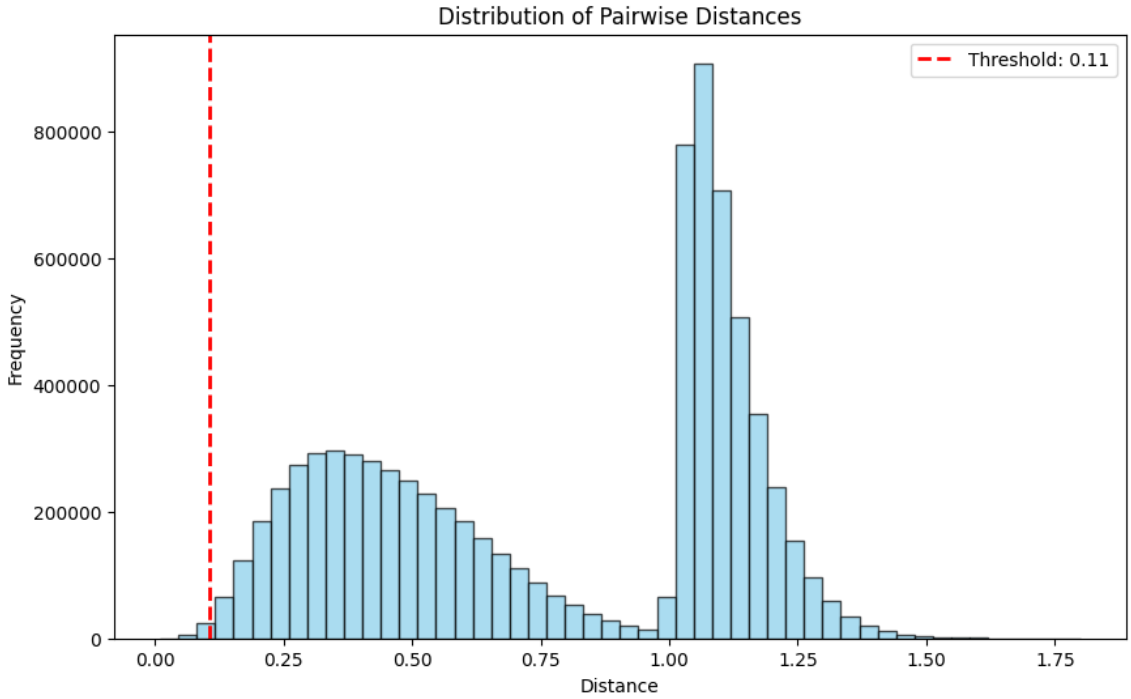


Figure 2.2: Distribution of pairwise distances between training instances. The vertical line marks the selected percentile threshold.

Table 2.1 further illustrates the relationship between the selected percentiles, their corresponding distance thresholds, and the resulting minimum cluster unit sizes for the

same case study. For example, the 0.25 percentile was selected as it balances a small distance threshold while showing a decent number of neighbors (a value believed to be neither too low nor too high for the problem).

Table 2.1: Relationship between selected percentiles of pairwise distances, their corresponding distance thresholds and the resulting "minimum cluster unit" values (average number of neighbors within each threshold).

Percentile	Distance Threshold	Minimum Cluster Unit
0.1	≈ 0.087	3
0.25	≈ 0.107	9
0.5	≈ 0.124	19
5	≈ 0.223	198

2.3 Benchmark

This section describes the benchmark used to assess the pointwise reliability methods. Related works [38, 43] have adopted a similar evaluation, in which the predictions of the model are grouped into reliability intervals using a pointwise reliability method, and the number of errors in each interval is analyzed to assess the effectiveness of reliability estimation. These studies typically apply pointwise reliability methods to models optimized for the problem.

In this work, the benchmarking strategy follows that presented in a previous study [39]. Unlike most related work, the benchmark was applied using a modest ML model, with a substantial presence of errors. This choice ensures the inclusion of a substantial proportion of both correct and incorrect predictions, reducing the bias that could be introduced when applying these metrics to models with very high accuracy.

Predictions are grouped into intervals based on their reliability scores, using 10% intervals (*e.g.* $[0.00, 0.10]$, $]0.90, 1.00]$). Within each interval, the predictions are classified according to their correctness and true class, as follows:

- Correct (0): label is 0 and prediction is correct.
- Incorrect (0): label is 0 and prediction is incorrect.
- Correct (1): label is 1 and prediction is correct.
- Incorrect (1): label is 1 and prediction is incorrect.

This categorization supports the evaluation of how well the methods distinguish between high and low reliability predictions. It also helps to verify whether higher reliability scores are associated with lower error rates, as expected. In addition, analyzing the distribution of errors between reliability intervals helps identify inconsistencies or limitations in the methods.

In the regression case study (Chapter 5), a similar approach is used. However, since the targets are continuous, correctness is defined by whether the predicted value falls within a defined tolerance threshold around the true value (*e.g.* [true $\pm \delta$], with δ selected based on domain knowledge). The evaluation also includes an analysis of the absolute prediction error across reliability intervals, helping to assess how reliability scores correspond to variability in prediction error magnitude.

Although visualizations are not included in this document, techniques such as the t-distributed Stochastic Neighbor Embedding [44] were used during the study to support the interpretation of the results. These visualizations were introduced in [39] as part of the original benchmarking framework. In that context, visual patterns of reliable and unreliable predictions provided additional support to the numerical analysis by showing how prediction errors and class distributions are organized in the dataset. In the present work, such visualizations are not analyzed[1] due to the use of repeated random data splits, which result in varying training and test sets across iterations and limits the consistency of visual representations.

[1] Examples of the visualizations are presented in Appendix A.

3 Case Study: Patient Treatment

This chapter presents a case study to evaluate pointwise reliability methods in a classification context. The analysis is based on the work introduced in [39], where the same dataset and pointwise reliability methods are evaluated. The code used to implement these methods and the resulting visualization is available in the GitHub repository [45].

The evaluation includes five methods based on the density and local fit principles, specifically Subtractive Clustering, DBSCAN, Distance Based, ICM, and Density and Local Fit. The details of the implementation of the pointwise reliability methods are described in Chapter 2.2. The evaluation follows the benchmark detailed in the previous chapter, where the predictions are grouped into reliability intervals and their correctness is assessed based on the true labels.

3.1 Dataset and Model

The dataset used in this study is the "Patient Treatment Classification Dataset" [46], which contains anonymized health records from a private hospital in Indonesia. Each instance represents a patient and includes laboratory test results, demographic features, and a binary label indicating whether the treatment decision was "in care" or "out care".

This public dataset contains 4412 instances. The class distribution is moderately imbalanced, with approximately 60% of samples labeled as "out-care" (class 0). The original set of features includes numerical laboratory measurements, such as hemoglobin, erythrocyte, leukocyte, thrombocyte, hemocyte (HCT), mean corpuscular hemoglobin (MCH), mean corpuscular hemoglobin concentration (MCHC), mean corpuscular volume (MCV), age, and categorical variable sex.

A Pearson correlation analysis was performed to evaluate the potential redundancy between features. As shown in Figure 3.1,where the label is represented by SOURCE, hemocrit showed a strong correlation with hemoglobins, and MCH was highly correlated with MCHC. Given this overlap, hemocrit and MCH were removed from the feature set to avoid introducing redundant information into the model.

The final feature set (Table 3.1) includes seven numerical variables (hemoglobin, erythrocyte, leukocyte, thrombocyte, MCHC, MCV, age) and one categorical variable (sex).

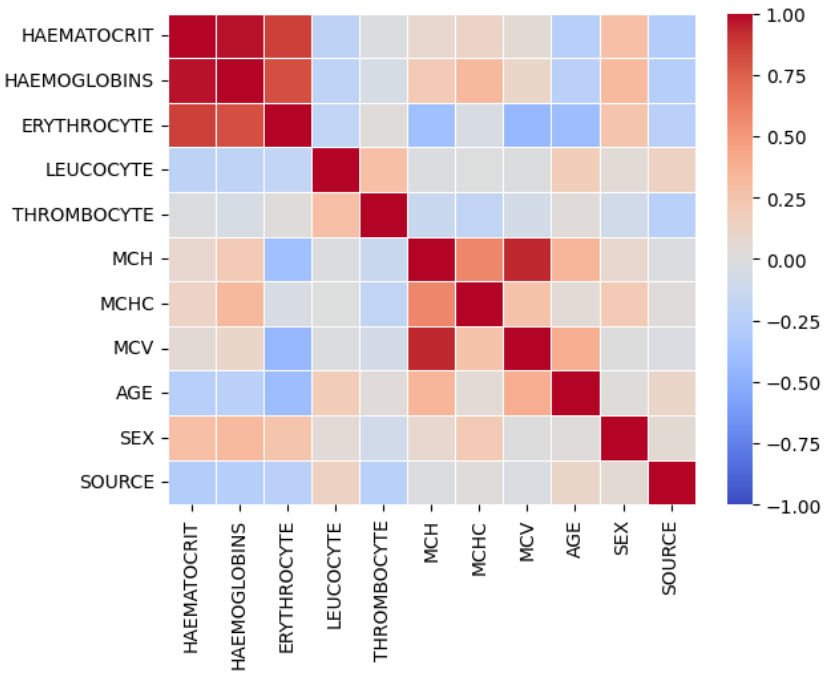


Figure 3.1: Pearson correlation matrix.

The summary of descriptive statistics of the numerical features is shown in Table 3.1.

Table 3.1: Descriptive statistics of numerical features.

Feature	**Mean**	**Std**	**Min**	**Max**
Haemoglobins	12.7	2.0	3.8	18.9
Erythrocyt	4.5	0.7	1.4	7.8
Leucocyte	8.7	4.9	1.1	76.6
Thrombocyte	257.5	114.0	10.0	1183.0
MCHC	33.3	1.2	26.0	39.0
MCV	84.6	6.8	54.0	115.6
AGE	46.6	21.7	1	98

To evaluate pointwise reliability, the same procedure was repeated ten times. In each iteration, the dataset was randomly split into a training set (80%) and a test set. A simple logistic regression classifier [47] was trained[1], and all pointwise reliability methods were applied to its predictions. All numerical features were scaled to the range $[0, 1]$, and the categorical feature sex was encoded as binary (0 for female, 1 for male).

It should be noted that the classifier is not the central component in this study. Since the objective is to evaluate the behavior of the pointwise reliability methods rather than to optimize the performance of the model, the classifier is treated as a black box. Using a complex model that achieves perfect accuracy would be unrealistic in a real world scenario and unsuitable for the benchmark, as it would leave little room for differen-

[1]The classifier was implemented using the `LogisticRegression` class from the `scikit-learn` Python library.

tiating between reliable and unreliable predictions. In this context, a simple model is sufficient to support the evaluation.

The classifier achieved an accuracy of 71.6% $\pm$ 1.2 and an area under the receiver operating characteristic curve of 0.758 $\pm$ 0.016 (mean $\pm$ standard deviation over 10 iterations), indicating moderate predictive performance, which is adequate for the benchmark evaluation.

3.2 Pointwise Reliability

This section assesses pointwise reliability methods applied to the patient treatment case study. The analysis includes five methods based on the density and local fit principles, namely Subtractive Clustering, DBSCAN, Distance Based, ICM, and Density and Local Fit, all described in Chapter 2.2. These methods are model agnostic and can be applied independently of the classification model.

All methods are evaluated using the benchmark strategy presented in Chapter 2.2. Predictions are grouped into reliability intervals and the distribution of errors is analyzed to assess the ability of each method to distinguish between reliable and unreliable predictions. Each method is evaluated over 10 iterations. The original benchmark design can be found in Correia *et al.*[39], where a single validation split was used and the analysis was complemented with t-distributed Stochastic Neighbor Embedding (t-SNE) visualizations. Although multiple runs prevent their use in the current aggregated results, visualizations from a single fold are included in Appendix A to support interpretation. For example, the visualizations reveal that clustering based methods fail to form meaningful patterns across reliability intervals, while the Density and Local Fit method assigns high reliability to predictions in homogeneous regions, supporting the quantitative analysis presented in this chapter.

To ensure comparability among methods, the threshold parameters were selected using the procedure described in Section 2.2.10, which uses the 0.25 percentile of the pairwise distance distribution. This resulted in a distance threshold of 0.107 and a minimum cluster size of 9.

3.2.1 Subtractive

Subtractive clustering (Section 2.2.1) serves as the first pointwise reliability method under evaluation. Figure 3.2 shows the error rates in each reliability interval, separated by predicted class. Each boxplot summarizes the variability in error across iterations and allows an assessment of how prediction correctness is distributed along the reliability intervals. For class 1, the error rate is consistently zero in all intervals where predictions are present. For class 0, there is no clear decreasing trend in error as the reliability

increases. In particular, the highest reliability interval does not present a lower error than the lowest one, and in several reliability intervals, its variance is substantial.

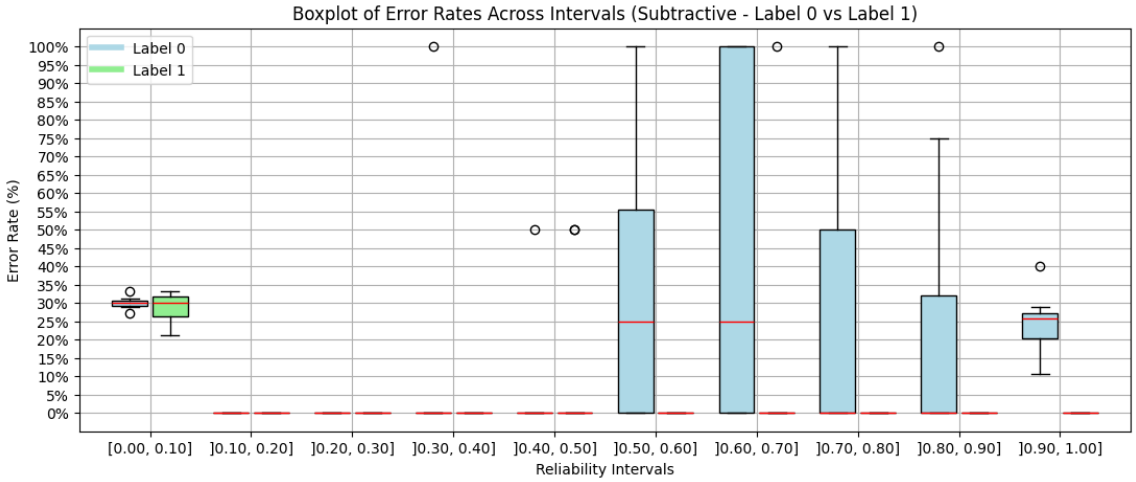


Figure 3.2: Boxplot of error rates across reliability intervals using the Subtractive method, separated by predicted label.

The distribution of the predictions across the intervals is presented in Table 3.2, which reports the average number of instances per iteration assigned to each reliability interval for both predicted classes.

Table 3.2: Number of predictions for each label ($Y = 0, Y = 1$) in each reliability interval using the Subtractive method.

Interval	**Y = 0**	**Y = 1**
$[0.0, 0.1]$	298.8 ± 14	108.1 ± 12
$]0.1, 0.2]$	0 ± 0	0 ± 0
$]0.2, 0.3]$	0 ± 0	0 ± 0
$]0.3, 0.4]$	0.4 ± 0.6	0.2 ± 0.6
$]0.4, 0.5]$	0.8 ± 0.6	0.4 ± 0.8
$]0.5, 0.6]$	1.8 ± 1.9	0.4 ± 0.8
$]0.6, 0.7]$	1 ± 1	0.4 ± 0.5
$]0.7, 0.8]$	1 ± 1	0.2 ± 0.4
$]0.8, 0.9]$	1.8 ± 2.2	0.5 ± 0.8
$]0.9, 1.0]$	24.8 ± 7	1.4 ± 0.9

Most predictions are concentrated in the first interval $[0.0, 0.1]$, indicating that most instances were not assigned to any cluster and thus received the lowest reliability score. A smaller but consistent group of predictions appears in the last interval $]0.9, 1.0]$, which corresponds to high reliability. The remaining intervals contain very few predictions, with many intervals showing a number of instances close to zero.

These results indicate that the subtractive clustering method has limitations in estimating reliability. Although class 1 predictions show zero error when a reliability score is assigned, they are almost entirely absent from most intervals, suggesting that the

method fails to capture a meaningful structure for this class. For class 0, predictions are more widely distributed, but error rates do not decrease with increasing reliability. In particular, the high reliability group does not achieve better predictive accuracy than the low reliability group.

In general, the method struggles to distinguish between reliable and unreliable predictions. This is reflected in the limited use of intermediate reliability intervals, as well as the lack of alignment between reliability and error.

3.2.2 DBSCAN

The DBSCAN method, introduced in Section 2.2.2, estimates the reliability of predictions based on dense regions in the feature space. The results follow a similar pattern to those observed with the Subtractive Clustering method, where predictions tend to cluster at the extremes of the reliability interval.

Figure 3.3 shows the error rates per reliability interval, separated by predicted class.

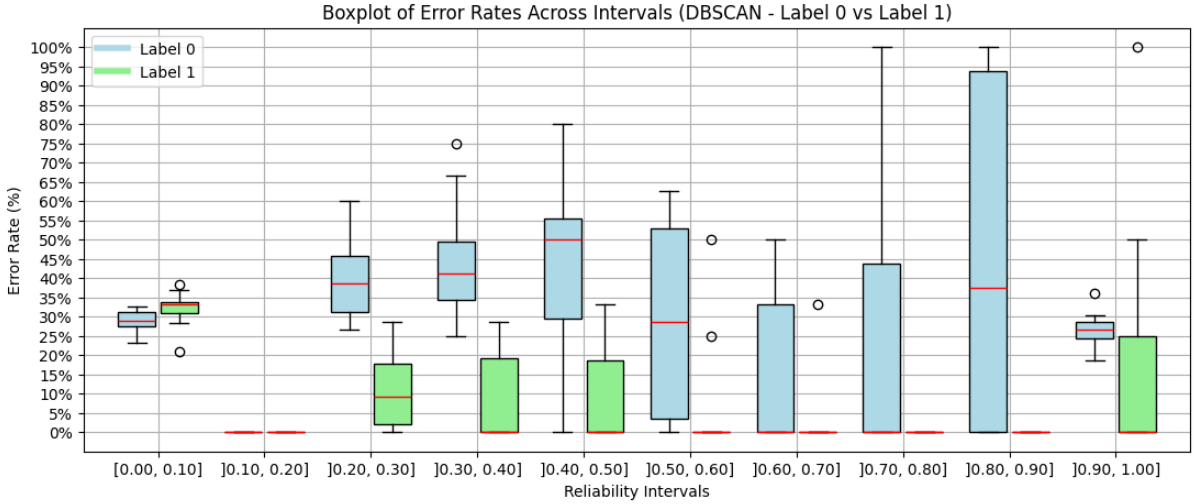


Figure 3.3: Boxplot of error rates across reliability intervals using DBSCAN [39].

For class 1, the median error rate remains zero in nearly all intervals. However, the variance within each interval is considerable (and considering that most intervals contain no or very few predictions for this class), the error estimates appear to be unstable. For class 0, the error rate shows substantial variability and does not follow a decreasing trend as reliability increases. In particular, the highest reliability interval $]0.9, 1.0]$ shows error levels comparable to lower intervals, indicating that the method does not consistently assign high reliability to correct predictions.

Table 3.3 presents the distribution of the predictions in the reliability intervals. Most predictions are concentrated in the interval $[0.0, 0.1]$, which corresponds to predictions that are not assigned to any cluster. A second cluster appears in the highest interval $]0.9, 1.0]$, mainly composed of predictions of class 0. Intermediate intervals are

sparsely populated for both classes, particularly for class 1, which rarely receives reliability scores above 0.3.

Table 3.3: Number of predictions for each label in every reliability interval using the DBSCAN method.

Interval	**Y = 0**	**Y = 1**
[0.0, 0.1]	214.3 ± 18	90.4 ± 9.6
]0.1, 0.2]	0 ± 0	0 ± 0
]0.2, 0.3]	20.3 ± 5.1	8.1 ± 2.5
]0.3, 0.4]	11 ± 5.1	5.2 ± 2.9
]0.4, 0.5]	7.5 ± 3.3	2.2 ± 1.3
]0.5, 0.6]	5.1 ± 2.6	2.6 ± 1.6
]0.6, 0.7]	4.7 ± 1.7	1.3 ± 0.9
]0.7, 0.8]	2.5 ± 2.5	0.3 ± 0.6
]0.8, 0.9]	1.9 ± 1.5	0.2 ± 0.4
]0.9, 1.0]	63 ± 13.6	1.3 ± 1

These results indicate that the DBSCAN method, similarly to the Subtractive method, struggles to produce reliability scores that align with prediction correctness. Although class 1 predictions are mostly correct when a score is assigned, they are nearly absent from most intervals. For class 0, the error rate does not improve with higher reliability.

The high concentration of predictions in the lowest interval shows that DBSCAN tends to classify many instances as noise. While this behavior encourages conservative reliability estimates, it also results in limited granularity and poor discrimination between reliable and unreliable predictions. Overall, the method shows weak performance.

3.2.3 Distance

The distance based method, introduced in Section 2.2.3, estimates reliability based on the number of neighboring instances within a fixed radius that share the predicted class. Compared to previous methods, it assigns reliability scores with a more continuous spread across the entire interval range.

Figure 3.4 shows the error rates in the reliability intervals, separated by predicted class. For class 1, the error rate shows a decreasing trend as reliability increases, with high reliability intervals achieving consistently low errors. This behavior aligns with the expected correlation between reliability and correctness. However, for class 0, the error rate is less consistent. The lowest and highest reliability intervals exhibit similar error levels, and there is no clear downward trend between the intervals. Furthermore, the variance in error rates remains high in several intervals, suggesting that the method does not consistently assign a higher reliability to more accurate predictions for this class.

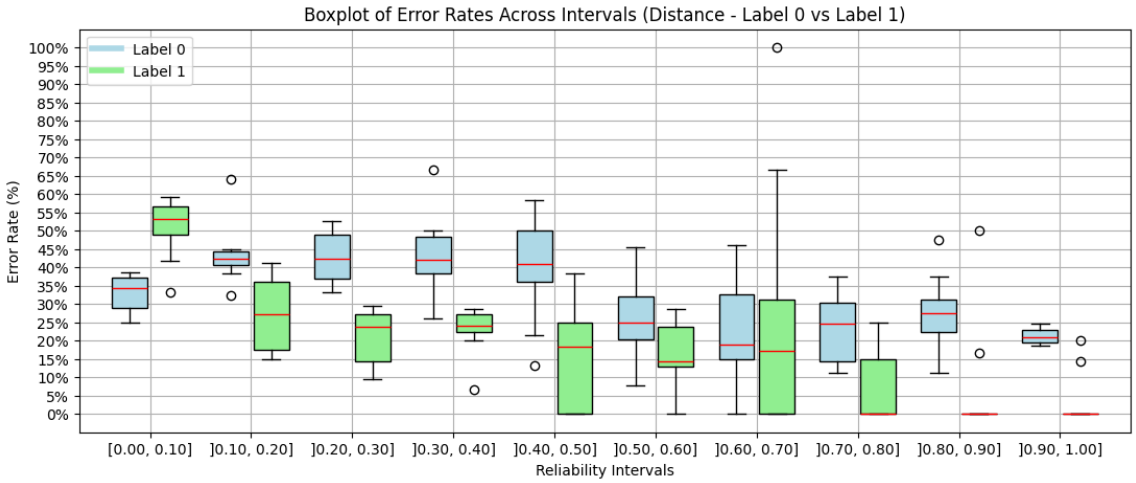


Figure 3.4: Boxplot of error rates across reliability intervals using the Distance Based method, separated by predicted label.

The distribution of the predictions in the reliability intervals are presented in Table 3.4. The predictions are more evenly spread across the full range compared to the Subtractive and DBSCAN methods. A moderate number of predictions are found in low reliability intervals, while the largest group is located in the highest interval $]0.9, 1.0]$, particularly for class 0. Intermediate intervals are also populated, indicating that the method can assign intermediate reliability scores with a reasonable frequency.

Table 3.4: Number of predictions for each label in every reliability interval using the Distance Based method.

Interval	**Y = 0**	**Y = 1**
$[0.0, 0.1]$	55.5 ± 8.1	31 ± 6.4
$]0.1, 0.2]$	36.8 ± 7.6	18.2 ± 3.3
$]0.2, 0.3]$	24.6 ± 5.3	17.6 ± 5
$]0.3, 0.4]$	20.5 ± 4.8	13.1 ± 3.1
$]0.4, 0.5]$	14.8 ± 4.6	7.9 ± 3.3
$]0.5, 0.6]$	15.1 ± 3.9	6.5 ± 2.1
$]0.6, 0.7]$	13 ± 3.6	3.9 ± 1.6
$]0.7, 0.8]$	10.8 ± 3.2	4.1 ± 1.5
$]0.8, 0.9]$	12 ± 3.4	2.5 ± 1.5
$]0.9, 1.0]$	127 ± 14.9	6.8 ± 3.1

Similarly to the previous methods, there is no clear structure in how reliability scores relate to the overall data distribution. This is particularly visible in the t-SNE visualization (Appendix A), where many highly reliable predictions appear in regions that contain instances from both classes. This suggests that the method does not effectively capture the class boundaries. However, instances with fewer nearby neighbors tend to receive low reliability scores. This reflects the ability of the method to detect local density, but also reveals a limitation in using proximity alone to determine reliability, without assessing whether dense regions are homogeneous.

While the method assigns reliability scores with greater granularity and offers a more balanced prediction distribution than previous approaches, it still has difficulty distinguishing between reliable and unreliable predictions (especially for class 0).

3.2.4 ICM

The ICM method, introduced in Section 2.2.4, estimates the reliability of a prediction by comparing the influence of neighboring instances that support the predicted class.

Figure 3.5 shows the error rates in the reliability intervals, separated by label.

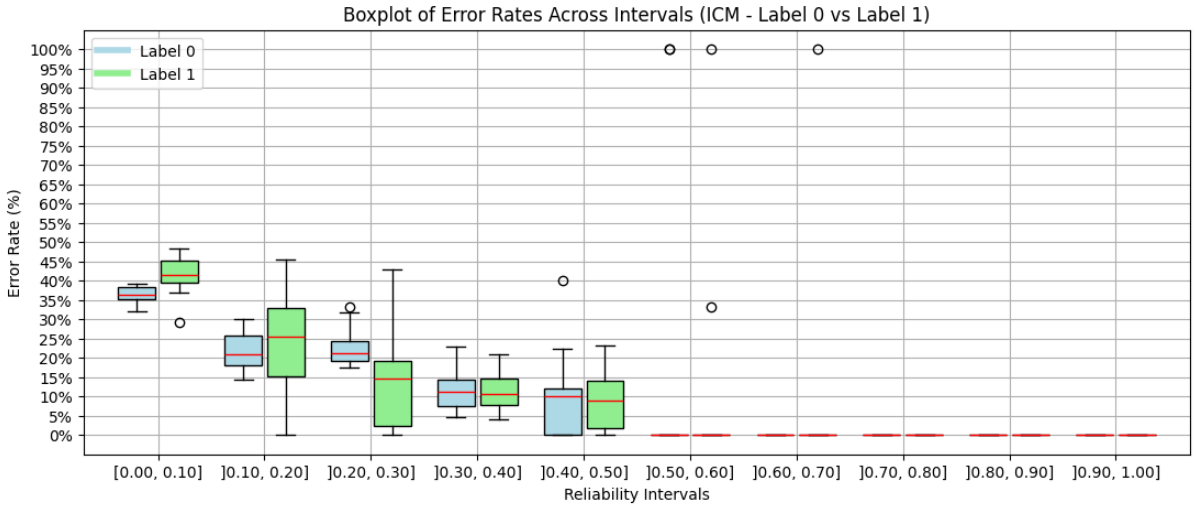


Figure 3.5: Boxplot of error rates across reliability intervals using ICM.

As shown in Figure 3.5, although predictions are concentrated in the lower half of the reliability interval ($[0.00, 0.50]$), there is a decreasing trend in error rates for both classes as the reliability interval increases. This indicates that the method can successfully identify intervals where the predictions are more reliable. A fluctuation (error variability) appears in the interval $]0.40, 0.50]$, but does not affect the overall trend.

Table 3.5 presents the distribution of the predictions in the reliability intervals. Most predictions are assigned to the interval $[0.0, 0.1]$, and only a small number appear above 0.5. No predictions are placed in the highest reliability intervals. This distribution is likely a consequence of the strict model criteria, which makes a reliable classification for the interval $[0.5, 1.0]$ practically nonexistent.

These results suggest that the ICM method is effective in distinguishing between more and less reliable predictions, as reflected by the consistent drop in error rates with increasing reliability. However, it also shows that the method is too restrictive, which may limit its practical use.

Table 3.5: Number of predictions for each label in every reliability interval using the ICM method.

Interval	Y = 0	Y = 1
[0.0, 0.1]	211.6 ± 7.2	55.6 ± 7.5
]0.1, 0.2]	45 ± 10.2	14.8 ± 4.1
]0.2, 0.3]	22.7 ± 3.2	10.8 ± 4.2
]0.3, 0.4]	43.7 ± 6.6	17.2 ± 4
]0.4, 0.5]	7.1 ± 2.3	11.5 ± 3.1
]0.5, 0.6]	0.3 ± 0.4	1.1 ± 0.8
]0.6, 0.7]	0 ± 0	0.3 ± 0.6
]0.7, 0.8]	0 ± 0	0.3 ± 0.6
]0.8, 0.9]	0 ± 0	0 ± 0
]0.9, 1.0]	0 ± 0	0 ± 0

3.2.5 Density and Local Fit

The Density and Local Fit method, described in Section 2.2.5, combines local density estimation with neighborhood agreement to assess the reliability of individual predictions. Among the evaluated methods, it achieves the most balanced distribution of reliability scores and the most consistent relationship between reliability and prediction accuracy.

Figure 3.6 shows the error rates in the reliability intervals, separated by predicted class. For both class 0 and class 1, the error rate decreases consistently as the reliability increases. In the highest reliability intervals, class 1 reaches zero error, while class 0 maintains one of its lowest error levels. Although fluctuations appear in intermediate intervals, the overall trend shows that the error rate decreases as the reliability interval increases.

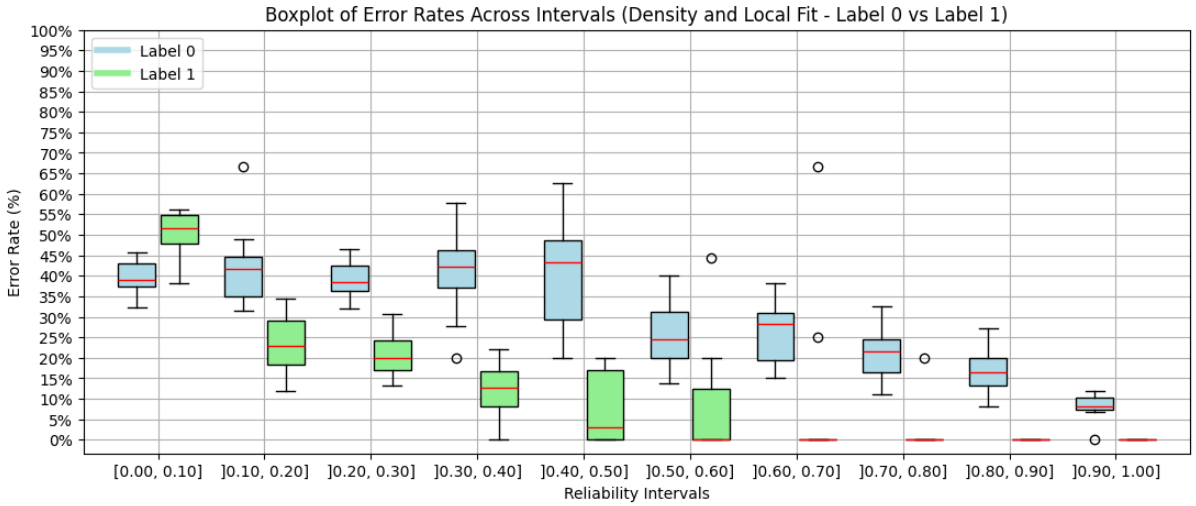


Figure 3.6: Boxplot of error rates across reliability intervals using the Density and Local Fit method, separated by predicted label [39].

