Contents lists available at ScienceDirect

Artificial Intelligence In Medicine

journal homepage: www.elsevier.com/locate/artmed

Research paper

Dimensionality reduction and ensemble of LSTMs for antimicrobial resistance prediction

Àlvar Hernàndez-Carnerero [a,*], Miquel Sànchez-Marrè [a], Inmaculada Mora-Jiménez [b], Cristina Soguero-Ruiz [b], Sergio Martínez-Agüero [b], Joaquín Álvarez-Rodríguez [c]

[a] Department of Computer Science (CS), Intelligent Data Science and Artificial Intelligence Research Center (IDEAI-UPC), Universitat Politècnica de Catalunya (UPC), Campus Nord, Edif. Omega, C. Jordi Girona, 1-3, 08034 Barcelona, Spain

[b] Department of Signal Theory and Communications, Telematics and Computing Systems, Rey Juan Carlos University, Fuenlabrada 28943, Spain

[c] Intensive Care Department, University Hospital of Fuenlabrada, Fuenlabrada 28943, Spain

ARTICLE INFO

Keywords:
Data-driven dimensionality reduction
LSTM-based prediction
Multivariate time series
Antimicrobial resistance prediction
Intensive Care Unit
Pseudomonas aeruginosa

ABSTRACT

Bacterial resistance to antibiotics has been rapidly increasing, resulting in low antibiotic effectiveness even treating common infections. The presence of resistant pathogens in environments such as a hospital Intensive Care Unit (ICU) exacerbates the critical admission-acquired infections. This work focuses on the prediction of antibiotic resistance in *Pseudomonas aeruginosa* nosocomial infections at the ICU, using Long Short-Term Memory (LSTM) artificial neural networks as the predictive method. The analyzed data were extracted from the Electronic Health Records (EHR) of patients admitted to the University Hospital of Fuenlabrada from 2004 to 2019 and were modeled as Multivariate Time Series. A data-driven dimensionality reduction method is built by adapting three feature importance techniques from the literature to the considered data and proposing an algorithm for selecting the most appropriate number of features. This is done using LSTM sequential capabilities so that the temporal aspect of features is taken into account. Furthermore, an ensemble of LSTMs is used to reduce the variance in performance. Our results indicate that the patient's admission information, the antibiotics administered during the ICU stay, and the previous antimicrobial resistance are the most important risk factors. Compared to other conventional dimensionality reduction schemes, our approach is able to improve performance while reducing the number of features for most of the experiments. In essence, the proposed framework achieve, in a computationally cost-efficient manner, promising results for supporting decisions in this clinical task, characterized by high dimensionality, data scarcity, and *concept drift*.

1. Introduction

After the discovery of penicillin by Alexander Fleming in 1928, which initiated the development and administration of antimicrobial drugs [1], bacteria have been evolving to resist antimicrobials. Though this evolution is naturally produced by bacteria mutations, it has been increasingly accelerated by the selective pressure generated by the widespread, and not always appropriate, use of antibiotics [2]. Bacterial resistance has grown to a point where current antibiotics may be no longer able to treat common infections, becoming antimicrobial resistance a global public health crisis [3].

In a hospital's Intensive Care Unit (ICU), due to factors such as the serious medical conditions of the patients with compromised immune systems, the cross-transmission among patients, and the high use of antimicrobial drugs, there is a growing risk for nosocomial infections, i.e. infections acquired during the ICU stay [4,5]. Furthermore, the presence of resistant pathogens increases the impact of these infections. The current study is focused on *Pseudomonas aeruginosa* because it is one of the most common bacteria associated with nosocomial infections in the ICUs [6].

In order to treat an infection, clinicians usually carry out a culture to identify the causative bacteria. Afterward, a susceptibility test or antibiogram is normally performed for each particular bacterium, which informs about bacterium susceptibility to a group of tested antibiotics. The antibiogram result for a particular bacterium is a set of pairs antibiotic tested — susceptibility of the bacterium to the antibiotic tested [7]. The antibiogram is used as a guide in selecting targeted antibiotic therapy, and also to monitor the bacterial resistance in the ICU as a whole [7]. The antibiogram results usually require 48 h. Owing to the critical condition of patients in the ICU, the early availability of the antibiogram result is crucial to provide the proper treatment for

[*] Corresponding author.
E-mail address: alvar.hernandez@upc.edu (À. Hernàndez-Carnerero).

https://doi.org/10.1016/j.artmed.2023.102508
Received 8 February 2022; Received in revised form 21 November 2022; Accepted 4 February 2023
Available online 8 February 2023

infectious disease patients [8]. To support clinical decisions related to *Pseudomonas aeruginosa* bacterium, we propose in this work the use of state-of-the-art Machine Learning (ML) techniques to obtain an early prediction of the antibiogram results.

Studies have proposed the use of ML for generating rules to improve the antibiotics administration programs, based on past recommendations for the patient's treatment [9]. Also, ML coupled with Feature Selection (FS) methods has been used to predict the general trend of antimicrobial resistance in a hospital [10]. Regarding the prediction of antimicrobial resistance of particular strains using ML, different sources of data have been considered. For instance, recently bacteria *Whole Genome Sequences* have been used for resistance prediction [11–13]. However, the present study makes use of Electronic Health Records (EHR) of ICU patients for the antibiogram result prediction, since EHR data are easier to obtain than genomics data. In fact, EHR data have been used for predicting antimicrobial resistance in previous studies, applying a number of techniques such as regression-based methods [14,15], Random Forest [16], Naïve Bayes, Decision Tree, K-Nearest Neighbors [17], Support Vector Machines, Multi-Layer Perceptron [18–20], and Long Short-Term Memory (LSTM) [21]. The study [17] introduces the *concept drift* in this specific field, arising when predicting antimicrobial resistance with EHR data. It refers to a phenomenon in which the data distribution changes over time, and a set of techniques such as *windowing* and dynamic selection of models are proposed to get around this challenge. In [15,16,20], the intrinsic temporal nature of the task is considered, and the methods are trained on the first instances temporally ordered (training set) and evaluated on the most recent instances (test set). In addition, the *windowing* technique is used to overcome the *concept drift* in [15,16]. Several studies applied feature importance methods and identified highly relevant EHR variables informing about the results of previous susceptibility tests associated with the patient for whom the antibiogram result is to be predicted [16,20]. Furthermore, in [15] it is suggested that information linked to the rest of the ICU patients could be useful to predict bacterial resistance of cultures for a particular patient in the ICU.

The data analyzed in the current study have been also used in [15,16,19,21–24], although considering different time frames, features, and instance structure. Recently, the work in [21] suggested tackling the problem by considering instances as time series modeling ICU patients because of the temporal dynamics of the EHR data. Following a similar idea, the current study represents instances as individual cultures characterized by *multivariate time series*.

In the literature, many approaches have been proposed for the classification of instances characterized by time series. These approaches can be summarized in four main paradigms: the model-based or generative methods, the distance-based methods, the feature-based methods, and the end-to-end methods [25,26]. The *model-based methods* firstly build a model for the instances associated with a particular class, commonly using AutoRegressive (AR) method [27,28]. Secondly, the new instance (based on time series) is evaluated with the generated models to find the most probable class for the new instance. However, the AR method can only deal with time series that satisfy the stationary assumption. The *distance-based methods* couple a classifier algorithm with a distance function measuring the difference between two time series (two instances). A widespread model using this methodology is Nearest Neighbors (NN) with the Dynamic Time Warping (DTW) distance [29]. Unfortunately, a drawback of DTW is that, in its basic form, its calculation is quadratic on the time complexity [30]. The *feature-based methods* firstly reduce the dimensionality of the time series by extracting a set of features and afterward use these features to train a classifier. An algorithm achieving good performance using this paradigm is HIVE-COTE [31], an ensemble of 37 classifiers with different feature representations including a Hierarchical Vote system. The main disadvantage of HIVE-COTE is the very high computational cost, making it impractical on real big data mining tasks. Note that the performance of the feature-based methods are highly dependent on the feature extraction procedure. Finally, unlike the feature-based methods, the *end-to-end methods* directly deal with the raw time series for classification without prior processing. These methods use Artificial Neural Networks (ANNs) and have gained relevance in classification tasks involving time series because of their promising performance, becoming the state-of-the-art in a variety of tasks (for instance exoplanet transit detection [32]). Among the different architectures, the Recurrent Neural Network (RNN) [33] design allows it to work with time series data in a natural way thanks to their feedback connections. In the current study, we use LSTMs because they are a type of RNNs capable to handle an arbitrary number of instants of time in the time series. Also, LSTMs are the state-of-the-art for classification using EHR data with time series [21,34]. Because of the natural temporal ordering in which cultures are performed, instances in training are the oldest cultures in the sequence and test instances are always the most recent ones in the sequence. Furthermore, the relevant information of previous susceptibility test results and information about co-admitted ICU patients is considered and adapted in the time series scheme. Co-admitted patients are those who stay in the ICU with the patient p whose culture is being considered. The current study devises a dimensionality reduction method based on the use of LSTM, and does not require expert knowledge. It consists of two stages: In the first stage, three feature importance techniques are adapted to the handled data and the importance provided by the three techniques is averaged. These features are sorted by importance and, in the second stage, an algorithm is proposed to calculate the number of selected features. In addition, in the present study, an ensemble of ANNs is applied with the aim of improving performance and decreasing variance.

The rest of the paper is as follows. Section 2 describes the data considered in this study. In Section 3 the methods and the experimental setting are explained, whereas the Results are presented in Section 4. Finally, conclusions and future work are discussed in Section 5.

2. Data considered

Data considered in this study have been extracted from the EHR in the ICU service of the University Hospital of Fuenlabrada (UHF). They were provided as an anonymized data set and correspond to a period of 15 years (from July 2004 to May 2019). Our data set contains 764 cultures in which the *Pseudomonas aeruginosa* bacteria was detected. These cultures belong to 277 patients, implying that some patients have assigned more than one culture. Every culture has the associated antibiogram results.

The study focuses on nosocomial infections, which are those acquired during the ICU stay [35]. Formally, they are defined as infections arising after 48 h of hospital admission [17] (the ICU admission, in our case). Because of that, all the considered instances are associated with patients such that their ICU stay is longer than 48 h. In this way, each instance represents a culture of a particular patient in which the *Pseudomonas aeruginosa* bacterium has been detected. Regarding the treatment of this bacterium, six families of antibiotics are especially relevant: Aminoglycosides (AMG), Carbapenems (CAR), 4th Generation Cephalosporins (CF4), Extended-spectrum penicillins (PAP), Polymyxins (POL), and Quinolones (QUI) [36,37]. Focusing on these families, the present work considers six binary targets, one for each family. Every target indicates whether the culture's bacterium is susceptible or resistant to the respective antimicrobial family. Predictive data of the instances contain information registered in the EHR of the corresponding patient, represented as a multivariate time series. A multivariate time series is a sequence of data where there is more than one observation for each time step (the temporal distance among time steps is one day in our data set).

