





Article

Approaches for Dealing with Seasonality in Clinical Prediction Models for Infections

Bernardo Cánovas-Segura ^{1,*}, Antonio Morales ^{1,†}, Jose M. Juarez ^{1,†} and Manuel Campos ^{1,2,†}¹ MedAI-Lab, University of Murcia, 30100 Murcia, Spain; morales@um.es (A.M.); jmjuarez@um.es (J.M.J.); manuelcampos@um.es (M.C.)² Murcian Bio-Health Institute (IMIB-Arrixaca), 30120 Murcia, Spain

* Correspondence: bernardocs@um.es

† These authors contributed equally to this work.

Abstract: The quantitative effect of seasonality on the prevalence of infectious diseases has been widely studied in epidemiological models. However, its influence in clinical prediction models has not been analyzed in great depth. In this work, we study the different approaches that can be employed to deal with seasonality when using white-box models related to infections, including two new proposals based on sliding windows and ensembles. We additionally consider the effects of class imbalance and high dimensionality, as they are common problems that must be confronted when building clinical prediction models. These approaches were tested with four datasets: two created synthetically and two extracted from the MIMIC-III database. Our proposed methods obtained the best results in the majority of the experiments, although traditional approaches attained good results in certain cases. On the whole, our results corroborate the theory that clinical prediction models for infections can be improved by considering the effect of seasonality, although the techniques employed to obtain the best results are highly dependent on both the dataset and the modeling technique considered.

Keywords: seasonality; concept drift; clinical prediction models; high dimensionality; class imbalance; infectious diseases



Citation: Cánovas-Segura, B.; Morales, A.; Juarez, J.M.; Campos, M. Approaches for Dealing with Seasonality in Clinical Prediction Models for Infections. *Appl. Sci.* **2023**, *13*, 8317. <https://doi.org/10.3390/app13148317>

Academic Editors: Antonio Fernández-Caballero, Enno van der Velde and Giorgio Leonardi

Received: 26 May 2023

Revised: 27 June 2023

Accepted: 14 July 2023

Published: 18 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Seasonal variations in the prevalence of infectious diseases, which are commonly known as seasonality, have been widely documented and studied [1–4]. Although these variations are usually attributed to seasonal changes in humidity, temperature, or even different human behaviors throughout the year, the detailed causes of seasonality remain a recurrent research topic [1,2,5,6].

Seasonality is usually considered in epidemiological time series studies, in which the objectives are to estimate the number of future cases or assess the factors correlated with the spread of infections [2,7]. However, seasonal variations are rarely considered in clinical prediction models, which have objectives related to prognostic and health service research [8].

In this paper, we explore the main techniques used to address the challenge of seasonality in prediction models for infectious diseases. We focus on classification problems, and use the approach of simply ignoring season in the models as a gold standard. Two common approaches used to deal with seasonality [2,7–13], namely, adding the season as an additional feature and generating different models for different season, are considered in this work. Furthermore, we propose two algorithms based on common methods from the field of datastream mining research, namely, sliding windows and ensembles of models trained on different time periods. The effects of these approaches are studied with regard to both synthetic datasets and data related to infectious diseases extracted from the MIMIC-

III database [14]. This work greatly extends our preliminary proposal presented in [15]. The main contributions of this paper are:

- New approaches for dealing with seasonality based on sliding windows and ensembles;
- An extensive study of the effects of seasonality on clinical prediction models in the presence of high dimensionality and imbalanced data;
- Experimental settings based on freely available interpretable techniques and open data. With the aim of ensuring the reproducibility and usability of these results in future research, we have made the code developed for this work freely available at: <https://github.com/berncase/seasonality-rProject> (accessed on 25 May 2023).

The remainder of this paper is structured as follows. Section 2 presents a comprehensive analysis of the issue of seasonality in clinical data and discusses relevant research in the field. In Section 3, we explain the approaches employed to deal with seasonality that are considered in this work, including the two new proposed algorithms. In addition, we provide a detailed description of the two synthetic datasets with seasonal variations considered in this work along with the two clinical datasets extracted from the MIMIC-III database. Section 4 provides insights into the conducted experiments and their results, which are further discussed in Section 5. In Section 6, we outline the limitations of this work and highlight future research directions. Finally, Section 7 presents the conclusions drawn from this study.

2. Related Work

It is widely accepted that seasonal variations are a common trait of many infectious diseases [1]. As stated in [8], several methods are employed in epidemiological studies to examine the effect of seasonality, including the statistical comparison of two different time periods, geometrical models assuming sinusoidal cyclic patterns, and generalized linear models in which seasonality is included as an extra term. These approaches make it possible to assess the effect of seasonality on the number of cases and identify factors that might be associated with these variations [2,7].

When the outcome of the model is to predict a condition regarding a particular patient, a common strategy is to include the season among the possible features to be explored. For example, the season can be included as an additional feature with four possible values (spring, summer, autumn, and winter) in northern and southern countries [9] or two values (dry and wet) in models for countries with subtropical or tropical climates [10]. Another strategy is to build separate models for each season [11], or at least different models for summer and winter [12,13].

From a different perspective, we can consider seasonality in clinical data as a particular case of concept drift in a datastream. Datastream mining is a recent research field that is focused on the development of models over huge amounts of online data obtained from sources such as sensors, bank transactions, or social networks [16,17]. When dealing with datastreams, certain particularities must be considered. For example, it is assumed that the whole stream can neither be stored in memory nor accessed repeatedly, signifying that the algorithms working with it can manage only a limited number of data items at the same time or even that the model must be trained by observing each item only once [18]. Another common assumption is that the datastream is non-stationary, which means that the distribution of the features and/or the target outcomes varies over time [16]. This aspect of data is commonly known as *concept drift* [19–21], and is an open and fast-growing research topic [22]. This effect often occurs in clinical research, in which the study of clinical models capable of evolving over time, known as dynamic models, is a recent and active topic [23]. A more comprehensive study of the state of the art in these research lines can be found in [16,22,23].

However, problems arise when attempting to apply these approaches to open clinical data in order to validate and share results. First, de-identification policies force the removal of most of the elements from real timestamps when sharing clinical datasets in order to ensure patient privacy [24]. In the case of the MIMIC-III Database [14], which is

the public data source used in this work, the date of each patient's admission is randomized, although both the day of the week and the season shown in the original source are maintained. Consequently, approaches that rely on strict temporal ordering of the data cannot be readily applied in such cases. Another challenge is that many of these algorithms are not designed to work with a relatively small number of samples, which is common when working with certain diseases. Furthermore, the need for interpretable models and the effects of high dimensionality and class imbalance are not usually considered in the design and validation of these frameworks.

In this paper, we propose and evaluate two approaches based on sound strategies from datastream mining that we have adapted to the particular problem of seasonality in open clinical data, as described in the following sections.

3. Materials and Methods

In this section, we describe the data mining techniques and the datastream mining frameworks considered in this work, along with the modifications proposed to deal with the problem of seasonality.

3.1. White-Box Models

The aim of a clinical model is to provide support when making clinical decisions, not to replace the experts who make them. Therefore, the interpretability of the model and the ability to understand the rationale behind its predictions are crucial for its acceptance, even if this results in a slight decrease in its accuracy. Models that are easily understood and applied by users are commonly referred to as *white-box models* or *interpretable models*. Examples of such models include logistic regressions and decision trees, which are widely used in clinical settings. On the other hand, certain Artificial Intelligence approaches, such as deep learning and bagging (when using a large number of members [25,26]) generate complex models, often referred as *black-box models*, which are more difficult to interpret [27]. These models have historically been less well accepted despite producing very accurate predictions [28].

In this work, we experiment with both logistic regression and decision trees. Logistic regression is one of the most common techniques employed to build clinical prediction models [29]. These models have the following structure:

$$\log\left(\frac{\mu}{1-\mu}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

where $\mu = P(Y = 1)$, in which Y is an indicator variable denoting the occurrence of the event of interest and $x_1 \cdots x_p$ are the features from the dataset used in the model. The β_i components are usually estimated by means of maximum likelihood, although they are sometimes modified in order to avoid overfitting [30].

Decision tree models consist of a series of nested conditions that partition the feature space into homogeneous groups that can be assumed to be of a particular class with an acceptable margin or error [31]. These models can be represented as a set of *if-then* rules, or more traditionally as an inverted-tree graph in which the split conditions are the nodes and the predicted classes are the leaves. As a result, these models are highly interpretable [31].

3.2. High Dimensionality

A common problem related to clinical datasets is that of having a high number of possible features and a modest number of observations. This may make the training process difficult and lead to less accurate models.

This effect is usually alleviated by using filters. One common approach consists of performing univariate analyses between each candidate predictor and the outcome class [32]. In these cases, experts recommend the use of the chi-squared test or Fisher's exact test for nominal variables and a univariate logistic regression model or two-sample Student's *t*-test for continuous variables. Those predictors that have a low *p* value, commonly $p < 0.05$, are then considered interesting for the training process.

Other more complex filters are available as well; for example, the Fast Correlation-Based Filter (FCBF) [33] estimates the relevance of each feature with regard to the target outcome along with whether it is redundant for any other relevant feature. Only those features considered highly relevant and non-redundant are used to train the model.

Furthermore, it is a common practice to use the Least Absolute Shrinkage and Selection Operator (LASSO) approach when working with logistic regression. LASSO seeks a balance between model complexity and accuracy by imposing a constraint on the sum of the absolute size of the regression coefficients [34]. This allows certain coefficients to reach zero during the search for the optimal solution, allowing them to be removed from the model. This results in less overfitted models.

The modern algorithms used to create decision trees, such as C5.0, include the option of *winnowing*, or removing the predictors that are considered to be unimportant before creating the model [31].

In this work, we used LASSO to carry out experiments with logistic regression and activate the winnowing option for decision trees. We additionally tested the use of only those features with $p < 0.05$ after a Fisher test or *t*-test and the use of FCBF to discard highly redundant features for both logistic regression and decision tree models.

3.3. Class Imbalance

Another common problem when developing clinical prediction models is that the datasets are skewed towards one of the values of the target outcome. For example, it is typical for fewer patients to be infected by a bacterial species than to not be infected or be infected by a different one. In these situations, models tend to be biased towards the most frequent value, and minority cases, which usually have high relevance, may be ignored [35].

There is a wealth of approaches for dealing with the problem of class imbalance. In this work, we focus on those that have little or no impact on the interpretability of the resulting models. In particular, we experiment with undersampling and oversampling. When using undersampling, all the observations of the minority class are considered for the training dataset, while only a random sample of the majority class is used until a selected ratio of minority class is attained. On the contrary, in the oversampling strategy all the observations of the majority class are considered and the samples from the minority class are randomly repeated until the selected ratio with respect to the majority class is attained.

Undersampling and oversampling do not alter the values of the data samples, nor do they generate synthetic samples; consequently, they have no impact on model interpretability, which is why they are used in this work rather than other more complex approaches.

3.4. Proposed Adaptations to Deal with Seasonality

In this work, we follow a datastream mining approach; therefore, we consider seasonality as a particular case of concept drift. Furthermore, we assume the following:

- Our observations do not follow a strict temporal order (owing to, e.g., de-identification processes); therefore, we cannot apply common techniques used to address concept drift.
- It is possible to determine or estimate the month in which each observation was made; additionally, we consider the case in which only the season of the observation is known.
- We have sufficient observations with which to build models based on data from a limited number of months, or seasons, if we cannot attain that level of detail.

These assumptions are utilized in order to adapt several well known datastream mining frameworks, as described below.

3.4.1. Sliding Windows

One of the earliest techniques proposed to address concept drift was the use of sliding windows [21]. The underlying hypothesis of this approach is that the newest datapoints are more useful than the older ones when attempting to predict the target outcome to the

point that the oldest ones can be discarded when training a prediction model. Therefore, only those data points within a determined time interval (i.e., the time window) are used to build the prediction model. This framework has been the starting point for many other methods, and is commonly used as a baseline in the evaluation of new algorithms [36].

However, the traditional sliding window approach and its subsequent improvements require an ordered datastream. If it is not possible to estimate the real timestamp of the data, or if they are not ordered by occurrence, then these approaches cannot be applied.

We propose an adaptation of the sliding window approach focused on dealing with the problem of seasonality in those cases in which only the month or the season of the data are known. The proposed method is formally described in the functions *WindowTraining* (Algorithm 1) and *WindowPredict* (Algorithm 2). Let us assume that it is necessary to make a prediction for a datapoint d_m belonging month m and that in our training dataset we know the month in which each observation was made. We first partition our training dataset by the month of its observations, after which we create a window of a predefined size w around the data obtained in m . This window contains data from w months; therefore, it includes the training data for month m , the previous $\frac{w-1}{2}$ months, and the subsequent $\frac{w-1}{2}$ months, where w is assumed to be an odd number.

Algorithm 1 *WindowTraining* (monthly/seasonal)

Input: w : Size of the window with $w \in \{1, 3, 5, 7, 9, 11\}$ in the monthly version, or $w = 3$ in the seasonal version

Input: \mathcal{D} : Training dataset

Output: \mathcal{K} : A set with a model for each month/season

- 1: Aggregate data in \mathcal{D} by month/season
 - 2: **for each** month/season m **do**
 - 3: $D_m \leftarrow$ data of w months/seasons from \mathcal{D} , gathering data from month/season $m - \frac{w-1}{2}$ to month/season $m + \frac{w-1}{2}$
 - 4: $K_m \leftarrow$ model trained using D_m
 - 5: Include K_m in \mathcal{K}
 - 6: **return** \mathcal{K}
-

Algorithm 2 *WindowPredict* (monthly/seasonal)

Input: d_m : Observation in month/season m for prediction

Input: \mathcal{K} : Set of trained models, one for each month/season

Output: p_{d_m} : Prediction of the observation d_m

- 1: $m \leftarrow$ extract month/season from d_m
 - 2: Select $K_m \in \mathcal{K}$ trained by using data from the window centred on m
 - 3: $p_{d_m} \leftarrow$ prediction of K_m for observation d_m
 - 4: **return** p_{d_m}
-

We illustrate our proposal with the example depicted in Figure 1a,b. In our example, we assume a window of three months, i.e., $w = 3$. First, we centre the sliding window on January (Figure 1a). Our first training dataset is composed of datapoints from December, January, and February, and the model trained with it will be used to predict observations from January. By sliding the window, we can create up to twelve different models (one per month). Figure 1b shows an example of a prediction for a datapoint d_m belonging to February. According to the month of the observation to be predicted, the particular model trained when the window was centered on that month (February, in this example) is used to make the prediction.