Table 3.6 shows the number of predictions per reliability interval. Compared to other methods, the predictions are more evenly distributed throughout the full reliability range. Although a moderate number of predictions fall into the lower intervals, the higher intervals also show a significant number of instances, particularly for class 0. This pattern demonstrates the ability of the method to assign reliability values with greater granularity without being overly conservative or skewed toward low scores.

Table 3.6: Number of predictions for each label in each reliability interval using the Density and Local Fit method.

Interval	**Y = 0**	**Y = 1**
[0.0, 0.1]	65 ± 8.1	40.3 ± 7.8
]0.1, 0.2]	36.6 ± 6.7	21.4 ± 5.1
]0.2, 0.3]	25.5 ± 4.8	15.5 ± 2.9
]0.3, 0.4]	21.4 ± 4.5	12.0 ± 3.8
]0.4, 0.5]	17.2 ± 4.9	8.4 ± 3.7
]0.5, 0.6]	22 ± 5.4	5.3 ± 2.2
]0.6, 0.7]	30.6 ± 7.6	3.5 ± 1.5
]0.7, 0.8]	44.1 ± 5.9	2.9 ± 1.2
]0.8, 0.9]	43.3 ± 6.8	1.1 ± 1.6
]0.9, 1.0]	24.7 ± 3.7	1.2 ± 0.7

These results confirm that the method can distinguish between reliable and unreliable predictions more effectively than the other approaches evaluated. Its ability to distribute predictions across reliability intervals, coupled with a clear decrease in error rates, shows that the method captures a meaningful structure in the feature space.

Despite the overall consistency, a few misclassifications still occur in high reliability intervals, and some intermediate intervals show mild variability in error. These could potentially be addressed by refining internal parameters, such as adjusting neighborhood thresholds or using different parameters for both classes.

3.2.6 Discussion of the Results

The results of the five evaluated methods reveal distinct patterns in how prediction reliability is assessed. Clustering based methods, such as Subtractive Clustering and DBSCAN, tend to assign most predictions to the lowest reliability interval $[0.0, 0.1]$, with sparse or inconsistent assignments to higher intervals. These methods also present unstable error rates across intervals, which often shows high error variance even in high reliability intervals. These findings suggest that methods relying on clustering may not be sufficient to assess pointwise reliability.

The distance based method, compared to the clustering methods, shows that the predictions are spread more evenly across reliability intervals. For the less prevalent class (class 1), the method shows a general decrease in error rates as the reliability increases.

However, for class 0, the trend is less consistent. The high error variance in intermediate and high reliability intervals highlights a lack of robustness, suggesting that proximity alone does not fully capture the reliability of the predictions.

The ICM method shows an error rate decrease as reliability increases, and the method shows a relatively stable behavior across iterations. However, its conservative algorithm leads to very few predictions classified in high reliability intervals. This restrictiveness, while reducing overconfidence, limits its practical use.

Among all methods, the Density and Local Fit approach achieves the most consistent results. The predictions are well distributed across the reliability intervals and the error rates decrease steadily for both classes. For class 0, the error rate drops from approximately 39% in the lowest interval to 7% in the highest. For class 1, the error rate reaches 0% in the $[0.8, 0.9]$ and $[0.9, 1.0]$ intervals. Although the number of class 1 predictions in these intervals is small, the consistency of this trend across iterations supports the effectiveness of the method. These results align with the findings reported in [38], which showed that the method can separate more reliable from less reliable predictions.

This case study shows the limitations of clustering based methods when used in isolation and the value of methods that combine density estimation with local fit principles. Although some approaches demonstrate strengths in specific scenarios, such as the ICM method which shows a decreasing trend in prediction errors as reliability increases and has achieved favorable results in other implementations [25, 43]. However, in this context, only the Density and Local Fit method maintain consistent performance across classes and reliability intervals.

Future improvements may include adjusting thresholds or refining the pointwise reliability algorithms. Additional methods based on the local fit principle should also be considered to increase the scope and relevance of the benchmark.

4 Case Study: GRACE

This chapter presents a second case study to evaluate pointwise reliability methods in a classification task. The evaluation follows the benchmark described in Chapter 2.2 and applies the same five pointwise reliability methods introduced in Chapter 3, namely Subtractive Clustering, DBSCAN, Distance Based, ICM and Density and Local Fit. This case study comprises a different dataset, complementing the previous findings by allowing for the evaluation of method consistency in different data domains.

4.1 Dataset and Model

This section describes the sources, features, and preparation steps of the datasets used in this case study. The study uses a clinical dataset from the Cardiology Intensive Care Unit of Coimbra Hospital and University Center. The dataset includes patients admitted between 2009 and 2016 with acute coronary syndrome and includes diagnoses such as ST-elevation myocardial infarction (STEMI). Each row corresponds to a single patient, and the target variable indicates with class 1 that the patient has died and class 0 indicates survival. The dataset contains 1544 patients, with an unbalanced distribution (approximately 30% for class 1).

Table 4.1 presents the summary statistics for the features of the dataset after removing entries with null or invalid values (resulting in 1363 entries from the original 1544) and before any normalization.

Table 4.1: Descriptive statistics of features after removing entries with missing or invalid values, showing the 25th percentile (Q1) and 75th percentile (Q3).

Feature (unit)	**Mean**	**Std**	**Q1**	**Q3**
Age (years)	68.01	13.21	58.0	78.0
Heart rate at admission (bpm)	76.66	18.60	65.0	86.0
Systolic BP at admission (mmHg)	134.02	27.56	118.0	150.0
Killip class at admission (I to IV)	1.33	0.68	1.0	1.0
Creatinine at admission (µmol/L)	110.23	112.04	69.05	109.5
Maximum Killip class (I to IV)	1.45	0.81	1.0	2.0
Troponin at admission (ng/L)	20.21	63.02	0.11	10.6
STEMI (0 or 1)	0.36	0.48	0.0	1.0

The predictors used are aligned with those used in the GRACE risk score [48], a well known framework for short term mortality and myocardial infarction risk stratification in acute coronary syndrome. GRACE assigns points to a set of clinical variables

(age, systolic blood pressure, heart rate, creatinine, Killip class, STEMI, cardiac arrest at admission, and elevated cardiac markers), each contributing to a cumulative score. The total score is then assigned to three risk levels (low, intermediate, and high). In this study, the GRACE score itself is not computed, instead, its individual variables are used directly as predictors (in a logistic regression) to predict mortality, as they are considered well suited to the problem [25, 43, 49].

To train the model, admission variables were mapped to the corresponding GRACE components. Age, systolic blood pressure, heart rate, creatinine, and STEMI were taken from admission measurements. Elevated cardiac markers were captured through the admission troponin indicator. Cardiac arrest at admission is represented in the dataset by Killip equal to IV (which represents cardiac arrest).

The same procedure described in Chapter 3 was used to assess the pointwise reliability. The process was repeated ten times. In each iteration, the dataset was randomly split into a training set (90%) and a test set with stratified sampling to preserve the class ratio. In each repetition a simple logistic regression classifier was trained[1], and all pointwise reliability methods were applied to its predictions using the test set. All numerical features were scaled to the range $[0, 1]$. No hyperparameter tuning or class weighting was applied, since the objective was not to maximize discrimination but to evaluate whether pointwise reliability methods can distinguish between reliable and unreliable predictions. Across the ten repetitions, the classifier achieved an accuracy of 76% (standard deviation of 1%) and an area under the receiver operating characteristic curve of 0.80 ± 0.01.

4.2 Pointwise Reliability

The same five pointwise reliability methods evaluated in Chapter 3 are applied in this case study, following the benchmark described in Chapter 2.2. Each prediction is assigned a reliability score and the predictions are grouped into reliability intervals. Within each interval, the distribution of correct and incorrect predictions is analyzed separately for each predicted class.

The parameter selection procedure described in Section 2.2.10 was applied to this study. Based on the distribution of pairwise distances in the training data, the 0.75 percentile value was selected, resulting in a distance threshold of 0.089 and a minimum cluster size of 9.

The remainder of this section mirrors the structure presented in Chapter 3, providing an analysis for each pointwise method, followed by a discussion of the results.

[1] The classifier was implemented using the `LogisticRegression` class from the `scikit-learn` Python library.

4.2.1 Subtractive

The subtractive method (Section 2.2.1) was evaluated using the benchmark strategy detailed in Section 2.3, based on ten independent iterations. The predictions were divided by predicted label and grouped into reliability intervals. The distribution of errors was analyzed across these intervals.

Figure 4.1 shows the error rates in each reliability interval, separated by predicted class. As in the Patient Treatment case study, the majority of class 1 predictions achieve zero error in most intervals where predictions are present. However, in contrast to the previous dataset, class 0 show a more pronounced concentration of errors in the lowest reliability interval $[0.0, 0.1]$. The intermediate intervals for both classes are sparsely populated, and the error rates do not show a consistent downward trend as the reliability increases. Several high reliability intervals ($]0.8, 1.0]$) still exhibit non zero error for class 0, and despite increasing the error rate in the $]0.8, 1.0]$ interval for class 0, the error is relatively low (with some considerable variance) compared to the previous case study.

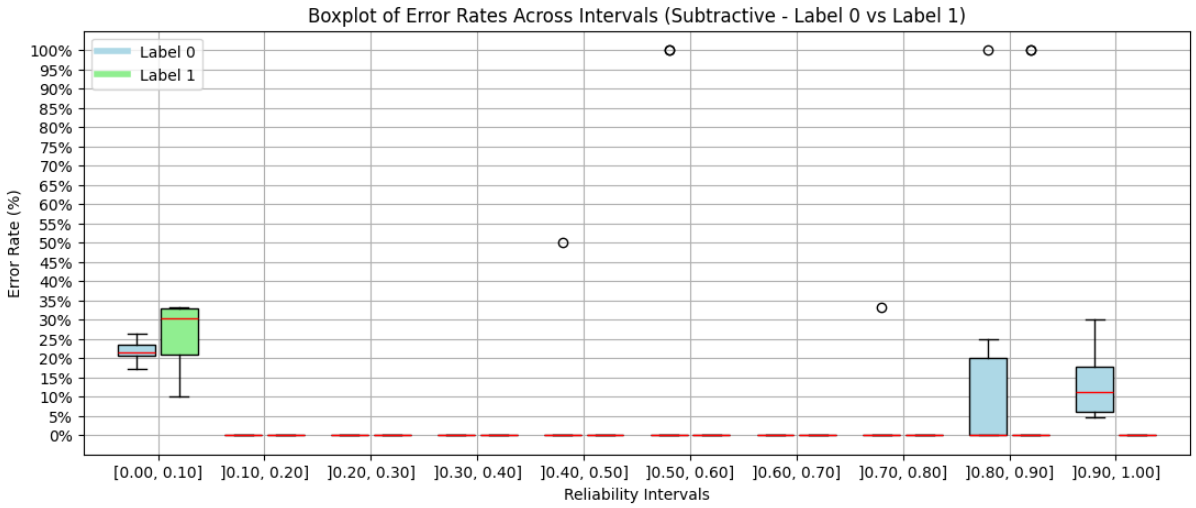


Figure 4.1: Boxplot of error rates across reliability intervals using the Subtractive method, separated by predicted label.

The distribution of the predictions across the intervals is presented in Table 4.2, which presents the average number of instances per iteration assigned to each reliability interval for both predicted classes. The pattern mirrors that seen in the Patient Treatment case study, where the majority of predictions are concentrated in the lowest interval $[0.0, 0.1]$, reflecting instances not assigned to any cluster and classified with the lowest reliability score. A smaller proportion of the predictions, mostly from class 0, appears in the highest interval $]0.9, 1.0]$. The intermediate intervals contain very few predictions.

Table 4.2: Number of predictions for each label ($Y = 0, Y = 1$) in each reliability interval using the Subtractive method.

Interval	Y = 0	Y = 1
[0.0, 0.1]	88.6 ± 4.9	27.4 ± 3.6
]0.1, 0.2]	0.0 ± 0.0	0.0 ± 0.0
]0.2, 0.3]	0.0 ± 0.0	0.0 ± 0.0
]0.3, 0.4]	0.0 ± 0.0	0.0 ± 0.0
]0.4, 0.5]	0.4 ± 0.7	0.0 ± 0.0
]0.5, 0.6]	0.3 ± 0.7	0.0 ± 0.0
]0.6, 0.7]	0.5 ± 0.8	0.4 ± 1.0
]0.7, 0.8]	1.1 ± 1.2	0.0 ± 0.0
]0.8, 0.9]	2.3 ± 2.1	0.4 ± 0.5
]0.9, 1.0]	15.6 ± 5.2	0.0 ± 0.0

The subtractive clustering method in the GRACE dataset exhibits the same limitations observed in the Patient Treatment case study. The method provides limited granularity in reliability estimation, with most predictions concentrated at the extremes of the reliability scale and little use of intermediate values. While class 1 predictions tend to have no errors when assigned to high reliability intervals, they are rare outside the lowest interval. For class 0, error rates remain present even among the highest reliability predictions. These findings indicate that for this case study, similar to the result obtained in Chapter 3, subtractive clustering struggles to discriminate effectively between reliable and unreliable predictions.

4.2.2 DBSCAN

The DBSCAN method, introduced in Section 2.2.2, shows similar patterns to those observed with the Subtractive Clustering method. Figure 4.2 shows the error rates per reliability interval, separated by predicted class.

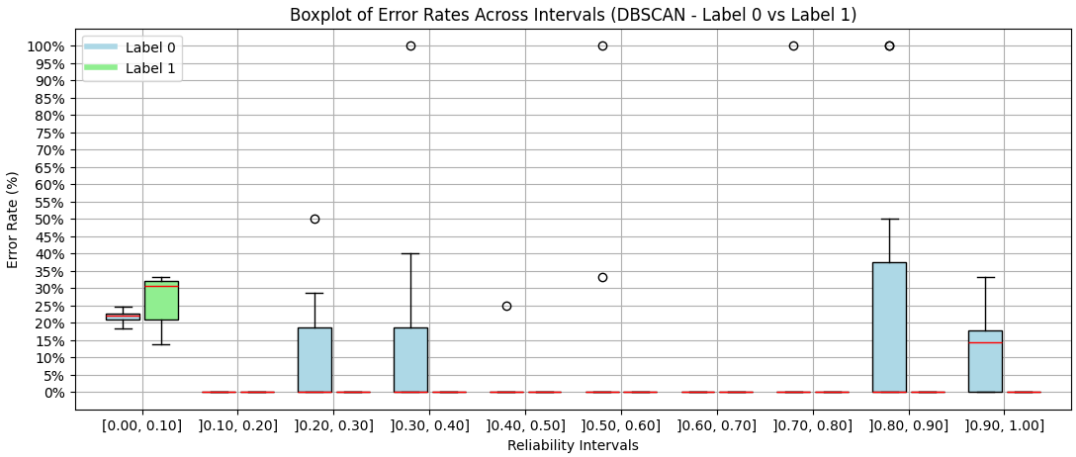


Figure 4.2: Boxplot of error rates across reliability intervals using the DBSCAN method, separated by predicted label.

For class 1, the median error rate remains zero in all populated intervals (except for the lowest reliability interval), However, as can be seen in Table 4.3, there are no predictions in addition to the first interval. For class 0, the error rate shows considerable variability and there is no consistent decrease in error with increasing reliability.

Table 4.3 presents the distribution of the predictions in the reliability intervals. Most predictions are concentrated in the lowest interval $[0.0, 0.1]$. A smaller cluster appears in the highest interval $]0.9, 1.0]$, composed essentially of class 0 predictions. The intermediate intervals are sparsely populated for class 0 , and class 1 just presents prediction in the first interval. However, DBSCAN separates the predictions for each interval slightly better than the subtractive method.

Table 4.3: Number of predictions for each label in every reliability interval using the DBSCAN method.

Interval	**Y = 0**	**Y = 1**
[0.0, 0.1]	89.5 ± 6.4	28.2 ± 3.3
]0.1, 0.2]	0.0 ± 0.0	0.0 ± 0.0
]0.2, 0.3]	5.2 ± 2.0	0.0 ± 0.0
]0.3, 0.4]	2.8 ± 1.3	0.0 ± 0.0
]0.4, 0.5]	1.8 ± 2.3	0.0 ± 0.0
]0.5, 0.6]	1.0 ± 1.2	0.0 ± 0.0
]0.6, 0.7]	0.4 ± 0.7	0.0 ± 0.0
]0.7, 0.8]	1.1 ± 1.1	0.0 ± 0.0
]0.8, 0.9]	1.0 ± 0.8	0.0 ± 0.0
]0.9, 1.0]	6.0 ± 2.5	0.0 ± 0.0

The GRACE results further support the conclusions obtained for DBSCAN in the previous case study (Chapter 3). While class 1 predictions show no errors for most reliability intervals, the number of predictions in these intervals is very small (inexistent in this case study). For class 0, higher reliability scores do not guarantee lower error rates. The predictions are concentrated at the extremes of the reliability interval and there is no clear evidence that the method is successful in identifying reliable predictions. These findings indicate that the method has a limited capability to distinguish between reliable and unreliable predictions.

4.2.3 Distance

The distance method (Section 2.2.3) assigns reliability scores according to the proportion of nearby instances that share the predicted label. The method distributes predictions across the entire reliability range more evenly than the clustering based approaches and, consistent with what is observed in Chapter 3, it lacks the ability to separate reliable and unreliable predictions.

Figure 4.3 shows the error rates for each reliability interval, separated by predicted

class. For class 1, a gradual reduction in error is visible as the reliability increases, followed by a considerable proportion of predictions observed without errors in intervals above $]0.4, 0.5]$. This is consistent with the general tendency observed in the Patient Treatment dataset. Class 0, however, does not follow the same pattern, error rates vary substantially between intervals, and high reliability intervals exhibit noticeable error. Variability within intervals is also visible, suggesting that local density alone may not consistently align with prediction correctness for this class.

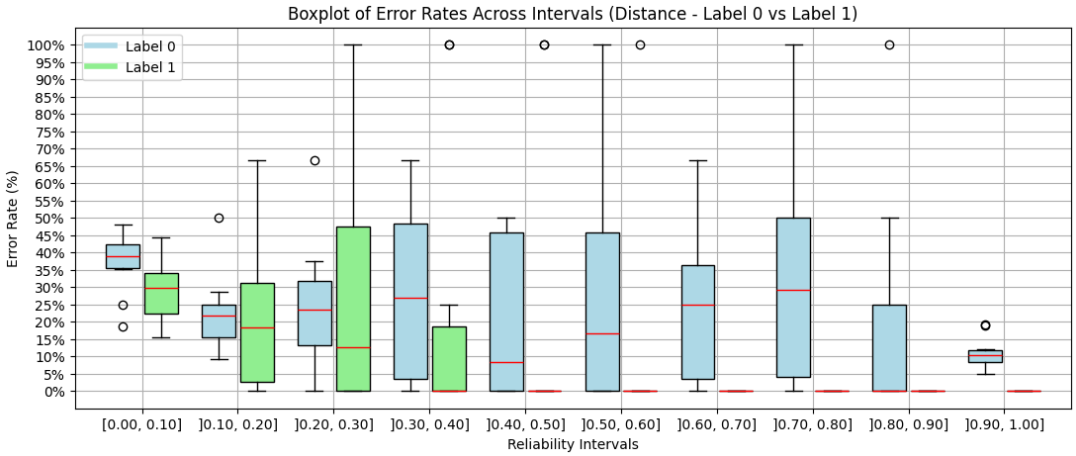


Figure 4.3: Boxplot of error rates across reliability intervals using the Distance Based method, separated by predicted label.

The average number of predictions per reliability interval is shown in Table 4.4. Predictions are present in all intervals, with low and high reliability ranges ($[0.0, 0.1]$ and $]0.9, 1.0]$) that contain the highest counts, especially for class 0 (possibly due to the class imbalance, which causes a greater number of clusters for this class). The intermediate intervals hold a moderate and consistent number of predictions for both classes, which is a contrast to the Subtractive and DBSCAN methods where intermediate values were rarely assigned.

Table 4.4: Number of predictions for each label in every reliability interval using the Distance Based method.

Interval	**Y = 0**	**Y = 1**
$[0.0, 0.1]$	19.5 ± 4.2	18.2 ± 2.9
$]0.1, 0.2]$	11.4 ± 3.8	4.9 ± 3.0
$]0.2, 0.3]$	8.3 ± 3.7	2.2 ± 1.4
$]0.3, 0.4]$	5.3 ± 2.1	1.2 ± 1.2
$]0.4, 0.5]$	3.9 ± 1.9	0.6 ± 0.5
$]0.5, 0.6]$	2.8 ± 1.5	0.5 ± 0.5
$]0.6, 0.7]$	3.8 ± 1.5	0.2 ± 0.4
$]0.7, 0.8]$	3.5 ± 2.5	0.2 ± 0.4
$]0.8, 0.9]$	3.4 ± 1.8	0.1 ± 0.3
$]0.9, 1.0]$	46.9 ± 6.0	0.1 ± 0.3

The Distance Based method offers a more smooth distribution of reliability scores than the subtractive and DBSCAN approaches and shows a clearer association between high reliability and low error for class 1. This may be related to the fact that class 1 represents only 30% of the dataset, and finding clusters of this class is a strong indicator of reliability. For class 0, however, this relationship is less consistent, with some high reliability predictions still producing errors. Compared with clustering methods, distance based can spread the predictions in all the reliability intervals, but the observed variability indicates that distance information alone does not fully capture prediction reliability.

The results further support the findings from Chapter 3, relying on proximity alone to determine reliability, without assessing whether dense regions are homogeneous, limits the ability of the method to distinguish between reliable and unreliable predictions.

4.2.4 ICM

The ICM method (Section 2.2.4) compares the weighted influence of neighbors that agree with the predicted label to those that do not. The general behavior is similar to what was observed in the Patient Treatment case study, with predictions concentrated in the lower half of the reliability scale and error rates generally decreasing as reliability increases.

Figure 4.4 presents the error rates per reliability interval, separated by predicted class. For the interval $]0.10, 0.50]$, both classes show a steady reduction in error, although there is noticeable variability in the error rate across all intervals.

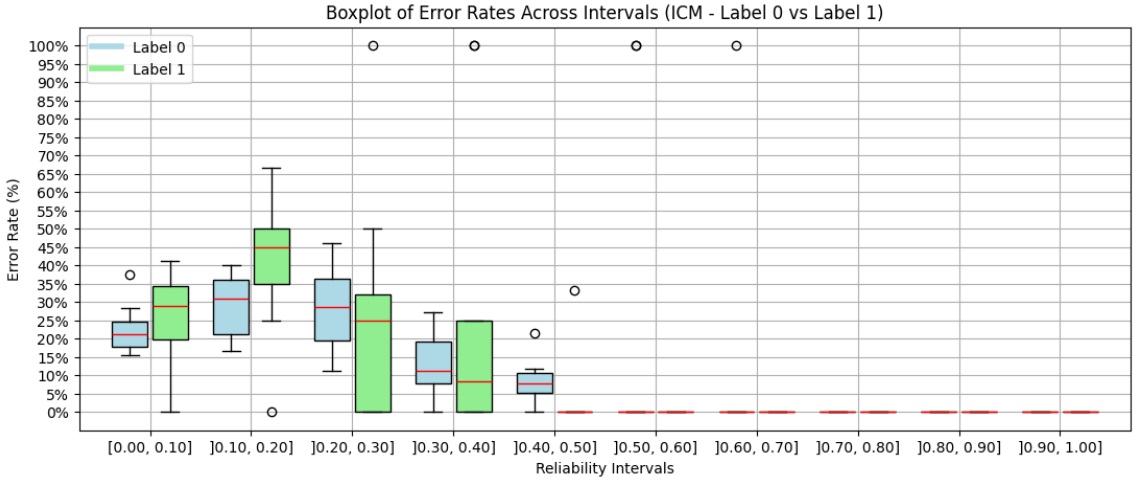


Figure 4.4: Boxplot of error rates across reliability intervals using the ICM method, separated by predicted label.

Table 4.5 shows the distribution of the predictions. As in Chapter 3, most of the predictions fall into the lowest reliability interval, which suggests that even if the method was capable of distributing the predictions more evenly across the entire reliability range

(assuming the same error distribution), most of the instances would still be assigned a score below 0.5 and thus be considered unreliable.

Table 4.5: Number of predictions for each label in every reliability interval using the ICM method.

Interval	**Y = 0**	**Y = 1**
$[0.0, 0.1]$	52.7 ± 6.3	16.7 ± 3.6
$]0.1, 0.2]$	11.6 ± 3.9	3.8 ± 1.4
$]0.2, 0.3]$	11.1 ± 3.7	2.9 ± 1.9
$]0.3, 0.4]$	16.2 ± 3.3	2.4 ± 1.7
$]0.4, 0.5]$	16.7 ± 3.0	2.3 ± 1.1
$]0.5, 0.6]$	0.4 ± 0.5	0.1 ± 0.3
$]0.6, 0.7]$	0.1 ± 0.3	0.0 ± 0.0
$]0.7, 0.8]$	0.0 ± 0.0	0.0 ± 0.0
$]0.8, 0.9]$	0.0 ± 0.0	0.0 ± 0.0
$]0.9, 1.0]$	0.0 ± 0.0	0.0 ± 0.0

These results further support the conclusion obtained in Chapter 3. Within the range in which the method assigns predictions, the error rate decreases consistently for both classes. However, the method barely assigns any predictions to the interval $[0.5, 1.0]$, which limits the granularity of the reliability estimation. If the scores were distributed more evenly throughout the entire range ($[0, 1]$), the decreasing error observed in the first half of the reliability scale would likely extend further, resulting in a more stable and informative method.

4.2.5 Density and Local Fit

The error rates for the Density and Local Fit method (Section 2.2.5) are shown in Figure 4.5.

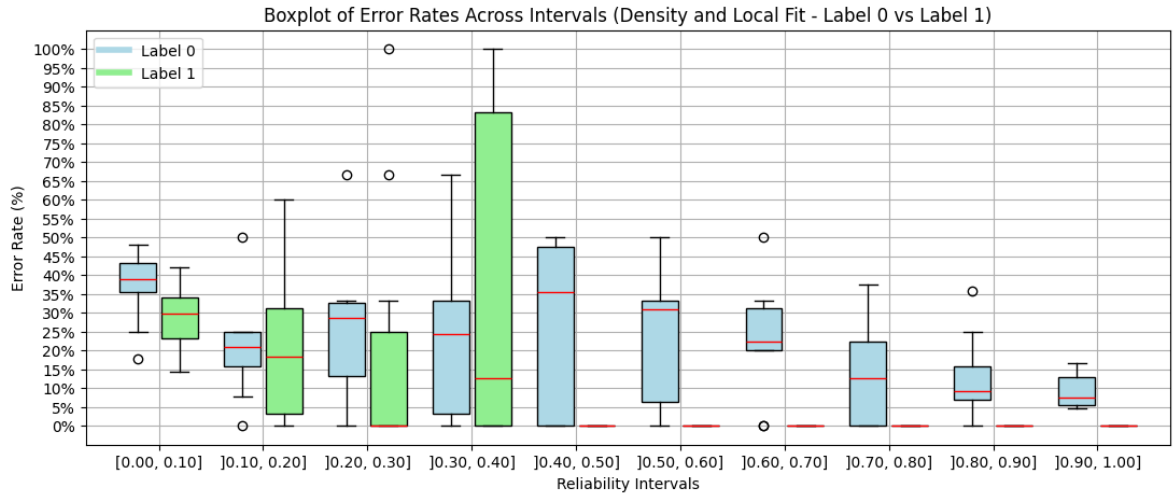


Figure 4.5: Boxplot of error rates across reliability intervals using the Density and Local Fit method, separated by predicted label.

The method achieves the most balanced distribution of reliability scores and the most consistent correspondence between reliability and prediction accuracy among all evaluated method. For class 0, the error rate shows a clear and steady decrease as the reliability increases, particularly from the interval $[0.5, 1.0]$ onward, where predictions start to be deemed highly reliable. In the highest reliability intervals, class 1 reaches zero error, while class 0 attains one of its lowest error levels. Although some variability remains in the middle intervals, the overall trend reflects a strong inverse relationship between reliability and error rate.

Table 4.6 shows the number of predictions per reliability interval. The predictions are distributed across the entire reliability range, with both low and high reliability intervals containing a well balanced number of instances compared to other methods. Class 0 predictions are more frequent in the highest intervals, without showing a high error rate, while class 1 predictions tend to be concentrated in the lower and mid reliability ranges.

Table 4.6: Number of predictions for each label in every reliability interval using the Density and Local Fit method.

Interval	**Y = 0**	**Y = 1**
[0.0, 0.1]	20.2 ± 3.5	18.8 ± 2.4
]0.1, 0.2]	11.3 ± 3.8	5.3 ± 2.9
]0.2, 0.3]	8.0 ± 3.7	1.5 ± 1.0
]0.3, 0.4]	5.6 ± 2.1	1.6 ± 1.2
]0.4, 0.5]	4.2 ± 1.9	0.1 ± 0.3
]0.5, 0.6]	3.9 ± 1.3	0.4 ± 0.5
]0.6, 0.7]	6.4 ± 2.0	0.2 ± 0.4
]0.7, 0.8]	9.8 ± 2.1	0.3 ± 0.5
]0.8, 0.9]	19.8 ± 4.4	0.0 ± 0.0
]0.9, 1.0]	19.6 ± 4.6	0.0 ± 0.0

These results confirm the ability of the method to assess the reliability of individual predictions. Consistent with the findings of Chapter 3, the method assigns reliability scores that are well distributed throughout the range and maintains a clear decreasing trend in error rates. Furthermore, the number of predictions in each reliability interval is more balanced compared to other methods. As in the previous case study, a small number of misclassifications appear in high reliability intervals for class 0, and some fluctuations occur in mid range intervals, indicating that there is still room for improvement. In this particular use case, further refinements should also account for the class imbalance, as the pattern for class 1 is less consistent than in the previous study.

4.2.6 Discussion of the Results

The results observed in the GRACE dataset are consistent with those of the Patient Treatment case study (Chapter 3). Clustering based methods failed to assign consistent

reliability scores, with most predictions concentrated in low reliability intervals and little variation in error rates across the range. The Distance based method performed slightly better, but still showed inconsistencies, particularly for the minority class. The ICM method showed a more consistent decrease in error with increasing reliability, although it rarely assigned predictions with reliability values above 0.5.

As in the previous case study, the Density and Local Fit method stands out as the most effective approach. It achieves a well balanced distribution of predictions across the full range of reliability intervals and maintains a clear decreasing trend in error rates for both classes. Performance is particularly strong in the $[0.5, 1.0]$ interval, where reliability is considered high, class 1 reaches zero error while class 0 shows one of its lowest error levels among all methods. These results further confirm the potential of the method to assess pointwise reliability across different data domains.