In this work, fixed-length sequences of 30 days are used to train the network. Since instances are characterized by time series with potentially different lengths depending on the patient's ICU stay, it is needed to truncate or pad each sequence so that they all have the same

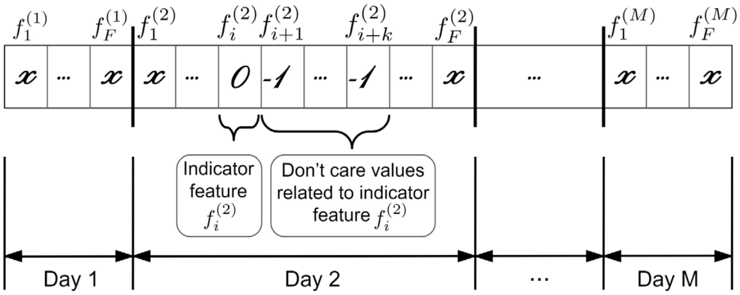


Fig. 1. Scheme of one instance with F features and M days, including the representation of the indicator feature $f_i^{(2)}$ (corresponding to Day 2) and its associated k features (from $f_{i+1}^{(2)}$ to $f_{i+k}^{(2)}$) filled with *don't care values*.

length. As previously mentioned, the last day linked to a particular instance corresponds to the day the particular culture is performed. In order to truncate the time series of an instance containing more than 30 days, the 29 days immediately previous to the day the culture is performed are considered. In the case of instances containing less than 30 days, pre-padding with '−1' is used to fill the values of the time series in these days.

As a result, data characterizing every instance can be represented as an array of length *number_of_days* × *number_of_features*, with the $number\_of\_features$ detailed in Section 2.1. Instances consider patient data from the day of the first ICU admission to the day the culture is predicted.

2.1. Features of the time series

In total, the number of features used is 398. This high number is the result of using One-Hot Encoding to represent those variables which originally are categorical, obtaining a new feature for each category (modality) of the original variable. The variables with several modalities are indicated as *feature_name-[modality]*. The variables and corresponding modalities are listed in the Annex. All features consider information always previous to the date and time when the culture is performed.

Owing to the nature of the data and their encoding, there is a high number of unknown values. Unknown values encompass both *missing* and *don't care values*. While *missing* values do not provide information about the features, *don't care values* tells us that the feature values are irrelevant to the instance's class [38]. In the medical field, it is relevant to differentiate *missing* from *don't care values*, since not all the features may be related to the patient diagnosis [38]. In the current study, the values of the subsets of features *"Admissions information"*, *"Culture and antibiogram information"* and *"ICU admissions information"* (see Sections 2.1.2 and 2.1.3), will be *don't care values* when the patient whose antibiogram result is being predicted is not currently staying in the ICU, or the patient has not performed any culture, or there are not any co-admitted patients, respectively. Since the feature values are normalized between 0 and 1, the value assigned to *don't care values* is −1, outside the normalized range to distinguish them from actual values. For an explicit representation of these values, one indicator feature is created for each of the three subsets of features previously mentioned. They are named *admission_indicator*, *cultures_indicator*, and *icu_admission_indicator*, respectively, which denote whether the rest of the features in the same subset of features (see Sections 2.1.2 and 2.1.3) are coded as *don't care values* or not. Fig. 1 depicts the structure of one instance in the data set, with the representation of an indicator feature and its respective *don't care values*.

The features in the present study are organized in three broad groups, named *temporal features*, *patient features*, and *ICU features*. They are described below.

2.1.1. Temporal features

For each particular day d, some *temporal features* are considered. Features such as the complete date or the year are not used, since they introduce an explicit ordering among instances, and previous studies [15,16] have shown that their use worsens the results obtained.

- **day_week** specifies the day of the week with an integer from 0 to 6 representing days from Monday to Sunday, respectively.
- **day_month** represents the day of the month.
- **month** represents the month of the year with an integer from 0 to 11 representing months from January to December.

2.1.2. Patient features

The *patient features* just consider the information of the patient whose culture and respective antimicrobial resistance, is being predicted, named patient p. Each value of these features makes reference to a specific day d. For a clearer explanation, *patient features* are divided into four sets of features (next described): *"Admissions information"*, *"Culture and antibiogram information"*, *"Antibiogram results"*, and *"Treatment information"*.

Admissions information takes into account data about the admissions of the patient p. The following features are created:

- **admission_indicator** denotes whether the patient p is staying in the ICU on day d or not, with a 1 and a 0 respectively. In the case the patient is not staying in the ICU, the rest of the features considered in *"Admissions information"* set are *don't care values*, represented by −1.
- **origin-[modality]**: Its value is 1 if the ICU stay starts on the day d and the clinical origin of the patient is the one indicated by the modality of the feature. Otherwise, its value is 0.
- **destination-[modality]**: A value of 1 denotes that the stay ends on the day d and the clinical destination of the patient is the one indicated by the modality of the feature. Otherwise, the 0 value is used.
- **reason_admission-[modality]**: Its value is 1 if the reason for admission for patient p is the one indicated by the modality of the feature. Its value is 0 otherwise.
- **patient_category-[modality]**: There are two types, medical and surgical, with the value 1 indicating the appropriate category for the patient p. Otherwise, a 0 is considered.
- **age**: Its value is the age of the patient p on day d.
- **gender**: Its value is 0 or 1 if the patient p is female or male, respectively.

Culture and antibiogram information includes data of the previous cultures containing *Pseudomonas aeruginosa* for the patient p. Regarding the day when the prediction is performed (the last day considered in the time sequence), just the culture which result is being predicted is considered, regardless there were several cultures on the same day:

- **cultures_indicator** denotes whether the patient p has had at least one culture performed on the considered day d or not, with a 1 and a 0 respectively. When no culture has been performed, the rest of the features considered in *"Culture and antibiogram information"* set are *don't care values*, represented by -1.
- **culture_type- [modality]** : The features identified as "culture_type-[modality]" refer to the kind of tissue sample. Some of them are: "*blood*", "*urine*", "*exudate_wound*", "*abscess*", among others. The value registered for each culture_type-[modality] refers to the number of cultures carried out to the patient p in the considered day d.
- **culture_type_grouped- [modality]** : number of cultures performed to the patient p in the considered day d with the type of culture grouped indicated by the feature.
- **culture_type_grouped_2- [modality]** : number of cultures performed to the patient p in the considered day d with the type of culture grouped 2 indicated by the feature.
- **antibiogram_antibiotic- [modality]** : number of antibiograms linked to the patient p in the considered day d such that the antibiotic indicated by the feature has been tested for resistance.
- **antibiogram_family- [modality]** : number of antibiotics tested in antibiograms belonging to the family indicated by the feature, for the patient p and considered day d.

Antibiogram results express the results of the cultures containing *Pseudomonas aeruginosa* and antibiograms of the patient *p*. The following features are created:

- **pseudomonas_aeruginosa_detected**: it is 1 if there is at least one culture containing *Pseudomonas aeruginosa* for patient p and day d. Otherwise, it is 0.
- **resistance- [modality]** : refers to results of the antibiograms carried out 48 h before day d. Three values are possible for every antimicrobial family af: 0 denotes that it has been found susceptible bacteria but no resistant bacteria; 1 indicates that there are results with resistant bacteria; -1 is used to code that the af family has not been tested.

Treatment information details the antibiotics administered to the patient *p*, and whether they have received mechanical ventilation. The following features are created:

- **antibiotics- [modality]** : It is 1 if the antibiotic indicated by the modality is administered to patient *p*, and 0 otherwise.
- **mechanic_ventilation**: It is 1 if the patient receives mechanic ventilation, and 0 otherwise.

2.1.3. ICU features

The *ICU features* consider information of *co-admitted patients* (*ca-p*) to the patient p whose culture is being considered (see definition of co-admitted patients in Section 1). Again, each value makes reference to a particular day d. *ICU features* are divided into four sets, described below: *"Number of co-admitted patients"*, *"ICU admission information"*, *"ICU antibiogram results"*, and *"ICU treatment information"*.

The set **number of co-admitted patients** contains just the feature **co-admitted_patients**, representing the number of *ca-p* in a particular day d.

The group designated as **ICU admission information** takes into account data about admissions of *ca-p*. The following features are created:

- **icu_admission_indicator** denotes whether there is at least one *ca-p* on day d (value 1) or not (value 0). When there are no *ca-p*, the rest of the features in the *"ICU admissions information"* set are *don't care values*, represented by -1.
- **icu_origin- [modality]** : number of *ca-p* that start their ICU stay in day d and their clinical origin is the one indicated by the feature modality.
- **icu_destination- [modality]** : number of *ca-p* finishing their stay in day d, being their clinical destination the one indicated by the feature modality.
- **icu_reason_admission- [modality]** : number of *ca-p* whose reason of admission matches with the one indicated by the feature modality.
- **icu_patient_category- [modality]** : number of *ca-p* whose patient category matches with the one indicated by the feature modality.
- **icu_age**: mean age of the *ca-p* in day d.
- **icu_gender- [modality]** : number of *ca-p* with the gender indicated by the feature modality.

The set **ICU antibiogram results** refer to the cultures containing *Pseudomonas aeruginosa* and antibiograms assigned to the *ca-p*. The following features are created:

- **icu_pseudomonas_aeruginosa_detected**: number of *ca-p* for whom at least one culture containing the *Pseudomonas aeruginosa* bacterium is detected in the considered day d.
- **icu_resistance- [modality]** : its value is calculated as follows. First, for each *ca-p*, a value is calculated following the criterion presented in **resistance- [modality]** . Then, if the values for all the *ca-p* patients are -1, the value of **icu_resistance- [modality]** is also set to -1. Otherwise, it is the sum of all the values different from -1.

The group named **ICU treatment information** details the antibiotics administered to the *ca-p*, and whether they have received mechanical ventilation. The following features are created:

- **icu_antibiotics- [modality]** : its value is 1 if there is at least one *ca-p* for whom the antibiotic indicated by the modality of the feature has been administered in the considered day d. Otherwise, its value is 0.
- **icu_mechanic_ventilation**: number of *ca-p* who are receiving mechanical ventilation in the considered day d.

2.2. Data analysis

Six different subsets are generated, one for each target (antibiotic family). Since the susceptibility tests may have not been carried out simultaneously for the six antibiotic families, each subset can have a different number of cultures. Thus, the number of instances is 755, 643, 749, 749, 483, and 708 for the AMG, CAR, CF4, PAP, POL, and QUI subsets, respectively. Fig. 2 depicts the temporal evolution of the susceptibility to each of the antibiotics families considered. In general, it is observed that the fraction of resistant instances increases as time evolves. It is also observed that the number of cultures performed is lower in the most recent years. Finally, some particularities are found in specific antibiotic families. For instance, there are no instances in the POL family until 2007, probably because until then its use was not necessary to treat *Pseudomonas aeruginosa* infections. On the other hand, the QUI family has a low number of instances in 2018 and none in 2019. This may be caused because the QUI family is no longer tested in antibiograms because of the low effectiveness of this antibiotic to deal with *Pseudomonas aeruginosa* (note the high rate of resistant instances from 2014).