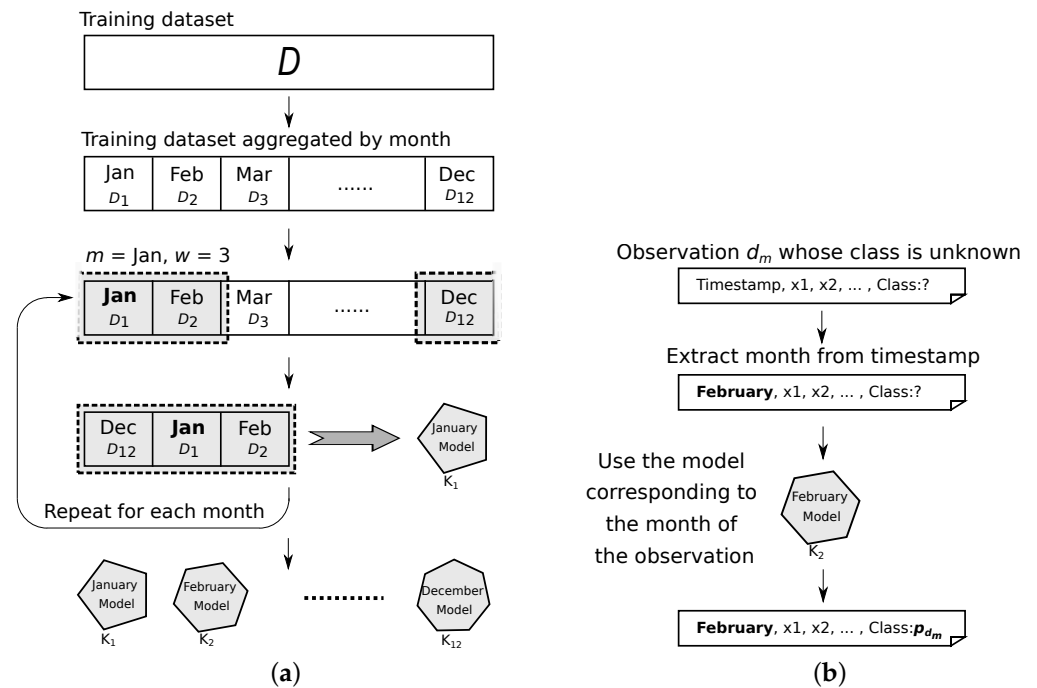


Figure 1. Examples of our proposed adaptation of the sliding window algorithm for seasonality; (a) shows the steps to train the models assuming a window of three months, while (b) shows the steps for estimating the class to which a new observation belongs.

3.4.2. Ensembles

The ensemble approach consists of training models with data from different time intervals, then combining their outputs when predicting a new observation [16,37,38]. In this case, models created with older data can be restored and used again when the correlations between the outcome class and the features return to a previous state in the stream (i.e., recurring concepts). The literature contains many variations of this framework depending on the strategy used to maintain older models, update them, or merge their results into the final output [16].

However, this generic framework and its subsequent refinements require a dataset with a strict order between its observations; therefore, it cannot be applied to de-identified clinical data.

We propose an adaptation of the ensemble framework to deal with the problem of seasonality in open clinical data. The method is formalized by two functions, *EnsembleTraining* (Algorithm 3) and *EnsemblePredict* (Algorithm 4). Let \mathcal{D} be the dataset available for training, where D_i corresponds to the data for the i -th month and where $i = \{1, \dots, 12\}$. We build a prediction model for each month using the data regarding that specific month (*EnsembleTraining*, lines 1–6). The model trained using D_i is denoted as K_i .

It is then necessary to calculate the weights used to combine the outputs of the models in the ensemble (*EnsembleTraining*, lines 7–12). This is done by again iterating over the training dataset, and each model K_i is tested on the training data of every month D_j , where $j = \{1, \dots, 12\}$. The Root Mean Squared Error (RMSE) is then estimated for each pair (K_i, D_j) .

Algorithm 3 *EnsembleTraining* (monthly/seasonal)

Input: n : number of models of the ensemble. $n = 12$ for the monthly ensemble, $n = 4$ for the seasonal ensemble

Input: \mathcal{D} : Training dataset

Output: \mathcal{K} : set of models, one for each month/season

Output: W : $n \times n$ weights matrix.

```

1: Aggregate data in  $\mathcal{D}$  by month/season
2:  $R \leftarrow$  Initialize an empty  $n \times n$  matrix to store root-mean squared errors from models
3: for each month/season  $i$  do
4:    $D_i \leftarrow$  subset of  $\mathcal{D}$  gathering data from month  $i$ 
5:    $K_i \leftarrow$  model trained using  $D_i$ 
6:   Include  $K_i$  in  $\mathcal{K}$ 
7:   for each month/season  $j$  do
8:      $D_j \leftarrow$  subset of  $\mathcal{D}$  including data from month  $j$ 
9:      $R[i, j] \leftarrow$  Root mean squared error obtained when applying  $K_i$  to  $D_j$   $\triangleright$  In case
        $R[i, j] = 0$ , we assume  $R[i, j] = 10^{-9}$ 
10:  for each month/season  $i$  do
11:    for each month/season  $j$  do
12:       $W[i, j] \leftarrow \frac{1}{\sum_{i=1}^n \frac{1}{R[i, j]}}$ 
13: return  $\mathcal{K}, W$ 

```

Algorithm 4 *EnsemblePredict* (monthly/seasonal)

Input: d_m : Observation in month/season m for prediction

Input: \mathcal{K} : Set of trained models, one for each month

Input: W : Weight matrix

Input: n : number of models of the ensemble. $n = 12$ for the monthly ensemble, $n = 4$ for the seasonal ensemble

Output: p_{d_m} : Prediction of the observation d_m

```

1:  $m \leftarrow$  extract month/season from  $d_m$ 
2:  $p_{d_m} \leftarrow \sum_{i=1}^n W[i, m] \text{Prediction}(K_i, d_m)$ 
3: return  $p_{d_m}$ 

```

It is worth mentioning that (1) the lowest error is usually obtained when $i = j$, and the model K_i is tested using the same data employed to build it (i.e., D_i); and (2) the highest error usually occurs when it is tested with data in which the seasonal effect has the greatest effect on the correlations when compared to D_i , say, from month z . Therefore, it is necessary to calculate the weights such that, when a new observation from month i is predicted, the output from model K_i will have the highest weight, with the contrary being the when the observation is from month z . This is done as follows: after the RMSE for all the pairs (K_i, D_j) has been calculated, we calculate a weight matrix W using the weight of each model i when used to predict data from month j ($W[i, j]$):

$$W[i, j] = \frac{1}{\sum_{j=1}^{12} \frac{1}{R[i, j]}} \quad (1)$$

where $R[i, j]$ is the RMSE estimated for the model K_i when applied to the training data from month j . In the case of $R[i, j] = 0$ for any combination of i and j , we assume that $R[i, j] = 10^{-9}$.

A table with the weights is stored along with the ensemble. When it is necessary to predict the outcome of a new observation d_m (*EnsemblePredict* function), the final prediction provided by the ensemble is

$$p_{d_m} \leftarrow \sum_{i=1}^n W[i, m] \text{Prediction}(K_i, d_m), \quad (2)$$

where $\text{Prediction}(K_i, d_m)$ is the prediction of model K_i for the observation d_m and m is the month to which d_m belongs. If only the season of each observation is known, then the ensemble is composed of four models (one per season), while the rest of the algorithm is similar.

Figure 2a shows a graphical explanation of the steps of the *EnsembleTraining* function when training a monthly ensemble. Figure 2b shows an example of execution of the *EnsemblePredict* function when used to predict an observation belonging to February.

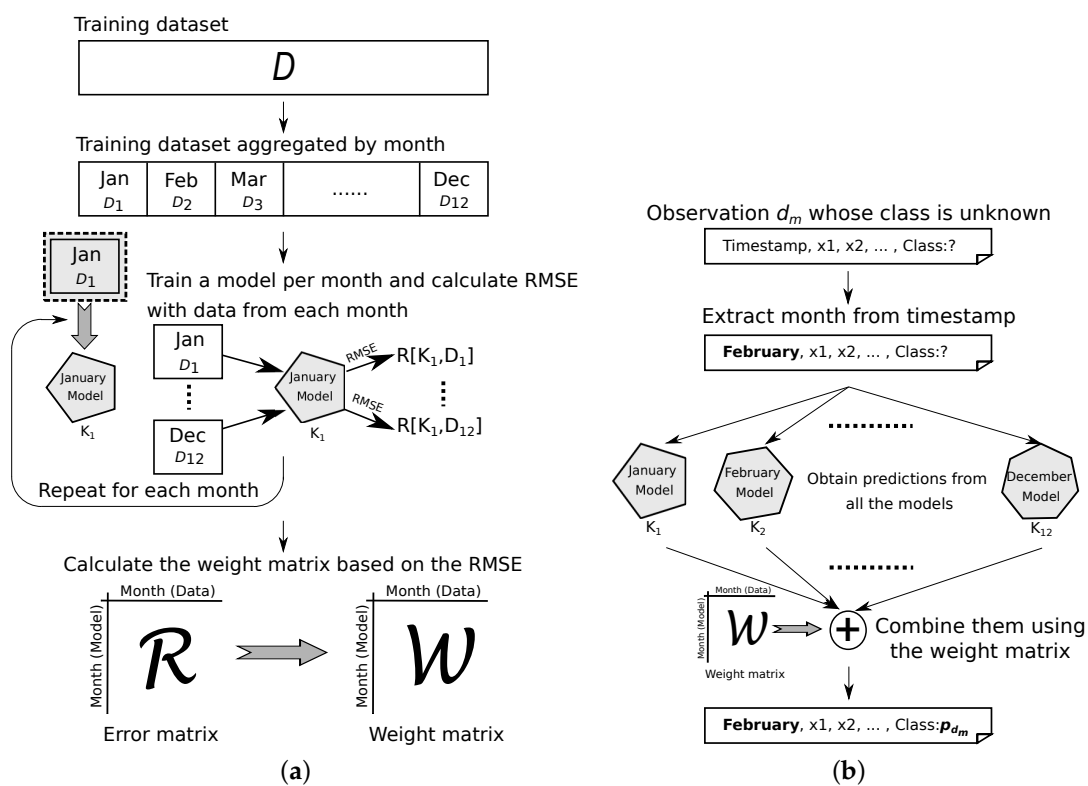


Figure 2. Examples of our proposed adaptation of the ensemble algorithm for seasonality: (a) shows the steps for training the models in a monthly ensemble, while (b) shows the steps for estimating the class to which a new observation belongs.

3.5. Dataset Description

We considered two synthetic datasets and two real-world datasets extracted from clinical open data, specifically from the MIMIC-III database [14].

3.5.1. Synthetic Datasets

We created two synthetic datasets in order to simulate two different seasonal variations. Let us assume the model defined by the equation $k_1x_1 + k_2x_2 = y$, where x_1 and x_2 are random variables within the range $[-10, 10]$. The values of k_1 and k_2 represent the unknown factors of the model that need to be estimated. Because our focus is on binary classification models, we introduce a categorical column named *class* as the binary outcome. This column takes two possible values: *non-negative* when $y \geq 0$, and *negative* otherwise.

Our aim was to simulate a concept drift during winter. This was achieved by varying the value of k_1 between 0 and 1 throughout the year and calculating the value of k_2 as $k_2 = 1 - k_1$. If $k_1 = 1$, then $k_2 = 0$, and the outcome y depends only on the value of x_1 . The contrary occurs when k_1 reaches 0. Consequently, each observation includes a *timestamp* attribute that is used to vary the values of these factors according to the season. here, this attribute ranges from the 1 January 2100 to 31 December 2199, similar to the dates in the MIMIC-III database, although our methods consider only the month or season of each particular date.

Furthermore, we added extra variables that are not directly related to the model in order to simulate the problem of high dimensionality. We included ten random variables, r_1 to r_{10} , that have a uniform distribution with values in $[-10, 10]$. In addition, we included another ten variables $c_j^1, j \in \{1, \dots, 10\}$ correlated with x_1 and ten more variables $c_j^2, j \in \{1, \dots, 10\}$ correlated with x_2 . These variables were calculated as follows:

$$c_j^i = x_i + \epsilon, \epsilon \sim U(-2, 2). \quad (3)$$

As such, these c_j^i variables have values similar to those that really affect the outcome of the model, with an additional small random error such that they are not perfectly correlated.

The synthetic datasets eventually contain a total of 33 columns: one nominal column that indicates the class of the observation (*non-negative* or *negative*), two columns (x_1 and x_2) that really affect the class, ten absolutely random columns (r_1, \dots, r_{10}), and twenty columns ($c_1^1, \dots, c_{10}^1, c_1^2, \dots, c_{10}^2$) correlated with x_1 or x_2 .

A class imbalance of ten to one was then simulated in order to increase the complexity of the dataset. After each dataset had been generated, we assumed that the *non-negative* class was the minority one and randomly removed samples until there were ten *negative* samples for each *non-negative* sample. A total of 5500 rows were generated per dataset (5000 *negative* samples and 500 *non-negative* samples).

We made two different assumptions about how seasonality affects data, which were simulated by varying k_1 and k_2 over time:

- In the *condensed dataset*, we assumed that x_1 does not affect y (i.e., $k_1 = 0$ and $k_2 = 1$) except in winter, when y becomes gradually affected by x_1 following a Gaussian curve with its maximum centered in the middle of the season (i.e., $k_1 = 1$ and $k_2 = 0$ exactly in the middle of winter). In this case, the effects of seasonality are strictly present only during winter.
- In the *sinusoidal dataset*, we assumed that k_1 varies following a sinusoidal function that reaches its maximum in the middle of winter and that its effects decrease slightly until reaching its minimum in the middle of summer. The use of sine curves to represent seasonal variation is quite common in epidemiological studies regarding the seasonal occurrence of infectious diseases [8]. In this case, the effects of seasonality are present throughout the year, with the main differences being between winter and summer.

Figure 3 shows a sample of the values of x_1 and x_2 along with the outcome class and the changes in k_1 and k_2 for the *condensed* and *sinusoidal* datasets. Furthermore, Figure 4 shows a graph concerning the correlation between the numeric attributes of both datasets. As shown in these plots, c_1^1, \dots, c_{10}^1 features are highly correlated with x_1 ; the same occurs with c_1^2, \dots, c_{10}^2 and x_2 , while r_1, \dots, r_{10} are not correlated with either x_1 and x_2 , as intended.