It should also be noted that the Density and Local Fit method still produces a small number of misclassifications in high reliability intervals. Although error rates remain low, their presence suggests potential for further improvements. Possible improvements include refining the hyperparameter selection process, adjusting the hyperparameter of the method separately for each class (since class imbalance appears to influence prediction distribution and error patterns for the minority class), or adapting the algorithm to incorporate weighted contributions so that closer instances have a greater contribution in the reliability computation.

In conclusion, the findings confirm that the results from Chapter 3 align with GRACE case study. The Density and Local Fit method, which combines density and local fit principles, demonstrates superior capability in identifying reliable predictions, while the unsupervised clustering based methods consistently produce the weakest results.

5 Case Study: Hospital Admissions

This chapter presents a case study on hospital admission forecasting. Unlike the previous assessments (Chapters 3 and 4), this case study follows a different structure motivated by the availability of a dataset with a considerable amount of information.

Before implementing a pointwise reliability assessment, an exploratory phase was conducted to characterize the data, define the problem, and verify its feasibility for resolution using Machine Learning (ML) models. This preliminary validation ensured that the research objectives were aligned with the characteristics of the dataset and that the proposed modeling approach was appropriate for the problem.

Forecasting hospital admissions plays a crucial role in public health management and in optimizing hospital resources. Forecasting models help mitigate overcrowding in healthcare facilities, improve patient care, and enable proactive operational planning [15, 16]. However, forecasting hospital admissions is inherently challenging due to the complexity of seasonality and external influences such as environmental and weather variables [50, 51, 52, 53].

The complexity of daily admission patterns is often increased by the overlap of multiple seasonal effects, including weekly and annual cycles [50]. Consequently, the selection of an appropriate forecasting methodology requires consideration of temporal resolution and the appropriate characteristics (selected variables). Short term forecasts (one month or less) are traditionally more accurate than long term forecasts (for example, one year), especially when using classic time series approaches such as Auto-Regressive Integrated Moving Average (ARIMA) models.

Seasonal ARIMA (SARIMA) and its variant SARIMAX (with exogenous regressor) have served as base models in healthcare forecasting due to their simplicity and effectiveness in capturing seasonal patterns and trends [15]. Despite their common use, ARIMA models often show limited performance, especially for longer forecast horizons and when multiple seasonal patterns are involved [50, 51]. In recent years, Machine Learning (ML) techniques have emerged as an alternative [54, 55, 56, 51]. Methods such as Random Forest (RF) and eXtreme Gradient Boosting (XGB) are particularly effective, demonstrating good performance compared to traditional forecasting models while being more efficient (in terms of resources) than more complex techniques such as neural networks.

Environmental factors, such as air pollution and weather conditions, significantly influence hospital admissions related to respiratory problems, with numerous studies

confirming these associations. For example, Weeberb *et al.* [57] showed that, in Brazil, extreme temperatures significantly increase hospital admissions, especially among vulnerable groups such as the elderly. Similarly, exposure to air pollutants, particularly particulate matter ($PM_{2.5}$), nitrogen dioxide (NO_2) and ozone (O_3), was strongly associated with increased hospital admissions, highlighting the importance of integrating pollution data into models [58]. Another important factor (especially in forecasts) is the lag effect, where past environmental conditions influence current hospital admissions. Studies by Tadano *et al.* [52] and Pawar *et al.* [56] suggest that ideal lag times typically vary between three and seven days after exposure, increasing the accuracy of the model by capturing the delayed effects of pollutants and weather conditions on respiratory health outcomes.

This study aims to predict daily patient admissions related to respiratory issues in Belo Horizonte (Brazil), using a dataset with an interval of five years (2015-2019). The evaluation periods chosen (30 days and 1 year) allow for a comparison of performance between SARIMA/SARIMAX and ML algorithms (XGB and Random Forest). These intervals were selected to assess short term (where traditional forecasting models typically perform best) and long term forecasting capabilities, aligned with short / medium and long term operational planning [50, 16].

The analysis focuses on two main aspects. First, the performance of the models is evaluated over short and long term horizons using temporal validation. Second, the benchmark procedure described in Section 2.3 is applied to assess the effectiveness of pointwise reliability methods. In this case study, an adapted version of the Density and Local Fit method is included, due to its promising results in classification tasks (Section 2.2.5). Furthermore, methods specifically designed for regression are evaluated, namely CONFIVE, CONFINE, and their proposed variants (denCONFIVE and iqrDenCONFIVE), as described in Sections 2.2.6-2.2.9.

Model training was conducted exclusively using data from the first four years of the dataset. The final year was reserved for testing both the models and the pointwise reliability methods.

The following sections of this chapter detail the data sets used, the comparison of the forecasting models employed, the results, and finally, the assessment of the reliability of the individual forecasts of the ML model.

The code, visualizations, and data are available in the GitHub repository[59].

5.1 Dataset

This section describes the sources, features, and preparation steps of the datasets used in this study. The data consist of hospital admissions specifically related to respiratory

conditions, extended with environmental datasets (including weather conditions and air quality measures). All datasets (all data used) were preprocessed and combined into a final dataset that contained daily records, allowing subsequent analysis.

The central objective was to predict daily hospital admissions focusing on respiratory problems. The primary data source (with data related to hospital admissions) was the *Sistema de Informações Hospitalares do Sistema Único de Saúde* provided through the *Plataforma de Ciência de Dados aplicada à Saúde* (PCDaS)[1]. The data from PCDaS was originally collected from the *Departamento de Informação e Informática do Sistema Único de Saúde* [2] and processed monthly into enriched records. For this study, Belo Horizonte (Minas Gerais, Brazil) hospital admission data were selected from 2015 to 2019. The dataset included detailed patient information such as admission date and cause of admission. Data were filtered to include only cases where the primary diagnosis was respiratory diseases ("*Doenças do aparelho respiratório*").

Weather data were acquired through the *Instituto Nacional de Meteorologia* from the station of Pampulha in Belo Horizonte[3], covering 2015 to 2019. The original hourly records were aggregated into daily values to provide variables such as daily maximum and minimum temperatures, maximum and minimum humidity. Additional temporal variables including day of the week, weekends, and holidays were integrated, providing potential predictors for admissions variability.

Daily air quality metrics were obtained from the *Instituto de Energia e Meio Ambiente*, focusing on data from the Amazonas monitoring station in Belo Horizonte[4].

The pollutant values correspond to the standardized Air Quality Index (IQAr - *Índice de Qualidade do Ar*), which reflects the public health impact of exposure to various atmospheric pollutants[5]. The IQAr are not raw concentrations but processed index values based on pollutant thresholds. The IQAr values are computed using the formula 5.1:

$$IQAr = \frac{I_{high} - I_{low}}{C_{high} - C_{low}} \times (C - C_{low}) + I_{low} \tag{5.1}$$

where C is the observed concentration of pollutant, C_{low} and C_{high} are the lower and upper concentration breakpoints of the range in which C falls, and I_{low} and I_{high} are the corresponding index values for that range (the breakpoints of the IQAr category, as shown in Table 5.1). The reported daily IQAr corresponds to the maximum value of the pollutant index among those measured.

It is important to note that the air quality data in this study originate from a single

[1] `https://pcdas.icict.fiocruz.br`; Accessed: 17 May 2025
[2] `https://datasus.saude.gov.br/`; Accessed: 17 May 2025
[3] `https://tempo.inmet.gov.br/TabelaEstacoes`, station A521; Accessed: 17 May 2025
[4] `https://energiaeambiente.org.br/qualidadedoar/`; Accessed: 17 May 2025
[5] `https://www.gov.br`; Accessed: 17 May 2025

monitoring station, which limits the representation of pollutant exposure in the city. In addition, pollutant data include missing values and (for real use cases) are often unavailable for long term forecasts, reducing their utility as predictive features in that context. However, in short term scenarios, where current or recent pollutant data are available, these features may offer valuable information.

Despite these limitations, previous studies [58, 52, 56, 53] support the inclusion of environmental factors. For example, Requia *et al.* [58] reported that a 10 $\mu g/m^3$ increase in $PM_{2.5}$ is associated with a 3.28% increase in hospital admissions for respiratory diseases. Likewise, a 10 ppb (parts per billion) increase in NO_2 correlates with a 35.26% increase in hospital admissions related to respiratory problems.

Table 5.1 summarizes the IQAr classification ranges for the pollutants considered in this study, namely PM_{10}, O_3, and sulfur dioxide (SO_2).

Table 5.1: IQAr classification ranges by pollutant type.

IQAr Category	**PM_{10} ($\mu g/m^3$, 24h)**	**O_3 ($\mu g/m^3$, 8h)**	**SO_2 ($\mu g/m^3$, 24h)**
Good (0-40)	0-45	0-100	0-40
Moderate (41-80)	46-100	101-130	41-50
Poor (81-120)	101-150	131-160	51-125
Very Poor (121-200)	151-250	161-200	126-800
Terrible (>200)	>250	>200	>800

In summary, hospital admissions datasets, meteorological conditions, and air quality were integrated into a single structured dataset composed of daily records from 2015 to 2019. Additional variables derived from calendar information, such as indicators for weekends, holidays, and days after holidays, were included based on findings from a previous study [16].

Table 5.2 shows the descriptive statistics of the continuous variables used. The table also includes a description for the number of admissions.

Table 5.2: Descriptive statistics of continuous variables.

Variable	**Count**	**Mean**	**Min**	**25%**	**50%**	**75%**	**Max**
Admissions	1826	47.78	11.0	37.0	47.0	59.0	94.0
Temp Max (°C)	1826	28.31	17.8	26.3	28.5	30.5	37.7
Temp Min (°C)	1826	17.79	7.3	16.1	18.4	19.7	24.1
Temp Mean (°C)	1826	22.28	13.6	20.5	22.3	24.1	29.6
Humidity Max (%)	1826	82.26	42.0	77.0	84.0	90.0	94.0
Humidity Min (%)	1826	40.26	10.0	32.0	39.0	48.0	91.0
PM_{10}	1457	19.91	4.0	13.0	18.0	24.0	65.0
O_3	1265	52.44	14.0	41.0	50.0	62.0	176.0
SO_2	1091	18.75	0.0	8.0	12.0	24.0	83.0

Table 5.3 shows the remaining features in the dataset and their description.

Table 5.3: Description of categorical variables.

Variable	**Description**
DayOfWeekNum	Day of the week (0=Sunday to 6=Saturday)
WeekOfYear	Week number of the year (1 to 53)
Month	Calendar month (1=January to 12=December)
IsWeekend	Boolean variable: 1 if weekend, 0 otherwise
IsHoliday	Boolean variable: 1 if national holiday, 0 otherwise
IsDayAfterHoliday	Boolean variable: 1 if the day follows a holiday, 0 otherwise

In addition, to make comparisons during the study, some lagged variables are also included in the dataset, such as the meteorological and air quality parameters of previous days, to capture potential delayed effects on admission.

Figure 5.1 shows boxplots representing the distribution of daily admissions for respiratory related hospitalizations in Belo Horizonte (2015-2019), grouped by (a) day of the week, (b) week of the year, (c) month, and (d) year.

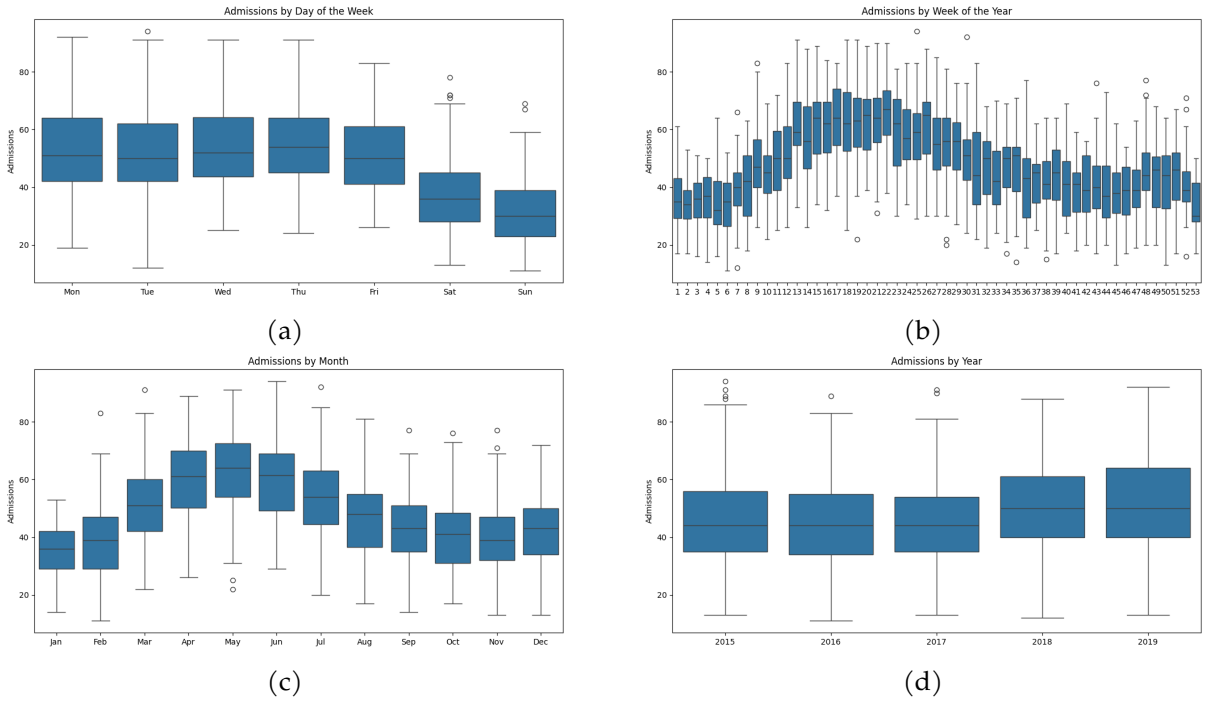


Figure 5.1: Distribution of daily hospital admissions due to respiratory related admissions in Belo Horizonte, grouped by day of the week (a), week of the year (b), month (c), and year (d).

Daily hospitalizations vary significantly by day of the week, with lower medians observed during weekends, especially on Sundays. This suggests a behavior of lower hospital demand or limited hospital availability on these days. Weekly trends reveal a seasonal structure, with higher hospitalization rates concentrated between weeks 10 and 30 (March to August), reflecting an increase in hospitalizations during the colder months. The monthly patterns align with the weekly structure, peaking between April

and June and declining by the end of the year (which further supports the seasonal impact). Annual distributions appear stable overall, although there is a slight upward trend in hospitalizations in 2018 and 2019.

These visualizations provide strong evidence of the existence of multiple seasonal effects (daily, weekly, monthly), justifying the use of time series models with seasonal components and the inclusion of "calendar" variables in predictive models.

Figure 5.2 shows the Pearson correlation matrix with the relationships between daily hospital admissions and various predictors, including meteorological and environmental variables (using categorical and continuous predictors). The strongest (negative) correlations with admissions are observed for the weekend, with coefficients of -0.52. This suggests (as also observed in Figure 5.1) that fewer admissions occur on weekends, likely reflecting patterns in healthcare demand and availability of services.

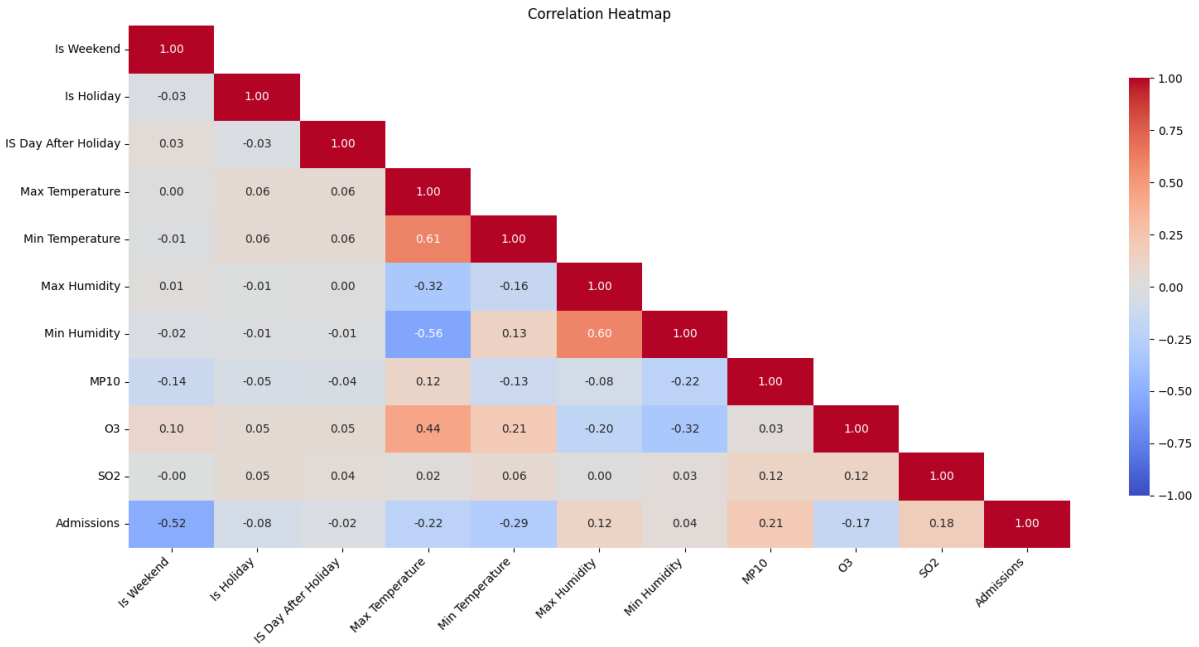


Figure 5.2: Correlation heatmap showing Pearson correlation coefficients between selected features and daily hospital admissions due to respiratory problems.

For meteorological variables, minimum temperature shows the highest (negative) correlation with admissions (-0.29) followed by maximum temperature (-0.22). In terms of humidity, considering only the correlation, just maximum humidity can have an impact on admissions. Regarding air pollution metrics (PM_{10}, SO_2, and O_3), the correlation matrix shows weak correlations with admissions, with PM_{10} showing the highest positive correlation of 0.21.

Overall, while no single variable shows a strong linear correlation with admissions, several exhibit modest associations that support their inclusion as features in predictive models (also, they have been identified as influential factors in previous studies). These relationships also reinforce the complexity of the problem.

Additionally, to investigate the influence of lagged environmental variables, as explored in previous studies [52, 56, 53], a correlation analysis was conducted between lagged predictors and "current" day hospital admissions.

Table 5.4 shows the Pearson correlation coefficients between pollutants and meteorological variables at lags of 0 to 7 days, in relation to hospital admissions.

Table 5.4: Pearson correlation between daily hospital admissions and environmental variables at lags 0 to 7.

	Lag 0	**Lag 1**	**Lag 2**	**Lag 3**	**Lag 4**	**Lag 5**	**Lag 6**	**Lag 7**
PM_{10}	0.21	0.14	0.06	0.04	0.04	0.05	0.12	0.16
O_3	−0.17	−0.11	−0.07	−0.06	−0.07	−0.1	−0.13	−0.17
SO_2	0.18	0.17	0.17	0.17	0.17	0.16	0.16	0.17
Max Temp	−0.22	−0.23	−0.24	−0.25	−0.25	−0.25	−0.24	−0.26
Min Temp	−0.29	−0.30	−0.29	−0.31	−0.30	−0.28	−0.27	−0.27
Max Hum	0.12	0.12	0.14	0.15	0.14	0.15	0.14	0.15
Min Hum	0.03	0.03	0.03	0.03	0.05	0.05	0.06	0.09

While the results (in Table 5.4) suggest minimal variation in the strength of the correlations between different lags, closer inspection leads to additional considerations. For variables that exhibit a higher absolute correlation with admissions, such as minimum and maximum temperature (and for PM_{10} and O_3, which show a decrease in correlation with admissions as the lag increases), the pairwise correlations among different lags of the same variable reveal a steady decline. As shown in Table 5.5, the correlation between lag 1 and lag 3 for maximum humidity, maximum temperature, and PM_{10} drops from 0.71, 0.73, and 0.73 at lag 1 to 0.49, 0.47, and 0.38 at lag 3, respectively.

Table 5.5: Pearson correlation between each feature and its lagged values (lags 1 to 7).

	Lag 1	**Lag 2**	**Lag 3**	**Lag 4**	**Lag 5**	**Lag 6**	**Lag 7**
Max Hum	0.71	0.57	0.49	0.42	0.38	0.34	0.30
Min Temp	0.73	0.56	0.47	0.42	0.39	0.36	0.35
PM_{10}	0.73	0.48	0.38	0.37	0.34	0.31	0.29
SO_2	0.99	0.98	0.98	0.98	0.98	0.98	0.97

This finding may suggest some degree of importance in including recent past values for predictive purposes. As noted in the literature, environmental factors often have a short term delayed effect on respiratory health, with the strongest influence typically observed within the first three days after exposure [52, 56].

An exception to this behavior is SO_2. Despite its consistent correlation with admissions on all lags, further examination reveals that SO_2 exhibits extremely high autocorrelation over its lagged values (with pairwise correlations near 1.00 from lag 0 to lag 7), suggesting minimal temporal variation. This may reflect the low variability of this pollutant in the specific monitoring station used in this study. Another factor that may bias this result is the high number of missing values for SO_2.

Based on these observations and recommendations from previous literature, two different approaches are considered, using only current day environmental variables and incorporating lagged predictors from day 0 to day 3. In both cases, the variables are studied independently to assess their contribution to the forecasting performance.

5.2 Time Series and Machine Learning

This section presents an assessment of the forecasting methods applied to hospital admissions. The objective is to evaluate the performance of traditional time series models (specifically SARIMA and SARIMAX) in comparison with ML approaches such as RF and XGB. The evaluation is carried out over two different forecast horizons: 30 days (short term) and 360 days (long term). The dataset has been divided into a training/validation set (the first four years) and a test set (the last year).

The statistical models used in this study differ in the type of input data they consider. The SARIMA model relies exclusively on the past values of the target variable (hospital admissions), while SARIMAX incorporates additional exogenous variables. ML models (Sections 5.2.2 and 5.2.3) were trained using a broader set of features from the dataset, including calendar based temporal variables (*e.g.*, day of the week) and meteorological indicators.

5.2.1 SARIMA(X)

To serve as a baseline for the ML models, variations of the ARIMA model were developed, specifically the SARIMA and SARIMAX configurations. ARIMA is a statistical model for time series forecasting that combines three components: autoregressive terms (AR), differencing (I) and moving averages (MA). It is expressed in the form of ARIMA(p, d, q), where p is the number of autoregressive terms, d is the number of differences needed to make the series stationary, and q is the number of lagged forecast errors in the prediction equation.

SARIMA extends the ARIMA model by incorporating seasonal patterns and is denoted as SARIMA$(p, d, q)(P, D, Q)_s$, where P, D, and Q represent the seasonal autoregressive, differencing, and moving average terms, respectively, and s indicates the length of the seasonal cycle (for example, $s = 7$ for weekly seasonality). SARIMAX (SARIMA with eXogenous variables) introduces external regressors (such as temperature and pollution indices) into the SARIMA [60].

To determine the appropriate level of differencing (d) for the ARIMA models, an Augmented Dickey-Fuller (ADF) test was performed [60]. This statistical procedure assesses whether a time series is stationary (mean, variance, and autocorrelation remain constant over time). For the original series, the ADF test results in a test statistic of

-2.7452 (p-value = 0.0665), which does not provide sufficient evidence to reject the null hypothesis of non-stationarity. After first differencing, the test statistic dropped to -8.9545 (p-value = 0.0000), indicating strong evidence of stationarity in the transformed series. Based on these results, a first-order differencing ($d = 1$) was adopted.

To further support the selection of autoregressive and moving average terms, Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) [60] plots were analyzed for both the original and difference series, as shown in Figure 5.3. Spikes in the ACF plot suggest the presence of MA components, while the spikes in the PACF indicate potential AR components. In the original series, both ACF and PACF plots show slow decay, consistent with non-stationary data. After differencing, the plots stabilize, and only a few lags show significant spikes (suggesting low order AR and MA). Following previous studies that modeled daily hospital admission with weekly seasonality [50], the seasonal period was established at $s = 7$ (note that longer seasonal periods were not explored as the complexity of the model increases substantially).

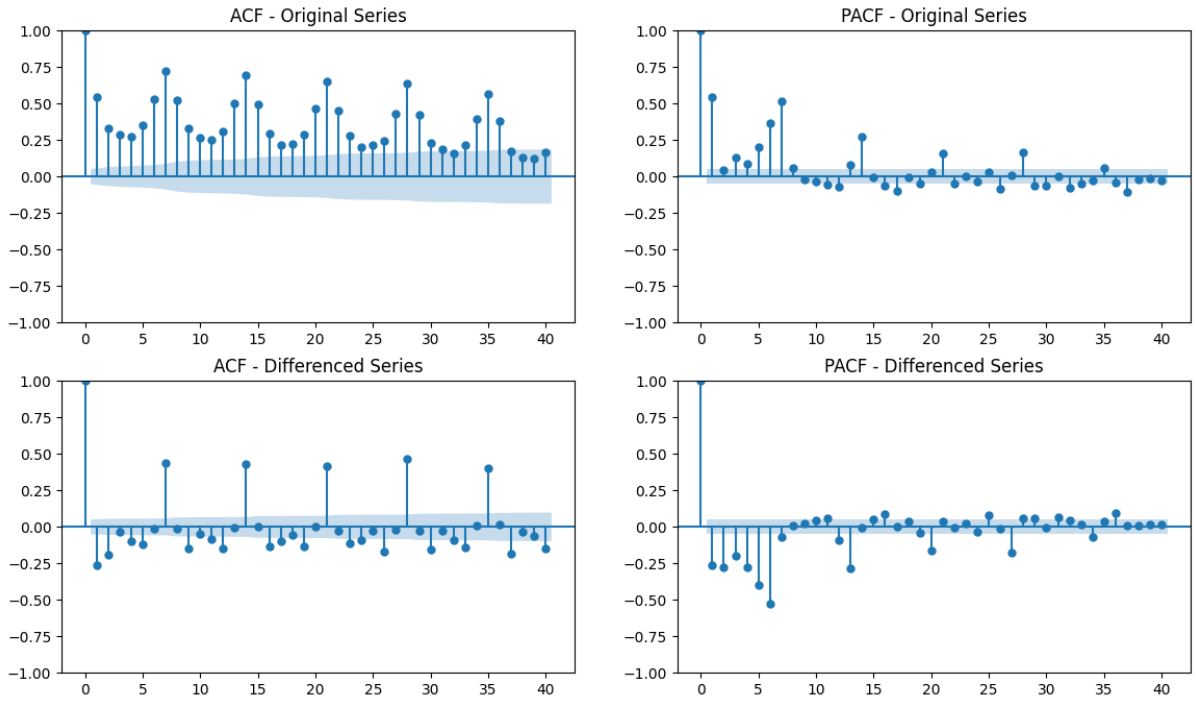


Figure 5.3: ACF and PACF for the original and first-differenced hospital admissions time series.

Following the initial analysis, a grid search for the SARIMA model was performed using the *auto_arima* function from the *pmdarima* (Python) library. The configuration for the grid search included AR and MA terms ranging from 0 to 15, with the differencing order (d) fixed at 1, based on the ADF test. Seasonal AR and MA terms ranged from 0 to 2, with seasonal differencing (D) allowed up to 1. The seasonal period (s) was fixed at 7 to reflect the identified weekly seasonality. The configuration for the model with the best results (using the Akaike Information Criterion [60]) was SARIMA(2,1,1)(1,0,1)[7]. The selected model passed the Ljung-Box test (p = 0.91)

[60], indicating that there is no significant autocorrelation in residuals (differences between the actual observed values and the values predicted by the model). Most parameters were statistically significant at the 5% level, except for the first non seasonal AR term. In general, the model met the key diagnostic checks.

For the 360 day forecast horizon, the SARIMA model achieved a mean absolute error (MAE) of 11.18, a root mean squared error (RMSE) of 13.95, and a mean absolute percentage error (MAPE) of 22.13%. The coefficient of determination (R^2) was 0.19, and the Pearson correlation coefficient was 0.55. These results indicate that, while the model was able to capture the general seasonal pattern (showing a moderate correlation), it had difficulty to capture the daily variability.

Figure 5.4 shows the predicted and actual values over time. Although the model captures the overall seasonal pattern, it tends to smooth daily fluctuations (values are around the median and do not capture daily peaks). As a result, the predicted values do not fully reflect the changes observed in the actual admissions.

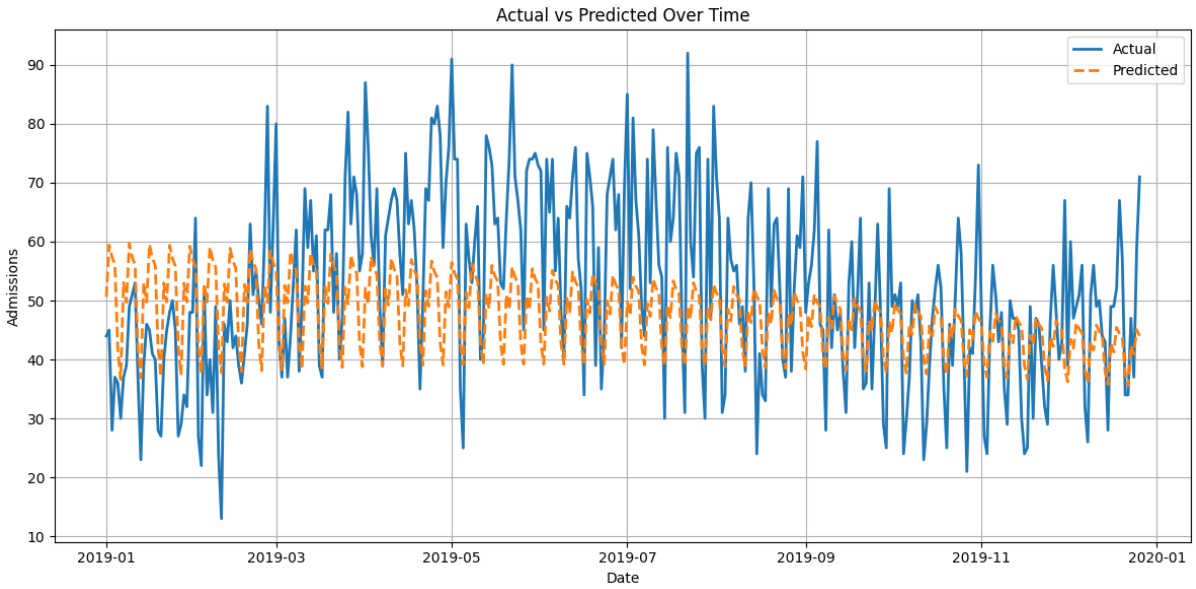


Figure 5.4: SARIMA - 360 day forecast vs. actual admissions.

For the 30 day test window, the performance of the model decreased. MAE increased to 11.81, RMSE to 13.08, and MAPE to 33.08%. The R^2 dropped to -1.55, although the correlation remained relatively high at 0.75.

Figure 5.5 shows the same behavior observed in the 360 day forecast, the model captures the overall seasonal pattern but tends to smooth the daily fluctuations. The figure also helps to explain the high correlation despite the poor error metrics. Although the model follows the general direction of the variations, it does not accurately capture their magnitude, resulting in larger errors.