Regarding the unknown values, our data set includes both *missing* and *don't care* values, *missing* values are found in *resistance-[modality]* and *icu_resistance-[modality]* features. The -1 value is used to encode those features for which the corresponding antimicrobial family has not been tested (i.e., there is no susceptibility result). Table 1 shows the percentage of days with unknown values for each feature or set of features. It is observed that the number of *don't care* values in *"Admissions information"* is relatively low taking into account the data sparsity. Percentages in *"ICU admissions information"* are similar to those in *"Admissions information"*, since patients for whom the culture

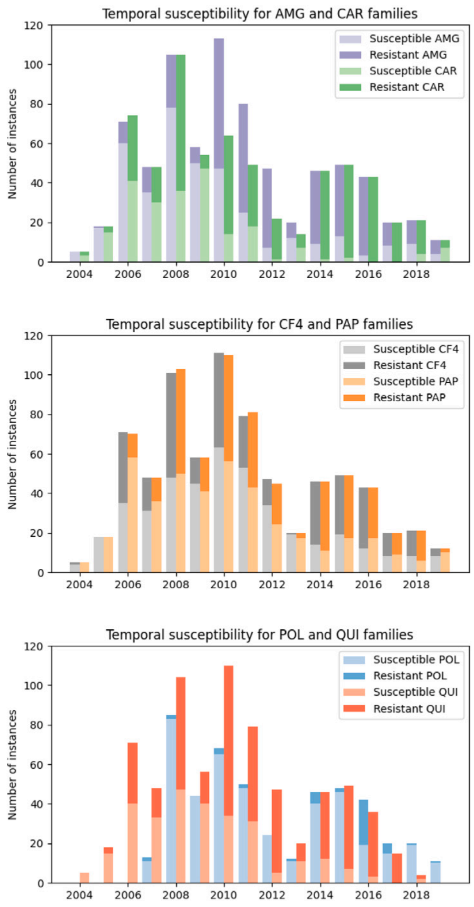


Fig. 2. Evolution in the number of susceptible and resistant instances for each antimicrobial family. Top panel comprises AMG and CAR data subsets. Middle panel refers to CF4 and PAP. Bottom panel includes POL and QUI.

Table 1
Percentage of days (considering 30 days) with unknown values for each feature (row) and data set (column). *Adm.* stands for *"Admission information"*, *ICU Adm.* stands for *"ICU admissions information"*, *Cul.* stands for *"Culture and antibiogram information"*, *Res. [...]* stands for *resistance-[modality]* and *ICU Res. [...]* stands for *icu_resistance-[modality]* (co-admitted patients).

	AMG	CAR	CF4	PAP	POL	QUI
Adm.	38.7	37.1	38.6	38.7	37.6	38.3
ICU Adm.	38.7	37.1	38.6	38.7	37.6	38.3
Cul.	90.1	89.7	90.1	90.1	89.4	89.9
Res. AMG	94.0	93.6	94.0	94.0	93.3	93.8
Res. CAR	94.7	94.1	94.7	94.7	94.1	94.6
Res. CF4	94.0	93.6	94.0	94.0	93.3	93.8
Res. PAP	94.0	93.7	94.0	94.0	93.4	93.9
Res. POL	96.0	95.8	96.0	96.0	94.5	95.9
Res. QUI	94.2	93.8	94.2	94.2	93.6	93.8
ICU Res. AMG	91.5	91.6	91.5	91.5	90.8	91.3
ICU Res. CAR	92.7	92.5	92.7	92.7	92.2	92.6
ICU Res. CF4	91.5	91.6	91.5	91.5	90.9	91.3
ICU Res. PAP	91.5	91.6	91.5	91.5	90.9	91.3
ICU Res. POL	94.3	94.3	94.3	94.3	93.0	94.3
ICU Res. QUI	91.8	91.9	91.8	91.8	91.2	91.4

results are predicted are always with at least one co-admitted patient during their ICU stay. *"Culture and antibiogram information"* has a much higher percentage of *don't care values*. This happens because cultures are not performed everyday. The features named *resistance-[modality]* obtain even higher percentages of unknown values than *"Culture and antibiogram information"* because not all the antimicrobial families are always tested in the same culture. On the other hand, features like *icu_resistance-[modality]* also get higher percentages of unknown values than *"Culture and antibiogram information"*, but they are slightly lower than those of *resistance-[modality]* because *icu_resistance-[modality]* features refer to co-admitted patients, and it is likely that they count with antibiograms. several patients, increasing the chances of having different antibiotics tested.

3. Methods and experimental setting

The problem at hand is tackled as a multivariate time series classification task suffering from high dimensionality, with 398 features and a relatively low number of instances, ranging from 483 to 755. Because of that, dimensionality reduction is applied to mitigate the model complexity and potentially improve the model performance.

The selection of features is performed independently for each antimicrobial family data set, considering wrapper FS methods for two reasons [39]. The first is that they usually achieve better predictive performance than other methods (such as filter FS methods), although they are computationally more expensive. The second reason is that, since wrapper methods take into account the same approach later used for classification, they can offer a better performance than the one provided by filtering feature selection methods.

In the current work, we first propose to calculate indices to measure the *feature importance* by adapting some wrapper methods in the literature; then, we propose an algorithm using these indices to carry out the *selection of the most relevant features*. The proposed FS algorithm is entirely data-driven, not requiring expert knowledge. This allows us to apply it to any field of study.

The following sections describe how the LSTM neural networks are built so that it can later be used by the wrapper feature importance methods (Section 3.1). Our proposal for feature importance is detailed in Section 3.2, and the proposed FS algorithm is described in Section 3.3. With the aim of reducing the performance variability caused by data scarcity, an ensemble of LSTM is considered in Section 3.4.

3.1. Artificial neural network architecture

The LSTM is a type of RNN able to work with sequential data or time-series data. Unlike traditional feed-forward ANNs, RNNs have feedback connections that enable using the information of past inputs when generating an output [40]. Nevertheless, classical RNNs face the problem of *vanishing gradients* and *exploding gradients* using backpropagation, and they are not capable of learning when tackling sequences with more than 5–10 time steps up to the target prediction [41]. The ability of LSTMs to forget and keep information makes them robust to the vanishing gradient problem, allowing them to handle an arbitrary number of time steps. This is possible due to the cell state and the operations carried out by the gates in the LSTM cells [41], depicted in Fig. 3. The cell state can be seen as the "memory" of the network. The forget gate consists of a sigmoid layer and decides what information should be removed from the cell state. The input gate (composed of a sigmoid layer), together with the tanh layer, store new information in the cell state. The output gate, which consists of a sigmoid layer, decides what parts of the cell state are outputted.

As described in Section 2, this work considers six families of antibiotics. A binary classification task (susceptible or resistant) is performed with six LSTM models, each one independently trained with a different antimicrobial family data set. The network architecture, depicted in Fig. 4, contains two stacked LSTM layers and 100 units per layer, which were empirically selected. More specifically, the shape of the input layer is 30 (time steps) × 398 (features). The LSTM layer has 100 units, each using the Rectified Linear Unit (ReLU) activation function instead of the default tanh function depicted in Fig. 3, outputting a different value for each time step in the input data (this allows the output to have the appropriate shape so that it can be fed to the next

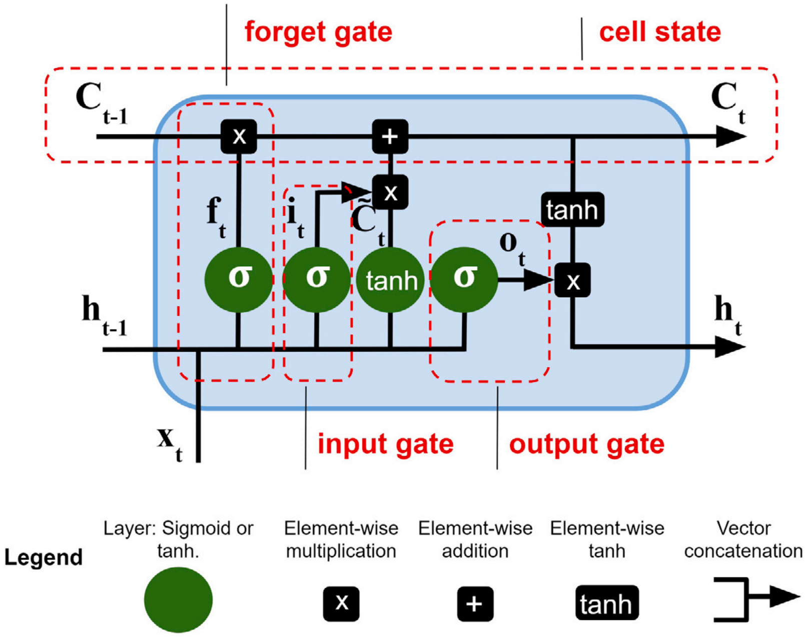


Fig. 3. Conventional LSTM cell diagram with usual activation functions. (Developed by the authors).

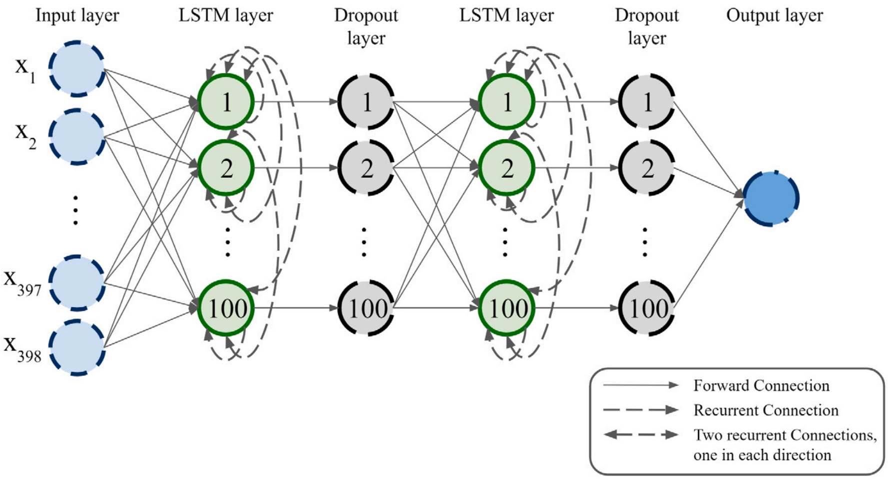


Fig. 4. Diagram of the LSTM architecture used in the experiments.

LSTM layer). The ReLU activation function is broadly used in ANNs because the model using ReLU is generally less computationally costly to train and usually achieves equal or better performance than other activation functions [42]. In addition, the ReLU can avoid the vanishing gradient problem, which is an advantage over the tanh function [43]. A dropout layer with rate of 0.2 is applied on the 100 units to mitigate overfitting [44]. Next, a new LSTM layer, again with 100 units and ReLU activation functions is considered, now returning a single value per unit (an output from the end of the sequence). A dropout layer, as before, with 100 units and a rate of 0.2 follows this new LSTM layer. Finally, the output layer consists of one sigmoid activation function to predict the binary target.

Instances in every data set are temporally ordered (from the oldest to the most recent ones) by the date of the culture result being predicted, and they are split in the training, validation, and test sets. Afterward, instances in each set are randomly shuffled. In this way, the model performance is evaluated in the same way as it would be done in the real setting, that is, the instances in the training set are previous in time to the instances in the validation set, and validation instances are temporally previous to instances in the test set. *Validation* is applied in all the experiments [45]. Also, in order to keep the independence among sets, the patients considered in the training set are discarded from the patients included in the validation and test sets. Table 2 details the number of instances per class (susceptible/resistant) on every data set for the training, validation and test sets. Data from 2004 to 2012 were considered for the training set. Instances in years from 2013 to 2019 were split in two parts: first half for the validation set and second half for the test set. Note that the sum of instances per data set is lower than the total indicated in Section 2.2 since instances linked to patients considered in the training set were discarded from the validation and test sets.