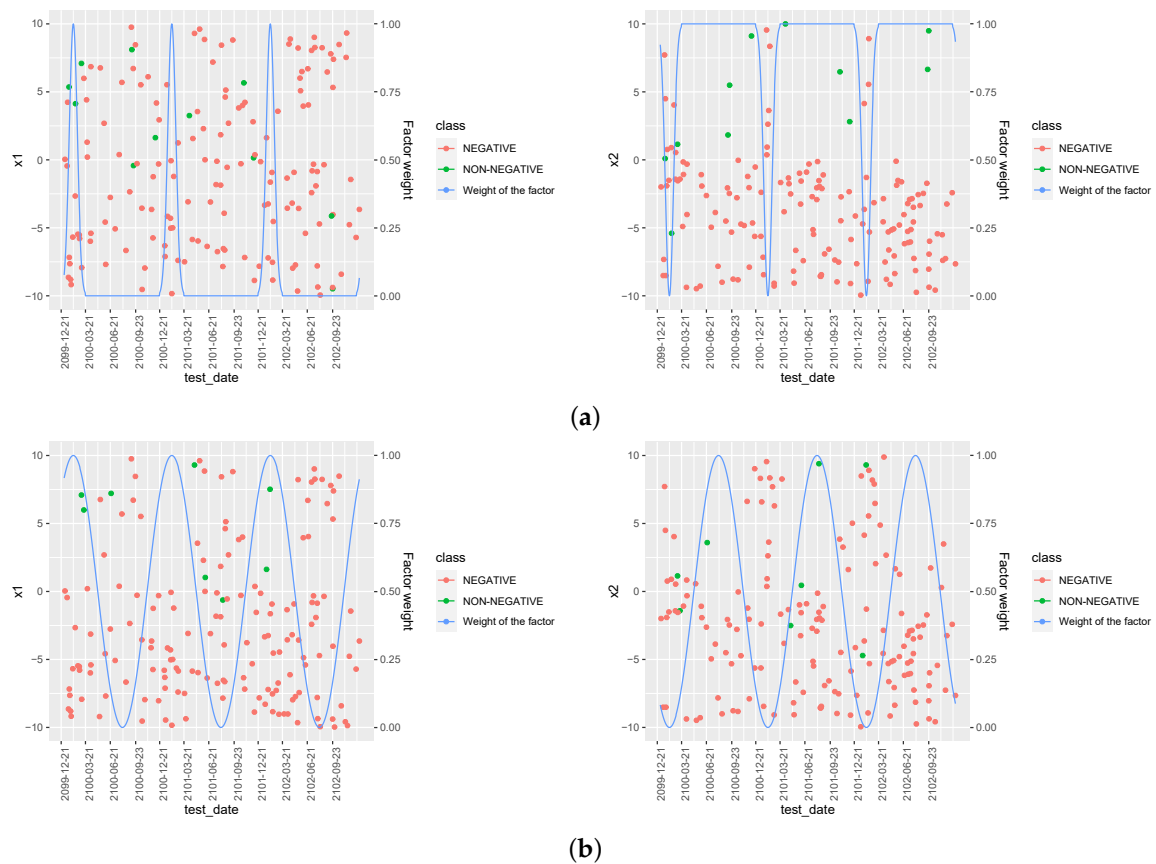


Figure 3. Values of x_1 and x_2 in relation to the date of the datapoint and the outcome class for (a) the *condensed* and (b) the *sinusoidal* datasets. The variation in factors k_1 and k_2 is displayed, clearly showing the effect of the corresponding variables on the class when their value is 1 and the loss of the effect when it is 0. The graphics are displayed as an example, and contain only a small part of the datasets.

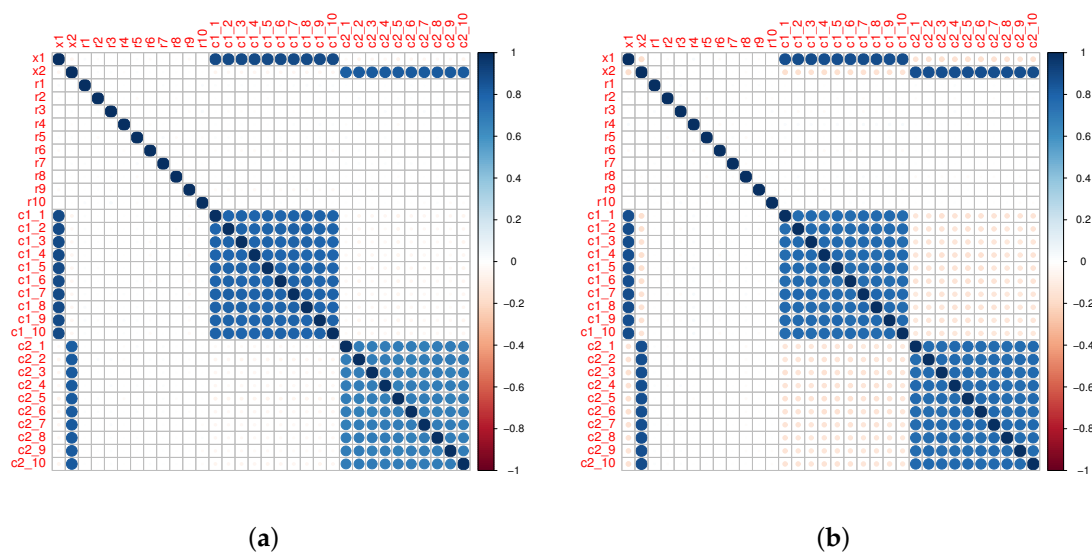


Figure 4. Correlation between features of (a) the *condensed* and (b) the *sinusoidal* datasets. x_1 and x_2 are the features that affect the final class, while $c_1^1, \dots, c_{10}^1, c_1^2, \dots, c_{10}^2$ are clearly correlated with one of them; r_1, \dots, r_{10} are simple random variables.

3.5.2. Clinical Datasets

We extracted two datasets from the MIMIC-III database in order to test the performance of these techniques with real hospital data. MIMIC-III is a freely available database containing data regarding hospital admissions to the clinical care units of the Beth Israel Deaconess Medical Center in Boston, Massachusetts [14]. It includes a wide variety of data, including demographics, microbiology cultures, laboratory tests, and bedside monitoring.

Data Extraction

It should be noted that the aim of this work is not to develop a precise clinical model; rather, it is to test the performance of the different approaches employed to deal with seasonality in this context. Therefore, we designed a query for MIMIC-III in order to extract a dataset containing generic data related to infections that might be suitable for our study.

The query was specifically designed to retrieve the first positive microbiology test for each microorganism and sample type in every admission. It collected various demographic data (age, gender, insurance, marital status, ethnicity), data related to microbiology tests (microorganism found, type of sample, date of the test), and hospital stay data (admission type and location, previous ICU stays, current service at the time of test ordering), as well as the mean, maximum, and minimum values of the white blood cell count and lactate within a 24 h time windows on the day the sample was obtained. The code of the query is depicted in Appendix A.

We generated two datasets from the results of the query, each of which was focused on a different species of bacteria. The target outcome was to predict whether the microorganism isolated in each test belonged to the species of bacteria being studied.

In the *Acinetobacter* dataset, we focused on bacteria belonging to the *Acinetobacter* species. These are responsible for many healthcare-associated infections (HCAIs), and multiple studies suggest the existence of clear seasonal variations in these infections [3]. In this dataset, we assumed that those microbiology tests that were positive for *Acinetobacter* sp., *Acinetobacter baumannii*, or *Acinetobacter baumannii* complex belonged to the *positive* class and that the others were *negative*.

Another bacteria species with clear seasonal variations is *Streptococcus pneumoniae*, for which infections are known to occur more frequently in cold seasons [39]. Using a similar strategy, we generated an *S. pneumoniae* dataset in which those microbiology tests in which *S. pneumoniae* were detected were considered as *positive* and the others were considered as *negative*.

The time when the microbiology sample was obtained was considered as the temporal reference in these datasets. The data in the MIMIC-III database have been de-identified in order to protect the patients' confidentiality, which implies that the available date is not the real one. The de-identification procedure randomly shifts the real date into the future, sometime between the years 2100 and 2200. However, the season is preserved (i.e., an observation made during winter will be shifted to a winter month in the future), making these data appropriate for our work.

Data Preprocessing

We carried out further transformations in both datasets in order to adapt them to the techniques used in this work. The patients' ages were stratified as adult (between 16 and 65) or elderly (65 and over). Only the microbiology tests concerning sputum, bronchoalveolar lavage, blood culture, and swab were considered, as these are the samples related to systemic and respiratory infections caused by the studied bacteria types. Only the most frequent values for *admission location*, *marital status*, *ethnicity*, and *current service* were considered, while the rest were labeled as *other*.

Next, we discarded those cases in which any of the selected attributes were missing in order to obtain a consistent dataset. To facilitate model development, we converted each multilevel categorical attribute into multiple Boolean attributes. However, we decided

not to normalize the continuous attributes, as doing this could potentially impact the interpretability of the models.

Appendix B provides a summary of the attributes available in both the *Acinetobacter* and *S. pneumoniae* datasets along with their representation in each class. Both datasets have a noticeable class imbalance (i.e., 6301 negative vs. 61 positive cases in the *Acinetobacter* dataset and 6280 negative vs. 82 positive cases in the *S. pneumoniae* dataset) and high dimensionality (i.e., a total of 52 features after the aforementioned transformations).

4. Experiments and Results

In this section, we analyze the impact of the aforementioned methods and their combinations when creating models for the high-dimensional imbalanced datasets proposed above.

4.1. Experimental Settings

We experimented with five different approaches used to deal with seasonality in data:

- *None*: we built a single model that ignores the season; the results of this experiment were used as a gold standard for comparison.
- *Season as a feature*: we built a single model that includes the season as an additional feature for prediction.
- *Model per season*: we built isolated models, one for each season; for a given observation to be predicted, the model corresponding with the relevant season is used.
- *Monthly/seasonal window (3,5,7,9)*: as explained earlier, a sliding window was adapted to account for seasonality; we experimented using windows with lengths of 3, 5, 7, and 9 months and with a window containing both the season of the observation to be predicted and the adjacent ones (i.e., a seasonal window).
- *Monthly/seasonal ensemble*: we used an ensemble model that aggregates the output of different models for each prediction, as explained previously; we experimented with an ensemble of twelve models (i.e., one model per month) and four models (i.e., one model per season).

Figure 5 provides an overview of the workflow employed in the experimental settings. We adopted a training–validation–testing strategy [40]. First, the dataset was split into training/validation (80% of the data) and testing datasets (20% of the data). We followed a random sampling strategy only within each class group in order to preserve the overall class distribution of the data. We then generated 100 datasets from the original training/validation subset using a sampling with replacement strategy.

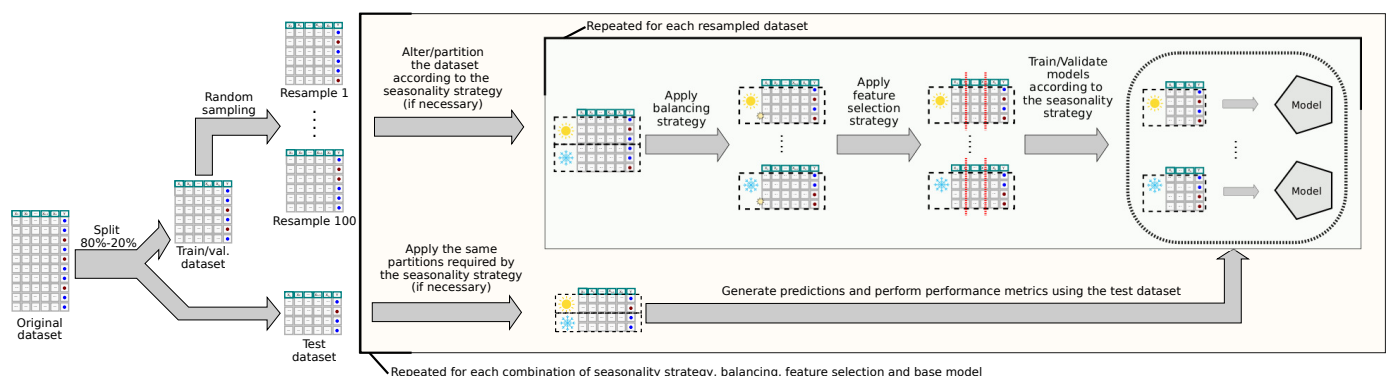


Figure 5. Workflow followed when performing the experiments.

Each particular combination of techniques was then applied to each resampling of the training/validation dataset. The first step consisted of applying the technique related to seasonality. Several of these approaches might imply the creation of various models over different subsets of data (e.g., a model based on data for winter only). After the training/validation resample had been partitioned (if necessary), we then applied the

balancing approach and the feature selection algorithm, and the model was eventually built using either logistic regression or the C5.0 algorithm for tree models. In addition to the tested approaches for feature selection, we applied the LASSO technique to the logistic regression models and winnowing to the decision trees in all our experiments. These techniques are common approaches which are used together with the aforementioned modeling techniques in the presence of high dimensionality, and as such are suitable for inclusion in this scenario.

We consequently trained 100 models per combination of approaches and dataset (three options for feature selection, five options for class balancing, ten options for seasonality, and two techniques for interpretable models), that is, a total of 30,000 models per dataset.

Each model we created was then used to predict the test datasets, and the resulting area under the receiver operating characteristic curve (AUC) was stored. This made it possible to obtain 100 AUC results per combination of techniques. A *t*-test was then performed in order to calculate the mean and 95% confidence intervals per combination used in the remainder of the analysis. The number of predictors included in these models was studied as an approximation with which to evaluate the differences in complexity between models.

All the experiments were performed using the R platform version 4.0.2 and RStudio version 1.3.1093. The LASSO models were fitted using the *glmnet* R package [41,42]. The α parameter, which controls the elastic net behavior, was set to $\alpha = 1.0 - 10^{-5}$ in order to obtain a LASSO effect and ensure numerical stability [42]. The λ we eventually used was $\lambda = \lambda_{1se}$, that is, the value obtained by the model had an error within one standard error from the minimum when performing ten-fold cross validation on the training/validation dataset, as suggested in [42]. The decision tree models were created using the C50 package [43], with active winnowing and the remaining parameters set to their default values (trials = 1, rules = false, subset = true, bands = 0, noGlobalPruning = false, CF = 0.25, minCases = 2, fuzzyThreshold = false, sample = 0, earlyStopping = true). The *Biocomb* package [44] was used for the experiments with FCBF. In this case the *threshold* parameter was set to 0 for an initial safe approach, as suggested by [45]. The implementations of Fisher's exact test and Student's *t*-test used in the filter by the *p* value were those included in the *stats* package of the R platform.

4.2. Seasonality in Data

In order to assess whether seasonality has an effect on the relationships between the predictors and the target outcome, we performed a univariate test between each feature and the class using data from different seasons.

Figure 6a–d show the changes in the *p* values of features when partitioning the datasets by the season of the observed data. These figures include the traditional cut-off of $p = 0.05$, shown as a dashed line. In the condensed and sinusoidal datasets, the effect is clear; the feature x_1 has its maximum relevance (lowest *p* value) in winter, as expected, to the point that it is above 0.05 during the rest of the year in the condensed dataset (Figure 6a) and during summer in the sinusoidal dataset (Figure 6b).

With regard to the clinical datasets, the two features with the minimum *p* values in each season were selected in order to study their variation during the rest of the year. Again, clear variations are present among seasons. For example, the *min_lactate* feature in the *Acinetobacter* dataset has a low *p* value (high relevance) in spring and winter, yet its *p* value would lead it to be discarded as a relevant feature in summer. In the *S.pneumoniae* dataset, the fact that the specimen is of the sputum type is more relevant in summer than in the winter, while the contrary occurs with respect to the datapoint patient concerning whether the patient received ICU service before.