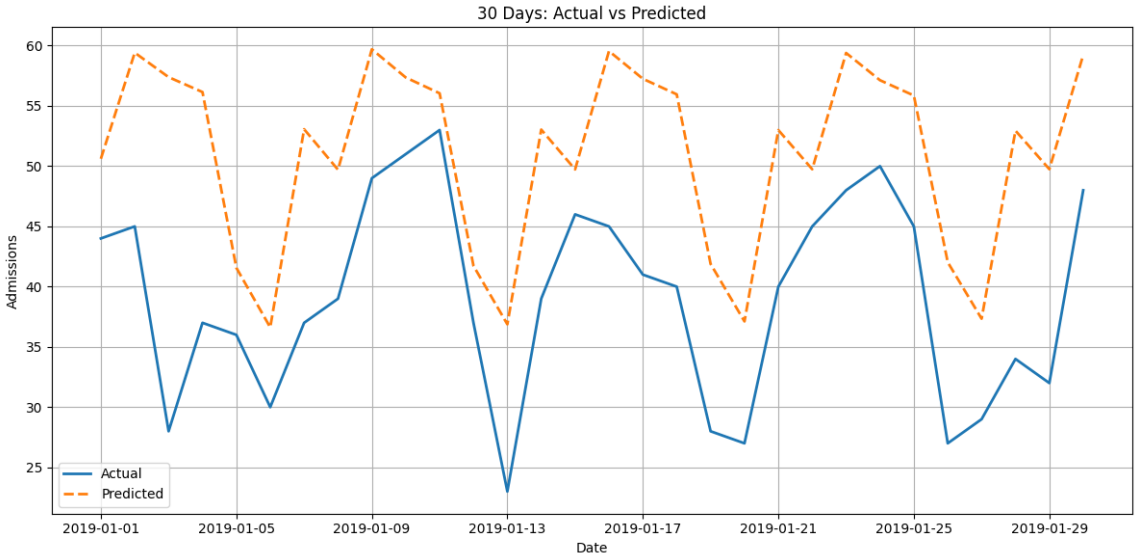


Figure 5.5: SARIMA - 30 day forecast vs. actual admissions.

The negative R^2 observed in the short term test can be explained by the scatter plot in Figure 5.6. The SARIMA model fails to capture daily fluctuations and (as noticed before) predicts values toward the mean. As a result, the errors are larger than if they forecast based on a mean (for 30 days).

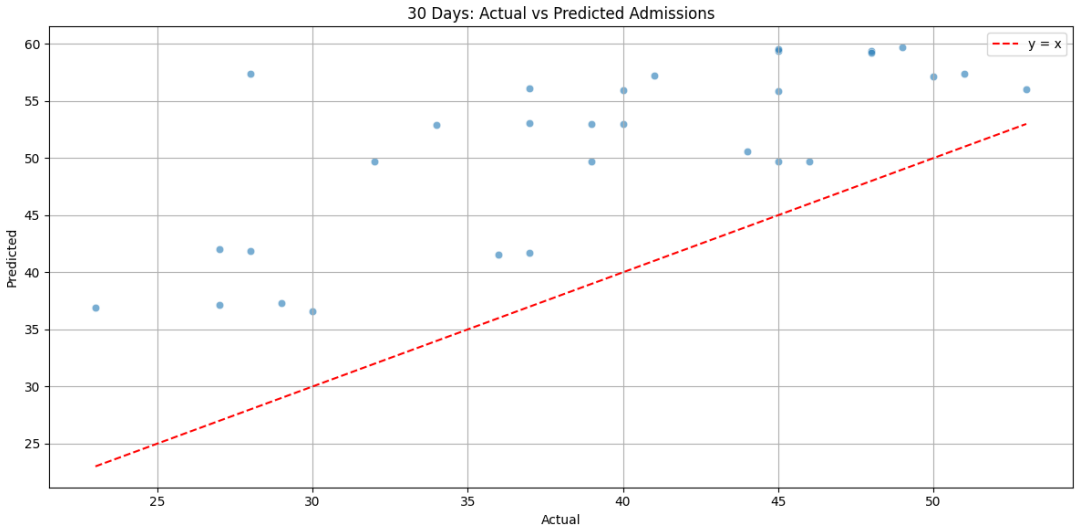


Figure 5.6: SARIMA - Actual vs. predicted admissions (30 day test).

To extend the SARIMA model and incorporate external factors, a **SARIMAX** model was developed using the same internal configuration (SARIMAX(2,1,1)(1,0,1)[7]) as SARIMA. For external information (exogenous regressors), binary indicators for weekends, holidays, and the day after holidays, together with temperature and humidity variables, were tested using Granger causality tests [61]. Granger tests identified weekends, maximum and minimum temperatures, and maximum humidity as statistically significant (p-value < 0.05), consequently these were the only regressors included in the model.

The SARIMAX model showed a slight improvement in long term forecasts. For the 360 day test, it achieved an MAE of 11.11, RMSE of 13.78, and MAPE of 21.83%. The R^2 increased to 0.21, with a correlation of 0.52. In Figure 5.7, which shows the predicted and actual values over time, it is evident that (despite the improved numerical metrics), SARIMA and SARIMAX exhibit similar patterns. Both models capture the overall seasonal trend but tend to smooth daily fluctuations.

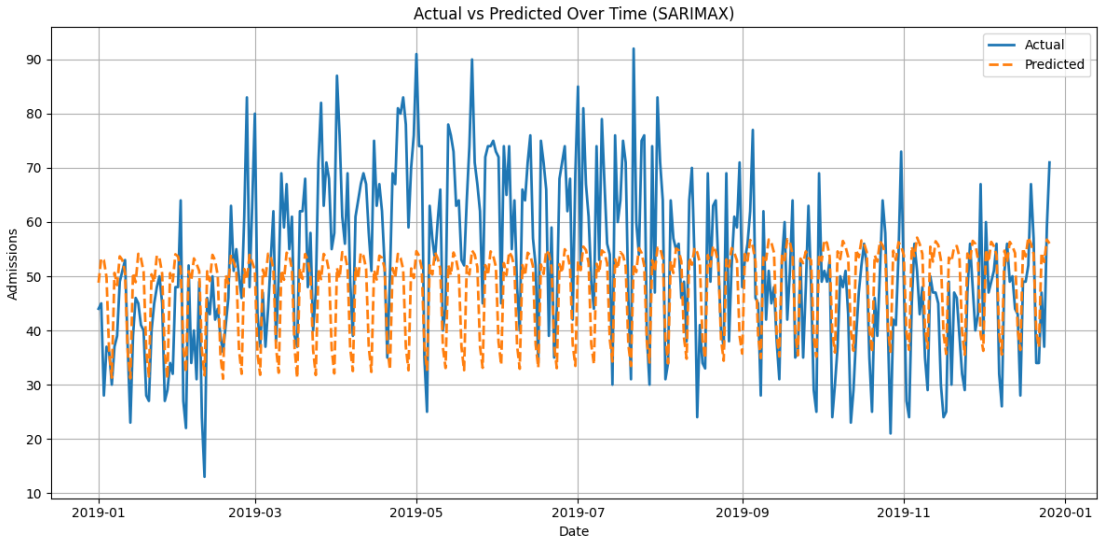


Figure 5.7: SARIMAX - 360 day forecast vs. actual admissions.

During the 30 day test, SARIMAX achieved substantially lower errors than SARIMA, namely MAE (7.76 vs. 11.81), RMSE (9.53 vs. 13.08), and MAPE (21.83% vs. 33.08%). The R^2 remained negative (-0.36), the correlation was comparable at 0.74.

Figure 5.8 shows the predicted and actual values over time for the forecast of 30 days. Similarly to SARIMA, the SARIMAX model still fails to capture the day-to-day variation, but is closer to the actual number of admissions.

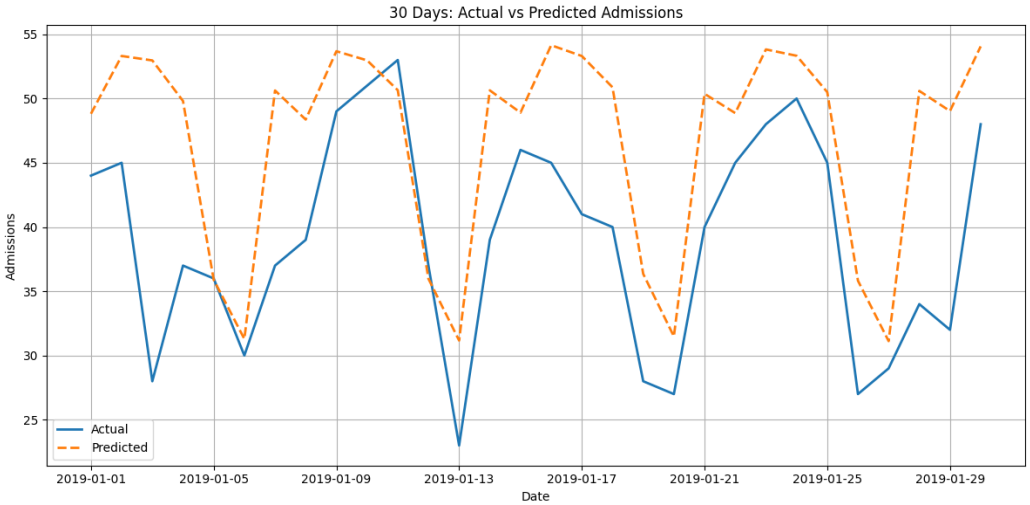


Figure 5.8: SARIMAX: 30 day forecast vs. actual admissions.

Overall, SARIMA captured the seasonal pattern of the data but struggled with daily variation and (as expected) delivered modest performance over the long term horizon. Incorporating exogenous variables into the SARIMAX model resulted in small improvements in long term predictions and more noticeable improvements in short term forecast accuracy.

5.2.2 Random Forest Regressor

Random Forest (RF) [62] is an ensemble learning algorithm that builds multiple decision trees during training and outputs the mean prediction (in regression tasks) of the individual trees. It combines the principles of bagging (bootstrap aggregation) and random feature selection to improve the results and reduce overfitting. Each tree is trained on a randomly sampled subset of the training data, and at each split, a random subset of features is considered.

The RF model was implemented using the *RandomForestRegressor* class from the *scikit-learn* Python library. The input features for the model were selected based on the variables described in Tables 5.2 and 5.3, and include both calendar based and meteorological indicators. The final feature set consisted of *DayOfWeekNum*, *WeekOfYear*, *Month*, *IsWeekend*, *IsHoliday*, *temp_max*, *temp_min*, and *humidity_max*.

The training procedure was repeated ten times using different initializations (*random_state*). For each iteration, a grid search with 5-fold cross-validation was used to identify the optimal combination of hyperparameters. The grid included the number of estimators ($n\_estimators = [20, 40, 60, 80, 100, 120, 140, 160]$), maximum tree depth ($max\_depth = [\text{none}, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]$), and minimum samples required to split a node ($min\_samples\_split = [2, 4, 6, 8, 10, 12, 14, 16]$). The final model ($n\_estimators = 140$, $max\_depth = 6$, and $min\_samples\_split = 16$) was selected based on the RMSE calculated from the test set.

Table 5.6 summarizes the evaluation metrics (10 iterations) for the RF model over the 360 day test horizon. The model demonstrates substantial improvements compared to SARIMA and SARIMAX models, particularly in terms of lower forecast errors.

Table 5.6: Random Forest - Test set performance metrics (360 days).

MAE	**RMSE**	**MAPE (%)**	**R^2**	**Correlation**
8.21 ± 0.03	10.99 ± 0.04	15.58 ± 0.07	0.49 ± 0.004	0.78 ± 0.003

Figure 5.9 shows the results of the best RF model (that is, the iteration with the lowest RMSE in the test set) compared to the actual values. The visual comparison shows the ability of the model to align with the actual values, capturing both seasonal patterns and short term variation. Nevertheless, the model tends to smooth daily variation (there are clear peaks in the data that the model are not capable of predict).

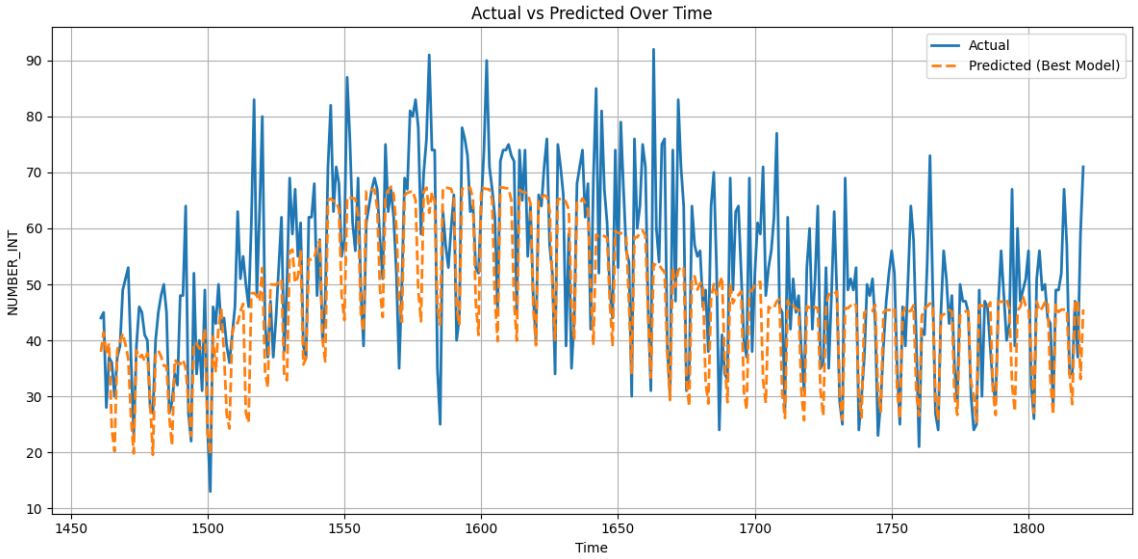


Figure 5.9: Random Forest - 360 day forecast vs. actual admissions.

Table 5.7 summarizes the evaluation metrics (10 iterations) for the RF model over the 30 day test horizon.

Table 5.7: Random Forest - Test set performance metrics (30 days).

MAE	**RMSE**	**MAPE (%)**	**R^2**	**Correlation**
6.53 ± 0.10	7.79 ± 0.11	16.44 ± 0.27	0.09 ± 0.03	0.70 ± 0.01

Although the error values are lower and the correlation are strong (suggesting that the model follows the variation in the data), the R^2 is relatively low, indicating that while predictions follow general trends, they often miss daily variation (peaks).

Figure 5.10 shows the results of the best RF model (that is, the iteration with the lowest RMSE in the test set) compared to the actual values for the 30 day forecast.

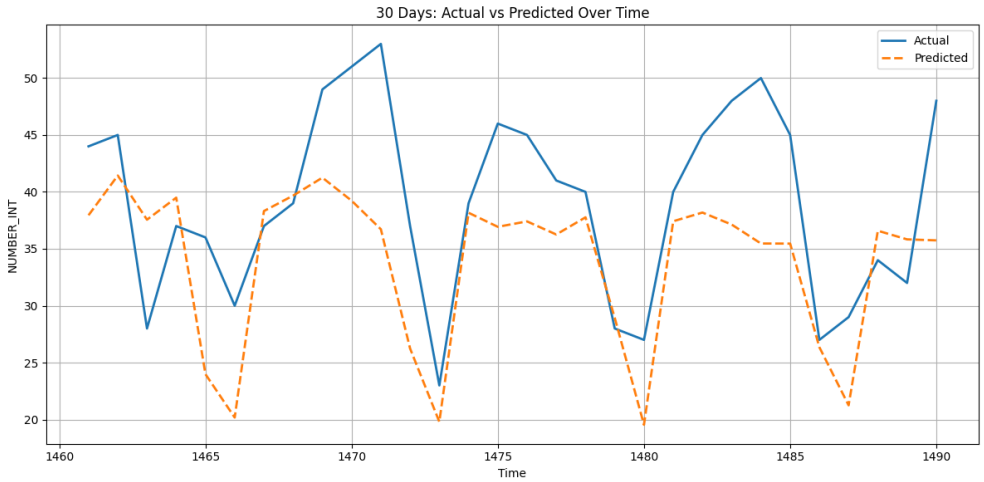


Figure 5.10: Random Forest - 30 day forecast vs. actual admissions.

As observed, while the model captures general trends and weekly seasonality, it tends to smooth peaks. In order to further investigate this limitation, Figure 5.11 shows a scatter plot of actual compared to predicted values.

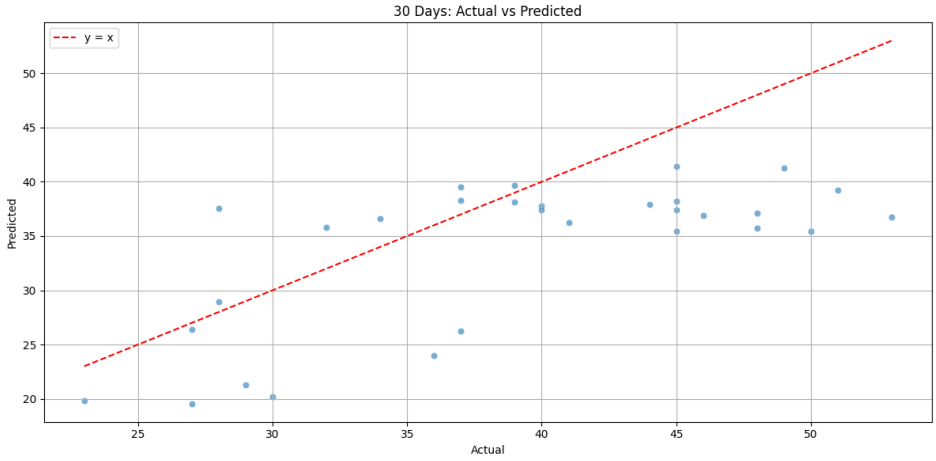


Figure 5.11: Random Forest - Actual vs. predicted admissions (30 day test).

As can be seen in the Figure 5.11, the points generally follow the expected trend, but higher values are often predicted as being lower than the actual ones (this is also true for the 360 day forecast). This indicates that the model tends to smooth the most extreme peaks, predicting values closer to the average. Another observation is that the R^2 score (metric used during training to determine the best model) drops slightly from validation (≈ 0.53) to testing (≈ 0.49). This decrease suggests that the model generalizes reasonably well and is not significantly overfitting or underfitting. The under prediction of peaks may reflect the inherent unpredictability of these events (data complexity), the absence of a key predictor in the current feature set, or even limitations of the regressor to deal with the complexity of the data (outliers).

Figure 5.12 shows the feature importance plot for the best performing model.

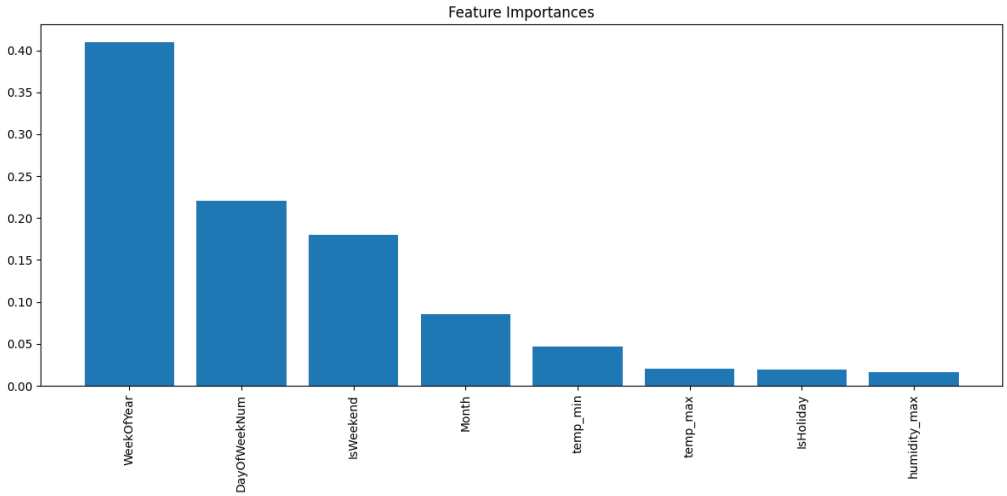


Figure 5.12: Random Forest - Feature importance extracted from the model.

As shown in Figure 5.12, the "calendar" features (*DayOfWeekNum*, *WeekOfYear* and *isWeekend*) were by far the most influential predictors. This suggests that the model relies heavily on periodic temporal patterns to make forecasts, reinforcing the importance of seasonality and structured calendar effects in hospital admission dynamics.

5.2.3 XGB Regressor

XGB [63] is an ensemble learning algorithm based on decision trees that uses a gradient boosting framework. XGB builds models sequentially, where each new tree corrects the residual errors of the previous ensemble. The algorithm incorporates regularization, shrinkage (learning rate), and subsampling to improve generalization and reduce overfitting.

XGB model was implemented using the *XGBRegressor* class from the *XGB* Python library. The training procedure was repeated ten times using different initializations. For each iteration, a grid search with 5-fold cross-validation was used to identify the optimal combination of hyperparameters. The grid included the number of estimators ($n\_estimators = [20, 40, 60, 80, 100, 120, 140, 160]$), maximum tree depth ($max\_depth = [2, 4, 6, 10, 12, 14, 16, 18, 20]$), learning rate ($learning\_rate = [0.01, 0.1, 0.2]$), subsample of the training instances ($subsample = [0.8, 0.9, 1.0]$) and the subsample ratio of columns ($colsample\_bytree = [0.8, 1.0]$).

Table 5.8 summarizes the results for the 360 day test set. The XGB results are close to those obtained with the RF model, but slightly lower in terms of errors (given the standard deviation, it can be considered insignificant).

Table 5.8: XGB - Test set performance metrics over 10 iterations (360 days).

MAE	**RMSE**	**MAPE (%)**	**R^2**	**Correlation**
8.25 ± 0.02	11.03 ± 0.03	15.63 ± 0.05	0.49 ± 0.002	0.78 ± 0.001

Table 5.9 summarizes the evaluation metrics (10 iterations) for the XGB model over the 30 day test horizon. The XGB shows slightly better results for errors than the RF model. However, as for the 360 day test, given the standard deviation, it can be considered insignificant.

Table 5.9: XGB - Test set performance metrics over 10 iterations (30 days).

MAE	**RMSE**	**MAPE (%)**	**R^2**	**Correlation**
6.50 ± 0.09	7.76 ± 0.10	16.41 ± 0.21	0.10 ± 0.02	0.71 ± 0.01

Figure 5.13 shows the results of the best XGB model compared to the actual values over the 360 day horizon. The results are similar to those obtained with RF (that is, the model captures both seasonal patterns and short term variation, but tends to smooth daily variation).

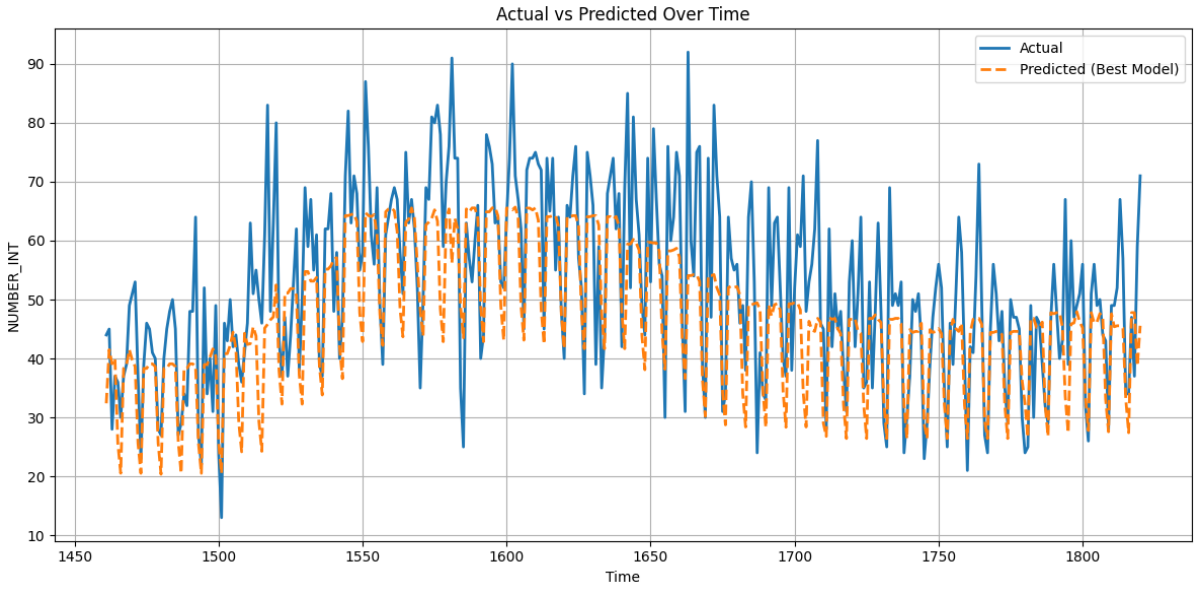


Figure 5.13: XGB - 360 day forecast vs. actual admissions.

Figure 5.14 shows the results of the best XGB model compared to the actual values for the 30 day forecast. The figure confirms the similarities between XGB and RF, with the visualization being practically the same.

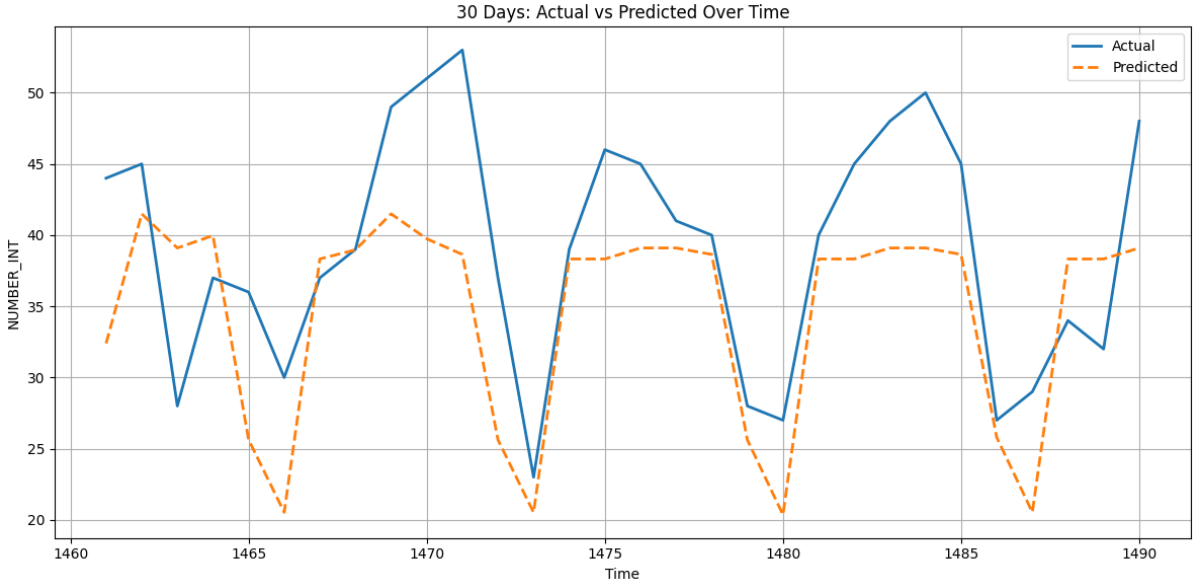


Figure 5.14: XGB - 30 day forecast vs. actual admissions.

Figure 5.15 shows the feature importance plot for the XGB model that achieved the best performance according to RMSE. As in the case of RF, features related to the calendar (such as *Month*, *IsWeekend*) dominate the importance rankings of the model. Although XGB distributes the feature importance slightly more evenly than RF, more than 90% of the total importance remains concentrated in this small subset of calendar based variables. This probably explains the similar behavior and performance of both models,

which appear to rely heavily on calendar patterns, highlighting the role of seasonality. The near absence of meaningful contributions from meteorological variables suggests that additional external predictors might be necessary to improve the model results (particularly to capture admission spikes).

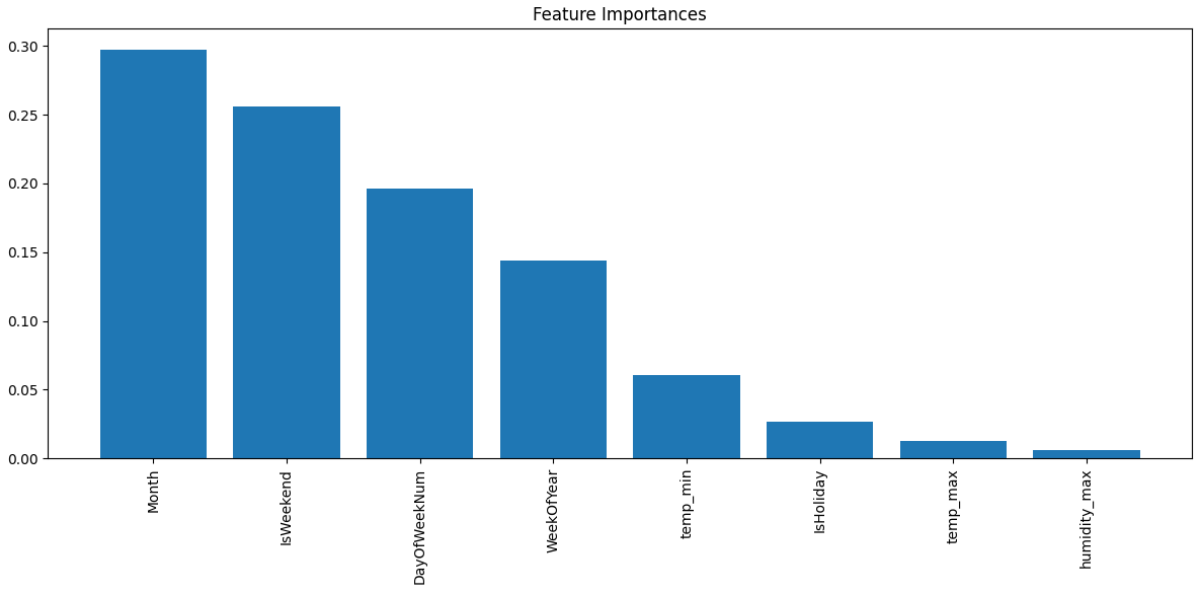


Figure 5.15: XGB - Feature importance extracted from the model.

5.2.4 Machine Learning using Lag and Pollutants

This section extends the assessment of ML models (RF and XGB) incorporating additional predictors derived from lagged meteorological and air quality values. The analysis is divided into two parts. The first part investigates the effect of including lagged meteorological variables (from day 0 to day 3), with performance compared to previously established models (Section 5.2.2 and 5.2.3). The second part evaluates the contribution of pollutants (PM_{10}, O_3, and SO_2). Due to missing values in the pollutant data (each affecting different dates), each pollutant was tested individually. This approach preserves the largest possible test set and avoids the data sparsity that would result from combining all pollutants into a single dataset.