The loss function *Binary Cross Entropy* was optimized using the Root Mean Square Propagation (RMSprop) algorithm, due to its good performance on temporal series prediction [46]. The considered learning rate was set to 10^{-4}, with a decay of 10^{-5} and 20 epochs for training. As illustrated in Fig. 5 for the AMG family, the validation loss increases from epoch 20, while the training loss continues decreasing, revealing overfitting on the training data. Though the values shown in Fig. 5 correspond to the AMG family data set, the behavior is very similar for the rest of the families.

Table 2
Number of instances per antibiotic family for the training, validation and test sets. *Sus./Res.* refers to susceptible/resistant instances. Years from 2004 to 2012 are considered in the training set. The validation and test sets contain the first and second half of instances from 2013 to 2019, respectively.

	Training		Validation		Test	
	Sus.	Res.	Sus.	Res.	Sus.	Res.
AMG	324	221	30	73	28	76
CAR	205	234	10	90	11	90
CF4	331	207	49	55	37	67
PAP	331	207	42	62	43	61
POL	275	9	90	8	67	31
QUI	250	288	28	55	7	77

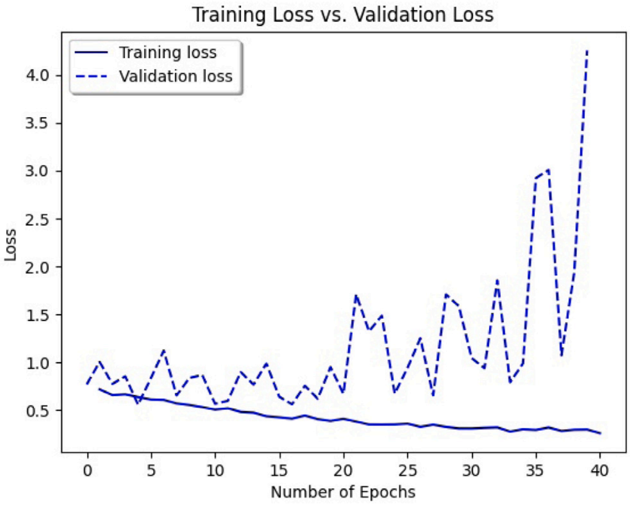


Fig. 5. Evolution of training and validation loss with the number of epochs when considering the network architecture in Fig. 4 and the AMG antimicrobial family.

The merit figures considered in the experiments are *Accuracy*, *Sensitivity* and *Specificity*, which are formulated as follows:

$$Accuracy = \frac{tp + tn}{tp + fn + tn + fp} \tag{1}$$

$$Sensitivity = \frac{tp}{tp + fn} \tag{2}$$

$$Specificity = \frac{tn}{tn + fp} \tag{3}$$

where tp are the true positives, tn are the true negatives, fp are the false positives, fn are the false negatives, and a positive is a resistant result and a negative is a susceptible result. The *Sensitivity* takes into account the accuracy among the resistant instances (i.e. the majority class in the validation and test sets, except for the POL antibiotic family), while the *Specificity* considers the accuracy among the susceptible instances (i.e. the minority class in the validation and test sets, except for the POL antibiotic family). The area under the Receiver Operating Characteristic (ROC) curve, abbreviated as AUC [47], is also used to assess the performance. The ROC curve is a graph in which the true positive rate is plotted on the y-axis and the false positive rate is plotted on the x-axis.

3.2. Feature importance methods

Owing to the intrinsic high cost of training LSTM networks, the feature importance methods implemented need to be time-efficient. Because of that, methods such as *Forward Feature Selection*, *Backward Feature Elimination*, and *Exhaustive Feature Selection* are not feasible in this context [48]. Three well-known feature evaluation techniques, namely *Sensitivity Analysis*, *Random Subset Feature Selection*, and *Permutation Feature Importance* have been adapted for handling the temporal data in the current study. We name the adapted methods as *Gradient Feature Importance*, *Random Subset Feature Importance*, and *Permutation Feature Importance*, respectively. Note that the importance values reported by each method are not comparable, since they are provided within different ranges. To estimate the general relevance of each feature, we normalize first these values so that they are in the interval $[0, 1]$ for each method, and then they are averaged. The figure of merit considered in the three methods is the average among *Accuracy*, *Sensitivity*, and *Specificity*. Due to the class imbalance observed in the validation and test sets, it is relevant to use these measures to evaluate the performance in each class.

Gradient feature importance is based on the ability to interpret the knowledge enclosed in the trained model. Interpretability is especially challenging when analyzing ANNs, often considered as black-boxes. Particularly, in this study the *Sensitivity analysis* is considered to measure the influence of features over the prediction [49,50]. It calculates the gradient (a partial derivative of a multivariate function) of the function g describing the model for the specific data point (an instance) being evaluated. The intuition behind it is measuring the relationship between changes in one feature value and those in the model prediction. A usual formulation for the *Sensitivity analysis* is the following:

$$S_i(\mathbf{x}) = \left(\frac{\partial g}{\partial x_i}\right)^2 \tag{4}$$

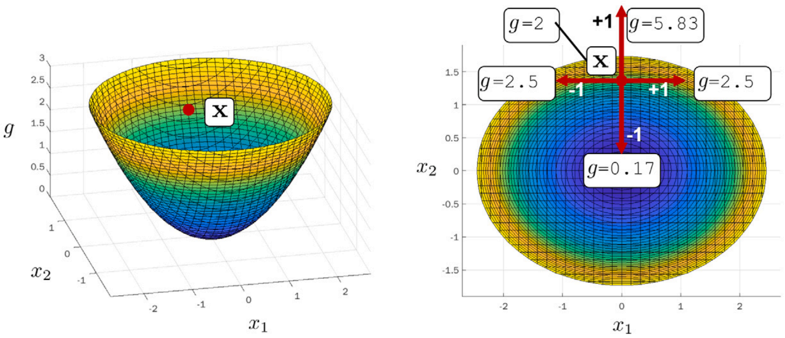
where S_i is the *Sensitivity analysis* value for the ith feature, $\mathbf{x}$ is the data point being evaluated, and $\frac{\partial g}{\partial x_i}$ is the partial derivative of g with respect to the ith feature at the specific point given by $\mathbf{x}$. The greater the value of $S_i(\mathbf{x})$, the more sensitive is the model in the particular data point $\mathbf{x}$ to the ith feature, and the more important is the ith feature. To get the general *Sensitivity analysis* value for a particular feature in the entire data set, the average over all data points (i.e. instances) in the validation set is performed.

The function g representing the LSTM model is complex and contains a high number of parameters (the weights and biases) and inputs (the number of features). In order to numerically illustrate the *Sensitivity analysis* measure, Fig. 6 shows a simple example of a quadratic function g with respect to the inputs when evaluated in a bi-dimensional data point $\mathbf{x}$ (representing an instance), and how g varies when each variable changes in one unit. For this example, the value of the function is more sensitive to changes in x_2, which is reflected in the *Sensitivity analysis* measure when computing Eq. (4).

The *Sensitivity analysis* has remarkably been applied for interpreting or explaining Convolutional Neural Networks in image classification, representing sensitivity values as heatmaps over evaluated images to indicate the relevant areas for classification [51]. Nevertheless, this process can be generalized for any ANN. For instance, in the field of time series prediction, a sensitivity value is calculated for every feature and time step of the evaluated instance. To get the general *Sensitivity analysis* value for every feature, values can be averaged over the time steps [52].

The implementation in this work is specified in the pseudo-code shown in Algorithm 1. The gradient for each feature and time step ($PD_{i,t}(\mathbf{x})$), mentioned in the pseudo-code, is calculated using the *GradientTape* context manager from the *Tensorflow* library. Since the initial LSTM weights may cause the relevance values to vary over repetitions, the calculation is carried out a reasonably high number of times (50, in this work), averaging values to help to obtain reproducible results.

Random subset feature importance is based on some of the core concepts of the *Random Subset Feature Selection* (RSFS) method [53,54]. Specifically, the *Random Subset Feature Importance* procedure selects random subsets of features of a fixed size in an iterative way, trains a model and obtains its performance when using the respective subset of features. In every iteration, this performance is stored for every feature in the selected subset. Differing from RSFF, the number of iterations is experimentally set high enough to make the relevance values converge



$$\boxed{g(x_1, x_2) \ = \ \frac{1}{2}x_1^2 + x_2^2}$$

$$\boxed{\mathbf{x} : (x_1 = 0, x_2 = \sqrt{2})} \quad \boxed{g(\mathbf{x}) = 2}$$

Fig. 6. Simple example of *Sensitivity analysis*.

Algorithm 1 Gradient Feature Importance pseudo-code

```
1: for $repetition = 1, 2, \dots 50$ do
2:     Set the random seed to initialize the LSTM weights to a random number.
3:     Train the LSTM model using the Training Set.
4:     for Every instance $\mathbf{x}$ in the Validation Set do
5:         Calculate the partial derivative for each feature $i$ and time step $t$ on instance $\mathbf{x}$. $PD_{i,t}(\mathbf{x})$
6:         Get the mean temporal value averaging over time steps. $PD_i(\mathbf{x}) = mean\_over\_t(PD_{i,t}(\mathbf{x}))$
7:         Compute sensitivity value $S_i(\mathbf{x}) = (PD_i(\mathbf{x}))^2$
8:     end for
9:     Calculate the mean sensitivity value for each feature $i$, by averaging over all instances in the Validation set.
10:    Calculate the LSTM performance on the Validation Set.
11: end for
12: Calculate weighted average of the sensitivity value over repetitions for each feature $i$. The weights are proportional to the performance of the LSTM in each repetition.
13: Sorting the features in decreasing order of the sensitivity value calculated in line 12, returning them.
```

Algorithm 2 Random Subset Feature Importance pseudo-code

```
1: for $num\_iterations = 1, 2, \dots 500$ do
2:     Set the random seed to initialize the LSTM weights to a random number.
3:     Randomly, select a subset of 40 features.
4:     Train the LSTM model using the Training Set, just considering the subset of features.
5:     Calculate the figure of merit of the LSTM on the Validation Set.
6:     Append the figure of merit obtained with the particular subset to each of the 40 features.
7: end for
8: Calculate the feature relevance for every feature $i$ as the mean of the figures of merit available for the feature $i$.
9: Sorting the features in decreasing order of the relevance value calculated in line 8, and return them.
```

while being computationally efficient. Finally, the performance values stored for every feature are averaged to get the mean relevance linked to that particular feature.

The implementation of the *Random Subset Feature Importance* is described as pseudo-code in Algorithm 2, where the chosen number of features in every subset is 40 and the number of repetitions is 500. In this way, the total number of feature relevance values calculated is *500 iterations* × *40 features = 20.000 values*. Note that, since there are 398 features in total, on average there are 50 values per feature, the same number as in the previous *Gradient Feature Importance* method.

Permutation feature importance is based on the concept that the importance of a feature is expressed by the increase in the model's prediction error after permuting the values of that feature in evaluation. It was originally proposed in [55,56] specifically for the *random forest* algorithm, and has been *generalized* to any predictive algorithm [57,58].

The first step of the *Permutation Feature Importance* method is to train the predictive model (in this case the LSTM) and to estimate the performance on a set of instances (validation set) different from those used in training. Then, just using the validation instances, the values of the feature whose importance value is being calculated are randomly permuted while the rest of the features remain unchanged. The new set is a modified validation set such that, though the marginal distribution of the analyzed feature is not altered, its relation with the target learned in training no longer holds, leaving this feature uninformative. The model performance is now computed using the modified validation set, and it is compared with the performance provided by the same model when using the original validation set. It is expected to obtain worse results with the modified validation set since training and validation data do not have the same relationship with the target. The more informative and useful for prediction the feature is, the worst will be the prediction after shuffling its values. This process is repeated for every feature [59].