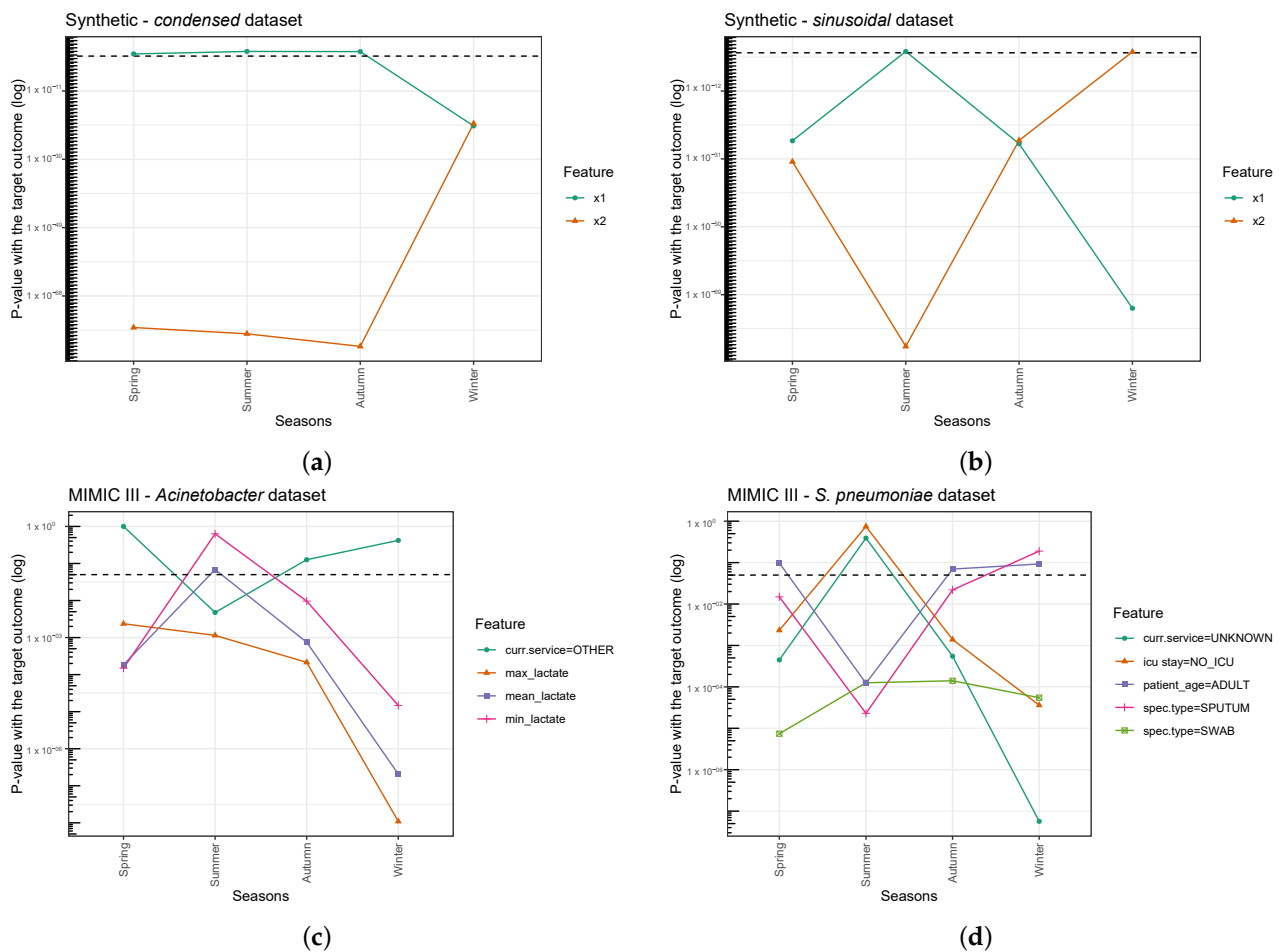


Figure 6. p -value of different features when considering data from a specific season only in (a) the condensed dataset, (b) the sinusoidal dataset, (c) the *Acinetobacter* dataset, and (d) the *S. pneumoniae* dataset. The dashed line represents the traditional cut-off of $p = 0.05$ used in feature selection.

4.3. Separate Analysis of the Effects of Different Approaches

We initially performed an analysis of the effects of using each set of techniques described in this work separately, i.e., feature selection to reduce high dimensionality, sampling to compensate class imbalance, and approaches for dealing with seasonality in data.

Figure 7a compares the results of each feature selection approach in terms of mean AUC, while Figure 7b compares the number of features included in the models, with the aim of illustrating the variations in model complexity.

The use of FCBF without combining it with other techniques tends to reduce the model AUC, with the exception of the condensed dataset and when used in logistic regression models. The p -value filter has less of an impact on the model AUC, and leads to a slighter reduction in model complexity.

The approaches based on decision trees obtained poor AUC results on all of the MIMIC-III datasets. In these cases, the high class imbalance led to tree models with only one node (zero features per model), resulting in all the observations being classified as belonging to the majority class regardless of the feature selection approach used.

The results when varying only the method employed to compensate the class imbalance are shown in Figure 8. When combined with logistic regression models, the balancing approaches achieve a similar or slightly lower AUC than the models without a balancing strategy. However, the performance improves when using decision trees, which is particularly relevant on the MIMIC-III datasets as the models are no longer empty. With regard to model complexity, undersampling leads to simpler models than oversampling.

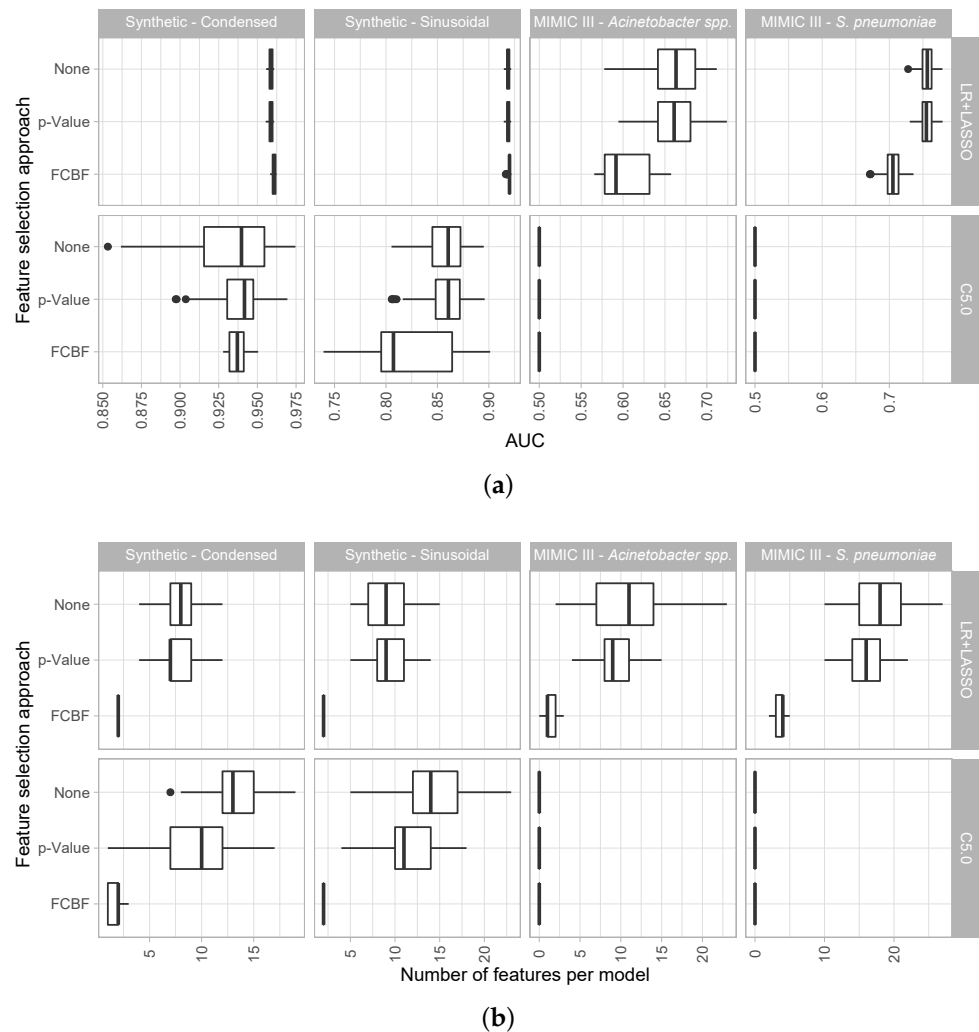


Figure 7. Values of (a) AUC and (b) number of features per model when using only a feature selection approach in each dataset.

Figure 9 shows the results obtained when using the seasonality approaches without any other preprocessing techniques. There are differences between the synthetic and clinical datasets. The use of ensembles, one model per season, or a three-month window improves the AUC results in the synthetic datasets, while leading to worse results on the MIMIC-III datasets with logistic regression. In these experiments, wider windows, the inclusion of the season as a feature, and even using no seasonal approach at all led to better performance. The results according to model complexity were more homogeneous. The use of the monthly ensemble and three-month window clearly reduced model complexity. Again, the high class imbalance in the MIMIC-III datasets led to one-node trees, resulting in models with a poor AUC.

These results indicate that the use of a seasonal ensemble leads to the same results as creating one model per season. Although they are different algorithms, their outputs are sufficiently similar that both the AUC and the number of features in the resulting models are identical.

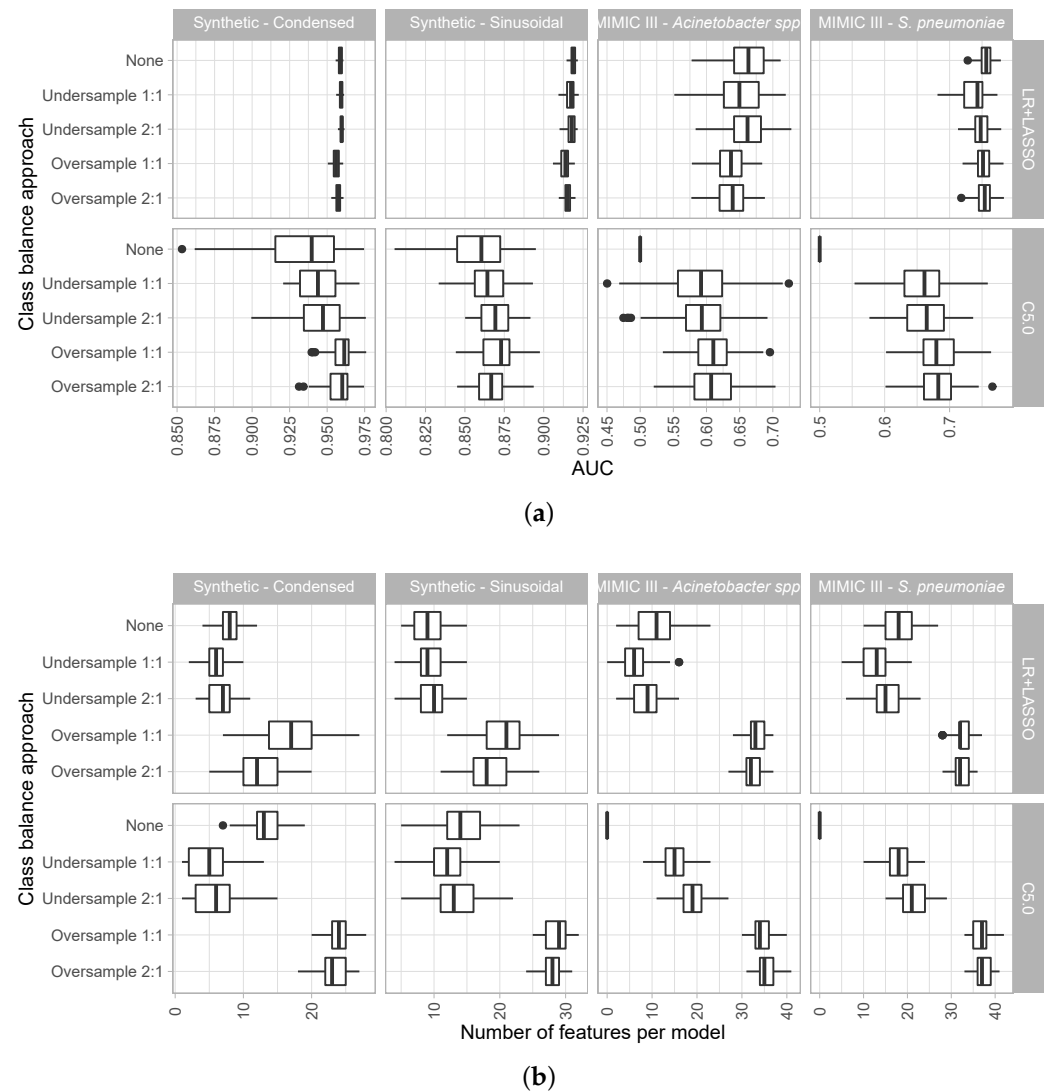


Figure 8. Values of (a) AUC and (b) number of features per model when using only a class balancing approach in each dataset, with 1:1 and 2:1 being the ratio among the different classes.

4.4. Analysis of the Effects of Different Approaches in Combination

Next, the different preprocessing and seasonal drift approaches were combined with the aim of improving the results obtained with them separately. We focused on each dataset and type of model in order to analyze them.

As the seasonal ensemble obtained the same results as the model-per-season approach, it was not included in this analysis in order to avoid repetition.

Note that it is not possible to build a model for certain datasets when no feature is able to attain the threshold set for the p -value filter ($p < 0.05$). Specifically, this occurred when combining the p -value filter with monthly ensembles and a few small sliding windows for the *Acinetobacter* and *S. pneumoniae* datasets.

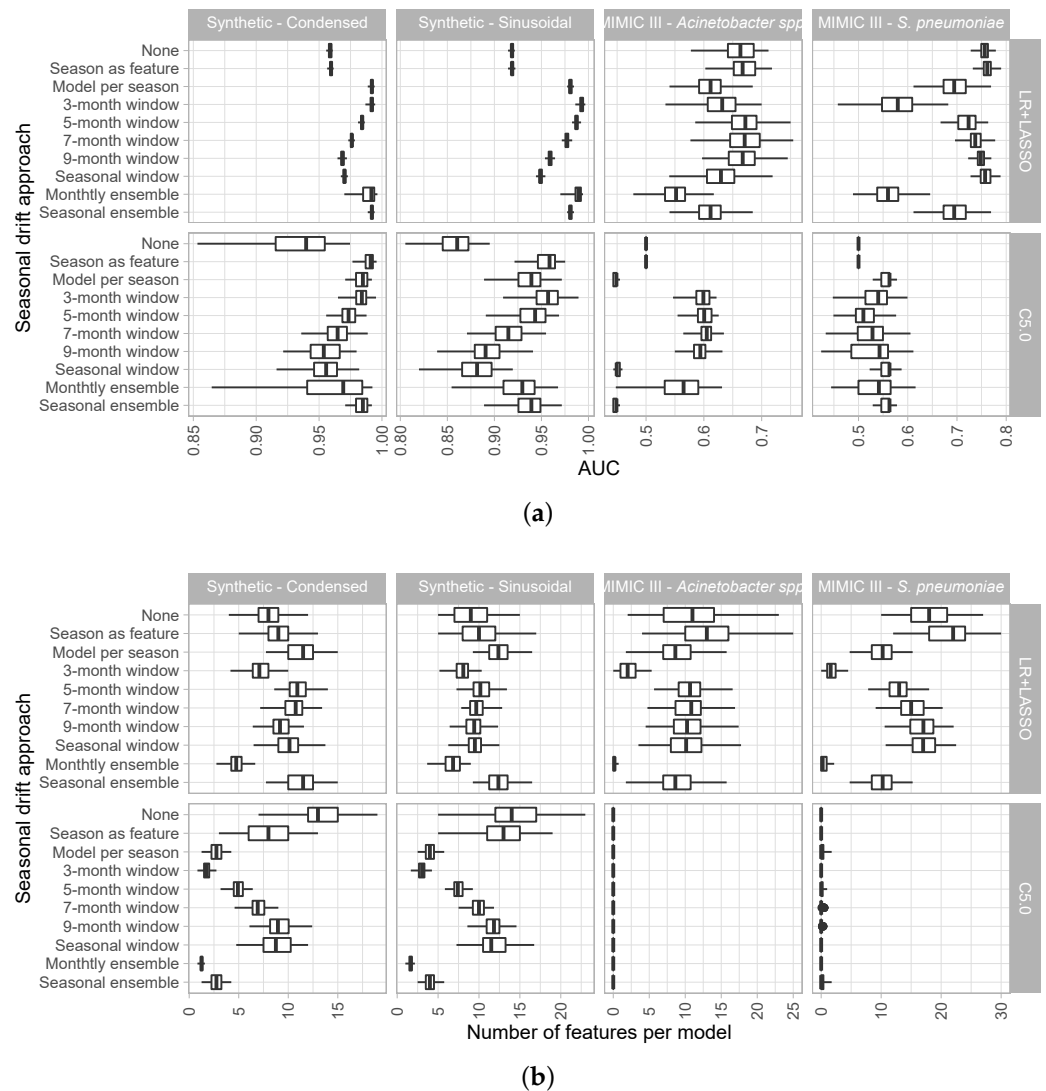


Figure 9. Values of (a) AUC and (b) number of features per model when using only a seasonal drift compensation approach in each dataset.