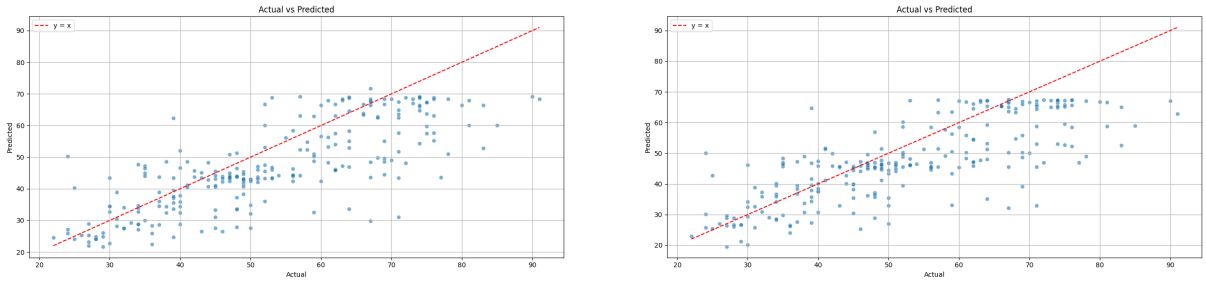
For the first set of tests, **lagged meteorological variables** (from day 0 to day 3) were incorporated into the dataset. Three configurations were evaluated: lagged minimum temperature, lagged minimum temperature and maximum humidity, and lagged minimum and maximum temperatures. These were compared with the baseline models introduced in Sections 5.2.2 and 5.2.3, with both RF and XGB trained using the same method (performance metrics and figures are provided in Appendix B for all tests).

In all configurations, the inclusion of lagged variables did not result in significant improvements over baseline models. In some cases, performance slightly decreased (although the variations are small). Feature importance plots (Appendix B) show that

"calendar" variables remain dominant in all models, contributing to approximately 90% of the predictive weight. Interestingly, lagged temperature features consistently received higher importance compared to the same features without lag, suggesting some potential to capture temporal patterns. Regardless, this added complexity did not improve the final results.

To evaluate the **contribution of pollutants**, separate tests were conducted for each variable (PM_{10}, O_3, and SO_2) using lagged values from day 0 to day 3. As in previous experiments with lagged meteorological variables, these features were added to the baseline predictor set and the models were trained following the same procedure described in Sections 5.2.2 and 5.2.3. Performance metrics are provided in the Appendix B.

Figure 5.16 shows the results for the RF model including the pollutant PM_{10} compared to the corresponding baseline. Visually, both models exhibit similar behavior, with no significant differences.



(a) (b)

Figure 5.16: Random Forest - Actual vs. predicted admissions using the model with pollutants (a) and the baseline (b).

The results (Appendix B) show a slight decline in performance across all metrics when air quality predictors are included in the models (similar to what was observed in tests with lagged temperature variables). One possible explanation relates to the limited quality of the pollutant data. Measurements were obtained from a single monitoring station and include a considerable number of missing values, which may have introduced noise and reduced the ability of the models to capture meaningful patterns.

Nonetheless, feature importance plots suggest that lagged pollutant variables tend to receive more predictive weight than the ones without lag (although their overall contribution remains low and still below features related to the temperature). These limitations, along with the lack of improvement in model accuracy, should not be interpreted as evidence that pollutant variables lack predictive relevance. Rather, they point to the importance of conducting additional analyses using more complete datasets, including measurements from multiple monitoring stations or geographically constrained studies (for example, using only hospitals located near the monitoring station). Future research could also explore the use of more complex models, such as neural networks, which have shown promising results in similar contexts [51, 52].

5.2.5 Discussion

Table 5.10 summarizes the performance of the four models (SARIMA, SARIMAX, RF, and XGB) over 360 days. The results suggest that ML models (RF and XGB) have better performance than traditional time series models over both forecast horizons. For the 360 day test, RF and XGB show significantly lower errors (for example, RMSE around 11 against 13.7 for SARIMAX - Table 5.10) and, despite slightly lower than the SARIMA models, stronger correlation with actual admissions, suggesting that the models are able to capture more accurately seasonal and short term variation.

Table 5.10: Test set performance metrics for 360 day forecast horizon over all models.

Model	MAE	RMSE	MAPE (%)	R^2	Corr.
SARIMA	11.18	13.95	22.13	0.19	0.55
SARIMAX	11.11	13.78	21.83	0.21	0.52
RF	8.21	10.99	15.58	0.49	0.78
XGB	8.25	11.03	15.63	0.49	0.78

The performance of the four models over a 30 day horizon is summarized in Table 5.11.

Table 5.11: Test set performance metrics for 30 day forecast horizon over all models.

Model	MAE	RMSE	MAPE (%)	R^2	Corr.
SARIMA	11.81	13.08	33.08	-1.55	0.75
SARIMAX	7.76	9.53	21.83	-0.36	0.74
RF	6.53	7.79	16.44	0.09	0.70
XGB	6.50	7.76	16.41	0.10	0.71

In the short term test (30 days), the accuracy of the SARIMA models decreased substantially, with SARIMA showing a negative R^2 and the highest MAPE (33%). SARIMAX mitigates the (bad) results, however, still have worst results than the ML approaches. RF and XGB maintain robust correlation values and achieve lower error metrics, indicating superior adaptability to short term variation (even if both tend to smooth peaks).

The similar performance of RF and XGB, as discussed in Section 5.2.3, can be attributed to their shared reliance on calendar predictors and the strong seasonality present in the data[6]. This dependence potentially explains the difficulty of the models in predicting admission peaks, suggesting that the inclusion of alternative or additional predictors, such as road traffic or lagged variables [16], could be necessary.

Regarding the use of pollutants, meteorological variables and past values to train the models, although the literature highlights their association with hospital admissions and reports positive results from their use in forecasting, their inclusion in this study

[6]Models were tested using a reduced set of predictors that excluded potentially redundant "calendar" variables. As shown in Appendix B, both RF and XGB exhibited only minimal performance changes compared to baseline models.

did not lead to performance improvements. One possible explanation is the quality and structure of the data, which come from a single monitoring station and contain missing records. This limits their representativeness and may introduce noise.

Despite these limitations, the feature importance analysis revealed that lagged variables (both meteorological and related to air quality) often received more weight than their non lagged counterparts. This suggest that, although they did not significantly impact performance in this study, lagged features may offer greater predictive value than same day values, as they correspond to delayed symptom manifestation. These conclusions do not invalidate the potential relevance of such indicators, but emphasize the need for more complete datasets, such as pollution measurements from multiple monitoring stations or datasets limited to hospitals located near the monitoring source.

5.3 Pointwise Reliability

This section assesses pointwise reliability methods applied to the hospital admissions use case. The analysis includes the Density and Local Fit method (Section 2.2.5), which was originally designed for classification tasks but is adapted here for regression due to its promising results in the previous experiments. In addition, methods specific for regression tasks are evaluated, such as CONFIVE, CONFINE, and the two proposed variants, denCONFIVE and iqrDenCONFIVE (Sections 2.2.6, 2.2.7, 2.2.8, and 2.2.9, respectively). All methods are evaluated using the benchmark strategy presented in Section 2.3. To ensure comparability between approaches, the threshold parameters were selected using the procedure described in Section 2.2.10, fixing them at the 0.5 percentile. This resulted in a distance threshold of 0.177 for methods relying on distance thresholds and a maximum of 7 neighbors (or cluster units).

5.3.1 Density and Local Fit

This method follows the approach described in Section 2.2.5, which combines density and local fit principles to estimate the pointwise reliability of individual predictions. While originally defined for classification problems, the method is adapted in this case study for the regression tasks by introducing a tolerance threshold to classify predictions as correct or incorrect. All components of the method (density, data agreement and ML agreement) are preserved, but instead of using class labels, correctness is defined based on a fixed absolute error threshold. Predictions are considered correct if their error is within ± 8 units. For example, for the ML agreement component, if the true value is x and the model predicts a nearby instance with a value between $x - 8$ and $x + 8$, the prediction is considered correct.

The threshold of 8 units reflects the average prediction error observed over 360 days (corresponding approximately to the MAE).

Figure 5.17 shows the distribution of the predictions and the corresponding error rates across the reliability intervals. Ideally, higher reliability scores should be associated with lower error, however, the results do not reflect this behavior. After a slight reduction in error between the first and fourth intervals, the error rate increases again in the middle range, reaching 1 in the $[0.60, 0.70]$ interval. The last two intervals, $[0.80, 0.90]$ and $[0.90, 1.00]$, do not show the lowest errors, and, in particular, the final interval has a considerable high error rate. These inconsistencies indicate that the method does not consistently assign higher reliability to more accurate predictions.

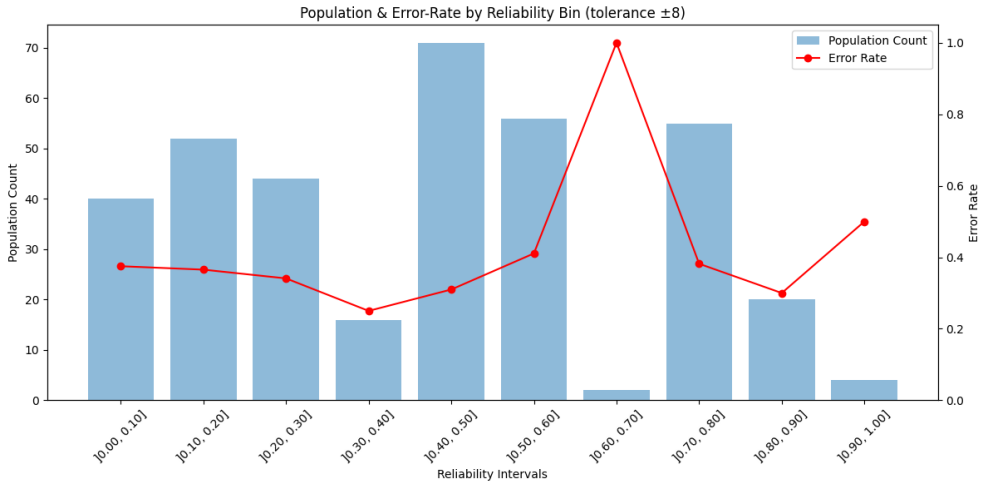


Figure 5.17: Population and error rate by reliability interval (tolerance± 8) using the Density and Local Fit method.

The scatter plot in the Figure 5.18 shows the relationship between absolute error and reliability score.

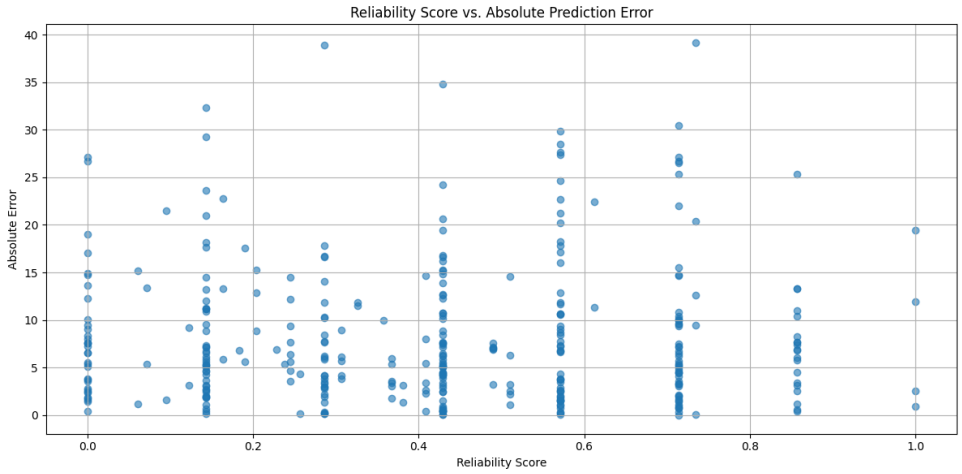


Figure 5.18: Absolute error in relation to the reliability score using the Density and Local Fit method.

Although lower errors are more frequent around mid to high reliability values, high absolute errors still appear throughout the reliability range (Figure 5.18). This dis-

persion shows that high reliability scores are not consistently aligned with accurate predictions.

The method fails to consistently align high reliability scores with low prediction errors, which limits its effectiveness in a regression context. To further investigate whether the approach could be improved, two additional configurations were considered. The first used the Gower distance to account for mixed type variables. The second relied exclusively on categorical features, given their high estimated relevance in the prediction model. However, neither variation led to a meaningful improvement. In the Gower version, high reliability scores continued to coexist with large errors, and no clear pattern was observed across reliability intervals. The categorical configuration produced reliability values concentrated in lower intervals and resulted in a flat error distribution, limiting its ability to distinguish between more reliable and less reliable predictions. The corresponding figures are presented in Appendix B.

5.3.2 CONFIVE

CONFIVE is the first method in this study designed specifically for regression tasks, as described in Section 2.2.6. The method estimates the reliability of a prediction based on the local variance of the target values of its k nearest neighbors. The idea is that if a prediction is surrounded by neighbors with similar results, it is more likely to be reliable. A normalization step maps the local variance to a score in the interval $[0, 1]$.

The evaluation uses the same threshold of ± 8 units to define correctness, following the setup used for the Density and Local Fit method.

Figure 5.19 presents the distribution of the predictions and the corresponding error rates across the reliability intervals.

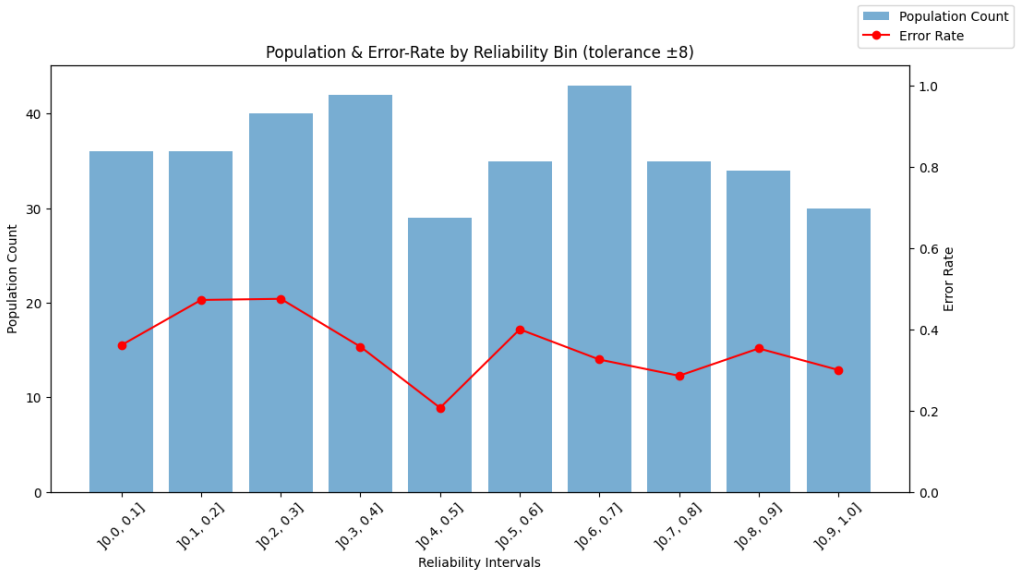


Figure 5.19: Population and error rate by reliability interval using CONFIVE method.

The error rate shows a decreasing tendency from the lowest to the middle intervals, reaching its minimum in $[0.40, 0.50]$. From $[0.60, 1.00]$, error rates remain stable, with only a slight decrease from 0.50 to 1.00. This is a favorable trend compared to the Density and Local Fit method, although the overall decline is modest, resulting in a small difference between the lowest and highest reliability groups in terms of error rate.

The scatter plot in Figure 5.20 shows the absolute prediction error in relation to the reliability score. Compared to the Density and Local Fit method, reliability scores are more widely distributed, but the improvement in this spread remains limited.

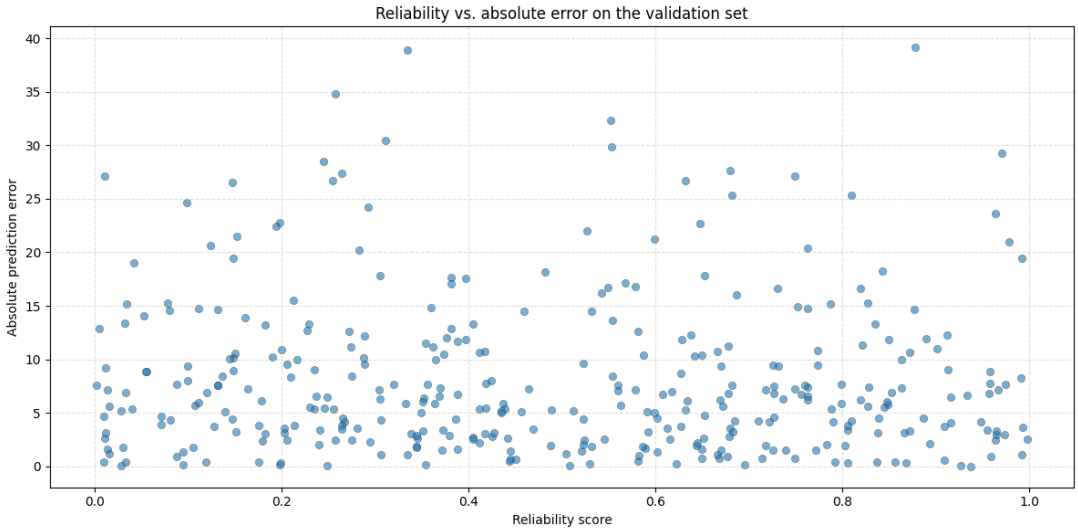


Figure 5.20: Absolute error in relation to the reliability score using CONFIVE method.

The results indicate that CONFIVE aligns the reliability scores with predictive accuracy more consistently than the classification based adaptation. However, contrary to the results presented in Chapter 3, where pointwise methods showed a clearer ability to separate reliable from unreliable predictions in classification tasks, the difference observed remains small. While the method shows improvement over the previous adaptation, it still lacks the ability to discriminate between high and low reliability predictions.

5.3.3 CONFINE

CONFINE, described in Section 2.2.7, computes the mean squared error between the predictions of the model and the true target values of the k nearest neighbors of a given instance. A lower local error indicates greater consistency and leads to a higher reliability score. The raw score is then normalized to the $[0, 1]$ interval.

The distribution of the predictions and associated error rates across the reliability intervals (using the ± 8 unit threshold to define the correctness) are shown in Figure 5.21. Similarly to the CONFIVE method, a decreasing trend in error rates is observed from the $[0.00, 0.10]$ to $[0.40, 0.50]$ reliability intervals. However, from $[0.50, 0.60]$ onward, the

error rate increases consistently across the upper reliability intervals. The final interval, intended to represent the most reliable predictions, exhibits a higher error rate than the lowest reliability interval. These results indicate that CONFINE does not effectively distinguish reliable from unreliable predictions in this case study.

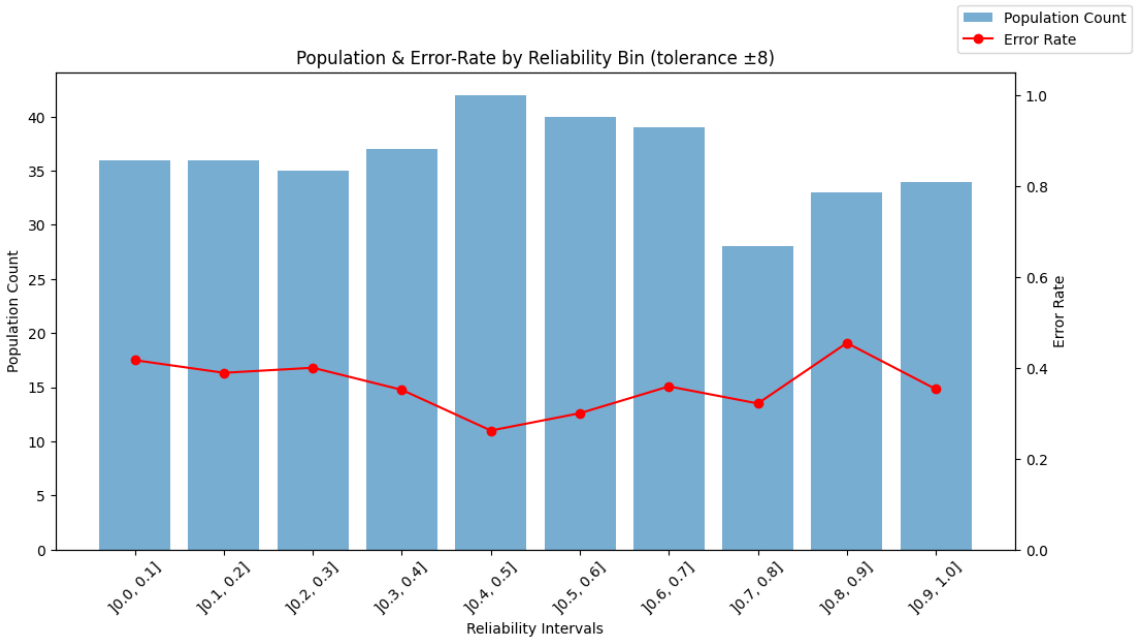


Figure 5.21: Population and error rate by reliability interval using CONFINE method.

Figure 5.22 shows the absolute prediction error in relation to the reliability score. No clear pattern is observed between reliability values and absolute error. Low and high errors appear uniformly throughout the reliability range, further demonstrating the limitations of the method to associate its reliability scores with prediction accuracy.

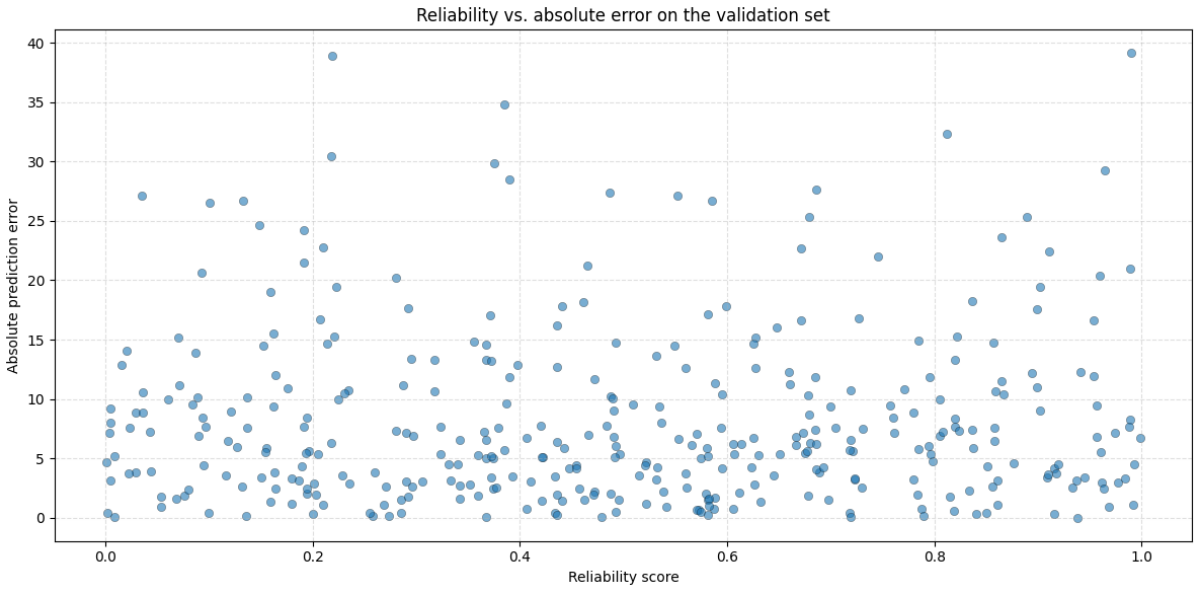


Figure 5.22: Absolute error as a function of the reliability score using the CONFINE method.

In general, CONFINE performs poorly in the current case study. Although originally designed for regression problems, the results of the method are less consistent than those obtained with CONFIVE, which showed a better separation between reliable and unreliable predictions.

5.3.4 denCONFIVE

Following the results of previous case studies, where the combination of density and local fit principles showed potential, the denCONFIVE method (Section 2.2.8) extends CONFIVE by integrating a density component. The final reliability score is computed as the product of the normalized local variance (as in CONFIVE) and a density measure based on the number of training instances within a distance threshold. The goal is to penalize predictions made in sparse regions, while still benefiting from the variance based reliability.

Figure 5.23 shows the error rate across reliability intervals using the ± 8 unit threshold.

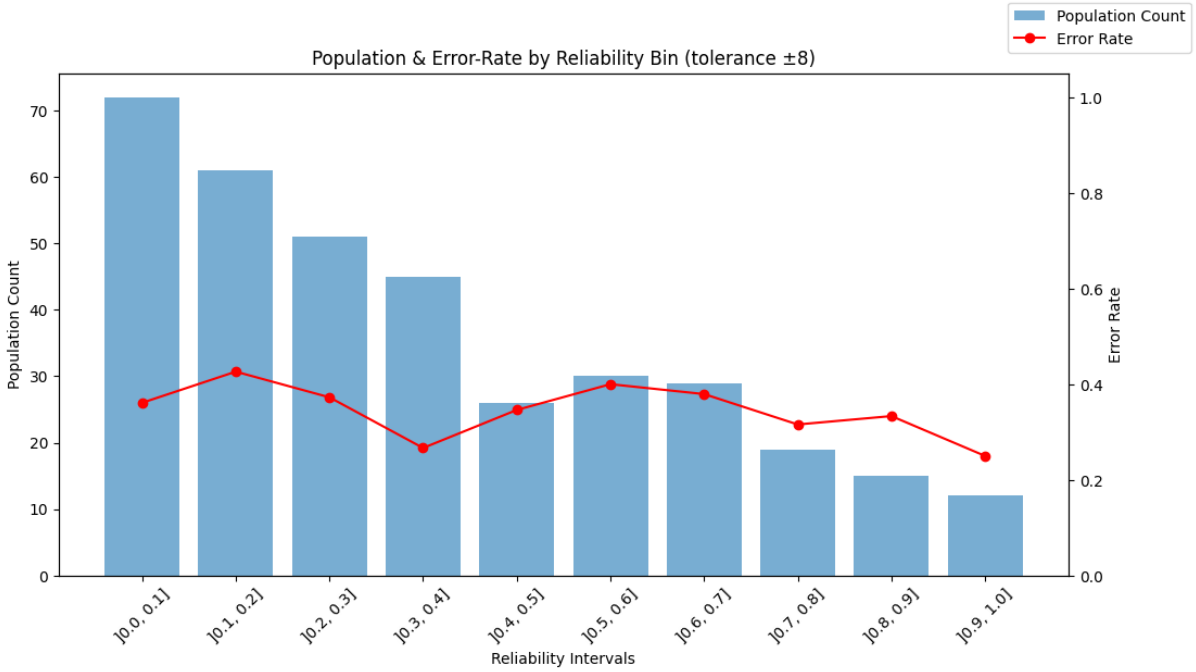


Figure 5.23: Population and error rate by reliability interval using the denCONFIVE method.

The overall pattern resembles the CONFIVE method, with a decrease in error rate from $[0.00, 0.10]$ to $[0.50, 0.60]$, followed by a continued decline through the higher intervals. Although some fluctuations remain, the transition between intervals is smoother, and the error rate in the highest reliability interval is the lowest observed. The method also concentrates a larger number of predictions in the low reliability intervals, which aligns with previous case studies and supports the idea that the method applies stricter reliability assignments.

The absolute prediction error in relation to the reliability score is shown in Figure 5.24. Contrary to previous methods, there appears to be a pattern in which high reliability predictions present lower absolute error. Predictions with high absolute error are still distributed across the range, but their frequency is reduced among the most reliable predictions. At the same time, a substantial number of instances classified with low reliability also present low absolute errors, which may suggest that the method has limited ability to differentiate between high and low reliability predictions.

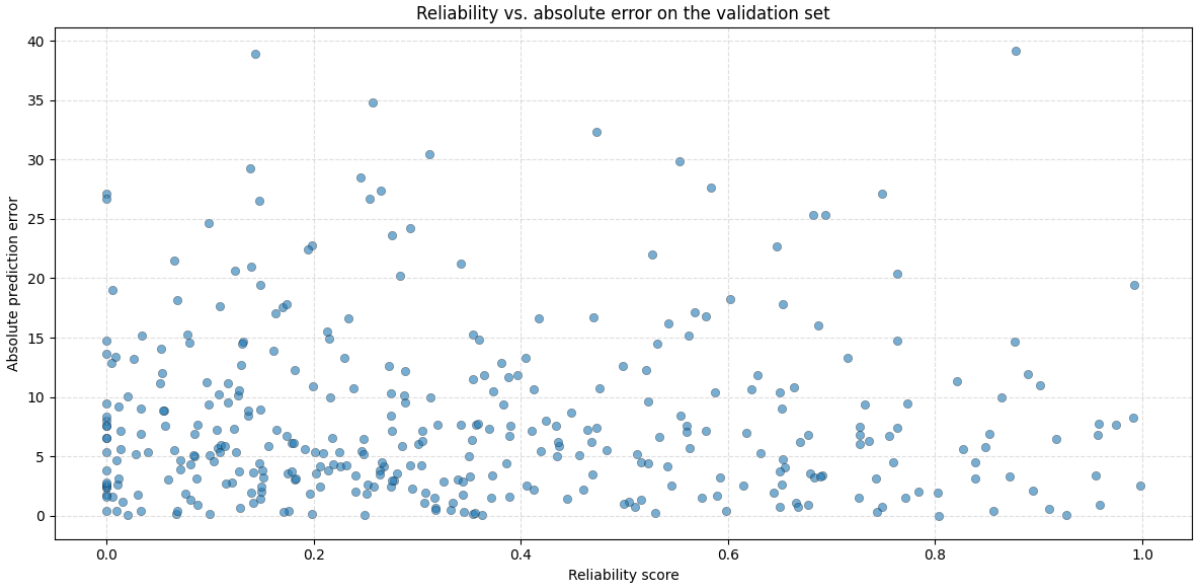


Figure 5.24: Absolute error as a function of the reliability score using the denCONFIVE method.

Among the methods evaluated so far, despite inconsistencies, denCONFIVE shows the most consistent alignment between predicted reliability and observed error.

5.3.5 iqrDenCONFIVE

The iqrDenCONFIVE method (Section 2.2.9) is an adaptation of denCONFIVE that modifies the normalization strategy for the local variance component, with the objective of preserving the differences in local variance while limiting the influence of extreme values.

The error rate in the reliability intervals using the ± 8 unit threshold is shown in Figure 5.25. The results follow a similar pattern to denCONFIVE, with a reduction in error from lower to higher reliability intervals. The method also appears to apply a more restrictive attribution of high reliability, as a low number of predictions are assigned to high reliability intervals. Although some variation is still visible, the transition is smoother and the general results appear slightly better compared to denCONFIVE.

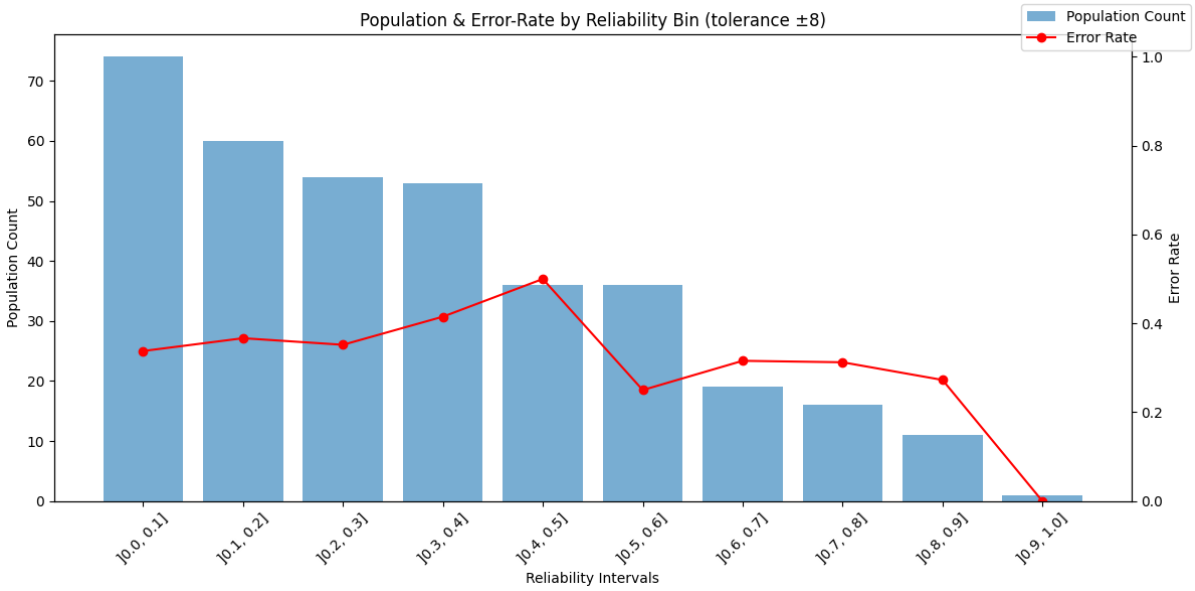


Figure 5.25: Population and error rate by reliability interval using the iqrDenCONFIVE method.