It is relevant to remark on the efficiency in cost of this method. Instead of retraining the model by removing one feature each time, which would be computationally intensive, the training is carried out just once, and feature importance values are computed by making just a series of predictions.

The implementation of the *Permutation Feature Importance* method is expressed as pseudo-code in Algorithm 3. The original and the modified validation set are compared using the figure of merit described at the beginning of Section 3.2 in order to be consistent with the implementation of the other two feature importance methods.

3.3. Feature selection approach

In this section, a robust and time-efficient algorithm is proposed to select the proper number of features. The objective is to find the minimum number of features maximizing the performance. For this

Algorithm 3 Permutation Feature Importance pseudo-code

```
1: for $repetition = 1, 2, \ldots 50$ do
2:     Set the random seed to initialize the LSTM weights to a random number.
3:     Train the LSTM model using the Training Set.
4:     Calculate performance figure of merit $PFM$ on the Validation Set $VS$.
5:     for Every feature $i$ do
6:         Permute values of the feature $i$ to generate a new Validation Set $VS_i$
7:         Calculate the performance $PFM_i$ when considering $VS_i$
8:         Calculate the importance of the feature $i$ as $FI_i = PFM - PFM_i$
9:     end for
10: end for
11: Calculate the mean importance value for each feature $i$, averaging over repetitions.
12: Sorting the features in decreasing order of the relevance value calculated in line 11, and return them.
```

purpose, we first analyze the performance variation as the number of features increases.

The importance of each feature is calculated by first normalizing the feature importance value given by each of the three aforementioned methods and averaging these three normalized values (see Section 3.2). With the features sorted in decreasing order according to the averaged feature importance, they are sequentially picked to be used in the prediction of the instances from the validation set, and the achieved *Accuracy*, *Sensitivity*, and *Specificity* are stored. That is, in the first place, the LSTM network is trained using just the most important feature, and its performance is stored. Afterward, the first and second most important features are used to train, and so on, until all the 398 features are considered. With that, 398 values of performance are obtained, one per number of features.

Though the number of features with the best tradeoff between mean and standard deviation performance could be chosen, the randomness in the network initialization and the data scarcity provide high variations in the performance. To mitigate this effect, we propose a recursive algorithm we have called *Recursive feature number selection* to find the interval in the number of features providing the best performance. The algorithm starts by considering the whole number of features and it is progressively focusing on narrower and narrower intervals until one of them (with the interval length lower than a predefined threshold) provides a performance which is better than the one of the neighboring intervals. The approach is based on calculating a performance value for every number of features. This performance value considers both the average of *Accuracy*, *Sensitivity* and *Specificity* and the standard deviation among the three. It is computed as follows:

$$PV_i = mean(Acc_i, Sen_i, Spe_i) - std(Acc_i, Sen_i, Spe_i) \tag{5}$$

where Acc_i, Sen_i, Spe_i are the *Accuracy*, *Sensitivity*, and *Specificity* values for the ith number of features. A low standard deviation is advantageous since it indicates agreement among figures of merit.

The performance measurement for an interval of N values (N different number of features) is formulated as the mean of the corresponding N performance values:

$$PM = \frac{1}{N}\sum_{i=1}^{N} PV_i \tag{6}$$

The implementation in pseudo-code is detailed in Algorithm 4.

Fig. 7 depicts an execution of the *Recursive feature number selection* algorithm when the $threshold$ is set to 10. Short vertical red lines (labeled with numbers) indicate the limits of each interval, step by step. In the first step (marked with the number 1), the whole interval from 1 to 398 is split into two parts, each on containing half of the values. Since the mean performance ($perf_first$ in Algorithm 4) is higher in the interval linked to the left part of the splitting point, the algorithm focuses on these values by recursively calling the function. This process is repeated until reaching the division indicated with the vertical line labeled as 6. The interval between lines labeled 6 and 5 is selected because its performance is better than the one between lines labeled 4 and 6. Since the last interval contains less than $threshold$ values, the algorithm finishes and returns the number of features indicated by the line labeled as 5 (corresponding to 36 number of features).

3.4. Horizontal voting ensemble

Apart from the high number of attributes, the small number of instances in the training set (from 284 to 545 training instances depending on the data set, as shown in Table 2) hinders model generalization and causes a high variance in performance. To address this problem, we use an ensemble of LSTMs following the approach of the *Horizontal Voting Ensemble* proposed in [60]. It consists of storing the trained model after every training epoch and, close to the end of the training process, obtaining a set of models to be used in the ensemble. That is, instead of predicting just with the model obtained after training, it uses an ensemble of models extracted from the partially trained neural network.

In our implementation, the models included in the ensemble are those generated from the 10th to the 30th epoch. Note that we have extended the number of epochs indicated in Section 3.1 for the purpose of using this procedure. Note that the *Horizontal Voting Ensemble* approach reduces the high computational cost involved when training the LSTM networks from scratch. Since LSTMs are trained just using the features obtained from the FS process, we experimentally checked that reducing from 100 to 50 units in both LSTM layers was beneficial for improving performance. The results detailed below (Section 4) consider the network architecture depicted in Fig. 4 with 100 units per layer when neither feature selection nor ensemble of networks is considered, and with 50 units per layer otherwise.

4. Results

The results of the experiments are detailed in the two following sections: Section 4.1 analyzes the final subset of selected features and evaluates the performance achieved when applying the dimensionality reduction method, and Section 4.2 assesses the outcome of predicting with an ensemble of LSTMs.

4.1. Dimensionality reduction

The number of features chosen per set of features is portrayed in Fig. 8 for each antimicrobial family. It is remarkable the high reduction in the number of features, from the original number of 398 to 36 (AMG), 18 (CAR), 67 (CF4), 24 (PAP), 55 (POL), and 49 (QUI) features. The sets of features not selected for any of the families are *"Number of co-admitted patients"*, *"ICU antibiogram results"*, and *"icu_mechanic_ventilation"*, suggesting that they are not useful in the prediction of antimicrobial resistance for a particular patient. This result may be reasonable since these three sets of features are not directly related to the patient whose antibiogram result is being predicted. Note also that features in the set *"Antibiogram results"* are selected in 4 out of the 6 antibiotic families. This fact confirms the findings in [16,20], which determine that a piece of highly useful information for prediction is the **previous resistance of bacteria to antibiotics** for the considered patient. The set of *"Temporal features"* are selected only in the case of the POL family, which was also observed in [16]. Note also that the *mechanic_ventilation* feature is selected for 2 families, which

Algorithm 4 Recursive feature number selection

```
Call function: recursive_number_feature_selection([1, 2, ..., 398], [mean_perf_1_feat, mean_perf_2_feats, ..., mean_perf_398_feats], [std_perf_1_feat, std_perf_2_feats, ..., std_perf_398_feats], threshold)

Define function: recursive_number_feature_selection($array\_number\_features$, $array\_mean\_perf$, $array\_std\_perf$, $threshold$):
if $length(array\_number\_features) < threshold$ then
    Return $array\_number\_features[last\_element]$
else
    $arr\_first\_num = array\_number\_features[beginning \rightarrow half]$
    $arr\_secnd\_num = array\_number\_features[half \rightarrow end]$
    $arr\_first\_mean = array\_mean\_perf[beginning \rightarrow half]$
    $arr\_secnd\_mean = array\_mean\_perf[half \rightarrow end]$
    $arr\_first\_std = array\_std\_perf[beginning \rightarrow half]$
    $arr\_secnd\_std = array\_std\_perf[half \rightarrow end]$
    $N \leftarrow length(arr\_first\_num)$;
     $perf\_first = \frac{1}{N}\sum_{i=1}^{N}(arr\_first\_mean[i] - arr\_first\_std[i])$
    $N \leftarrow length(arr\_secnd\_num)$;
     $perf\_secnd = \frac{1}{N}\sum_{i=1}^{N}(arr\_secnd\_mean[i] - arr\_secnd\_std[i])$
    if $perf\_first \geq perf\_second$ then
        $num\_feat$ = Call function:
                recursive_number_feature_selection($arr\_first\_num$,
                $arr\_first\_mean$, $arr\_first\_std$)
        Return $num\_feat$
    else
        $num\_feat$ = Call function:
                recursive_number_feature_selection($arr\_secnd\_num$,
                $arr\_secnd\_mean$, $arr\_secnd\_std$)
        Return $num\_feat$
    end if
end if
```

may be indicating the increased probability of acquiring resistance as a consequence of using mechanical ventilation during the ICU stay.

Table 3 shows that, for the majority of antibiotic families, the selected features refer to the patient's admission, the antibiotics administered to the patient and information about previous resistance of the patient's bacteria to antibiotics. It is relevant that one of the most selected sets of features is the one linked to the antibiotics administered to the patient. From a clinical viewpoint, this is explained by the *antibiotic selective pressure* phenomenon [61], which favors an increase in the resistant bacteria strains by selecting the populations surviving the use of the antibiotic. Furthermore, antibiotic resistance genes can be transferred among bacteria [62]. Features linked to the set of "ICU admission information" and those sets representing administered antibiotics to the co-admitted patients are in the table, possibly pointing towards cross-transmission among ICU patients, although in accordance with Fig. 8 they appear in a lower frequency. As previously, note the relevance of the feature *resistance_qui*, indicating the usefulness of the *"Antibiogram results"* set of features. Finally, remark the relevance of features related to respiratory issues (*reason admission-acute respiratory failure* and *reason admission-acute chronic respiratory failure*), also in accordance with the importance of the feature *mechanic_ventilation* previously indicated.

Tables 4 and 5 show the figures of merit on the validation set and the test set, respectively, when using the LSTM network before and after applying our FS approach. For each family, the network has been trained and evaluated 50 times, and the average and standard deviation of the provided figures of merit have been calculated. Repetitions are carried out because, due to the randomness in the ANN initialization, the performance can vary from one trained network to another one and it is convenient to average the values of the figures of merit over a number of repetitions to compensate for this variation among repetitions. In this particular case, 50 repetitions are enough for the mean and standard deviation to converge. In these tables, the results without feature selection (X) are calculated with 100 units per LSTM layer, and the results with feature selection (✓) are with 50 units per LSTM layer. It is observed that for both validation and test sets, **when FS is applied, not only the complexity of the problem is reduced, but the loss value is considerably reduced in all cases**. Also, the standard deviation of the values provided by the figures of merit is generally reduced with the proposed FS method. As expected, the results in the validation set are slightly better than those in the test set. This is caused by three reasons: the hyper-parameters are chosen using the validation set, the training set is temporally further from the test set

Table 3
Some of the selected features, sorted by the number of times they are selected in the different antibiotic families, from highest to lowest.