4.4.1. Synthetic Dataset—Condensed

The combinations that achieved the best AUC on the *condensed* dataset are ranked in Table 1, which shows the combinations with logistic regression models, and in Table 2, which shows the best results obtained by combinations with decision trees.

The approach that obtained the best mean AUC in logistic regression models was the use of one model per season, while including the season as a feature worked better for decision trees. The proposed sliding window approach using three-month windows showed promising results in both models. With regard to the filtering and balancing techniques, the combination of FCBF with 2:1 oversampling obtained the best mark with logistic regression, while oversampling with both ratios and no filter had the best results for decision trees. FCBF clearly led to simpler logistic regression models, with a mean of 1.08 features per model obtained in the best combination for logistic regression. However, FCBF did not achieve good AUC results in decision tree models. In those experiments, the simpler models were those based on a three-month window and one model per season. In all cases, the use of seasonal techniques obtained much better results than simply creating the model by ignoring seasonality.

Table 1. Top ten results obtained on the *condensed* synthetic dataset using logistic regression models.

Filter	Balancing	Seasonal Drift Approach	Mean AUC	Mean Features per Model
FCBF	Oversample 2:1	Model per season	0.995 (0.994, 0.995)	1.080 (0.993, 1.167)
FCBF	Undersample 2:1	Model per season	0.994 (0.994, 0.995)	0.623 (0.583, 0.662)
FCBF	Oversample 2:1	3-month window	0.993 (0.992, 0.994)	1.334 (1.278, 1.390)
<i>p</i> -Value	None	Model per season	0.993 (0.993, 0.993)	8.560 (8.346, 8.774)
FCBF	Undersample 2:1	3-month window	0.992 (0.992, 0.993)	0.724 (0.698, 0.751)
<i>p</i> -Value	None	3-month window	0.992 (0.991, 0.992)	5.991 (5.816, 6.166)
None	None	Model per season	0.992 (0.991, 0.992)	11.377 (11.069, 11.686)
FCBF	Oversample 1:1	Model per season	0.991 (0.991, 0.992)	3.025 (2.897, 3.153)
None	None	3-month window	0.991 (0.990, 0.992)	7.065 (6.822, 7.308)
<i>p</i> -Value	Undersample 2:1	Model per season	0.991 (0.990, 0.991)	7.740 (7.552, 7.928)
...
None	None	None	0.959 (0.959, 0.959)	7.790 (7.411, 8.169)

Table 2. Top ten results obtained on the *condensed* synthetic dataset using decision tree models.

Filter	Balancing	Seasonal Drift Approach	Mean AUC	Mean Features per Model
None	Oversample 1:1	Season as feature	0.992 (0.991, 0.992)	15.840 (15.482, 16.198)
None	Oversample 2:1	Season as feature	0.992 (0.991, 0.992)	14.730 (14.376, 15.084)
None	Oversample 2:1	3-month window	0.988 (0.988, 0.989)	5.056 (4.910, 5.202)
<i>p</i> -Value	Oversample 1:1	3-month window	0.988 (0.988, 0.989)	5.109 (4.959, 5.260)
<i>p</i> -Value	Oversample 2:1	3-month window	0.988 (0.988, 0.989)	4.896 (4.760, 5.031)
None	Oversample 1:1	3-month window	0.988 (0.987, 0.989)	5.269 (5.113, 5.426)
<i>p</i> -Value	Oversample 2:1	Model per season	0.988 (0.987, 0.988)	4.148 (4.058, 4.237)
None	Oversample 2:1	Model per season	0.987 (0.987, 0.988)	4.188 (4.099, 4.276)
<i>p</i> -Value	Oversample 1:1	Model per season	0.987 (0.987, 0.988)	4.510 (4.421, 4.599)
None	Oversample 1:1	Model per season	0.987 (0.986, 0.988)	4.595 (4.502, 4.688)
...
None	None	None	0.931 (0.925, 0.937)	12.920 (12.355, 13.485)

4.4.2. Synthetic Dataset—Sinusoidal

Tables 3 and 4 show the top ten combinations of techniques according to their mean AUC when applied to the *sinusoidal* dataset.

Our proposed sliding window approach obtains the best results in both the logistic regression and decision tree models with windows of shorter length (three and five months). Unlike the previous synthetic dataset, the use of any balancing strategy slightly worsens the results in logistic regression models, though it remains decisive for decision tree models. The filter based on the *p* value reduces the complexity of models with only a slight reduction or no reduction in model performance. Again, combinations of the studied techniques obtain the best results when compared to the model created without considering any of the techniques.

4.4.3. MIMIC-III *Acinetobacter* Dataset

Tables 5 and 6 show the top ten combinations of techniques according to their mean AUC obtained on the *Acinetobacter* dataset.

In this case, the proposed seven-month window without any feature selection or balancing technique attains the best results with logistic regression, while the monthly ensembles attain the best results for decision trees when combined with oversampling techniques. The slight reduction in dimensionality obtained when using the *p*-value filter is noteworthy, though in this case it additionally implies a slight reduction in AUC.

4.4.4. MIMIC-III *S. Pneumoniae* Dataset

Tables 7 and 8 show the top ten combinations of techniques according to their mean AUC obtained on the *S. pneumoniae* dataset.

The inclusion of the season as a feature led to the best results with the logistic regression models and to good results with decision trees. With regard to the latter, the seasonal window obtained the best results when combined with 1:1 undersampling and a p -value filter. The combination of these techniques did not drastically improve the results for logistic regression models in this dataset, to the point that the models trained with no more than LASSO and logistic regression are among the top ten results. In the case of decision trees, the use of any balancing strategy is again decisive with regard to obtaining a valid model.

Table 3. Top ten results obtained on the *sinusoidal* synthetic dataset with logistic regression models.

Filter	Balancing	Seasonal Drift Approach	Mean AUC	Mean Features per Model
None	None	3-month window	0.992 (0.991, 0.992)	7.907 (7.665, 8.150)
p -Value	None	3-month window	0.991 (0.990, 0.993)	6.112 (5.934, 6.291)
p -Value	Undersample 2:1	5-month window	0.989 (0.988, 0.989)	8.742 (8.574, 8.909)
None	Undersample 2:1	5-month window	0.988 (0.987, 0.988)	9.991 (9.777, 10.204)
p -Value	Undersample 1:1	5-month window	0.988 (0.987, 0.988)	8.287 (8.122, 8.452)
p -Value	None	5-month window	0.988 (0.987, 0.988)	9.057 (8.874, 9.241)
p -Value	Undersample 2:1	3-month window	0.987 (0.985, 0.990)	6.197 (6.023, 6.370)
None	None	5-month window	0.987 (0.987, 0.988)	10.280 (10.051, 10.509)
None	Undersample 1:1	5-month window	0.987 (0.986, 0.988)	9.542 (9.333, 9.750)
None	Undersample 2:1	3-month window	0.987 (0.984, 0.989)	7.533 (7.278, 7.789)
...
None	None	None	0.919 (0.918, 0.919)	9.600 (9.114, 10.086)

Table 4. Top ten results obtained on the *sinusoidal* synthetic dataset with decision tree models.

Filter	Balancing	Seasonal Drift Approach	Mean AUC	Mean Features per Model
p -Value	Oversample 2:1	3-month window	0.976 (0.974, 0.978)	6.862 (6.684, 7.041)
None	Oversample 2:1	3-month window	0.976 (0.974, 0.978)	6.947 (6.764, 7.131)
p -Value	Oversample 1:1	3-month window	0.976 (0.974, 0.978)	7.112 (6.905, 7.319)
None	Oversample 1:1	3-month window	0.975 (0.973, 0.977)	7.208 (6.993, 7.423)
FCBF	Oversample 2:1	3-month window	0.975 (0.973, 0.977)	2.859 (2.812, 2.906)
FCBF	Oversample 1:1	3-month window	0.973 (0.970, 0.976)	2.533 (2.492, 2.573)
p -Value	Oversample 1:1	5-month window	0.968 (0.967, 0.969)	14.316 (14.200, 14.431)
p -Value	Oversample 2:1	5-month window	0.968 (0.967, 0.969)	13.209 (13.110, 13.308)
None	Oversample 1:1	5-month window	0.968 (0.967, 0.969)	14.779 (14.647, 14.912)
None	Oversample 2:1	5-month window	0.967 (0.966, 0.968)	13.718 (13.613, 13.824)
...
None	None	None	0.852 (0.846, 0.859)	14.270 (13.451, 15.089)

Table 5. Top ten results obtained with logistic regression models on the *Acinetobacter* spp. dataset extracted from MIMIC-III.

Filter	Balancing	Seasonal Drift Approach	Mean AUC	Mean Features per Model
None	None	7-month window	0.671 (0.664, 0.678)	10.532 (10.034, 11.029)
None	None	5-month window	0.670 (0.664, 0.677)	10.793 (10.357, 11.229)
p -Value	None	7-month window	0.667 (0.660, 0.675)	9.562 (9.284, 9.839)
p -Value	None	5-month window	0.667 (0.660, 0.674)	9.411 (9.156, 9.666)
None	None	Season as feature	0.665 (0.659, 0.670)	12.990 (12.107, 13.873)
None	None	9-month window	0.664 (0.657, 0.671)	10.344 (9.780, 10.909)
p -Value	None	9-month window	0.663 (0.657, 0.670)	9.272 (9.003, 9.540)
p -Value	None	Season as feature	0.663 (0.658, 0.668)	10.400 (9.957, 10.843)
p -Value	Undersample 2:1	Season as feature	0.662 (0.656, 0.668)	8.740 (8.309, 9.171)
None	Undersample 2:1	Season as feature	0.661 (0.654, 0.669)	10.350 (9.475, 11.225)
...
None	None	None	0.660 (0.654, 0.666)	11.050 (10.193, 11.907)

Table 6. Top ten results obtained with decision tree models on the *Acinetobacter* spp. dataset extracted from MIMIC-III.

Filter	Balancing	Seasonal Drift Approach	Mean AUC	Mean Features per Model
None	Oversample 2:1	Monthtly ensemble	0.623 (0.612, 0.633)	8.107 (7.833, 8.380)
p-Value	Oversample 2:1	Monthtly ensemble	0.621 (0.611, 0.632)	8.093 (7.822, 8.365)
None	Oversample 1:1	Monthtly ensemble	0.619 (0.609, 0.628)	7.576 (7.317, 7.834)
p-Value	Oversample 1:1	Monthtly ensemble	0.618 (0.607, 0.628)	7.532 (7.281, 7.782)
p-Value	Oversample 1:1	5-month window	0.617 (0.609, 0.625)	22.009 (21.809, 22.209)
None	Oversample 1:1	5-month window	0.617 (0.609, 0.624)	22.712 (22.502, 22.923)
p-Value	Oversample 2:1	None	0.615 (0.607, 0.623)	31.960 (31.515, 32.405)
p-Value	Oversample 2:1	Season as feature	0.614 (0.606, 0.622)	34.660 (34.210, 35.110)
p-Value	Oversample 2:1	5-month window	0.614 (0.606, 0.622)	22.738 (22.539, 22.936)
p-Value	Oversample 1:1	None	0.613 (0.605, 0.620)	31.800 (31.392, 32.208)
...
None	None	None	0.499 (0.498, 0.500)	0.090 (−0.012, 0.192)

Table 7. Top ten results obtained with logistic regression models on the *S. pneumoniae* dataset extracted from MIMIC-III.

Filter	Balancing	Seasonal Drift Approach	Mean AUC	Mean Features per Model
None	Oversample 2:1	Season as feature	0.762 (0.759, 0.765)	35.310 (34.884, 35.736)
p-Value	Oversample 2:1	Season as feature	0.762 (0.759, 0.764)	33.790 (33.352, 34.228)
None	None	Season as feature	0.760 (0.758, 0.763)	21.310 (20.517, 22.103)
p-Value	Oversample 1:1	Season as feature	0.760 (0.757, 0.763)	34.800 (34.335, 35.265)
None	Oversample 1:1	Season as feature	0.760 (0.757, 0.763)	35.840 (35.442, 36.238)
p-Value	None	Season as feature	0.759 (0.757, 0.762)	18.150 (17.653, 18.647)
p-Value	None	Seasonal window	0.758 (0.756, 0.761)	14.435 (14.067, 14.803)
None	None	Seasonal window	0.757 (0.754, 0.760)	16.868 (16.323, 17.412)
None	None	None	0.755 (0.753, 0.757)	18.070 (17.311, 18.829)
p-Value	None	None	0.755 (0.752, 0.757)	16.120 (15.644, 16.596)

Table 8. Top ten results obtained with decision tree models on the *S. pneumoniae* dataset extracted from MIMIC-III.

Filter	Balancing	Seasonal Drift Approach	Mean AUC	Mean Features per Model
p-Value	Undersample 1:1	Seasonal window	0.690 (0.682, 0.698)	6.628 (6.271, 6.984)
p-Value	Oversample 2:1	Season as feature	0.685 (0.678, 0.692)	37.590 (37.231, 37.949)
None	Oversample 2:1	Season as feature	0.685 (0.678, 0.691)	40.480 (40.031, 40.929)
FCBF	Oversample 1:1	Season as feature	0.684 (0.677, 0.691)	9.780 (9.494, 10.066)
p-Value	Oversample 1:1	Season as feature	0.684 (0.677, 0.690)	37.950 (37.569, 38.331)
None	Oversample 1:1	Season as feature	0.683 (0.676, 0.690)	40.370 (40.008, 40.732)
p-Value	Oversample 1:1	None	0.682 (0.676, 0.689)	34.810 (34.429, 35.191)
p-Value	Oversample 2:1	None	0.682 (0.675, 0.688)	34.420 (34.061, 34.779)
None	Oversample 2:1	None	0.681 (0.675, 0.687)	37.290 (36.893, 37.687)
p-Value	Undersample 1:1	9-month window	0.681 (0.674, 0.687)	6.900 (6.652, 7.148)
...
None	None	None	0.502 (0.499, 0.505)	0.180 (−0.001, 0.361)

4.5. Analysis of the Impact of the Different Approaches on Interpretability

To the best of our knowledge, there is no widely accepted metric for the interpretability of a model. The number of features, which has been studied in the previous sections, could be a good approximation, as it is related to the complexity of the model. However, the use of approaches that involve a combination of multiple submodels may impact interpretability as well, even when interpretable submodels are used.