The absolute prediction error in relation to the reliability score is shown in Figure 5.26. The general pattern observed in denCONFIVE is also present, with lower absolute errors appearing more frequently at higher reliability levels. However, as with previous methods, the figure also reveals high error predictions within high reliability regions, as well as some low error predictions among the least reliable instances.

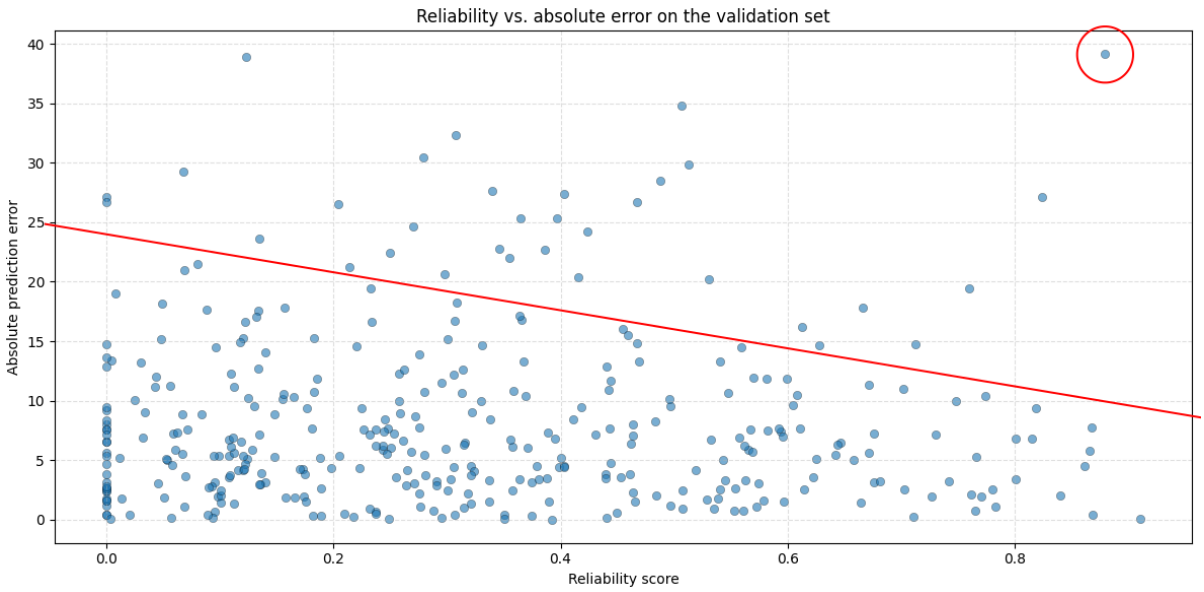


Figure 5.26: Absolute error as a function of the reliability score using the iqrDenCONFIVE method.

To better understand the presence of high reliability predictions with high absolute error, the test instance with the highest absolute error among those classified with high reliability was examined (highlighted in Figure 5.26). The method assigned to this instance a score within one of the highest reliability intervals, which is consistent with

its design: the local variance among its nearest neighbors is relatively low (based on the Min-Max scaling interval $[0, 250]$), which corresponds to a high reliability score.

Table 5.12 lists the feature values of the selected test instance (highlighted in bold) and its seven nearest neighbors. As shown, the neighbors are very similar in terms of features, yet their target values differ substantially. This suggests that the prediction error may be attributed to the incapacity of describing the target values with the current feature set, rather than to a failure in the reliability estimation itself, or that the instance may be an outlier.

Table 5.12: High error, high reliability instance (top row, in bold) and its 7 nearest neighbors.

Day	Week	Month	Weekend	Holiday	Max temp	Min temp	Humidity	y_{true}
0	**30**	**7**	**0**	**0**	**23.9**	**14.5**	**82.0**	**92**
0	30	7	0	0	24.8	15.3	81.0	51
0	28	7	0	0	24.3	15.7	83.0	64
0	29	7	0	0	24.4	13.3	80.0	58
0	30	7	0	0	24.9	14.6	77.0	55
0	28	7	0	0	21.4	13.8	82.0	59
0	25	6	0	0	24.7	14.1	83.0	59
0	31	7	0	0	22.2	14.4	76.0	47

A second case was also examined: an instance with low reliability that resulted in approximately zero absolute error (Table 5.13). Its nearest neighbors showed substantially higher variance in the target values, which combined with the density component supports the low reliability score assigned by the method. This case further confirms that the method responds appropriately to noisy local contexts, even when the regression model performs well.

Table 5.13: Low reliability, low error instance (top row, in bold) and its 7 nearest neighbors.

Day	Week	Month	Weekend	Holiday	Max temp	Min temp	Humidity	y_{true}
5	**3**	**1**	**1**	**0**	**32.7**	**21.8**	**71.0**	**31**
5	1	1	1	0	31.5	21.7	72.0	31
5	2	1	1	0	32.6	20.6	73.0	19
5	4	1	1	0	30.9	21.7	71.0	36
5	7	2	1	0	32.5	21.1	76.0	30
5	7	2	1	0	30.5	21.5	72.0	43
5	4	1	1	0	29.7	20.1	74.0	32
5	3	1	1	0	32.4	21.3	61.0	17

These examples illustrate that iqrDenCONFIVE behaves as intended. The method assigns high reliability to predictions made in dense and locally consistent regions and low reliability in more dispersed or variable contexts. However, due to the characteristics of the data and limitations of the regression model (specifically, features that may not fully explain the target), the reliability scores do not consistently separate high and low error predictions.

Regardless of the constraints, the method maintains a consistent association between reliability and predictive accuracy across most intervals and test instances.

5.3.6 Discussion of the Results

The performance of pointwise reliability methods varied depending on the underlying strategy. Methods that combine the density and local fit principle (such as denCONFIVE and iqrDenCONFIVE) showed the best results. These methods presented a more stable relationship between reliability scores and prediction error, even when the data included irregular or noisy patterns. This supports the findings of the classification case studies (Chapters 3 and 4), where the combination of principles also led to better results.

One of the main challenges appeared to be the presence of instances in the dataset with similar features but very different target values. The features used are likely not sufficient to fully explain the label, which limits the ability of the reliability methods to make accurate assessments. This may also help explain the constraints of the regression models. As discussed in Section 5.2, the models struggled with sharp changes in admission counts, suggesting that both prediction and reliability estimation may be affected by the same constraints.

The Density and Local Fit method, which previously showed promising results in the classification case studies, had difficulty separating reliable from unreliable predictions in this scenario. Methods developed specifically for regression, such as denCONFIVE, generally produced better results. Among them, iqrDenCONFIVE showed the best performance, offering a closer alignment between the reliability scores and the actual prediction error.

Although the results for this regression task are less clear than those of the classification use cases, they still offer useful insights. The methods showed meaningful patterns under less favorable conditions and justify further investigation. Future work could focus on more homogeneous datasets (*e.g.*, data from individual hospitals) or consider expanding the dataset with complementary information, such as additional air quality stations. Testing different models and incorporating domain specific knowledge can also improve the quality of reliability assessment.

6 Conclusions

The primary objective of this project was to investigate pointwise reliability measures for evaluating Machine Learning (ML) model predictions. The motivation for this work is based on the recognition that global performance metrics do not provide sufficient guidance on the reliability of individual predictions, which is a vital aspect for the adoption of ML models in critical applications. To address this issue, a set of model agnostic reliability methods was applied in three case studies, with the purpose of assessing their effectiveness in classification and regression tasks.

The first two case studies focused on clinical classification tasks that involve treatment decisions and mortality related to cardiovascular problems. The analysis revealed that clustering based methods, such as subtractive and DBSCAN, struggled to differentiate between reliable and unreliable predictions, as most were concentrated at the extremes of the reliability intervals. Distance based methods provided a more balanced distribution of scores, but produced inconsistent results. The ICM method showed a clearer decrease in error rates with higher reliability values, although its restrictive algorithm classifies all predictions as having low reliability, which limits its usability. Among all the approaches evaluated, the Density and Local fit method showed the most promising results. The method balanced prediction in all reliability intervals and showed a consistent relationship between higher reliability and lower error rates, making it a robust solution for classification problems.

The third case study shifted to the forecasting of hospital admissions and introduced additional challenges. In this context, models were evaluated to assess their forecasting performance. Traditional statistical models, such as SARIMA and SARIMAX, were able to capture seasonal patterns but failed to represent daily fluctuations, leading to relatively high errors. ML approaches, specifically Random Forest and XGB, consistently outperformed these baselines. They achieved lower error metrics and stronger correlations with actual admissions, demonstrating superior results in both short term and long term forecasts. Regarding the pointwise reliability, methods specific to regression tasks were also assessed. The overall performance in this case study was not as strong as that observed in the classification tasks. The CONFIVE provided modest results, while the CONFINE was less successful in aligning the reliability scores with the accuracy of the predictions. The most promising results were observed in denCONFIVE and iqrDenCONFIVE methods, which combine the density and local fit principles. The latter showed a better association between reliability scores and prediction errors, although some high reliability predictions still had large errors due to

dataset limitations, such as missing explanatory features.

In summary, the findings of this project show that pointwise reliability can improve trust in individual predictions of ML models. The work demonstrates that methods combining density and local fit principles generally produce more stable and meaningful reliability estimates than methods based on either principle alone. It also shows that ML models should be considered when forecasting hospital admissions, as they showed better results than traditional forecasting approaches.

Nevertheless, several limitations must be acknowledged. In the forecasting case study, the datasets had incomplete pollutant data and a restricted geographic scope that limited the analysis. Regarding pointwise reliability, while a broad set of methods was evaluated, it was not possible to cover all the approaches proposed in the literature, and parameter optimization for the methods was not explored in depth.

The limitations identified above point to several directions for future work.

For the forecasting problem, the evaluation should be expanded to more complete or representative datasets.

The pointwise reliability assessment would further benefit from a more consolidated dataset, because, despite showing promising results, those were not as strong as the results observed in the other case studies. In general, reliability assessment could also benefit from the integration of domain knowledge, which has proven to be effective in other research.

Finally, hybrid or adaptive reliability strategies that take into account the class distribution and the importance of features may also lead to better results.

As a result of this work, three publications were produced, each corresponding to one of the case studies presented. The first was published in a scientific journal and the others were presented at conferences:

1. *"Benchmarking Methods for Pointwise Reliability"* - published in Information[39];
2. *"Pointwise Reliability Assessment"* - published and presented at the Experiment@ International Conference 2025 (exp.at'25)[1];
3. *"Forecasting Respiratory-Related Hospital Admissions in Belo Horizonte: A Comparison of SARIMA(X) and Tree-Based ML"* - published and presented at the 31st Portuguese Conference on Pattern Recognition (RECPAD 2025)[2].

[1] `https://expat.org.pt/expat25/`; Accessed: 5 October 2025.
[2] `https://sites.google.com/view/recpad2025`; Accessed: 5 October 2025.

References

[1] A. Qayyum, J. Qadir, M. Bilal, and A. Al-Fuqaha, "Secure and robust machine learning for healthcare: A survey," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 156–180, 2021.

[2] M. Javaid, A. Haleem, R. Pratap Singh, R. Suman, and S. Rab, "Significance of machine learning in healthcare: Features, pillars and applications," *International Journal of Intelligent Networks*, vol. 3, pp. 58–73, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666603022000069

[3] B. Kompa, J. Snoek, and A. L. Beam, "Second opinion needed: communicating uncertainty in medical machine learning," *NPJ Digital Medicine*, vol. 4, no. 1, p. 4, 2021.

[4] S. Paredes, T. Rocha, S. Sousa, J. Henriques, J. Sousa, and L. Gonçalves, "Trust in machine learning models: Global and individual assessment," in *2024 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2024, pp. 535–542.

[5] J. Henriques, T. Rocha, P. de Carvalho, C. Silva, and S. Paredes, "Interpretability and explainability of machine learning models: Achievements and challenges," in *International Conference on Biomedical and Health Informatics 2022*, E. Pino, R. Magjarević, and P. de Carvalho, Eds. Cham: Springer Nature Switzerland, 2024, pp. 81–94.

[6] H. Wei, R. Xie, H. Cheng, L. Feng, B. An, and Y. Li, "Mitigating neural network overconfidence with logit normalization," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 23 631–23 644. [Online]. Available: https://proceedings.mlr.press/v162/wei22d.html

[7] A. C. Lorena, P. Y. A. Paiva, and R. B. C. Prudêncio, "Trusting my predictions: On the value of instance-level analysis," *ACM Comput. Surv.*, vol. 56, no. 7, Apr. 2024. [Online]. Available: https://doi.org/10.1145/3615354

[8] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[9] K. Lenhof, L. Eckhart, L.-M. Rolli, and H.-P. Lenhof, "Trust me if you can: a survey on reliability and interpretability of machine learning approaches for

drug sensitivity prediction in cancer," *Briefings in Bioinformatics*, vol. 25, no. 5, p. bbae379, 08 2024. [Online]. Available: https://doi.org/10.1093/bib/bbae379

[10] P. Schulam and S. Saria, "Can you trust this prediction? auditing pointwise reliability after learning," in *AISTATS 2019 - 22nd International Conference on Artificial Intelligence and Statistics*, 2020.

[11] G. Nicora, M. Rios, A. Abu-Hanna, and R. Bellazzi, "Evaluating pointwise reliability of machine learning prediction," *Journal of Biomedical Informatics*, vol. 127, 2022.

[12] E. Askanazi and I. Grinberg, "Analysis of machine learning prediction reliability based on sampling distance evaluation with feature decorrelation," *Machine Learning: Science and Technology*, vol. 5, no. 2, p. 025030, may 2024. [Online]. Available: https://dx.doi.org/10.1088/2632-2153/ad4231

[13] Z. Zhao, P. Gil, J. Loureiro, L. Petrella, and J. Henriques, "Knowledge-based reliability assessment of models with application to risk stratification," in *2024 IEEE 22nd Mediterranean Electrotechnical Conference (MELECON)*, 2024, pp. 299–304.

[14] J. Loureiro, L. Petrella, P. Gil, T. Rocha, S. Paredes, and J. Henriques, "Towards a comprehensive taxonomy of pointwise reliability assessment techniques for machine learning models," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–19, 2025.

[15] S. Grøntved, M. Jørgine Kirkeby, S. Paaske Johnsen, J. Mainz, J. Brink Valentin, and C. Mohr Jensen, "Towards reliable forecasting of healthcare capacity needs: A scoping review and evidence mapping," *International Journal of Medical Informatics*, vol. 189, p. 105527, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1386505624001904

[16] J. Mosselmans, "Refined forecasting of future hospital admissions for accurate operational planning decisions," Master's thesis, Faculty of Economics and Business Administration, Ghent University, 2023, available at https://libstore.ugent.be/fulltxt/RUG01/003/144/487/RUG01-003144487_2023_0001_AC.pdf.

[17] M. A. Bud, I. Moldovan, L. Radu, M. Nedelcu, and E. Figueiredo, "Reliability of probabilistic numerical data for training machine learning algorithms to detect damage in bridges," *Structural Control and Health Monitoring*, vol. 29, no. 7, p. e2950, 2022.

[18] C. De Maio, G. Fenza, M. Gallo, V. Loia, and C. Stanzione, "Toward reliable machine learning with congruity: a quality measure based on formal concept analysis," *Neural Computing and Applications*, vol. 35, no. 2, pp. 1899–1913, 2023.

[19] F. Valente, S. Paredes, J. Henriques, T. Rocha, P. de Carvalho, and J. Morais, "Interpretability, personalization and reliability of a machine learning based clinical

decision support system," *Data Mining and Knowledge Discovery*, vol. 36, no. 3, pp. 1140–1173, 2022.

[20] M. E. Hellman, "The nearest neighbor classification rule with a reject option," *IEEE Transactions on Systems Science and Cybernetics*, vol. 6, no. 3, pp. 179–185, 1970.

[21] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, 1996, pp. 226–231.

[22] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[23] G. Nicora and R. Bellazzi, "A reliable machine learning approach applied to single-cell classification in acute myeloid leukemia," in *AMIA annual symposium proceedings*, vol. 2020. American Medical Informatics Association, 2020, p. 925.

[24] J. C. Riquelme, J. S. Aguilar-Ruiz, and M. Toro, "Finding representative patterns with ordered projections," *Pattern Recognition*, vol. 36, no. 4, pp. 1009–1018, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S003132030200119X

[25] J. Sousa, S. Paredes, J. Henriques, L. Gonçalves, and T. Rocha, "Pointwise reliability metrics for machine learning model predictions: three different approaches," in *2024 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2024, pp. 550–555.

[26] F. P. Preparata and M. I. Shamos, *Convex Hulls: Basic Algorithms*. New York, NY: Springer New York, 1985, pp. 95–149. [Online]. Available: https://doi.org/10.1007/978-1-4612-1098-6_3

[27] J. Waa, J. Diggelen, M. Neerincx, and S. Raaijmakers, "Icm: An intuitive model independent and accurate certainty measure for machine learning," in *Proceedings of the 10th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART*, INSTICC. SciTePress, 2018, pp. 314–321.

[28] J. van der Waa, T. Schoonderwoerd, J. van Diggelen, and M. Neerincx, "Interpretable confidence measures for decision support systems," *International Journal of Human-Computer Studies*, vol. 144, p. 102493, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1071581920300951

[29] B. Kailkhura, B. Gallagher, S. Kim, A. Hiszpanski, and T. Y.-J. Han, "Reliable and explainable machine-learning methods for accelerated material discovery," *npj Computational Materials*, vol. 5, no. 1, p. 108, 2019.

[30] S. Briesemeister, J. Rahnenführer, and O. Kohlbacher, "No longer confidential: estimating the confidence of individual regression predictions," *PloS one*, vol. 7, no. 11, p. e48723, 2012.

[31] G. Adomavicius and Y. Wang, "Improving reliability estimation for individual numeric predictions: a machine learning approach," *INFORMS Journal on Computing*, vol. 34, no. 1, pp. 503–521, 2022.

[32] M. V. Trindade, "Reliability assessment of machine learning regression models," Master's thesis, Faculty of Sciences and Technology, University of Coimbra, 2024, available at https://hdl.handle.net/10316/117755.

[33] X. Zhang and I. Bose, "Reliability estimation for individual predictions in machine learning systems: A model reliability-based approach," *Decision Support Systems*, vol. 186, p. 114305, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923624001386

[34] P. D. Myers, K. Ng, K. Severson, U. Kartoun, W. Dai, W. Huang, F. A. Anderson, and C. M. Stultz, "Identifying unreliable predictions in clinical risk models," *NPJ digital medicine*, vol. 3, no. 1, p. 8, 2020.

[35] X. Xie, J. W. Ho, C. Murphy, G. Kaiser, B. Xu, and T. Y. Chen, "Testing and validating machine learning classifiers by metamorphic testing," *Journal of Systems and Software*, vol. 84, no. 4, pp. 544–558, 2011, the Ninth International Conference on Quality Software. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121210003213

[36] Q.-H. Luu, M. F. Lau, S. P. Ng, and T. Y. Chen, "Testing multiple linear regression systems with metamorphic testing," *Journal of Systems and Software*, vol. 182, p. 111062, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016412122100159X

[37] F. Valente, J. Henriques, S. Paredes, T. Rocha, P. de Carvalho, and J. Morais, "A new approach for interpretability and reliability in clinical risk prediction: Acute coronary syndrome scenario," *Artificial Intelligence in Medicine*, vol. 117, p. 102113, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0933365721001068

[38] J. Henriques, T. Rocha, S. Paredes, P. Gil, J. Loureiro, and L. Petrella, "Pointwise reliability of machine learning models: Application to cardiovascular risk assessment," in *9th European Medical and Biological Engineering Conference*, T. Jarm, R. Šmerc, and S. Mahnič-Kalamiza, Eds. Cham: Springer Nature Switzerland, 2024, pp. 213–222.

[39] C. Correia, S. Paredes, T. Rocha, J. Henriques, and J. Bernardino, "Benchmarking methods for pointwise reliability," *Information*, vol. 16, no. 4, 2025. [Online]. Available: https://www.mdpi.com/2078-2489/16/4/327

[40] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intelligent & fuzzy systems*, vol. 2, no. 3, pp. 267–278, 1994.

[41] S. K. Chandar, "Stock market prediction using subtractive clustering for a neuro fuzzy hybrid approach," *Cluster Computing*, vol. 22, no. Suppl 6, pp. 13 159–13 166, 2019.

[42] F. Sahigara, K. Mansouri, D. Ballabio, A. Mauri, V. Consonni, and R. Todeschini, "Comparison of different approaches to define the applicability domain of qsar models," *Molecules*, vol. 17, no. 5, pp. 4791–4810, 2012. [Online]. Available: https://www.mdpi.com/1420-3049/17/5/4791

[43] S. Sousa, S. Paredes, T. Rocha, J. Henriques, J. Sousa, and L. Gonçalves, "Machine learning models' assessment: trust and performance," *Medical & Biological Engineering & Computing*, vol. 62, no. 11, pp. 3397–3410, Nov. 2024. [Online]. Available: https://doi.org/10.1007/s11517-024-03145-5

[44] A. Cervati Neto, A. L. M. Levada, and M. Ferreira Cardia Haddad, "Supervised t-sne for metric learning with stochastic and geodesic distances," *IEEE Canadian Journal of Electrical and Computer Engineering*, vol. 47, no. 4, pp. 199–205, 2024.

[45] C. Correia, "Pointwise benchmark," https://github.com/Oak10/pointwise-benchmark, 2024, github repository, (commit b32496f). Accessed: 17 April 2025.

[46] M. Sadikin, "Ehr dataset for patient treatment classification," *Mendeley Data*, vol. 1, p. 2020, 2020.

[47] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, 3rd ed. John Wiley & Sons, 2013.

[48] P. de Araújo Gonçalves, J. Ferreira, C. Aguiar, and R. Seabra-Gomes, "Timi, pursuit, and grace risk scores: sustained prognostic value and interaction with revascularization in nste-acs," *European heart journal*, vol. 26, no. 9, pp. 865–872, 2005.

[49] J. Trovão, J. Henriques, L. Petrella, M. Trindade, and J. Loureiro, "Pointwise reliability assessment: Density vs. local fit methods," in *2024 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2024, pp. 556–563.

[50] A. Vieira, I. Sousa, and S. Dória-Nóbrega, "Forecasting daily admissions to an emergency department considering single and multiple seasonal patterns," *Healthcare Analytics*, vol. 3, p. 100146, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772442523000138

[51] C. N. Rocha and F. Rodrigues, "Forecasting emergency department admissions," *Intelligent Data Analysis*, vol. 25, no. 6, pp. 1579–1601, 2021. [Online]. Available: https://journals.sagepub.com/doi/abs/10.3233/IDA-205390

[52] Y. d. S. Tadano, E. T. Bacalhau, L. Casacio, E. Puchta, T. S. Pereira, T. Antonini Alves, C. M. L. Ugaya, and H. V. Siqueira, "Unorganized machines to estimate the number of hospital admissions due to respiratory diseases caused

by pm10 concentration," *Atmosphere*, vol. 12, no. 10, 2021. [Online]. Available: https://www.mdpi.com/2073-4433/12/10/1345

[53] J. S. d. Reis, R. L. Costa, F. D. d. S. Silva, E. D. F. de Souza, T. R. Cortes, R. H. Coelho, S. R. M. Velasco, D. J. D. Neves, J. F. Sousa Filho, C. E. C. Barreto, J. B. Cabral Júnior, H. S. dos Reis, K. R. Mendes, M. C. C. Lins, T. R. Ferreira, M. H. G. d. S. Vanderlei, M. F. Alonso, G. L. Mariano, H. B. Gomes, and H. B. Gomes, "Predicting asthma hospitalizations from climate and air pollution data: A machine learning-based approach," *Climate*, vol. 13, no. 2, 2025. [Online]. Available: https://www.mdpi.com/2225-1154/13/2/23

[54] H. Álvarez-Chaves, P. Muñoz, and M. D. R-Moreno, "Machine learning methods for predicting the admissions and hospitalisations in the emergency department of a civil and military hospital," *Journal of Intelligent Information Systems*, vol. 61, no. 3, pp. 881–900, 2023.

[55] C. Brossard, C. Goetz, P. Catoire, L. Cipolat, C. Guyeux, C. Gil Jardine, M. Akplogan, and L. Abensur Vuillaume, "Predicting emergency department admissions using a machine-learning algorithm: a proof of concept with retrospective study," *BMC Emergency Medicine*, vol. 25, no. 1, p. 3, 2025.

[56] B. Pawar, L. Garg, V. Prakash, R. Stevenson, M. Amaira, D. Soloducha, and A. Kraus, "Satellite-derived no2 data to predict hospital admissions: A machine learning-based approach," in *2024 International Conference on Emerging Technologies and Innovation for Sustainability* (*EmergIN*), 2024, pp. 491–496.

[57] W. J. Requia, A. M. Vicedo-Cabrera, E. de Schrijver, H. Amini, and A. Gasparrini, "Association of high ambient temperature with daily hospitalization for cardiorespiratory diseases in brazil: A national time-series study between 2008 and 2018," *Environmental Pollution*, vol. 331, p. 121851, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0269749123008539

[58] W. J. Requia, A. M. Vicedo-Cabrera, H. Amini, G. L. da Silva, J. D. Schwartz, and P. Koutrakis, "Short-term air pollution exposure and hospital admissions for cardiorespiratory diseases in brazil: A nationwide time-series study between 2008 and 2018," *Environmental Research*, vol. 217, p. 114794, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0013935122021211

[59] C. Correia, "hospital-adm-pw-rel," https://github.com/Oak10/hospital-adm-pw-rel, 2025, github repository, (commit 573ce04). Accessed: 17 August 2025.

[60] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[61] J. D. Hamilton, *Time series analysis*. Princeton university press, 2020.

[62] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[63] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 785–794. [Online]. Available: https://doi.org/10.1145/2939672.2939785

Appendices

Appendix A - Patient Treatment

Complementary results of pointwise reliability methods for the case study of patient treatment classification (Chapter 3).

t-SNE Visualizations

This section presents the t-distributed Stochastic Neighbor Embedding (t-SNE) visualizations for one fold of each pointwise reliability method (one iteration of the benchmark). All figures were originally introduced in Correia *et al.* [39] and support the quantitative findings.

The t-SNE visualization for the subtractive clustering method is shown in Figure A.1. Predictions are grouped by reliability interval and class.

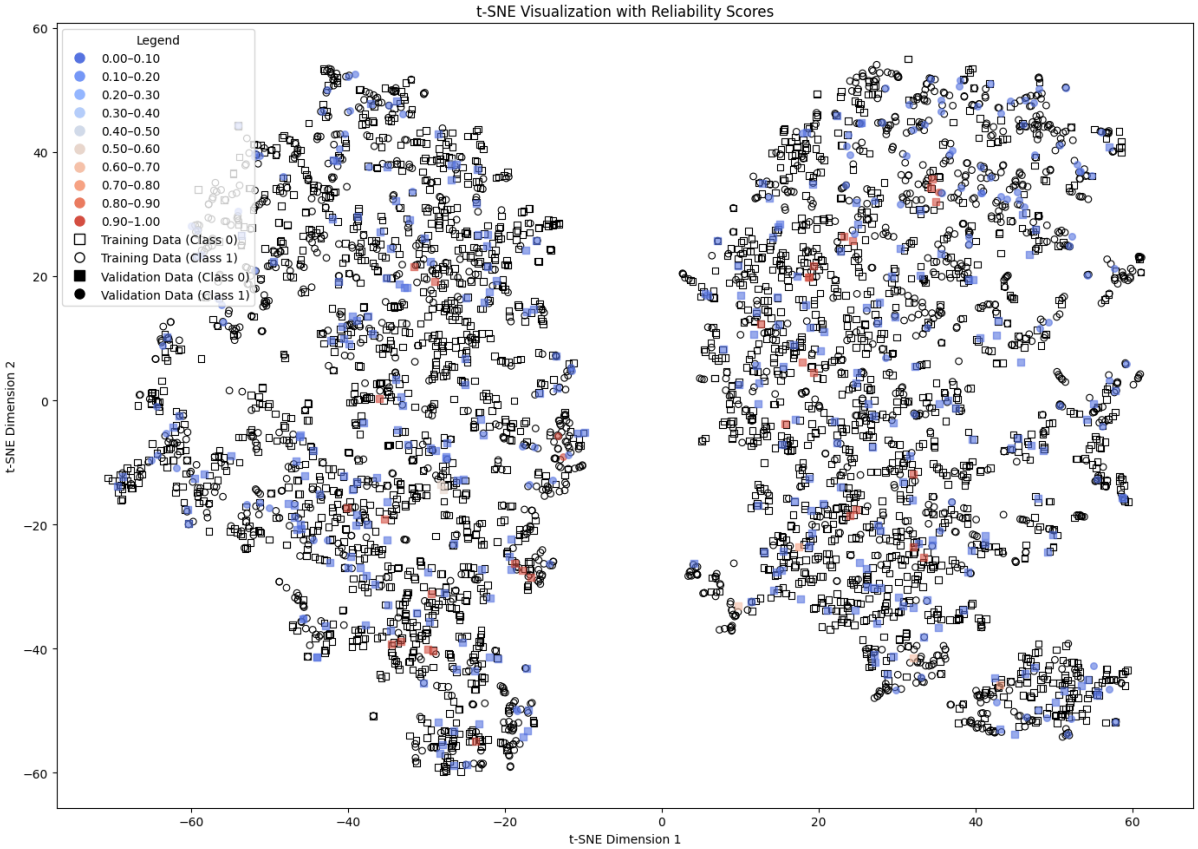


Figure A.1: t-SNE visualization of the training and validation datasets using the Subtractive Clustering method, categorized by reliability intervals and separated by class [39].

The visualization in Figure A.1 does not reveal any clear relationship between forecast reliability and local structure, with many instances marked as unreliable despite being surrounded by similar points.

Similar patterns are observed in Figure A.2 for the DBSCAN method. The visualization does not reveal a strong link between prediction reliability and class separation, further confirming the instability seen in the quantitative performance of the method.