Feature name	Antibiotic families	#
antibiotics-ciprofloxacin	AMG, CAR, CF4, POL, QUI	5
gender	AMG, CAR, CF4, PAP, QUI	5
reason_admission-acute_respiratory_failure	AMG, CF4, POL, QUI	4
antibiotics-colistin	AMG, CAR, POL, QUI	4
reason_admission-neuromuscular	AMG, CAR, POL, QUI	4
resistance_qui	AMG, CF4, PAP, QUI	4
reason_admission-hypovolaemia	AMG, CAR, CF4, QUI	4
antibiotics-doxycycline	AMG, CF4, PAP, QUI	4
antibiotics-amikacin	CAR, CF4, PAP, QUI	4
patient_category-medical_patient	CF4, PAP, POL, QUI	4
antibiotics-meropenem	AMG, CAR, QUI	3
antibiotics-clindamycin	AMG, CF4, POL	3
reason_admission-acute_chronic_respiratory_failure	AMG, POL, QUI	3
antibiotics-tigecycline	AMG, CF4, POL	3
icu_antibiotics-colistin	AMG, POL, QUI	3
icu_reason_admission-ischemic_cardiopathy	AMG, CF4, QUI	3
antibiotics-amphotericin_b_lipid	AMG, CAR, QUI	3
icu_antibiotics-erythromycin	AMG, CF4, POL	3
reason_admission-urgent_uncomplicated	CAR, CF4, QUI	3
icu_reason_admission-neurological_other	CAR, POL, QUI	3
antibiotics-aztreonam	CF4, PAP, POL	3
reason_admission-gastrointestinal_bleeding	PAP, POL, QUI	3
icu_antibiotics-ceftazidime	PAP, POL, QUI	3

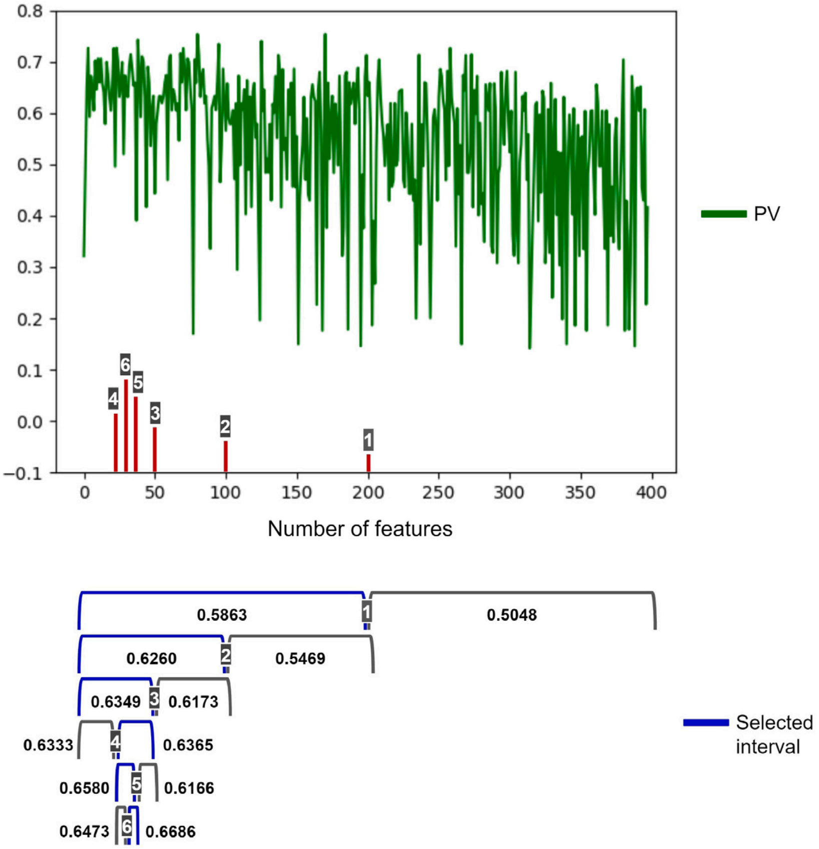


Fig. 7. The plot on the top shows the performance value (PV) with a green solid line, as the number of features increases. Inside the plot, the red ticks represent the steps of the *Recursive feature number selection* algorithm, with the number of steps showing on top of each red tick. Particularly, each red tick is placed on the splitting point in the interval to be divided. Below the plot, complementing the red ticks, a set of pairs of brackets represent each step of the algorithm. Each pair of brackets represent the two intervals in which the number of features is divided at every step. The number inside the bracket represents the performance measure (PM) for its respective interval. As mentioned in the text, the interval with the highest PM is selected and it is subsequently divided. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
Results for the validation set. Mean and standard deviation values for each antimicrobial family and merit figure when applying the LSTM, with (✓) and without (X) FS. The LSTM training and prediction is repeated 50 times, showing: Value of the loss function (Loss), "Area under the ROC curve" (AUC), "Accuracy" (Acc.), "Sensitivity" (Sen.) and "Specificity" (Spe.).

	FS	Loss	AUC	Acc.	Sen.	Spe.
AMG	X	0.91 ± 0.41	68.28 ± 6.21	62.8 ± 9.76	60.93 ± 22.27	67.33 ± 25.68
	✓	0.61 ± 0.07	76.21 ± 3.07	68.78 ± 5.70	65.23 ± 10.92	77.4 ± 13.27
CAR	X	0.92 ± 0.19	66.0 ± 4.70	54.82 ± 8.4	53.33 ± 9.89	68.2 ± 7.12
	✓	0.68 ± 0.03	70.38 ± 2.85	66.86 ± 2.14	67.6 ± 2.42	60.20 ± 1.40
CF4	X	0.89 ± 0.11	67.71 ± 3.35	55.69 ± 4.30	29.09 ± 8.67	85.55 ± 3.34
	✓	0.69 ± 0.03	72.8 ± 2.25	63.46 ± 3.09	36.87 ± 6.98	93.31 ± 6.59
PAP	X	1.08 ± 0.17	64.28 ± 3.18	49.33 ± 4.22	24.32 ± 7.91	86.24 ± 4.72
	✓	0.75 ± 0.03	67.71 ± 2.74	58.62 ± 3.87	37.16 ± 8.53	90.29 ± 8.15
POL	X	0.39 ± 0.11	66.58 ± 7.27	91.84 ± 0.0	0.0 ± 0.0	100.0 ± 0.0
	✓	0.34 ± 0.05	55.69 ± 6.76	91.84 ± 0.0	0.0 ± 0.0	100.0 ± 0.0
QUI	X	0.75 ± 0.18	72.47 ± 4.29	65.01 ± 4.68	61.49 ± 12.65	71.93 ± 14.55
	✓	0.56 ± 0.05	78.72 ± 3.58	70.58 ± 3.72	67.96 ± 5.92	75.71 ± 7.39

(*concept drift* also discussed in Section 4.2), and the relevant features are identified using the validation set. This difference is evident, for instance when comparing the average of **AUC** among the 6 families **without** FS in the validation set (67.55%) with the test set (60.53%), what is caused by the two first reasons commented. And it can also be observed the difference between the average of *AUC* values **with** FS for validation (70.25%) and test (67.7%), in this case caused by all three aforementioned reasons. A particular behavior in *AUC*, *Accuracy*, and *Sensitivity* is observed for the CF4 and PAP families since instances are usually classified as susceptible. This is probably due to the higher rate of susceptible instances in training, observed in Table 2. The family with a clearly different behavior is POL, which is an extreme case of the CF4 and PAP families, classifying all instances as susceptible, as shown with the 0 percentage in the Sensitivity figure of merit. Table 2 shows that the POL family only counts with 9 resistant instances in the training set. In these cases, applying a class balancing technique would be beneficial. These class particularities were also noticed in previous studies [16].

To determine whether there are statistically significant differences in prediction when using all features and the proposed FS, we perform a nonparametric statistical test [63,64] based on bootstrap resampling [65] for each antibiotic family. In particular, we check it by

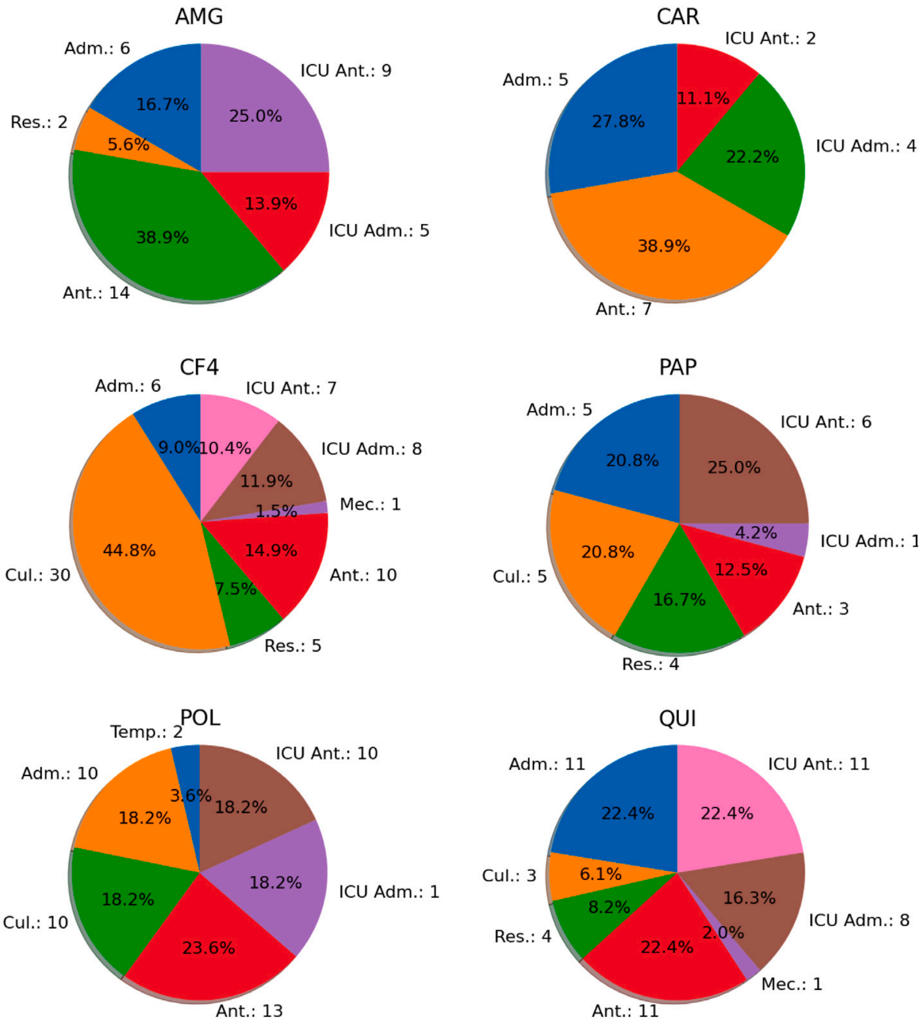


Fig. 8. Percentage (inside the pie chart) and number (next to the portion name) of features selected per antimicrobial family and set of features. Names of the sets of features refer to those specified in Section 2.1: "Temporal features" ("Temp."), "Admissions information" ("Adm."), "Culture and antibiogram information" ("Cul."), "Antibiogram results" ("Res."), features *"antibiotics-[modality]"* ("Ant."), feature *"mechanic_ventilation"* ("Mec."), "Number of co-admitted patients" ("Co. Pat."), "ICU admission information" ("ICU Adm."), "ICU antibiogram results" ("ICU Res."), features *"icu_antibiotics-[modality]"* ("ICU Ant."), and feature *"icu_mechanic_ventilation"* ("ICU Mec.").

Table 5
Results for the test set and each antimicrobial family. Mean ± standard deviation of the considered merit figures for 50 LSTM networks (each one with a different initialization), with (✓) and without (X) FS.