We provide an example using the models generated for the sinusoidal synthetic dataset. As a reminder, to generate this dataset we used a sinusoidal function to vary

the effect of the two main coefficients, x_1 and x_2 , in the outcome variable throughout the year. The coefficient x_1 was the most relevant factor in winter, while it had no effect on the outcome in summer. The feature x_2 had the opposite effect, being the most relevant in summer. The models selected here were trained with one of the samplings from the training/validation dataset without applying any filter or balancing strategy. By examining these models, we aim to highlight the differences in interpretability resulting from the different approaches to handling seasonality. The results shown here extend to both tree-based models and models generated with logistic regression; however, only the latter are shown in order to avoid duplication.

Figure 10 shows the values of the coefficients of the logistic regression model generated without applying any seasonal drift approach (Figure 10a) and when including the season as an additional feature (Figure 10b). These are common logistic regression models that are easily interpretable; in this case, x_1 and x_2 are much more relevant than the rest of the features. Among the other features selected, those starting with $c1_$ and $c2_$ refer to the c_i^1 and c_j^2 features added to complicate the dataset, which are highly-correlated with x_1 and x_2 , respectively. When the season is included as a feature (Figure 10b), it is included in the model, but it has little relevance compared to x_1 and x_2 .

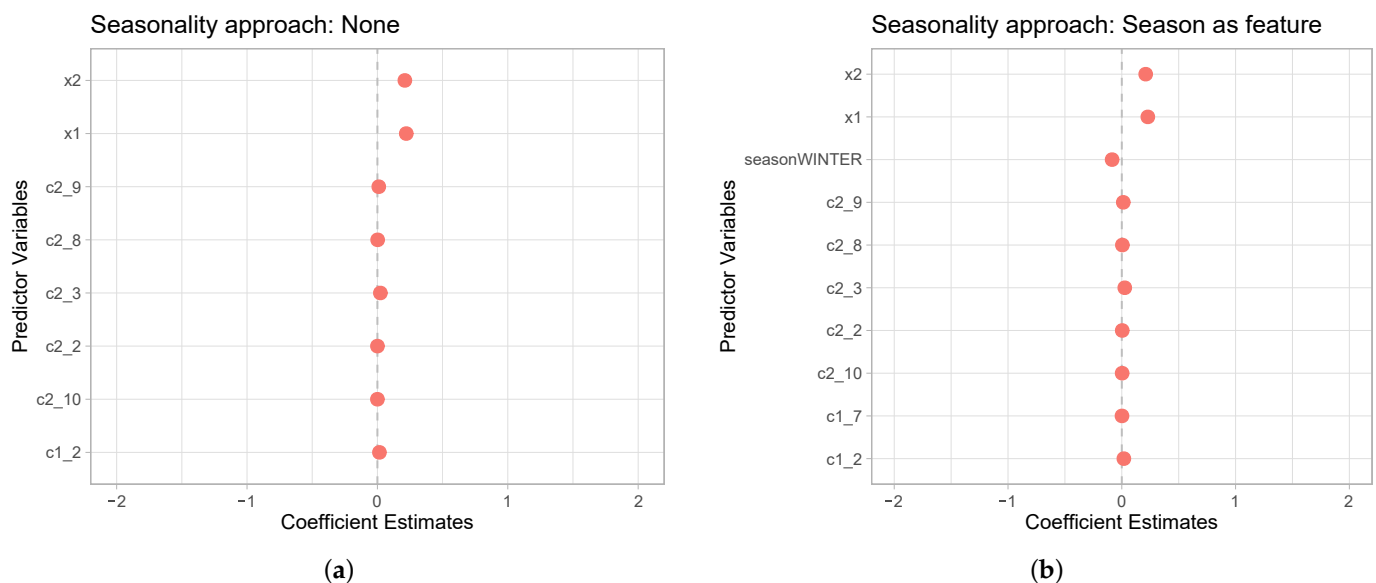


Figure 10. Coefficients of the logistic regression models obtained (a) without using any approach for dealing with seasonality and (b) when incorporating the season as a feature. The intercept of the models is not displayed for better clarity.

All these findings are consistent with the known behavior of the underlying model that generated the dataset; the yearly variations in the relevance of x_1 and x_2 are smooth, and as such the model must consider both in order to generate a prediction when the dataset is treated as a whole.

Figure 11 shows the models that were generated when using the strategy of creating a different model for each season. This model may appear more complicated to understand; however, because each model is used in a particular condition (i.e., to predict an observation of a particular season) and they can be interpreted separately, on the whole we consider it easy to interpret these models. For example, it can be observed that while the relevance of x_2 is high in the summer model, it does not even appear for consideration in the winter model, in which x_1 is by far the most relevant feature. Therefore, we can interpret this model, and even extract interesting information about the underlying impact of seasonality in the data.

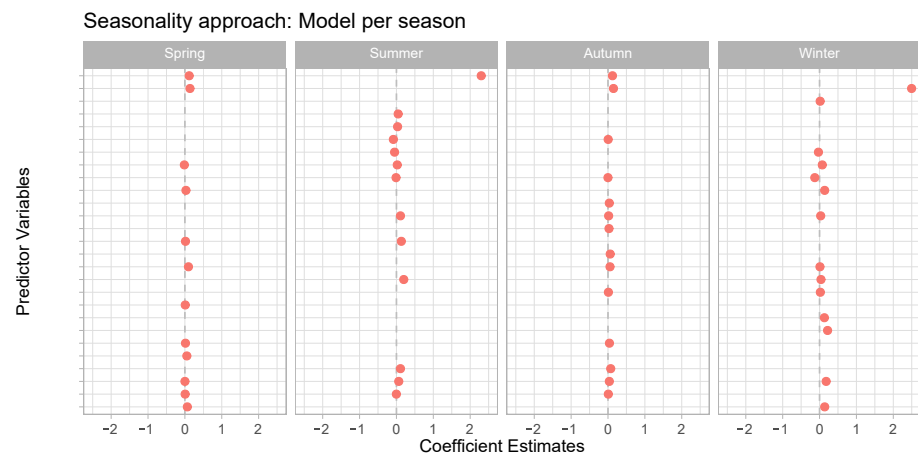


Figure 11. Coefficients of the logistic regression models obtained when creating a different model for each season. The intercept of the models is not displayed for clarity.

Figure 12 illustrates the models generated when using the seasonal window strategy. Although the models appear similar to the model-per-season approach, the amount of data used to build each model is larger; therefore, the impact of seasonality on the models cannot be appreciated as clearly. However, the increased relevance of x_2 in summer and x_1 in winter can be observed, though not as clearly as in the previous approach. Despite this, it is important to consider how these models were generated when analyzing their structure; with this context, they are interpretable and the reasoning behind each prediction can be easily traced.

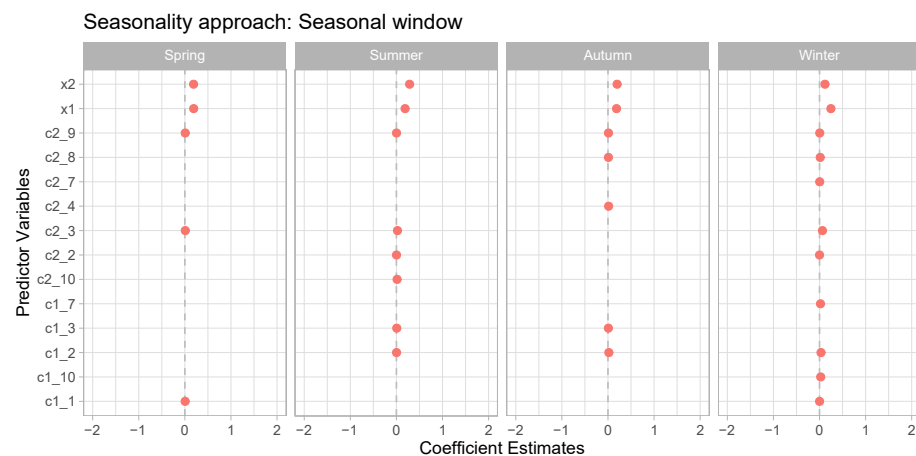


Figure 12. Coefficients of the logistic regression models obtained when using the seasonal window approach. The intercept of the models is not displayed for better clarity.

The results when using models for each month with the monthly window approach is shown in Figure 13. The models were made using a three-month sliding window approach; therefore, the effects of seasonality are not as diluted as in the previous example. Despite the increase in complexity, the change in relevance of x_1 and x_2 in each model are noticeable, and the behavior of the approach as a whole can be easily analyzed.

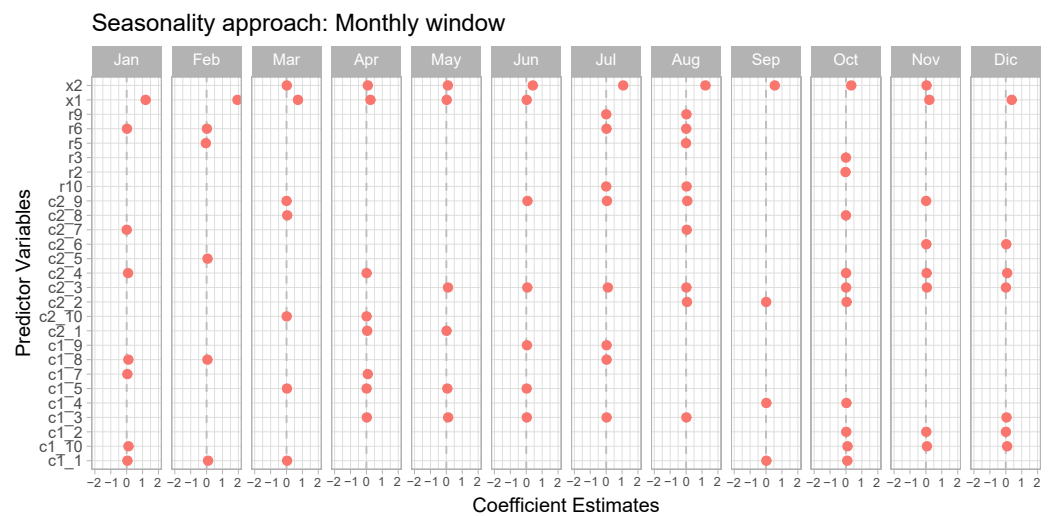


Figure 13. Coefficients of the logistic regression models obtained when using the monthly window approach with a three-month sliding window. The intercept of the models is not displayed for clarity.

The use of ensembles has a noticeable impact on interpretability, as can be appreciated in Figure 14. In this approach, we have to combine the outputs of the different models using a weight matrix, such as the one in Figure 14b. Even though the models can be analyzed independently (indeed, they are the same as those from the model-per-season approach), the combination matrix can be easily understood. For example, the matrix in Figure 14b suggests that for predicting data in summer or winter the output mainly relies on the models generated using data from these seasons, while in spring and autumn the output is mainly a mixture of the models for the other seasons. Although the whole model is more complicated than the previous approaches, it can be understood and analyzed.

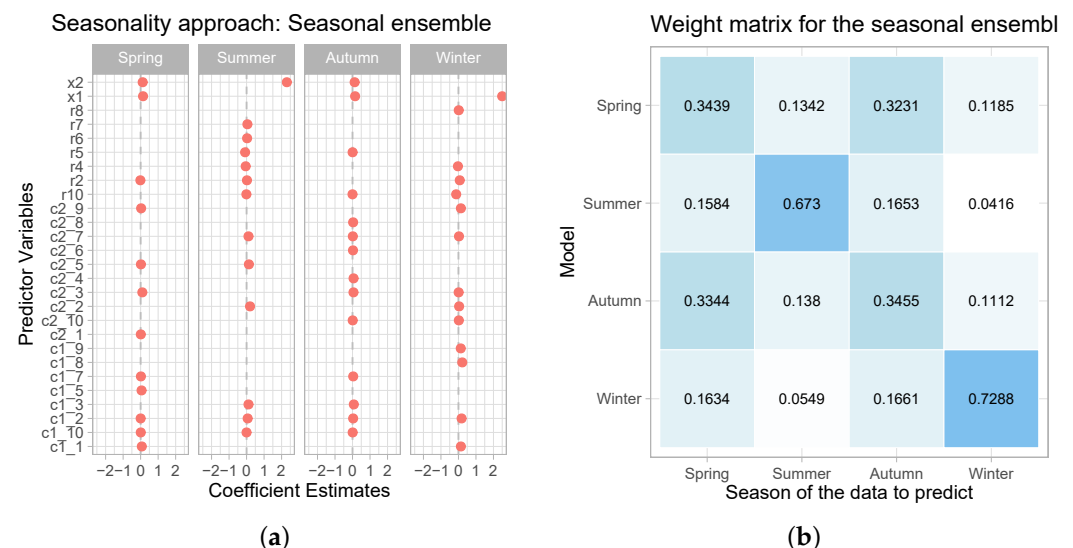
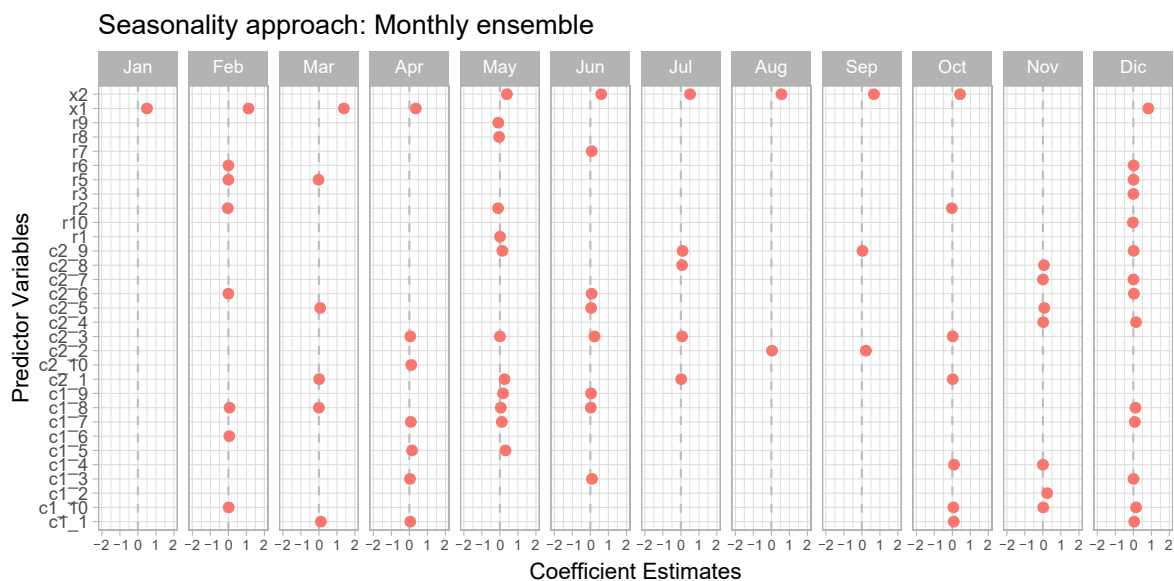