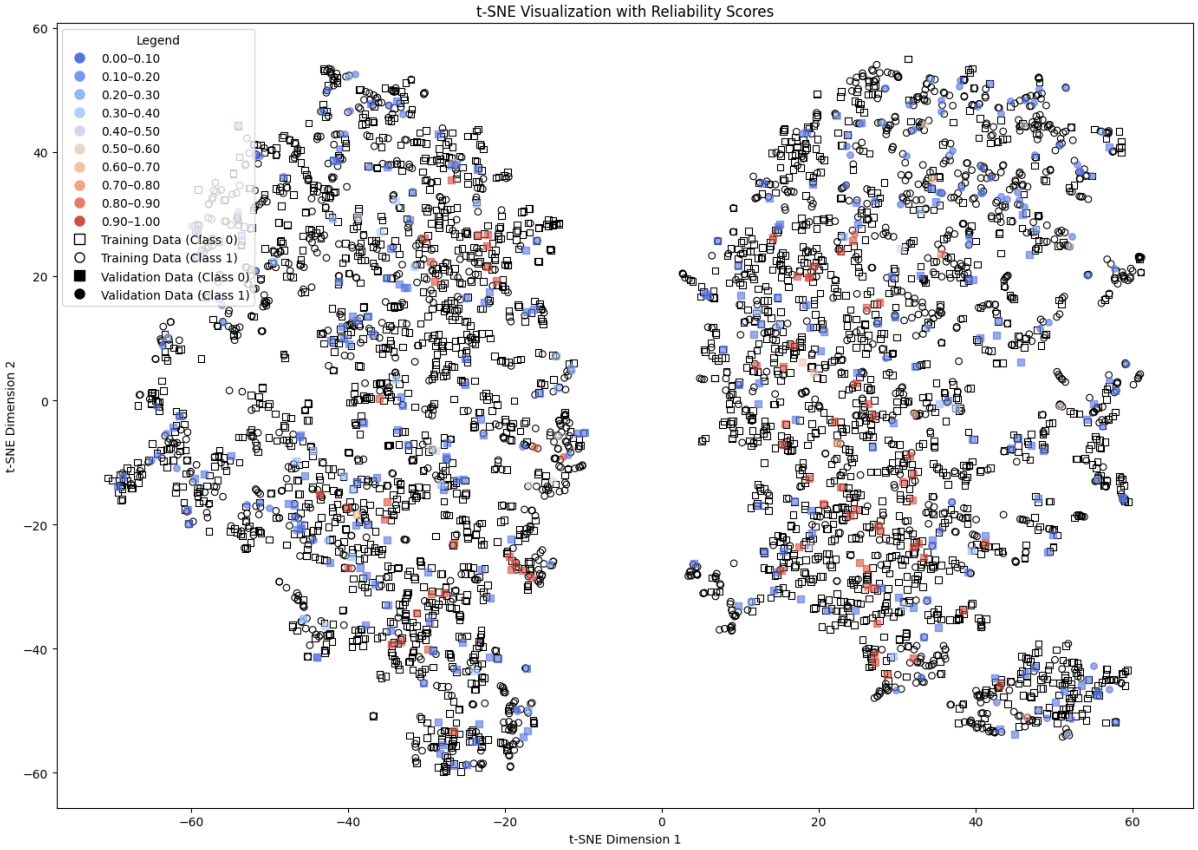


Figure A.2: t-SNE visualization of the training and validation datasets using the DBSCAN method, categorized by reliability intervals and separated by class [39].

The Distance Based method, shown in Figure A.3, assigns low reliability to isolated points, capturing local sparsity. However, high reliability predictions often occur in overlapping class regions.

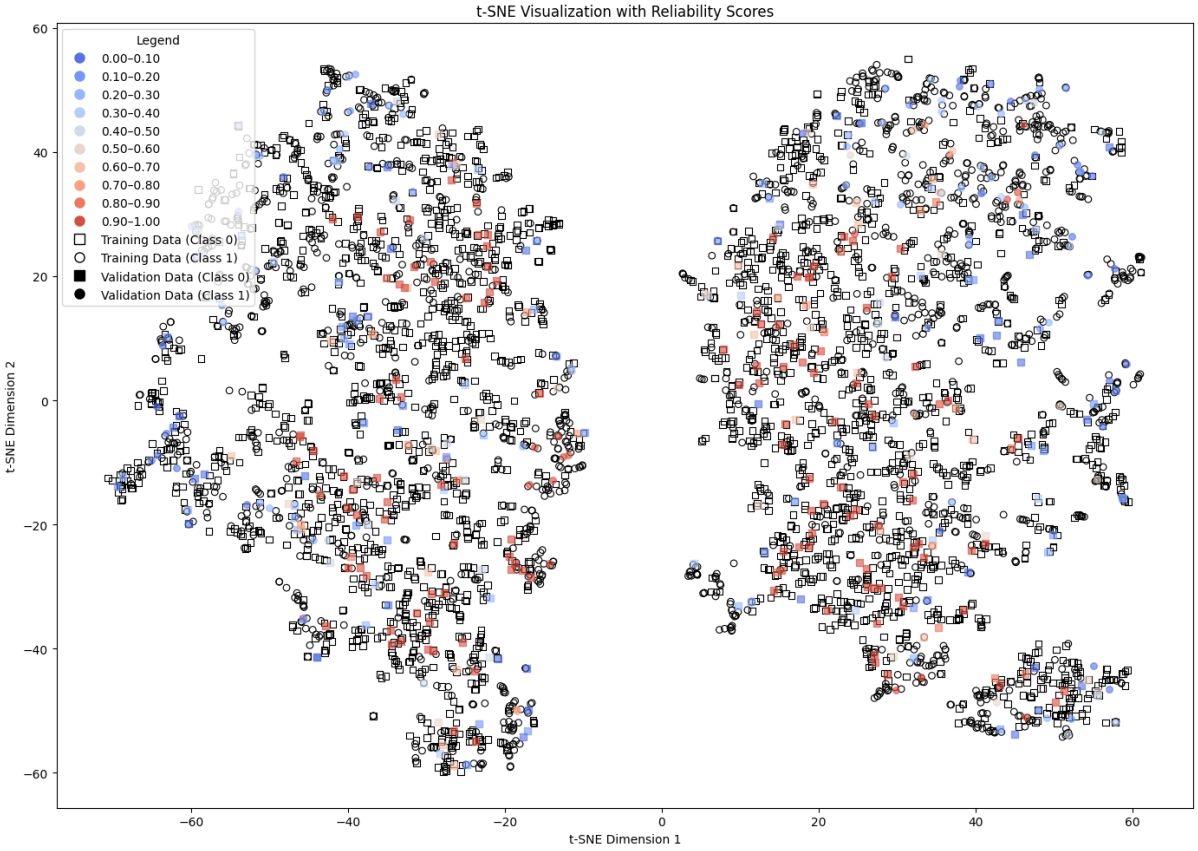


Figure A.3: t-SNE visualization of the training and validation datasets using the Distance Based method, categorized by reliability intervals and separated by class [39].

In Figure A.4, the ICM method is shown to assign low reliability even to instances surrounded by neighbors of the same class.

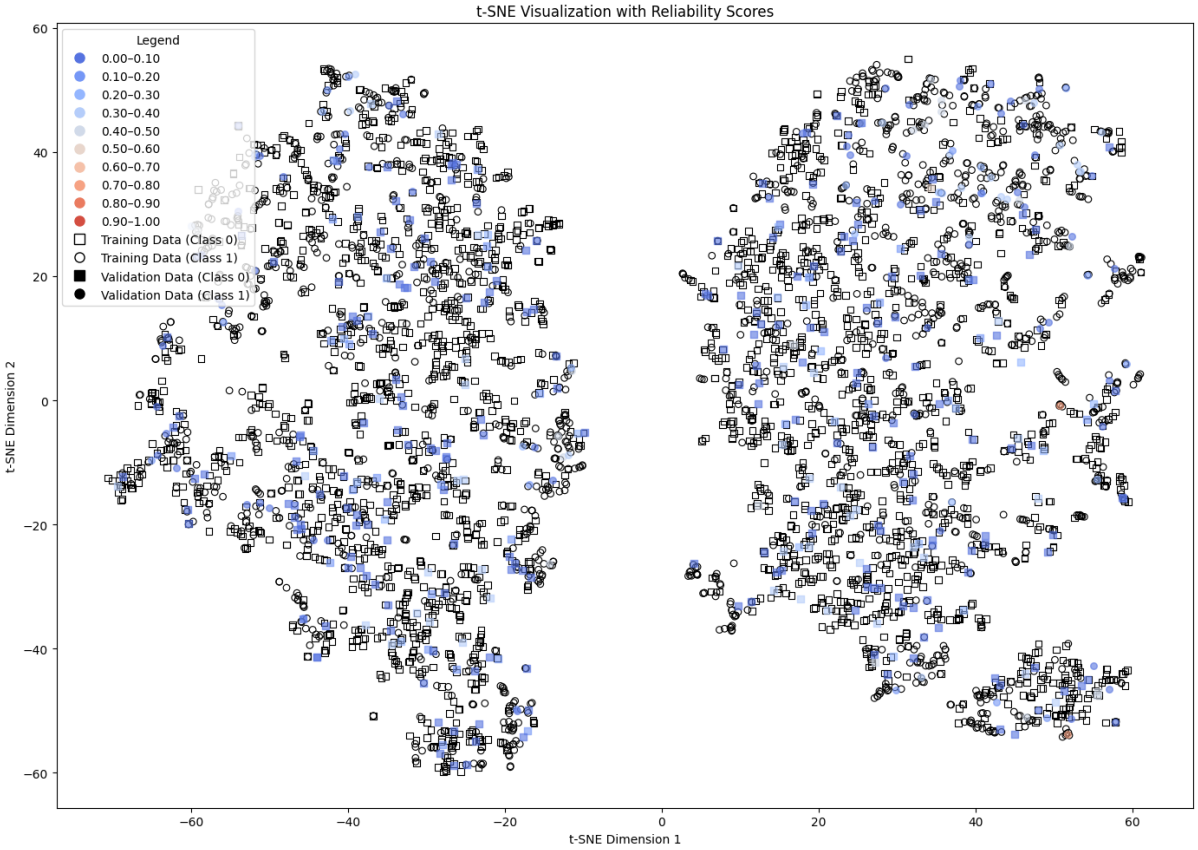


Figure A.4: t-SNE visualization of the training and validation datasets using the ICM method, categorized by reliability intervals and separated by class [39].

Figure A.5 shows that high reliability predictions from the Density and Local Fit method are typically located in dense, homogeneous regions, while low reliability predictions appear near class boundaries.

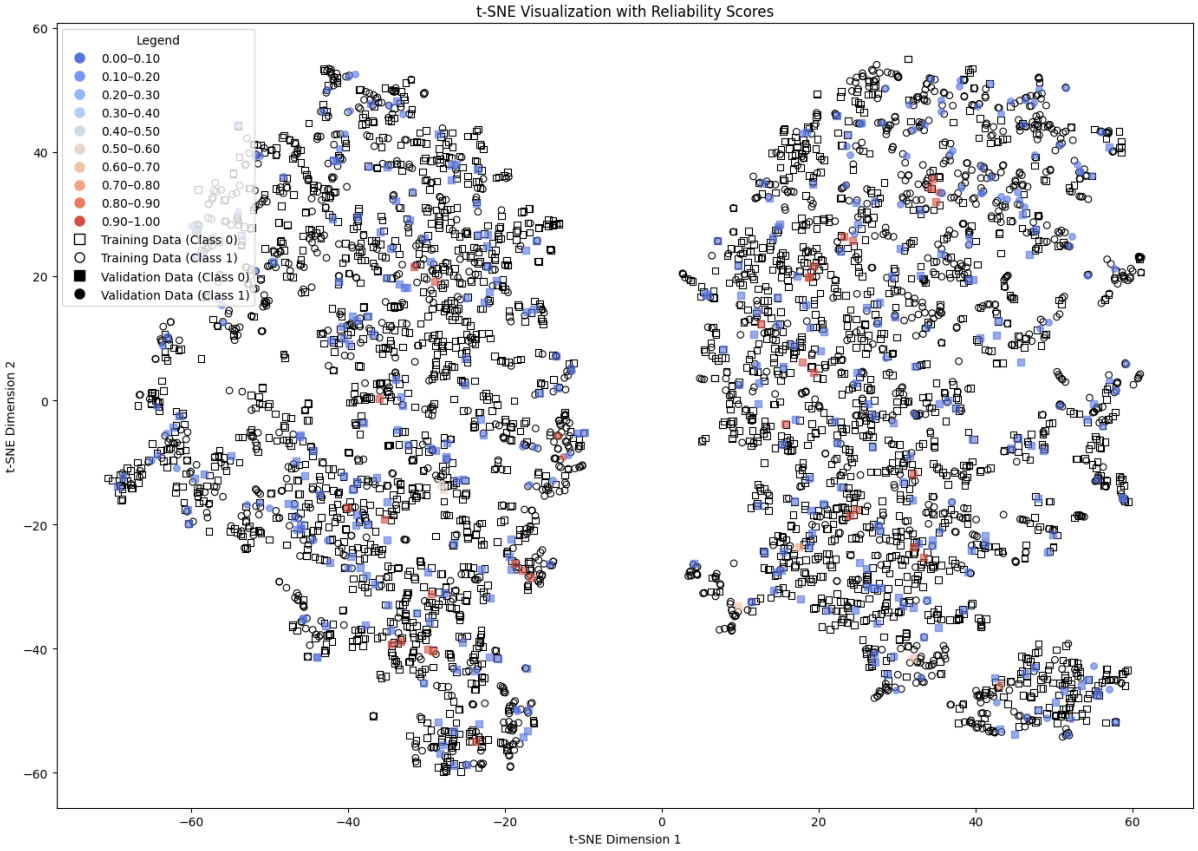


Figure A.5: t-SNE visualization of the training and validation datasets using the Density and Local Fit method, categorized by reliability intervals and separated by class [39].

Appendix B - Hospital Admissions

Complementary results from additional experiments with pointwise reliability methods and using models trained with reduced feature sets, lagged meteorological variables (temperature and humidity) and atmospheric pollutants (SO_2, PM_{10} and O_3) for the Hospital Admissions case study (Chapter 5).

Reduced Predictors

Table B.1 summarizes the performance of Random Forest (RF) and eXtreme Gradient Boosting (XGB) models trained using only a reduced set of predictors (*WeekOfYear*, *DayOfWeekNum*, *temp_min*, *temp_max*, *humidity_max*, and *IsHoliday*.)

Table B.1: Test set performance using reduced predictor sets (360 day horizon).

Model	**MAE**	**RMSE**	**MAPE (%)**	**R^2**	**Corr.**
RF (reduced)	8.18 ± 0.03	10.93 ± 0.04	15.54 ± 0.03	0.49 ± 0.003	0.78 ± 0.002
XGB (reduced)	8.23 ± 0.02	10.98 ± 0.04	15.63 ± 0.02	0.49 ± 0.004	0.78 ± 0.001

Figure B.1 shows the results of the best RF model using reduced predictor sets (that is, the iteration with the lowest RMSE in the test set) compared to the actual values.

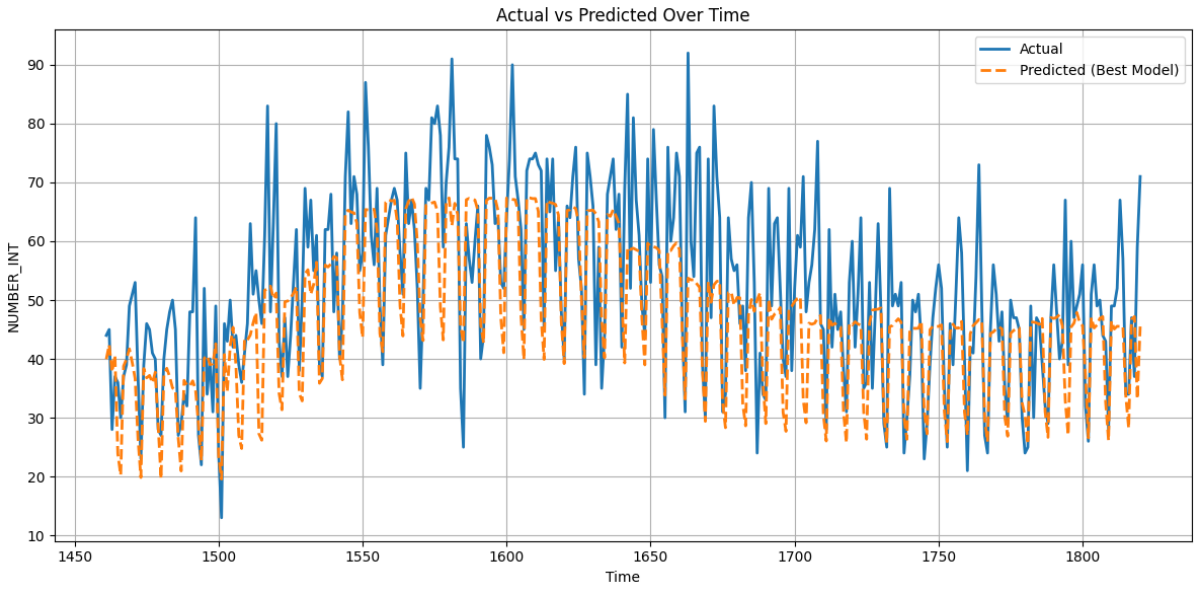


Figure B.1: RF reduced predictors - 360 day forecast vs. actual admissions.

Figure B.2 shows the plot of the importance of features for the RF model using reduced predictor sets (best results based on RMSE).

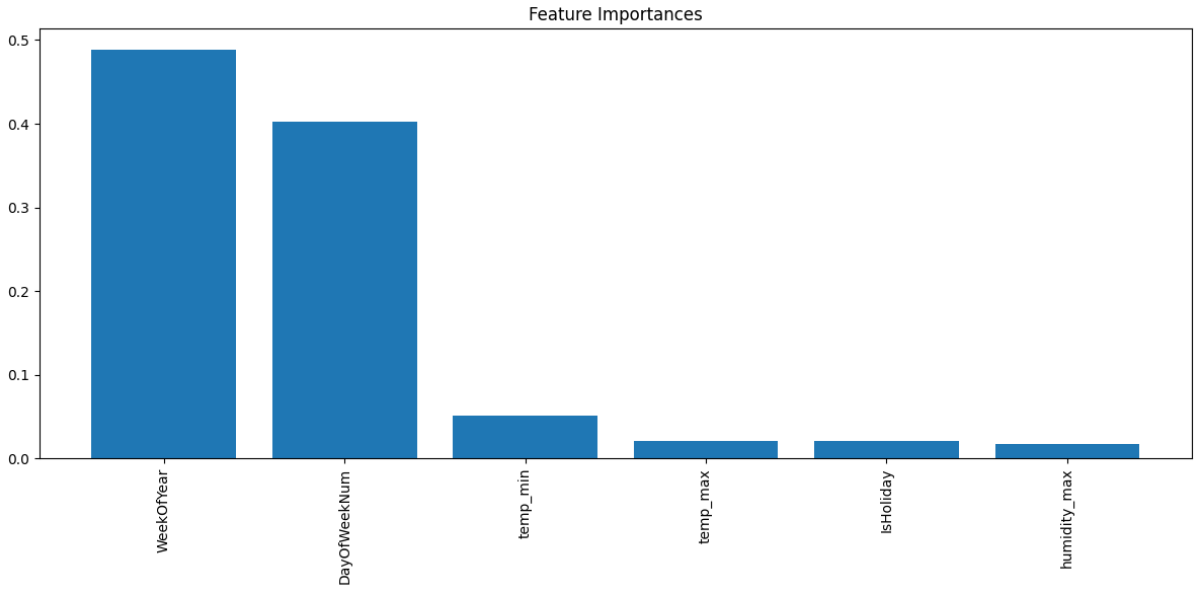


Figure B.2: RF reduced predictors - Feature importance.

Figure B.3 shows the results of the best XGB model using reduced predictor sets (that is, the iteration with the lowest RMSE in the test set) compared to actual values.

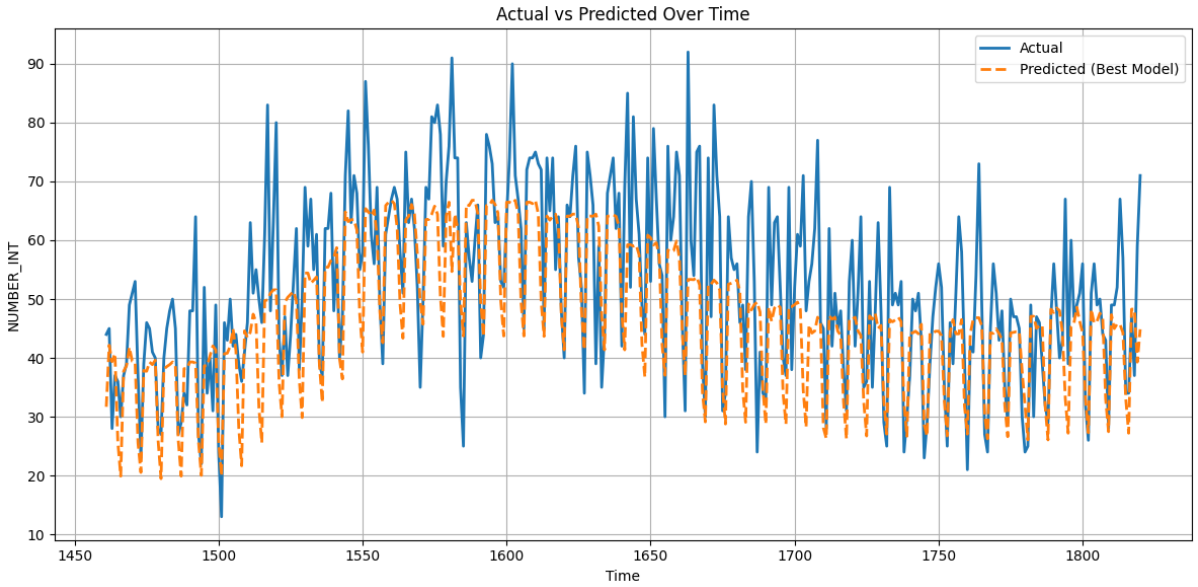


Figure B.3: XGB reduced predictors - 360 day forecast vs. actual admissions.

Figure B.4 shows the plot of the importance of features for the XGB model using reduced predictor sets (best results based on RMSE).

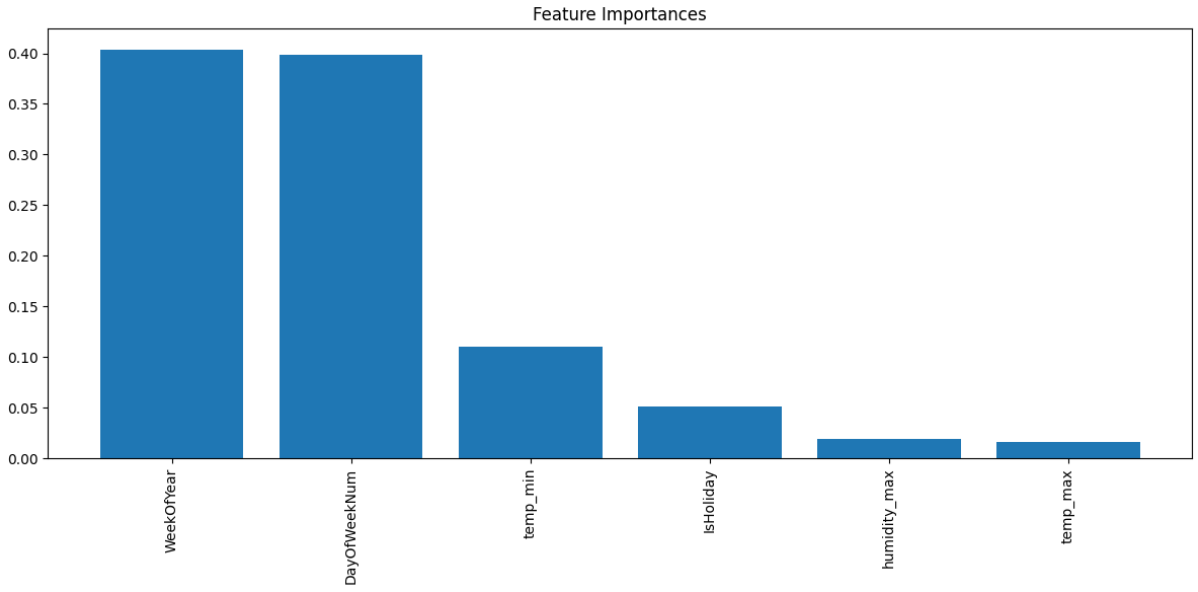


Figure B.4: XGB reduced predictors - Feature importance.

Table B.2 summarizes the performance of RF and XGB using various configurations of lagged meteorological variables (over 360 days). The baseline models (Sections 5.2.2 and 5.2.3) are evaluated alongside three extended configurations, models with lagged minimum temperature, lagged minimum temperature and maximum humidity, and lagged minimum and maximum temperatures. The table reports the mean and standard deviation for ten iterations for each configuration.

Table B.2: Test set performance metrics for 360 day forecast horizon over all models.

Model	**MAE**	**RMSE**	**MAPE (%)**	**R^2**	**Corr.**
RF (baseline)	8.21 ± 0.03	10.99 ± 0.04	15.58 ± 0.07	0.49 ± 0.00	0.78 ± 0.00
XGB (baseline)	8.25 ± 0.02	11.03 ± 0.03	15.63 ± 0.05	0.49 ± 0.00	0.78 ± 0.00
RF (tmp min)	8.21 ± 0.03	10.95 ± 0.02	15.61 ± 0.07	0.49 ± 0.00	0.78 ± 0.00
XGB (tmp min)	8.28 ± 0.00	11.05 ± 0.00	15.70 ± 0.00	0.48 ± 0.00	0.78 ± 0.00
RF (tmp min & humid max)	8.23 ± 0.04	11.01 ± 0.04	15.62 ± 0.07	0.49 ± 0.00	0.78 ± 0.00
XGB (tmp min & humid max)	8.27 ± 0.01	11.05 ± 0.01	15.67 ± 0.03	0.48 ± 0.00	0.78 ± 0.00
RF (tmp min & tmp max)	8.33 ± 0.05	11.10 ± 0.05	15.85 ± 0.12	0.48 ± 0.00	0.78 ± 0.00
XGB (tmp min & tmp max)	8.37 ± 0.01	11.13 ± 0.01	15.93 ± 0.02	0.48 ± 0.00	0.77 ± 0.00

Figure B.5 shows the results of the best RF model with lagged minimum temperature (that is, the iteration with the lowest RMSE in the test set) compared to the actual values.

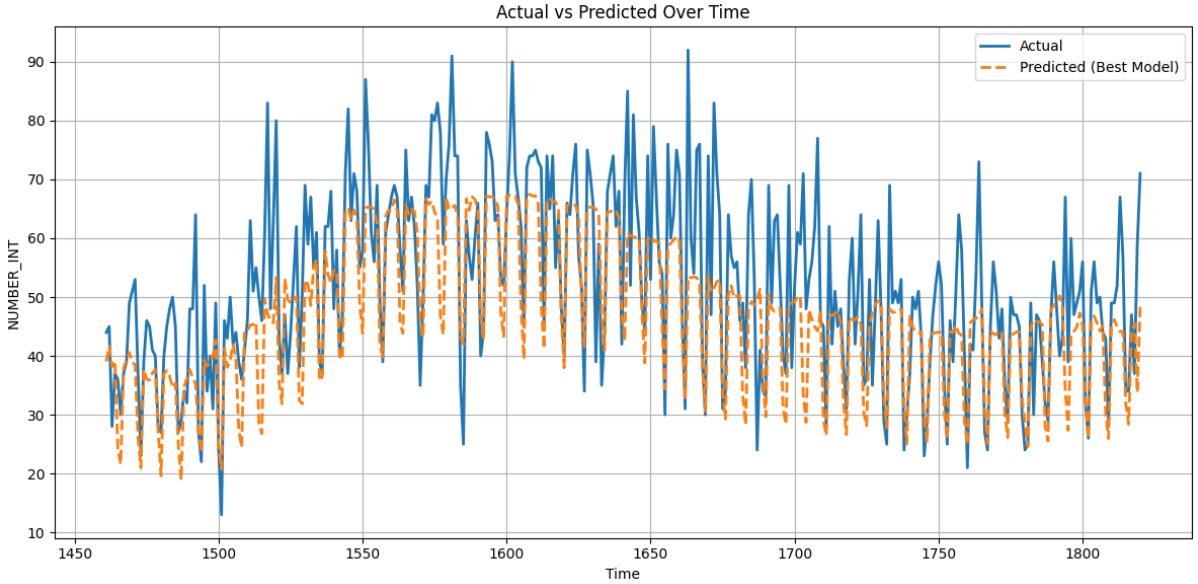


Figure B.5: RF with minimum temperature - 360 day forecast vs. actual admissions.

Figure B.6 shows the plot of the importance of features for the RF model with lagged minimum temperature (best results based on RMSE).

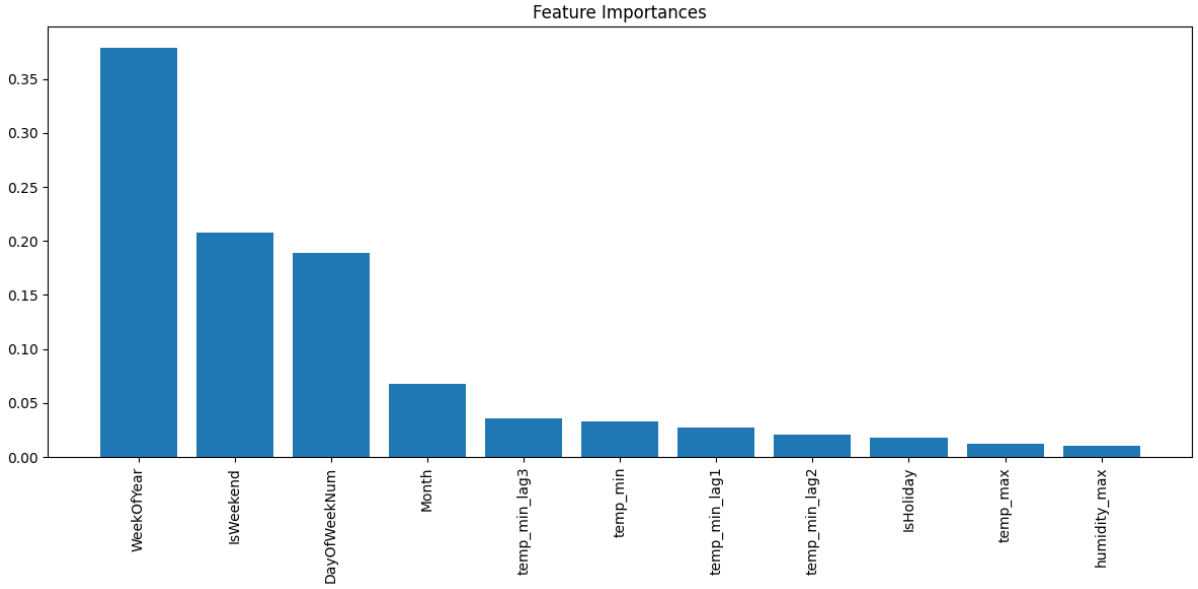


Figure B.6: RF with minimum temperature - Feature importance.

Figure B.7 shows the results of the best XGB model with lagged minimum temperature (that is, the iteration with the lowest RMSE in the test set) compared to actual values.