	FS	Loss	AUC	Acc.	Sen.	Spe.
AMG	X	1.51 ± 1.07	60.17 ± 11.99	53.08 ± 15.15	46.18 ± 28.94	71.79 ± 27.28
	✓	0.62 ± 0.10	76.6 ± 6.17	63.48 ± 11.14	58.24 ± 18.19	77.71 ± 13.84
CAR	X	0.84 ± 0.25	68.88 ± 7.08	53.78 ± 12.80	51.24 ± 16.24	74.55 ± 19.83
	✓	0.61 ± 0.03	65.93 ± 3.52	69.6 ± 1.29	72.33 ± 1.54	47.27 ± 3.64
CF4	X	1.27 ± 0.29	57.96 ± 3.94	43.73 ± 3.61	20.45 ± 8.06	85.89 ± 8.19
	✓	0.92 ± 0.06	62.77 ± 2.64	45.4 ± 3.62	24.33 ± 6.51	83.57 ± 7.07
PAP	X	1.24 ± 0.25	63.73 ± 4.53	50.94 ± 4.32	26.59 ± 9.74	85.49 ± 8.56
	✓	0.75 ± 0.04	71.19 ± 3.76	52.33 ± 5.66	26.69 ± 10.95	88.7 ± 6.01
POL	X	2.33 ± 1.41	37.13 ± 6.14	68.37 ± 0.0	0.0 ± 0.0	100.0 ± 0.0
	✓	2.15 ± 1.12	41.58 ± 6.05	68.37 ± 0.0	0.0 ± 0.0	100.0 ± 0.0
QUI	X	0.66 ± 0.27	75.31 ± 7.26	67.71 ± 12.03	67.01 ± 14.08	75.43 ± 17.15
	✓	0.52 ± 0.09	88.1 ± 4.06	69.24 ± 6.45	66.55 ± 7.13	98.86 ± 3.88

considering the *AUC* both for the validation and the test sets, which correspond to results in Tables 4 and 5. Note that the number of *AUC* observations in the original population is 50 (same number as the number of LSTM initializations). The hyper-parameters used for the statistical test based on bootstrap are: 70% of the number of *AUC* observations are selected in each bootstrap repetition, 2000 bootstrap repetitions, and alpha value (the likelihood that the true population parameter lies outside the confidence interval) of 0.05. Regarding Table 4, there is a statistically significant improvement when evaluating the validation sets with models performing FS for all the antimicrobial families except for the POL family. In this case, a higher AUC (statistically significant) is provided by models with all features. With regard to Table 5, the test shows better results (statistically significant) for all the antimicrobial families when performing FS, except for the CAR family (statistically significant higher AUC provided by models with all features).

In order to assess the effectiveness of the proposed algorithm for dimensionality reduction, it is compared with two other FS methods. The first one applies the test based on bootstrap, previously described, to every feature and then a voting procedure in order to account for the temporal aspect of the variables [65,66] (hereafter Bootstrap Voting FS method). The second FS method used for comparison is Group Lasso [66,67] (hereafter Group Lasso FS method), grouping time instants related to the same feature. Firstly, each FS method is applied to obtain the most relevant features for every antimicrobial family (specified in brackets below). The Bootstrap Voting FS method selects 99 features in the case of (AMG), 94 (CAR), 86 (CF4), 56 (PAP), 46 (POL) and 187 (QUI). The Group Lasso FS method selects 61 (AMG), 96 (CAR), 100 (CF4), 103 (PAP), 41 (POL) and 93 (QUI). As mentioned before, the Proposed FS method selects 36 (AMG), 18 (CAR), 67 (CF4), 24 (PAP), 55 (POL) and 49 (QUI). Note that our FS method selects a considerably fewer number of features than Bootstrap Voting and Group Lasso for all the antimicrobial families excepting POL. Afterward, using the new sets of selected features, the LSTM is trained (same ANN architecture as the one used with the proposed method, with 50 units per LSTM layer) and its performance is evaluated. In this manner, we gather the figures of merit obtained when using the features selected by each of the three methods (proposed method, Bootstrap Voting FS method and Group Lasso FS method). As in the previous paragraph, the *AUC* values are compared by applying a nonparametric test hypothesis (based on bootstrap resampling) to determine statistically significant differences between the proposed FS method and the other two approaches [63]. When evaluating the validation sets, the statistical test shows the following. Comparing our FS method with the Bootstrap Voting FS method, *AUC* values are higher (statistically significant difference) when using our approach for all the antimicrobial families. Comparing with Group Lasso FS, AUC values are higher (statistically significant difference) when using features selected with our approach for all the antimicrobial families excepting PAP and POL. For these two families there is not a statistically significant difference between the *AUC* provided when using either FS method. When evaluating the test sets, the statistical test shows the following. The *AUC* values when using our approach are higher (statistically significant difference) than those provided when considering the Bootstrap Voting FS method for all the antimicrobial families excepting AMG and POL (there is not a statistically significant difference when using either FS method). Comparing with the Group Lasso FS method, the *AUC* values are statistically higher when using features selected with our approach for all the antimicrobial families excepting CAR (no statistical difference) and POL. The LSTM network built for POL using features selected with Group Lasso FS provides higher AUC (statistically significant difference). To sum up, compared to Bootstrap Voting FS and Group Lasso FS, the use of the proposed FS method maintains or (in most cases) improves the *AUC* achieved, both in validation and test sets, while using considerably less features. The only exception is the POL family, for which the use of the features provided by Group Lasso FS achieves a statistically significant better performance than when considering the proposed FS method, also with a lower number of features. This different behavior with respect to other antimicrobial families can be caused by the high imbalance in training instances for the POL family, shown in Table 2. The proposed FS method exhibits a high capability for reducing the dimensionality in this task (from almost 400 features to 18–67) and also leads to a considerable increase in the performance, reducing the standard deviation among repetitions.

4.2. Ensemble prediction

The outcomes of applying the *Horizontal voting ensemble* are presented in Tables 6 and 7 both for the validation and test set, respectively. They illustrate the figures of merit by using just one network (X in "Ens." column) against an ensemble of 20 networks (✓ in "Ens." column). Again, the performance is calculated by carrying out 50 repetitions. Since this experiment just considers the features chosen in the FS process, the LSTM layers have a lower number of units (50, in this case). It is observed that the standard deviation of the merit figures is in general moderately reduced, which means that using the ensemble of classifiers allows decreasing variance to a small extent. Also, the loss value is slightly diminished in all of the antibiotic families, both for the validation and test set, implying that the performance is improved. As before, note that the validation results are better than the test results, although here the difference is less noticeable. As expected, the standard deviation is generally reduced both in validation and test when using the ensemble of networks. In validation, the *AUC* value is similar whether or not the ensemble is used. In test, the *AUC* averaged among the six families when not using an ensemble is (67.64%), and when using the ensemble, the *AUC* improves to (68,35%). As before, for every antimicrobial family we assess whether the difference in performance is statistically significant when using an ensemble of LSTMs. We use the nonparametric hypothesis test on the *AUC* values for this purpose: Regarding Table 6 (validation sets), there is no statistically significant differences between models with one LSTM and those considering an ensemble for all the antimicrobial families except for the QUI family. For the QUI family, a higher AUC (statistically significant) is provided by using network ensembles. As for Table 7, there is a statistically significant improvement when evaluating the test sets with models using network ensemble for all the antimicrobial families.

When analyzing these results it must be taken into account that we study antimicrobial resistance, a phenomenon that changes over time as bacteria mutates, which constitutes a temporal particularity of the data set. The features considered in this study refer to the EHR information of the analyzed patients. Since bacteria's mutations are not amongst the available features, the feature's values discriminating between classes may change over time [16]. This fact has been previously described as the *concept drift*, in which the concept of interest may depend on some hidden context not explicitly provided in the form of predictive features. Over time, it can cause a change in the underlying data distribution [68]. This phenomenon worsens the network performance when the training set is temporally far from the test set. A usually proposed solution for this problem is to apply *windowing*, which consists in learning from data in a window containing recent instances and predicting only instances in the immediate future, as temporally close as possible to the training instances [17]. In the current study, the *concept drift* has not been taken into account because it requires training LSTM networks multiple times, each for a different training window. Although the use of windowing could improve network performance, its computational cost is prohibitive. Nevertheless, the presented results show the potential capability of prediction even when a very small number of instances (with some amount of class imbalance) are used for training, making it harder to generalize. This promising performance, combined with different techniques, could make it possible to integrate deep neural networks and windowing to achieve accurate predictions.

5. Discussion and conclusions

This paper considers the prediction of antimicrobial resistance in *Pseudomonas aeruginosa* bacteria causing nosocomial infections at the ICU. In order to support physicians' decisions in clinical practice, we propose to carry out a binary classification task (susceptible/resistant). For the decision support implementation, it is suggested to model patients' EHR data as multivariate time series instances, using LSTM neural networks as classifiers. The considered features contain temporal information related to the time series, information about the admission, cultures, antibiograms, antibiotics administered and mechanical ventilation linked to the patient whose antimicrobial resistance's result is being predicted. Also, information linked to the ICU co-admitted

Table 6
Results for the validation set. Mean and standard deviation values for each antimicrobial family and merit figure applying FS, with just one LSTM network (X in “Ens” column) and with a network ensemble (✓). The training and prediction are repeated 50 times, showing: the value of the loss function (Loss), “Area under the ROC curve“ (AUC), “Accuracy” (Acc.), “Sensitivity” (Sen.) and “Specificity” (Spe.).

	Ens.	Loss	AUC	Acc	Sen	Spe
AMG	X	0.61 ± 0.08	75.78 ± 3.27	69.67 ± 3.69	67.04 ± 7.65	76.07 ± 10.74
	✓	0.58 ± 0.04	75.85 ± 3.08	69.51 ± 2.55	67.18 ± 5.72	75.2 ± 9.48
CAR	X	0.68 ± 0.04	70.44 ± 2.88	67.1 ± 1.89	67.84 ± 2.15	60.4 ± 2.8
	✓	0.67 ± 0.03	70.38 ± 2.68	67.14 ± 1.79	67.89 ± 2.02	60.4 ± 2.8
CF4	X	0.69 ± 0.04	72.86 ± 2.36	63.21 ± 3.19	37.6 ± 6.58	91.96 ± 6.65
	✓	0.69 ± 0.04	72.62 ± 2.06	63.62 ± 2.98	37.89 ± 6.59	92.49 ± 6.3
PAP	X	0.74 ± 0.03	67.64 ± 2.9	59.02 ± 4.21	37.55 ± 8.33	90.71 ± 8.0
	✓	0.74 ± 0.03	67.47 ± 2.54	59.38 ± 3.87	38.26 ± 7.85	90.57 ± 7.35
POL	X	0.35 ± 0.05	55.55 ± 7.71	91.84 ± 0.0	0.0 ± 0.0	100.0 ± 0.0
	✓	0.34 ± 0.04	54.75 ± 7.84	91.84 ± 0.0	0.0 ± 0.0	100.0 ± 0.0
QUI	X	0.56 ± 0.06	78.61 ± 3.9	71.06 ± 3.59	68.84 ± 5.44	75.43 ± 7.94
	✓	0.55 ± 0.04	78.97 ± 3.65	71.66 ± 3.6	69.96 ± 4.81	75.0 ± 8.11

Table 7
Results for the test set. Mean and standard deviation values for each antimicrobial family and merit figure applying FS, with just one LSTM network (X in “Ens” column) and with a network ensemble (✓). The training and prediction are repeated 50 times.