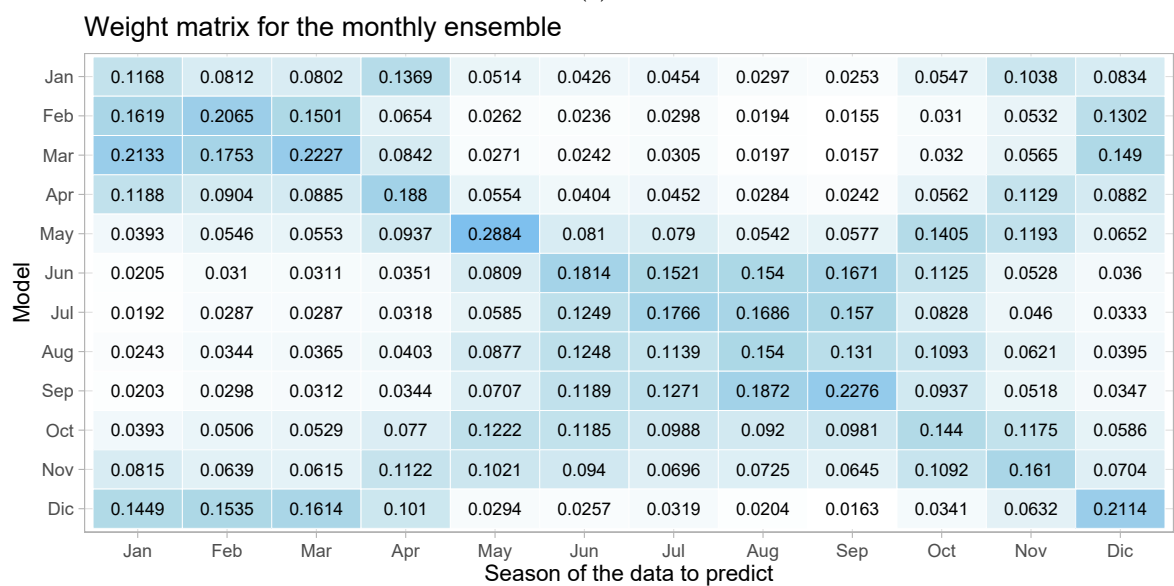
Figure 14. (a) Coefficients of the logistic regression models obtained in the seasonal ensemble approach and (b) the weight matrix required to obtain the result of the ensemble. The intercept of the models is not displayed for clarity.

The monthly ensemble, illustrated in Figure 15a,b, represents the most complex combination of all the studied approaches. Assessing the exact relevance of each feature on a specific output becomes challenging when using this approach. Nonetheless, the variations in the coefficients x_1 and x_2 can be observed across the monthly models. Additionally,

from the model matrix we are able to discern that the models adjacent to the month of the observation to be predicted have a greater impact on the output of the ensemble.



(a)



(b)

Figure 15. (a) Coefficients of the logistic regression models obtained in the monthly ensemble approach and (b) the weight matrix required to obtain the result of the ensemble. The intercept of the models is not displayed for clarity.

5. Discussion

According to the obtained results, no one specific approach or combination clearly outperforms the rest when seasonality, high dimensionality, and class imbalance are all present. However, the results provide useful information with which to discuss the advantages and disadvantages of each combination.

Despite the fact that all the experiments used LASSO or winnowing, in most cases the use of a feature selection technique reduced model complexity even more. In particular, FCBF drastically reduced the number of features; the effect of the filter based on the p value, while not as significant, was noticeable in most cases. Therefore, extra filtering techniques

appear to be advisable even in the presence of seasonality when reducing the complexity of the model is a critical requisite.

The use of feature selection techniques to reduce model complexity had differing effects on model performance. In the synthetic *condensed* dataset they clearly improved the AUC when combined with logistic regression, yet in most of the experiments they tended to slightly decrease model performance. It may be possible that when the underlying model is simple, as for the synthetic datasets, the reduction in complexity leads to the final models approximating the real ones. In the case of more complicated interactions and dependencies, as certainly occurs in clinical datasets, relevant features may be discarded, and the resulting models may lose accuracy. Therefore, the common trade-off between model simplicity and performance is present in these kinds of datasets.

In certain experiments combining techniques that severely reduced the number of observations used to train the models, the p -value filter was unable to select any features at all. For example, the combination of the p -value filter, monthly ensemble, and undersampling was unable to create valid models for most of the training/validation sampled datasets. As is well known, the P value is affected by the number of observations; thus, if no strong correlation is clear in the dataset, no feature is able to reach the cut-off value. Therefore, these combinations should be used with caution when the available training data are scarce. In the *Acinetobacter* dataset, the seven-month and five-month window approaches obtained the best results for logistic regression and the monthly ensemble obtained the best results for decision trees. The good results of month-based approaches rather than season-based ones with the MIMIC-III database may seem surprising, as the months in the timestamps of the MIMIC-III database are randomized to ensure patient confidentiality. However, because the season was maintained, the month of the randomized data was close to the real month; this may explain the good performance of these methods. Moreover, the drift in data might not occur precisely within an astronomical season, and may be delayed with respect to its boundaries or even occur multiple times throughout the year. Our proposed monthly window and monthly ensemble may be good options in these cases, provided that there is at least an approximate estimation of the month.

The best sizes for the sliding window approach changed depending on the dataset. While the best results for the *Acinetobacter* dataset were obtained with lengths of seven months, on the *condensed* and *sinusoidal* datasets a three-month window was the best option. Therefore, it is important to test with different windows sizes when using this approach for seasonality, as occurs with datastreams.

The particularities of the base model may have an impact on the performance of different seasonality strategies. For example, in decision trees it is possible that the splitting algorithm can obtain similar or even better results compared to the model-per-season or ensemble approaches if it is able to effectively utilize the season as a partitioning criterion. However, when the correlation between the season and the outcome variable is unclear or subject to drift the use of these techniques may help to obtain better results, as happened in some of our experiments. The effectiveness of feature selection and balancing strategies depends on the base model as well; for example, class balance techniques were fundamental to obtaining the best results with decision trees in all of our experiments, while they did not appear to be as essential when using logistic regression.

In all cases, the use of seasonal approaches combined with other techniques improved the resulting models with regard to both AUC and simplicity when compared to the direct application of logistic regression or decision trees. The proposed approaches for seasonality (sliding windows and ensembles) attained the best performance in five of eight combinations of datasets with modeling techniques and other traditional approaches in the rest of them. This supports the idea that multiple approaches should be considered when seasonality, high dimensionality, and class imbalance are all present in a clinical dataset.

6. Limitations and Future Work

Although we used interpretable models and techniques, the complexity of a final model can complicate its interpretation. For example, the best logistic regression models obtained for the *S. pneumoniae* dataset had a mean of 35.31 features in our experiments, which might be overwhelming for an expert to understand and apply. Moreover, as discussed in Section 4.5, the use of multiple models in windows and ensembles impacts the interpretability of the overall model. However, we consider this trade-off acceptable due to the reduced number of models and their interpretation being clear in terms of how they are applied, i.e., we use a different model each month/season.

Our discussion regarding the differences between each approach relies on the graphical representations of the results and rankings of the combinations with best performance. While performing statistical tests among all possible combinations can provide additional insights and detect statistical relevance, it is challenging due to the large number of combinations and the complexity of the data being compared. In light of these limitations, we opted for a clearer and more manageable approach to analyzing and discussing the results.

In our experiments, logistic regression models with LASSO usually obtained better results than those based on C5.0. However, it is important to note that decision trees offer a wide range of tuning possibilities, such as pruning heuristics and boosting, which were not extensively explored in this study. While further research should be carried out in order to determine the best interpretable model for a particular clinical problem, we believe that our experiments can provide valuable insights into the effects of seasonality techniques in both logistic regression and decision tree models.

In our future work, we intend to study further variations of the approaches presented here. One straightforward extension would be to use different sizes of sliding windows depending on the month for which the model is being created. This would make it possible to use wider windows in months with a low number of samples, allowing the creation of more robust models, and narrower ones in months with abundant data, allowing the creation of more precise models.

Furthermore, we intend to experiment with the adaptation of clinical datasets similar to those considered here for use with new algorithms developed for datastream mining, rather than adapting the algorithms to the datasets, which was the approach followed in this work.

7. Conclusions

In this work we have studied the problem of seasonality in clinical datasets, particularly when high dimensionality and class imbalance are present. We tested the combination of multiple techniques, including two new algorithms based on datastream mining research.

Regardless of the modeling technique used, our approaches clearly obtained the best results with two datasets, and with a third when combined with decision trees. The traditional approaches of model-per-season and season-as-feature obtained the best results in several of our experiments. The top techniques employed to deal with high dimensionality and class imbalance varied, leading us to conclude that the best approach for dealing with seasonality is highly dependent on the dataset and modeling technique; therefore, in future studies several techniques should be tested in order to obtain better clinical prediction models.

In spite of the differences in our results regarding the best approach, the use of any technique to deal with seasonality improved the resulting models in all of our experiments. Although traditional approaches achieved acceptable results, our experiments indicate that the use of the proposed techniques when developing clinical prediction models can lead to increased model performance in the presence of seasonality.

Author Contributions: Conceptualization: all authors; Methodology: all authors; Formal analysis and investigation: all authors; Writing—original draft preparation: B.C.-S.; Writing—review and editing: all authors; Funding acquisition: J.M.J. and M.C.; Resources: J.M.J. and M.C.; Supervision: J.M.J. and M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by the SITSUS project (Ref: RTI2018-094832-B-I00) and the CONFAINCE project (Ref: PID2021-122194OB-I00) supported by the Spanish Ministry of Science and Innovation, the Spanish Agency for Research (MCIN/AEI/10.13039/501100011033), and as appropriate by ERDF: A Way of Making Europe.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: While MIMIC-III is a freely accessible clinical database, several requirements must be fulfilled in order to gain access to it. More information is available at [14]. The code used to generate the synthetic datasets used in this work is freely available at <https://github.com/berncase/seasonality-rProject> (accessed on 25 May 2023).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FCBF	Fast Correlation-Based Filter
LASSO	Least Absolute Shrinkage and Selection Operator
RMSE	Root Mean Squared Error
HCAI	Health Care-Associated Infection
AUC	Area Under (the receiver operating characteristic) Curve

Appendix A. MIMIC-III Query Employed to Extract the Datasets Used in This Study

```

SELECT ft.*,                                -- First test by microorganism and specimen type
a.admission_type,
a.admission_location,
a.insurance,
a.marital_status,
a.ethnicity,
EXTRACT (YEAR FROM age(ft.test_time, p.dob)) as patient_age,
p.gender as gender,
min(lwbc.valuenum) as min_white_blood_cells,
avg(lwbc.valuenum) as mean_white_blood_cells,
max(lwbc.valuenum) as max_white_blood_cells,
min(llac.valuenum) as min_lactate,
avg(llac.valuenum) as mean_lactate,
max(llac.valuenum) as max_lactate,
max(icu.intime) as most_recent_icu_in,
min(icu.outtime) as most_recent_icu_out,
CASE
WHEN max(icu.intime) IS NULL THEN 'NO_ICU'
WHEN max(icu.intime) <= ft.test_time AND min(icu.outtime) IS NULL THEN 'IN_ICU'
WHEN max(icu.intime) <= ft.test_time AND min(icu.outtime) >= ft.test_time THEN 'IN_ICU'
WHEN min(icu.outtime) <= ft.test_time AND (ft.test_time - min(icu.outtime)) <=
INTERVAL '48_hours' THEN 'LAST_48H'
WHEN min(icu.outtime) <= ft.test_time AND (ft.test_time - min(icu.outtime)) >
INTERVAL '48_hours' AND (ft.test_time - min(icu.outtime)) <=
INTERVAL '7_days' THEN 'LAST_WEEK'
WHEN min(icu.outtime) <= ft.test_time AND (ft.test_time - min(icu.outtime)) >
INTERVAL '7_days' AND (ft.test_time - min(icu.outtime)) <=
INTERVAL '1_month' THEN 'LAST_MONTH'
ELSE 'MORE_THAN_1_MONTH'
END as icu_stay,
s.curr_service
FROM
(
-- First microorganism found by type of specimen and microorganism in each admission

```

```

SELECT DISTINCT m.subject_id, m.hadm_id, m.spec_itemid, m.spec_type_desc,
m.org_itemid, m.org_name, first_value(chartdate)
OVER (PARTITION BY m.hadm_id, spec_itemid, m.org_itemid
ORDER BY m.chartdate ASC) AS test_time
FROM microbiologyevents m
WHERE
m.org_itemid IS NOT NULL
ORDER BY m.hadm_id, m.org_name
) ft
LEFT JOIN admissions a ON (ft.hadm_id = a.hadm_id) -- patient's demographic data
LEFT JOIN patients p ON (ft.subject_id = p.subject_id) -- patient's age and gender
LEFT JOIN labevents lwbc ON ( -- White blood cells values
ft.hadm_id = lwbc.hadm_id
AND lwbc.itemid = 51301
AND lwbc.charttime >= (ft.test_time - INTERVAL '24_hours') AND
lwbc.charttime <= (ft.test_time + INTERVAL '24_hours'))
LEFT JOIN labevents llac ON ( -- Lactate values
ft.hadm_id = llac.hadm_id
AND llac.itemid = 50813
AND llac.charttime >= (ft.test_time - INTERVAL '24_hours') AND
llac.charttime <= (ft.test_time + INTERVAL '24_hours'))
LEFT JOIN icustays icu ON ( -- Previous ICU stays
ft.hadm_id = icu.hadm_id
AND icu.intime <= ft.test_time
)
LEFT JOIN ( -- Service in which the test was performed
SELECT *,
lead(curr_service, 1) OVER (PARTITION BY hadm_id ORDER BY transfertime ASC) next_service,
lead(transfertime, 1) OVER (PARTITION BY hadm_id ORDER BY transfertime ASC) next_transfertime
FROM (-- Union of services and admissions tables to have the complete history of
-- the patient's stay at the hospital
SELECT * FROM services
UNION
SELECT row_id, subject_id, hadm_id, admittime AS transfertime, '' AS prev_service,
'ADMISSION' AS curr_service FROM admissions
UNION
SELECT row_id, subject_id, hadm_id, disctime AS transfertime, '' AS prev_service,
'DISCHARGE' AS curr_service FROM admissions
) AS se
) AS s ON (
ft.hadm_id = s.hadm_id
AND ft.test_time BETWEEN s.transfertime AND s.next_transfertime
)
GROUP BY ft.subject_id,
ft.hadm_id, ft.spec_itemid, ft.spec_type_desc, ft.org_itemid, ft.org_name, ft.test_time,
a.admission_type,
a.admission_location,
a.insurance,
a.marital_status,
a.ethnicity,
patient_age,
gender,
s.curr_service