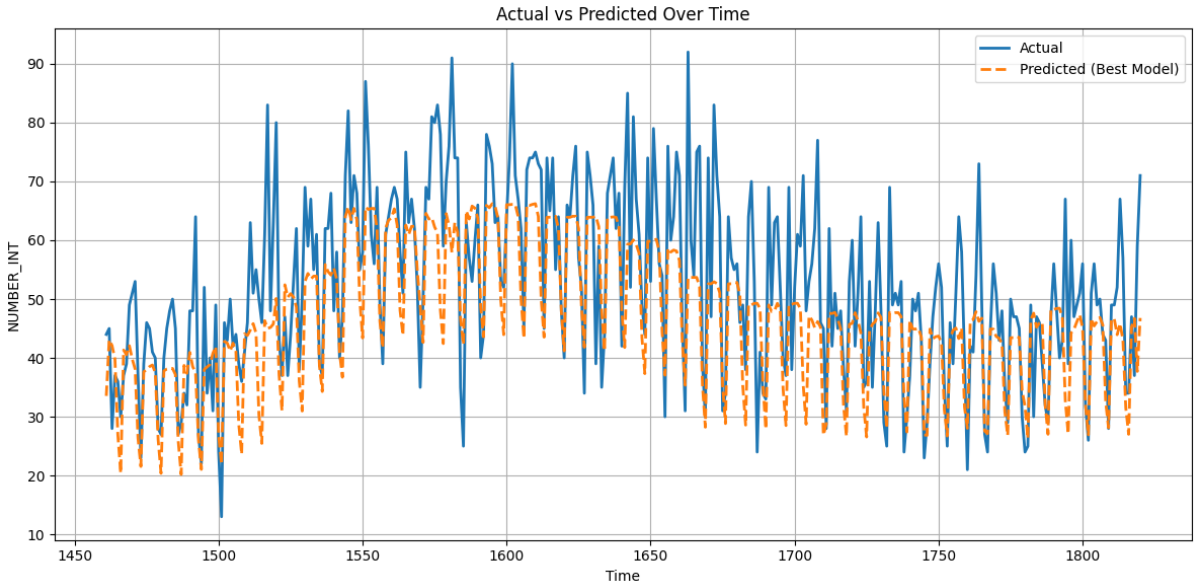


Figure B.7: XGB with minimum temperature - 360 day forecast vs. actual admissions.

Figure B.8 shows the plot of the importance of features for the XGB model with lagged minimum temperature (best results based on RMSE).

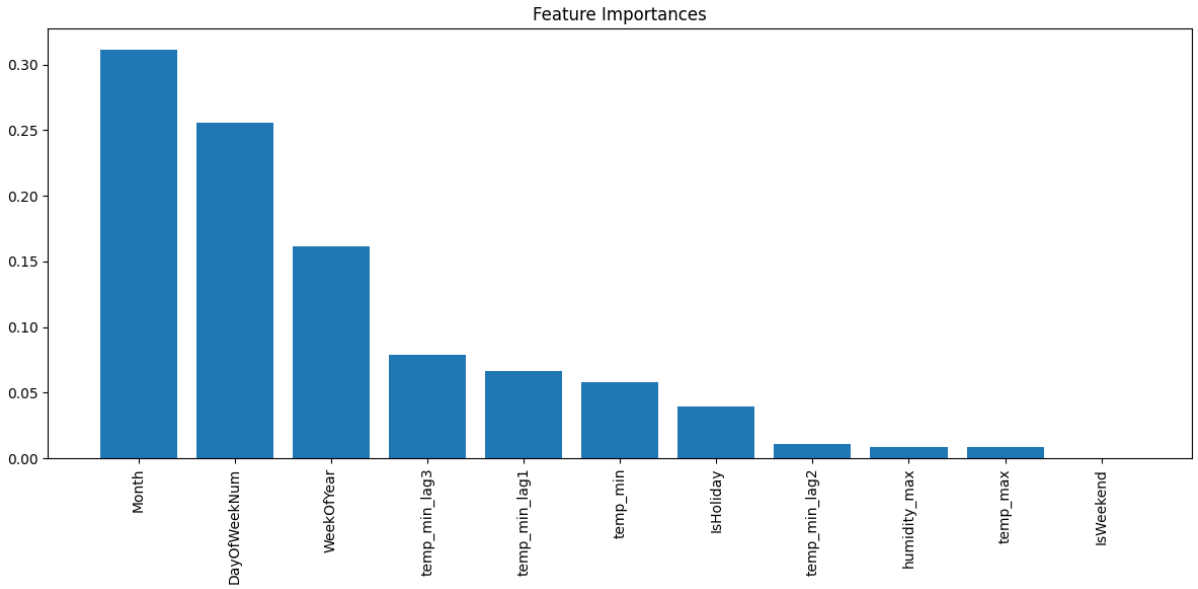


Figure B.8: XGB with minimum temperature - Feature importance.

Lagged temperature

Figure B.9 shows the results of the best RF model with lagged minimum temperature and maximum humidity compared to the actual values.

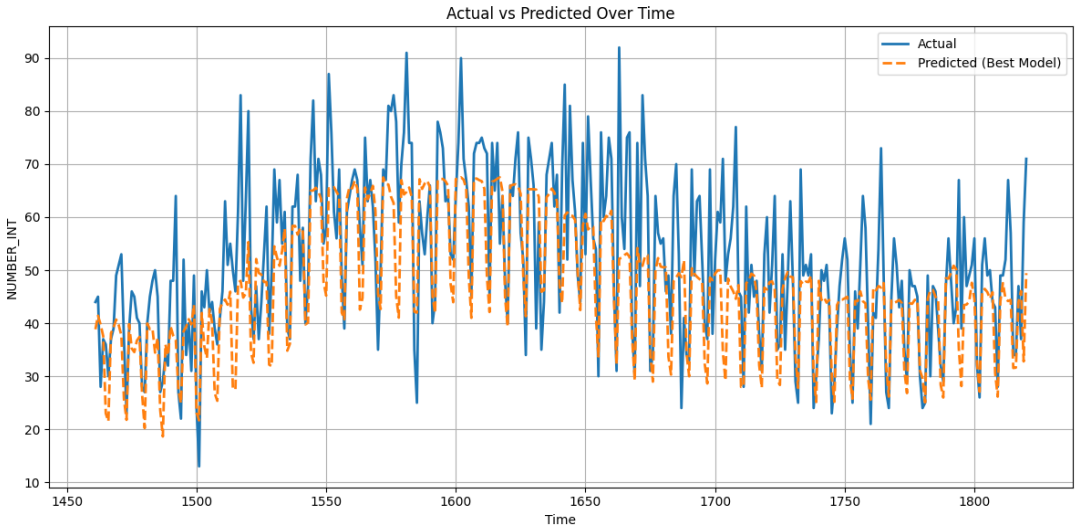


Figure B.9: RF with minimum temperature and maximum humidity - 360 day forecast vs. actual admissions.

Figure B.10 shows the plot of the importance of features for the RF model with lagged minimum temperature and maximum humidity.

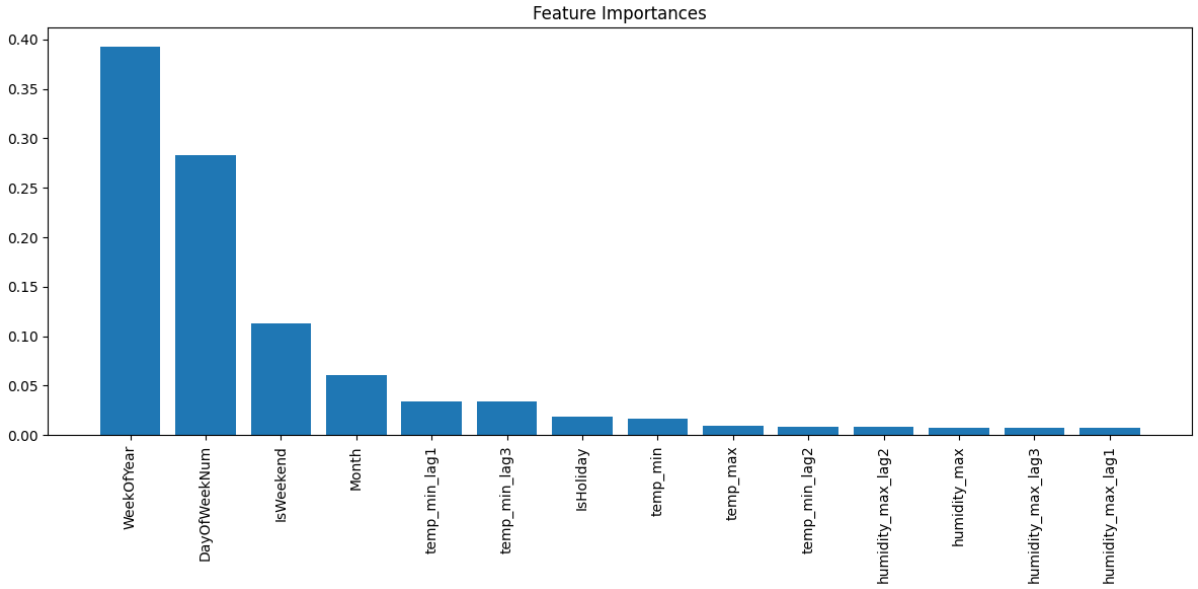


Figure B.10: RF with minimum temperature and maximum humidity - Feature importance.

Figure B.11 shows the results of the best XGB model with lagged minimum temperature and maximum humidity compared to actual values.

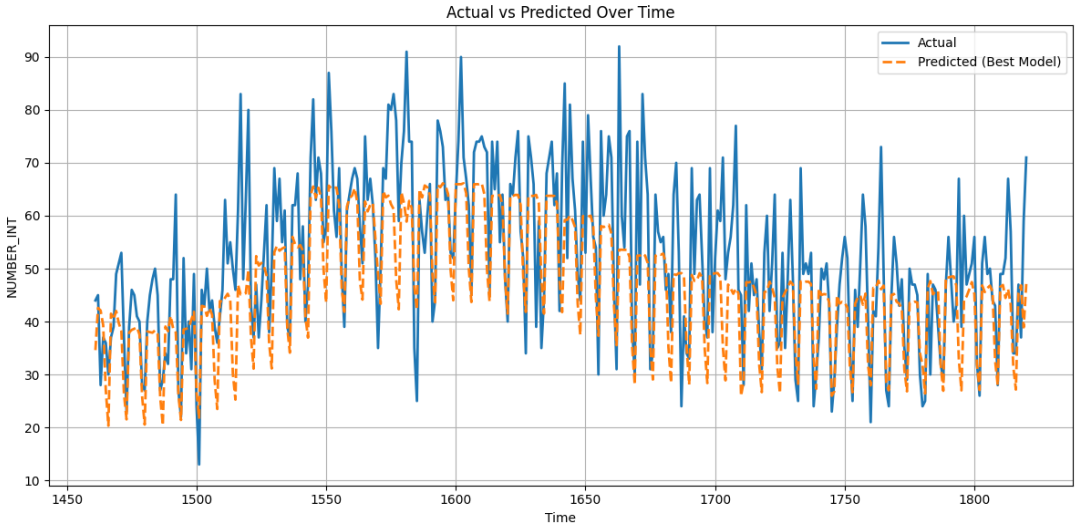


Figure B.11: XGB with with minimum temperature and maximum humidity - 360 day forecast vs. actual admissions.

Figure B.12 shows the feature importance plot for the XGB model with lagged minimum temperature and maximum humidity.

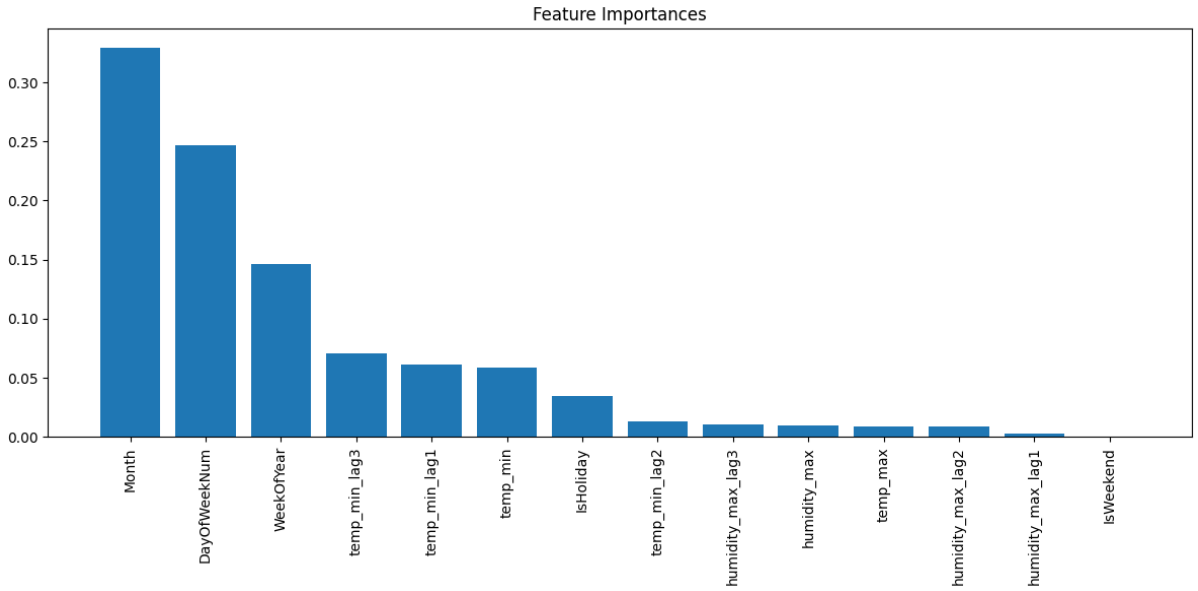


Figure B.12: XGB with minimum temperature and maximum humidity - Feature importance.

Figure B.13 shows the results of the best RF model with lagged maximum and minimum temperature compared to the actual values.

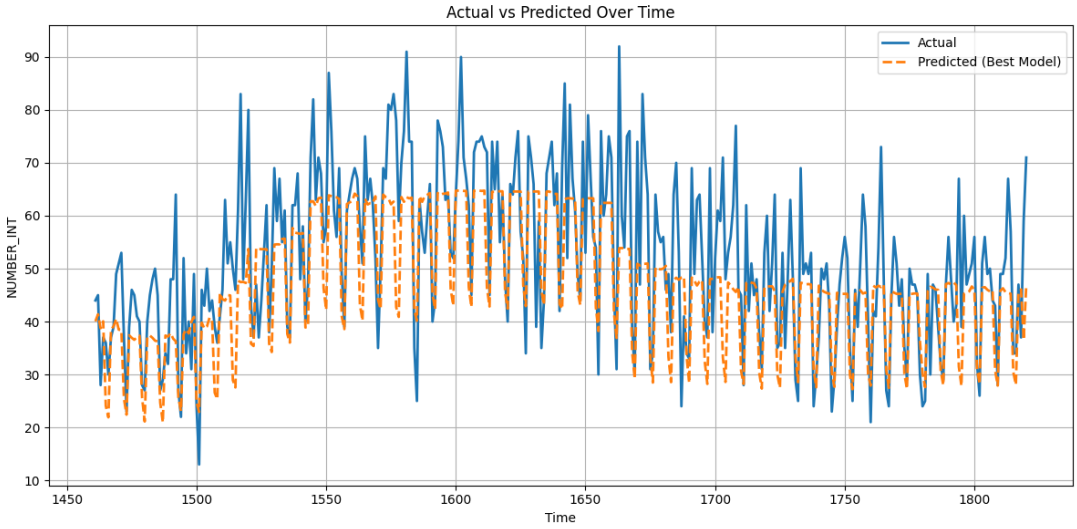


Figure B.13: RF with maximum and minimum temperature - 360 day forecast vs. actual admissions.

Figure B.14 shows the plot of the importance of features for the RF model with lagged maximum and minimum temperatures.

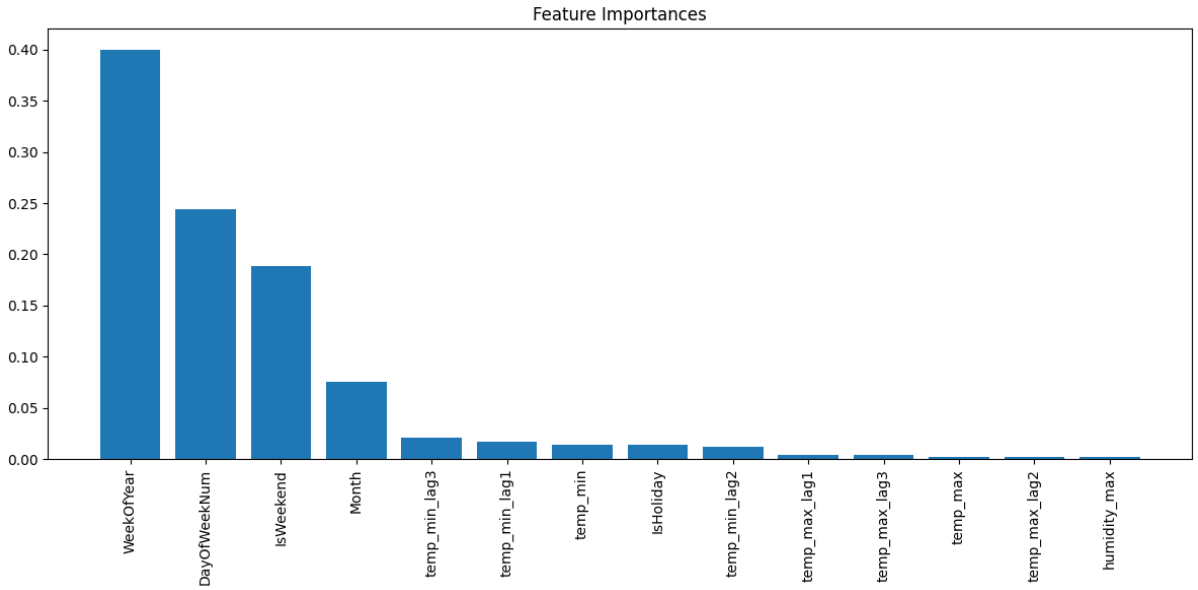


Figure B.14: RF with lagged maximum and minimum temperature - Feature importance.

Figure B.15 shows the results of the best XGB model with lagged maximum and minimum temperature compared to the actual values.

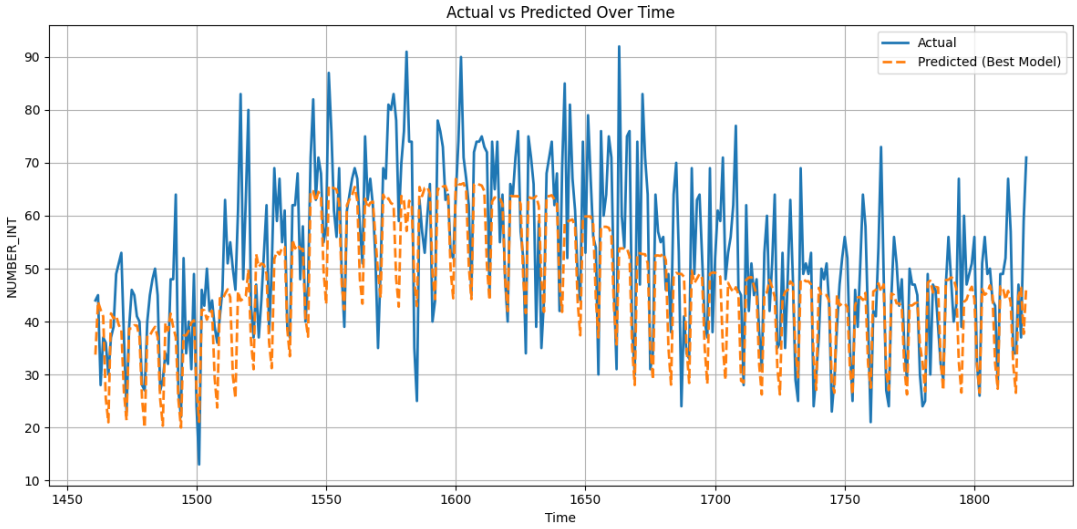


Figure B.15: XGB with maximum and minimum temperature - 360 day forecast vs. actual admissions.

Figure B.16 shows the feature importance plot for the XGB model with lagged maximum and minimum temperature.

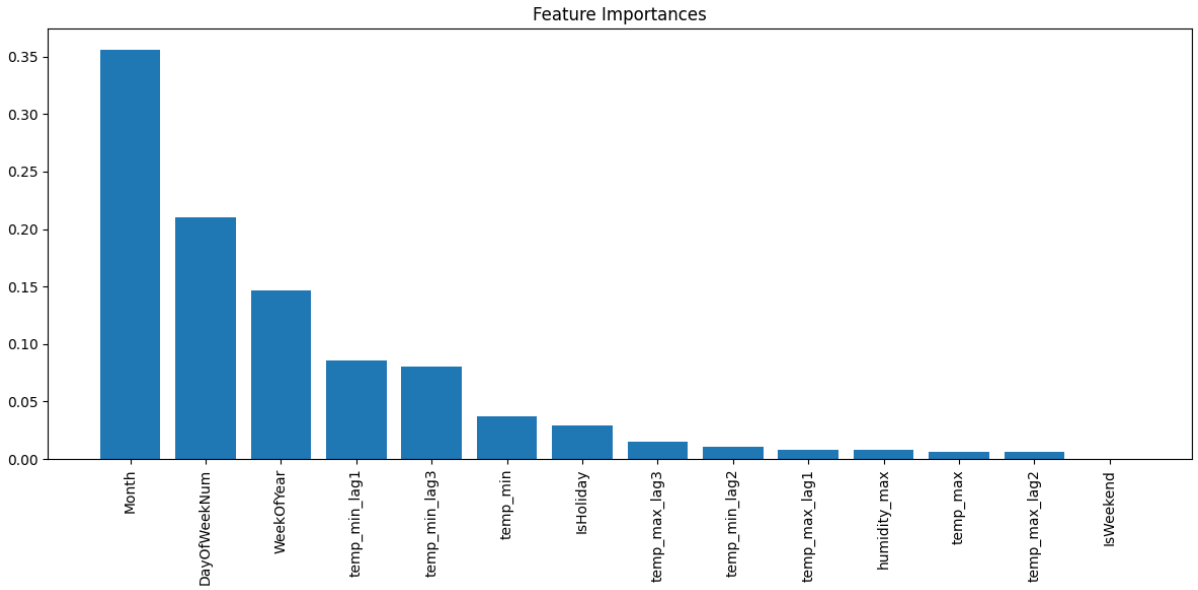


Figure B.16: XGB with lagged maximum and minimum temperature - Feature importance.

Pollutants

Table B.3 summarizes the performance of RF and XGB models trained with lagged pollutant variables (SO_2, PM_{10}, and O_3), combined with lagged minimum temperature (lags 0 to 3). Each pollutant contains missing values on different dates, and separate test sets were used for each case (the missing date was removed from the dataset). For consistency, the baseline models (Sections 5.2.2 and 5.2.3) were evaluated in the corresponding test sets for each pollutant. The table reports the mean and standard deviation for ten iterations (using the same training method described in Section 5.2.3), along with the results of the baseline model (with the best hyperparameters).

Table B.3: Test set performance metrics for RF and XGB with lagged pollutants.

Model	**MAE**	**RMSE**	**MAPE (%)**	**R^2**	**Corr.**
SO_2 - RF (baseline)	7.76	9.87	14.88	0.63	0.85
SO_2 - XGB (baseline)	7.85	10.10	14.93	0.61	0.84
SO_2 - RF (pollutant)	7.92 ± 0.10	10.24 ± 0.11	15.19 ± 0.18	0.60 ± 0.008	0.83 ± 0.002
SO_2 - XGB (pollutant)	8.26 ± 0.15	10.52 ± 0.15	15.85 ± 0.33	0.58 ± 0.01	0.83 ± 0.003
PM_{10} - RF (baseline)	8.26	11.16	15.62	0.48	0.77
PM_{10} - XGB (baseline)	8.34	11.17	15.73	0.48	0.77
PM_{10} - RF (pollutant)	8.69 ± 0.04	11.45 ± 0.05	16.50 ± 0.10	0.46 ± 0.005	0.78 ± 0.002
PM_{10} - XGB (pollutant)	8.85 ± 0.04	11.66 ± 0.07	16.69 ± 0.09	0.44 ± 0.006	0.77 ± 0.003
O_3 - RF (baseline)	9.01	11.83	15.71	0.52	0.82
O_3 - XGB (baseline)	8.97	11.62	15.64	0.53	0.82
O_3 - RF (pollutant)	9.50 ± 0.07	12.27 ± 0.10	16.78 ± 0.13	0.48 ± 0.008	0.79 ± 0.003
O_3 - XGB (pollutant)	9.13 ± 0.11	11.86 ± 0.14	15.93 ± 0.23	0.51 ± 0.01	0.81 ± 0.005

Additional Variants of the Density and Local Fit Method

Two alternative configurations of the Density and Local Fit method were evaluated. The first used the Gower distance (instead of computing Euclidean distance). The second applied the method using only categorical variables, which represented 90% of the estimated relevance in the model.

In the Gower based version, as shown in Figure B.17, the error rate in the different intervals did not follow a consistent decreasing pattern, and high reliability scores were not necessarily associated with lower errors. The corresponding scatter plot in Figure B.18 confirms that high absolute errors still occurred throughout the reliability intervals.

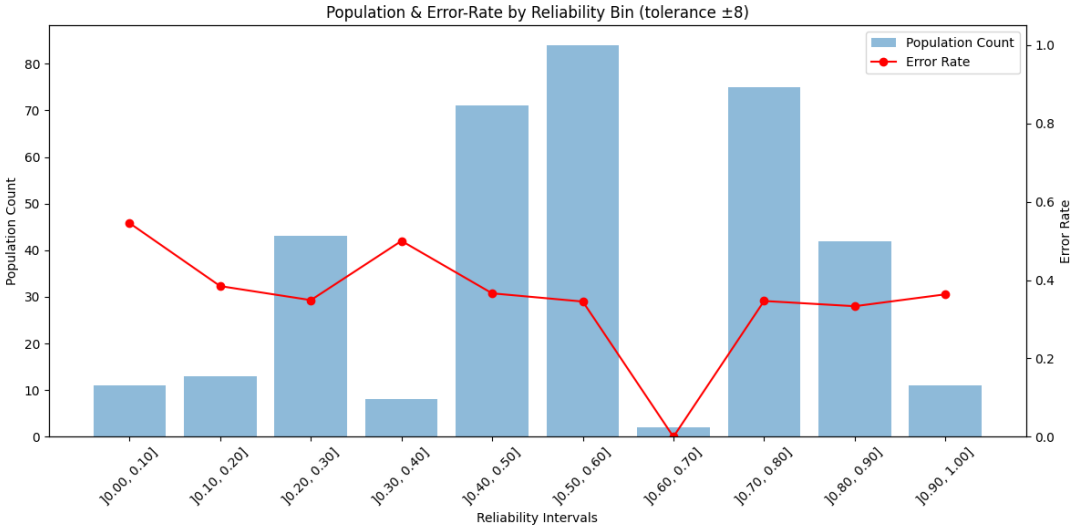


Figure B.17: Population and error rate by reliability interval (tolerance = ± 8) using the Gower distance.

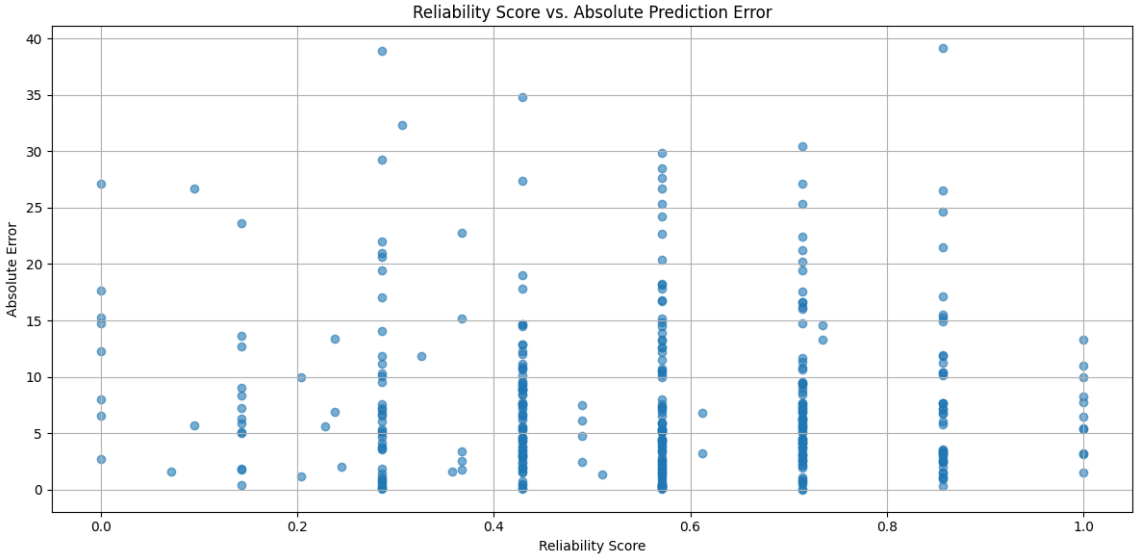


Figure B.18: Absolute error compared to reliability score using the Gower distance.

For the categorical only configuration, the reliability scores were concentrated in lower intervals and, as shown in Figure B.19, the error rate remained relatively flat.

Figure B.19: Population and error rate by reliability interval (tolerance = ± 8) using only categorical variables.

Figure B.20 shows that the absolute error distribution did not improve using only categorical variables, and the method lacked the ability to meaningfully separate more and less reliable predictions.

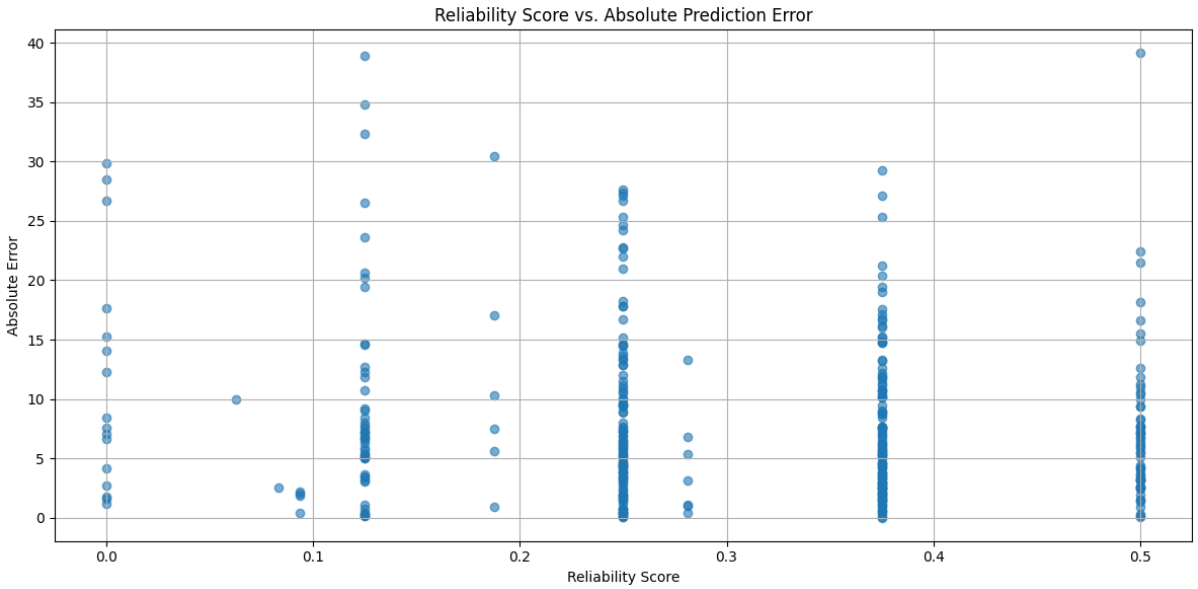


Figure B.20: Absolute error compared to reliability score using only categorical variables.

Instituto Superior de Engenharia
Politécnico de Coimbra