```

Appendix B. Description of the *Acinetobacter* and *S. Pneumoniae* Datasets

<i>Acinetobacter</i> Dataset					<i>S. pneumoniae</i> Dataset			
	NEGATIVE		POSITIVE		NEGATIVE		POSITIVE	
	N/Mean	Sd/%	N/Mean	Sd/%	N/Mean	Sd/%	N/Mean	Sd/%
spec_type_desc								
BLOOD CULTURE	4195	25.15%	15	9.43%	4154	25.00%	56	25.57%
BRONCHOALVEOLAR LAVAGE	1355	8.12%	20	12.58%	1355	8.15%	20	9.13%
SPUTUM	7784	46.67%	107	67.30%	7751	46.64%	140	63.93%
SWAB	3344	20.05%	17	10.69%	3358	20.21%	3	1.37%
admission_type								
ELECTIVE	1153	6.91%	7	4.40%	1155	6.95%	5	2.28%
EMERGENCY	15,134	90.74%	150	94.34%	15071	90.69%	213	97.26%
URGENT	391	2.34%	2	1.26%	392	2.36%	1	0.46%
admission_location								
CLINIC REFERRAL/PREMATURE	2827	16.95%	15	9.43%	2802	16.86%	40	18.26%
EMERGENCY ROOM ADMIT	9034	54.17%	86	54.09%	8971	53.98%	149	68.04%
OTHER	1796	10.77%	14	8.81%	1800	10.83%	10	4.57%
TRANSFER FROM HOSP/EXTRAM	3021	18.11%	44	27.67%	3045	18.32%	20	9.13%
insurance								
Government	372	2.23%	3	1.89%	366	2.20%	9	4.11%
Medicaid	1651	9.90%	14	8.81%	1632	9.82%	33	15.07%
Medicare	9711	58.23%	91	57.23%	9706	58.41%	96	43.84%
Private	4784	28.68%	50	31.45%	4764	28.67%	70	31.96%
Self Pay	160	0.96%	1	0.63%	150	0.90%	11	5.02%
marital_status								
DIVORCED	1163	6.97%	10	6.29%	1157	6.96%	16	7.31%
MARRIED	7591	45.52%	68	42.77%	7584	45.64%	75	34.25%
OTHER	1306	7.83%	19	11.95%	1294	7.79%	31	14.16%
SINGLE	4513	27.06%	42	26.42%	4487	27.00%	68	31.05%
WIDOWED	2105	12.62%	20	12.58%	2096	12.61%	29	13.24%
ethnicity								
BLACK/AFRICAN AMERICAN	1596	9.57%	19	11.95%	1601	9.63%	14	6.39%
HISPANIC OR LATINO	474	2.84%	5	3.14%	465	2.80%	14	6.39%
OTHER	1463	8.77%	13	8.18%	1460	8.79%	16	7.31%
UNKNOWN/NOT SPECIFIED	1290	7.73%	10	6.29%	1283	7.72%	17	7.76%
WHITE	11,855	71.08%	112	70.44%	11809	71.06%	158	72.15%
patient_age								
ADULT	8094	48.53%	77	48.43%	8033	48.34%	138	63.01%
ELDERLY	8584	51.47%	82	51.57%	8585	51.66%	81	36.99%
gender								
F	6655	39.90%	69	43.40%	6640	39.96%	84	38.36%
M	10,023	60.10%	90	56.60%	9978	60.04%	135	61.64%
min_white_blood_cells (numeric)	11.98	8.51	15.72	38.19	12.02	9.20	12.27	12.19
mean_white_blood_cells (numeric)	13.98	9.33	17.29	38.17	14.00	9.95	14.31	13.51
max_white_blood_cells (numeric)	16.10	11.54	19.06	38.23	16.13	12.02	16.56	15.27
min_lactate (numeric)	1.79	1.35	1.47	0.86	1.78	1.34	2.03	1.56
mean_lactate (numeric)	2.24	1.77	1.70	1.00	2.23	1.77	2.59	1.86
max_lactate (numeric)	2.78	2.52	1.94	1.27	2.77	2.51	3.30	2.46
icu_stay								
IN_ICU	9098	54.55%	106	66.67%	9110	54.82%	94	42.92%
LAST_48H	162	0.97%	4	2.52%	165	0.99%	1	0.46%
LAST_MONTH	632	3.79%	9	5.66%	640	3.85%	1	0.46%
LAST_WEEK	445	2.67%	6	3.77%	450	2.71%	1	0.46%
MORE_THAN_1_MONTH	148	0.89%	1	0.63%	149	0.90%	0	0.00%
NO_ICU	6193	37.13%	33	20.75%	6104	36.73%	122	55.71%
curr_service								
UNKNOWN	4905	29.41%	27	16.98%	4819	29.00%	113	51.60%
MED	5112	30.65%	65	40.88%	5133	30.89%	44	20.09%
OTHER	4783	28.68%	58	36.48%	4783	28.78%	58	26.48%
SURG	1878	11.26%	9	5.66%	1883	11.33%	4	1.83%
class								
NEGATIVE	16,678	100.00%	0	0.00%	16618	100.00%	0	0.00%
POSITIVE	0	0.00%	159	100.00%	0	0.00%	219	100.00%

References

1. Dowell, S.F. Seasonal Variation in Host Susceptibility and Cycles of Certain Infectious Diseases. *Emerg. Infect. Dis.* **2001**, *7*, 369–374. [CrossRef]
2. Imai, C.; Hashizume, M. A systematic review of methodology: Time series regression analysis for environmental factors and infectious diseases. *Trop. Med. Health* **2015**, *43*, 1–9. [CrossRef] [PubMed]
3. Richet, H. Seasonality in Gram-negative and healthcare-associated infections. *Clin. Microbiol. Infect.* **2012**, *18*, 934–940. [CrossRef] [PubMed]
4. Rodrigues, F.; de Luca, F.C.; da Cunha, A.R.; Fortaleza, C. Season, weather and predictors of healthcare-associated Gram-negative bloodstream infections: A case-only study. *J. Hosp. Infect.* **2019**, *101*, 134–141. [CrossRef]
5. Naumova, E.N. Mystery of Seasonality: Getting the Rhythm of Nature. *J. Public Health Policy* **2006**, *27*, 2–12. [CrossRef] [PubMed]
6. Schwab, F.; Gastmeier, P.; Meyer, E. The Warmer the Weather, the More Gram-Negative Bacteria—Impact of Temperature on Clinical Isolates in Intensive Care Units. *PLoS ONE* **2014**, *9*, e91105. [CrossRef]
7. Bhaskaran, K.; Gasparrini, A.; Hajat, S.; Smeeth, L.; Armstrong, B. Time series regression studies in environmental epidemiology. *Int. J. Epidemiol.* **2013**, *42*, 1187–1195. [CrossRef]
8. Christiansen, C.; Pedersen, L.; Sørensen, H.; Rothman, K. Methods to assess seasonal effects in epidemiological studies of infectious diseases—exemplified by application to the occurrence of meningococcal disease. *Clin. Microbiol. Infect.* **2012**, *18*, 963–969. [CrossRef]
9. Williams, D.J.; Zhu, Y.; Grijalva, C.G.; Self, W.H.; Harrell, F.E.; Reed, C.; Stockmann, C.; Arnold, S.R.; Ampofo, K.K.; Anderson, E.J.; et al. Predicting Severe Pneumonia Outcomes in Children. *PEDIATRICS* **2016**, *138*, e20161019. [CrossRef]
10. Steinhoff, M.C.; Walker, C.F.; Rimoin, A.W.; Hamza, H.S. A clinical decision rule for management of streptococcal pharyngitis in low-resource settings. *Acta Paediatr.* **2007**, *94*, 1038–1042. [CrossRef]
11. Godahewa, R.; Yann, T.; Bergmeir, C.; Petitjean, F. Seasonal Averaged One-Dependence Estimators: A Novel Algorithm to Address Seasonal Concept Drift in High-Dimensional Stream Classification. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020. [CrossRef]
12. Rocklöv, J.; Forsberg, B.; Meister, K. Winter mortality modifies the heat-mortality association the following summer. *Eur. Respir. J.* **2008**, *33*, 245–251. [CrossRef] [PubMed]
13. Sahota, H.; Barnett, H.; Lesosky, M.; Raboud, J.; Vieth, R.; Knight, J. Association of vitamin D-related information from a telephone interview with 25-hydroxyvitamin D. *Cancer Epidemiol. Biomarkers Prev.* **2008**, *17*, 232–238. [CrossRef] [PubMed]
14. Johnson, A.E.; Pollard, T.J.; Shen, L.; Lehman, L.W.H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L.A.; Mark, R.G. MIMIC-III, a freely accessible critical care database. *Sci. Data* **2016**, *3*, 160035. [CrossRef] [PubMed]
15. Cánovas-Segura, B.; Morales, A.; Juárez, J.M.; Campos, M. Seasonality in Infection Predictions Using Interpretable Models for High Dimensional Imbalanced Datasets. In Proceedings of the Artificial Intelligence in Medicine: 19th International Conference on Artificial Intelligence in Medicine, AIME 2021, Virtual Event, 15–18 June 2021; Tucker, A., Henriques Abreu, P., Cardoso, J., Pereira Rodrigues, P., Riaño, D., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2021; Volume 12721, pp. 152–156. [CrossRef]
16. Krawczyk, B.; Minku, L.L.; Gama, J.; Stefanowski, J.; Woźniak, M. Ensemble learning for data stream analysis: A survey. *Inf. Fusion* **2017**, *37*, 132–156. [CrossRef]
17. Muthukrishnan, S. Data streams: Algorithms and applications. *Found. Trends Theor. Comput. Sci.* **2005**, *1*, 117–236. [CrossRef]
18. Domingos, P.; Hulten, G. Mining High-Speed Data Streams. In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, USA, 20–23 August 2000; pp. 71–80. [CrossRef]
19. Schlimmer, J.C.; Granger, R.H., Jr.; Granger, R.H. Incremental learning from noisy data. *Mach. Learn.* **1986**, *1*, 317–354. [CrossRef]
20. Tsybal, A. *The Problem of Concept Drift: Definitions and Related Work*; Technical Report; Department of Computer Science, Trinity College: Dublin, Ireland, 2004.
21. Widmer, G.; Miroslav, K. Learning in the Presence of Concept Drift and Hidden Contexts. *Mach. Learn.* **1996**, *23*, 69–101. [CrossRef]
22. Barddal, J.P.; Gomes, H.M.; Enembreck, F.; Pfahringer, B. A survey on feature drift adaptation: Definition, benchmark, challenges and future directions. *J. Syst. Softw.* **2017**, *127*, 278–294. [CrossRef]
23. Jenkins, D.A.; Sperrin, M.; Martin, G.P.; Peek, N. Dynamic models to predict health outcomes: Current status and methodological challenges. *Diagn. Progn. Res.* **2018**, *2*, 23. [CrossRef]
24. Alder, S. De-identification of Protected Health Information: How to Anonymize PHI. Available online: <https://www.hipaajournal.com/de-identification-protected-health-information> (accessed on 25 May 2023).
25. Breiman, L. Bagging Predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
26. Caruana, R.; Lou, Y.; Gehrke, J.; Koch, P.; Sturm, M.; Elhadad, N. Intelligible Models for HealthCare. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Sydney, NSW, Australia, 10–13 August 2015. [CrossRef]
27. Lipton, Z.C. The Mythos of Model Interpretability. Available online: <https://arxiv.org/abs/1606.03490> (accessed on 25 May 2023).
28. Tu, J.V. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *J. Clin. Epidemiol.* **1996**, *49*, 1225–1231. [CrossRef]

29. Hastie, T.; Tibshirani, R. Generalized Additive Models. In *Encyclopedia of Statistical Sciences*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2006. [\[CrossRef\]](#)
30. Steyerberg, E. *Clinical Prediction Models*; Statistics for Biology and Health; Springer: New York, NY, USA, 2009.
31. Kuhn, M.; Johnson, K. *Applied Predictive Modeling*; Springer: New York, NY, USA, 2013.
32. Hosmer, D.W.; Lemeshow, S. *Applied Logistic Regression*; Wiley: Hoboken, NJ, USA 2000.
33. Yu, L.; Liu, H. Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. In Proceedings of the 20th International Conference on Machine Learning (ICML), Washington, DC, USA, 21–24 August 2003 ; pp. 1–8.
34. Tibshirani, R. Regression Selection and Shrinkage via the Lasso. *J. R. Stat. Soc. Ser. B (Methodol.)* **1996**, *58*, 267–288. [\[CrossRef\]](#)
35. Branco, P.; Torgo, L.; Ribeiro, R.P. A survey of predictive modeling on imbalanced domains. *ACM Comput. Surv.* **2016**, *49*, 31. [\[CrossRef\]](#)
36. Gama, J.; Žliobaitė, I.; Bifet, A.; Pechenizkiy, M.; Bouchachia, A. A survey on concept drift adaptation. *ACM Comput. Surv. (CSUR)* **2014**, *46*, 44. [\[CrossRef\]](#)
37. Wang, H.; Fan, W.; Yu, P.; Han, J. Mining concept-drifting data streams using ensemble classifiers. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003; Volume 2, pp. 226–235.
38. Tsymbol, A.; Pechenizkiy, M.; Cunningham, P.; Puuronen, S. Handling local concept drift with dynamic integration of classifiers: Domain of antibiotic resistance in nosocomial infections. In Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems, Salt Lake City, UT, USA, 22–23 June 2006; Volume 2006, pp. 679–684.
39. Schwab, F.; Gastmeier, P.; Hoffmann, P.; Meyer, E. Summer, sun and sepsis—The influence of outside temperature on nosocomial bloodstream infections: A cohort study and review of the literature. *PLoS ONE* **2020**, *15*, e0234656. [\[CrossRef\]](#)
40. Ripley, B.D.; Hjort, N.L. *Pattern Recognition and Neural Networks*, 1st ed.; Cambridge University Press: New York, NY, USA, 1995.
41. Friedman, J.; Hastie, T.; Tibshirani, R. Regularization Paths for Generalized Linear Models via Coordinate Descent. *J. Stat. Softw.* **2010**, *33*, 1–22. [\[CrossRef\]](#)
42. Hastie, T.; Qian, J.; Tay, K. An Introduction to Glmnet. Technical Report. Available online: <https://glmnet.stanford.edu/articles/glmnet.html> (accessed on 25 May 2023).
43. Kuhn, M.; Weston, S.; Culp, M.; Coulter, N.; Quinlan, R. C50: C5.0 Decision Trees and Rule-Based Models. Available online: <https://cran.r-project.org/web/packages/C50/index.html> (accessed on 25 May 2023).
44. Novoselova, N.; Wang, J.; Pessler, F.; Klawonn, F. Biocomb: Feature Selection and Classification with the Embedded Validation Procedures for Biomedical Data Analysis. Available online: <https://cran.r-project.org/web/packages/Biocomb/index.html> (accessed on 25 May 2023).
45. Yu, L.; Liu, H. Efficient Feature Selection via Analysis of Relevance and Redundancy. *J. Mach. Learn. Res.* **2004**, *5*, 1205–1224.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.