	Ens.	Loss	AUC	Acc	Sen	Spe
AMG	X	0.63 ± 0.15	76.0 ± 5.74	63.69 ± 10.77	58.55 ± 17.38	77.64 ± 11.58
	✓	0.61 ± 0.07	76.77 ± 5.57	64.25 ± 9.61	59.11 ± 15.09	78.21 ± 10.06
CAR	X	0.61 ± 0.03	65.83 ± 3.48	69.64 ± 1.0	72.47 ± 1.18	46.55 ± 2.95
	✓	0.61 ± 0.03	66.53 ± 3.28	69.74 ± 0.99	72.58 ± 1.21	46.55 ± 2.95
CF4	X	0.92 ± 0.06	63.04 ± 2.87	45.62 ± 3.36	24.6 ± 6.39	83.68 ± 7.01
	✓	0.9 ± 0.05	63.45 ± 2.64	45.58 ± 3.37	24.51 ± 7.05	83.73 ± 7.24
PAP	X	0.74 ± 0.03	71.1 ± 3.93	52.65 ± 5.14	27.61 ± 10.31	88.19 ± 5.88
	✓	0.73 ± 0.03	71.63 ± 3.47	53.02 ± 5.22	28.3 ± 9.93	88.09 ± 5.34
POL	X	2.26 ± 1.23	41.56 ± 6.06	68.37 ± 0.0	0.0 ± 0.0	100.0 ± 0.0
	✓	1.57 ± 0.41	42.32 ± 6.73	68.37 ± 0.0	0.0 ± 0.0	100.0 ± 0.0
QUI	X	0.50 ± 0.08	88.29 ± 3.71	70.24 ± 5.85	67.74 ± 6.45	97.71 ± 5.97
	✓	0.49 ± 0.07	89.37 ± 3.27	71.12 ± 5.52	68.65 ± 6.04	98.29 ± 5.45

patients about their admission, antibiogram results, antibiotics administered and mechanical ventilation is considered. Owing to the high dimensionality of the data set (~400 features), an efficient data-driven FS method is proposed by adapting three feature importance approaches from the literature (namely, *Gradient Feature Importance*, *Random Subset Feature Importance*, and *Permutation Feature Importance*), and developing a method to select the most appropriate number of features. On the other hand, because of the small number of instances and the consequent performance variance, an *Horizontal Voting Ensemble* is applied to reduce the model variability. The results of FS provide valuable information for both the data analysis and clinical point of view, by considering the features' temporal aspect. The features identified as most relevant are the antibiotics administered, the previous antimicrobial resistance, and the admission information of the patient whose antibiogram results are being predicted. This outcome reveals the presence of the *antibiotic selective pressure* phenomenon and confirms previous findings of recent antimicrobial resistance importance. Furthermore, our technique is able to capture the temporal dependencies among features in the data sequences and also provides very useful information both from the predictive and the clinical point of view, that could be applied to further studies. A nonparametric hypothesis test based on bootstrap resampling has been carried out to check the result's validity. It shows that the use of our feature selection method achieves a statistically significant improvement in performance for almost all of the families, both in validation and test. The only exceptions are POL in validation and CAR in test, for which it is significantly better to not carry out the proposed feature selection. The proposed dimensionality reduction technique is also compared with two other FS methods (bootstrap followed by voting, and Group Lasso). It is observed that, in most families, the proposed technique achieves a significantly better performance while using considerably less number of features. The ensemble of classifiers extracted from training is able to improve the performance while reducing the variance. A hypothesis test based on bootstrap resampling reveals that using an ensemble of networks for prediction does not provide a statistically significant improvement in the validation set. However, on the test set, the improvement in performance is statistically significant for all the antimicrobial families. In essence, the proposed methods and framework achieve, in a cost-efficient manner, promising results for the tackled task which is characterized by high dimensionality, data scarcity, a certain level of class imbalance, and *concept drift*. The proposed methodology and the experimental results have been critically analyzed from the clinical point of view with the help of clinicians. The features selected through dimensionality reduction are meaningful from the clinical perspective, since some of them evidence mechanisms in which bacteria become resistant in hospitals.

In the real hospital setting, there may be a discrepancy between the antibiogram result provided by the microbiology laboratory (susceptible/resistant) and what happens in reality in the ICU. This is due to the dynamic nature of antimicrobial resistance, which can vary over time based on multiple factors. In a nutshell, the results of the LSTM models only reflect how well they predict the antibiogram results (gold standard). Physicians can combine the prediction of the proposed method with their medical experience at the ICU to proceed with the decisions regarding the actual antimicrobial resistance.

In future studies, alternative FS algorithms able to deal with multivariate time series (such as methods based on Mutual Information [69, 70] or those based on the Granger causality discovery [71]) could also be considered. It is also interesting to apply class balancing techniques (such as undersampling, oversampling or instance weighting), especially in cases with high imbalance, for instance in the case of the POL antimicrobial family. Furthermore, because of the *concept drift*, it would be advantageous to carry out class balancing on delimited temporal windows, so that each temporally local data distribution is balanced.

Finally, it could also be beneficial to use ensemble learning by combining methods presented in this study together with remarkably accurate methods from previous studies (such as Logistic Regression and Random Forest [16]), so that the ensemble leverages the advantages of the different methods.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Availability of data and materials

The data used for this research comprise confidential patient health information, which is protected and may not be released unless approval by the Committee of Ethics of the University Hospital of Fuenlabrada (Spain).

Acknowledgments

This work has been partly supported by the Spanish Research Agency, grant numbers PID2019-106623RB-C41, AEI/10.13039/501100011033 (BigTheory), PID2019-107768RA-I00 (AAVis-BMR), by funding action by the Community of Madrid in the framework of the Multiannual Agreement with Rey Juan Carlos University in line of action 1 ‘‘Encouragement of Young Phd students investigation’’ Project Mapping-UCI (Ref F661), by the IDEAI-UPC Consolidated Research Group Grant from Catalan Agency of University and Research Grants (AGAUR, Generalitat de Catalunya) (2017 SGR 574) and by the Secretariat for Universities and Research of the Ministry of Research and Universities of the Government of Catalonia and the European Social Fund (2021 FI-B 00965).

Availability of data and materials

The data used for this research comprise confidential patient health information, which is protected and may not be released unless approval by the Committee of Ethics of the University Hospital of Fuenlabrada (Spain).

Annex

Below we list the features considered in this study specifying all the different modalities. The description of the feature meaning is provided in Section 2.1.

Temporal features

- day_week
- day_month
- month

Patient features

Admissions information

- admission_indicator
- origin-[modality], with modality being any of the following entries: anaesthesia, another_hospital, cardiology, dermatology, digestive, general_surgery, haematology, internal_medicine, nephrology, neurology, oncology, ophthalmology, other_floor, others, otorhinolaryngology, pneumology, psychiatry, resuscitation_unit, surgery_room, traumatology, urgencies, urology.
- destination-[modality], with modality being any of the following cases: cardiology, digestive, general_surgery, haematology, internal_medicine, mortuary, otorhinolaryngology, pneumology, psychiatry and urology.
- reason_admission-[modality], with modality being any of the following cases: acute_chronic_respiratory_failure, acute_renal_failure, alteration_level_of_consciousness, cardiac_insufficiency, cardio_respiratory_arrest, cardiovascular_other, eplilepsy, gastrointestinal_bleeding, hepatic_insufficiency, hypovolaemia, immunosuppressed_infection, infection_other, ischemic_cardiopathy, neurological_other, neuromuscular, pancreatitis, respiratory_other, scheduled_with_complications, serious_infection, severe_trauma, stroke, urgent_uncomplicated, urgent_with_complications, voluntary_intoxication
- patient_category-[modality], with modality being any of the following cases: medical_patient, surgical_patient.
- age
- gender

Culture and antibiogram information

- cultures_indicator
- culture_type-[modality], with modality being one of the following cases: abscess, abscess_abdominal, ascitic_fluid, blood, bronchoalveolar_lavage, bronchoaspirate, catheter_vascular, catheter_introductor_, catheter_telescoped, drainage, drainage_abdominal, exudate_axillary, exudate_inguinal, exudate_nasal, exudate_pharynx, exudate_pressure_ulcer_, exudate_rectal, exudate_wound, faeces, liquid_biliary, liquid_peritoneal, liquid_pleural, secretions, sputum, tissue_biopsy, urine
- culture_type_grouped-[modality], with modality being one of the following cases: blood, catheter, external_wound, faeces, internal_medium, liquid_abdominal, others, respiratory, sputum, surface, urine
- culture_type_grouped_2-[modality], with modality being either clinical_sample or surface
- antibiogram_antibiotic-[modality], with modality being one of the following cases: amikacin, amoxicillin_clavulanic, ampicilin_sulbactam, ampicillin, ampicillin_sulbactam, aztreonam, cefazolin, cefepime, cefotaxime, cefoxitin, cefpodoxime, ceftazidime, ceftriaxone, cefuroxime, cefuroxime_axetil, cefuroxime_axetil_1, cefuroxime_sodium, cephalotin, cephepima, chloramphenicol, ciprofloxacin, clindamycin, colistine, ertapenem, erythromycin, gentamicin, gentamicin_high_load_synergy, imipenem, kanamycin_high_load_synergy, levofloxacin, linezolid, meropenem, mezlocillin, minocycline, moxifloxacin, nalidixic_acid, nitrofurantoin, norfloxacin, ofloxacin, penicillin_g, phosphomycin, piperacilin_tazobactam, piperacillin, piperacillin_tazobactam, teicoplanin, tetracycline, ticarcilin, tigecycline, tobramycin, trimethoprim_sulfamethoxazole, vancomycin
- antibiogram_family-[modality], with modality being one of the following cases: amg, atf, car, cf1, cf2, cf3, cf4, gcc, gli, ibl, lin, mac, mon, otr, oxa, pap, pen, pol, qui, sul, ttc.

Antibiogram results

- pseudomonas_aeruginosa_detected
- resistance-[modality], with modality being one of the following cases: amg, car, cf4, pap, pol, qui

Treatment information

- antibiotics-[modality], with modality being one of the following cases: amikacina_j01gb, amoxicilina_clavulanico_j01cr, ampicilina_j01ca, anfotericina_b_lipidica_j02aa, anfotericina_b_liposomica_j02aa, anidulafungina, aztreonam_j01df, caspofungina_j02a1, cefazolina_j01da, cefepime_j01dc, cefotaxima_j01da, cefoxitina_j01da, ceftazidima_j01da, ceftriaxona_j01da, cefuroxima_j01da, ciprofloxacina_j01ma, claritromicina_j01fa, clindamicina_j01ff, clotrimazol_d01ac, cloxacilina_j01cf, colistina_j01xb, cotrimoxazol_j01ee, daptomicina_j01j3, doxiciclina_j01aa, eritromicina_j01fa, ertapenem_j01dh, fluconazol_j02ac, fosfomicina_j01xx, gentamicina_j01gb, imipenem_j01dh, levofloxacino_j01ma,linez-

olid, meropenem_j01dh, metronidazol_j01ca, metronidazol_j03ca, naftifina_d01ae, nistatina_a01ab, paramomicina_a07aa, penicilina_sodica_j01ce, piperaclina_j01c1a, piperacilina_tazobactam_j01cr, pirimetamina_p01bd, rifaximina_a07a, sulfadiazina_j03a, teicoplanina_j01xa, tigeciclina_p01bd, tobramicina_j01gb, tobramicina_j01gb, vancomicina_j01xa, voriconazol_j02ac
- mechanic_ventilation

ICU features

Number of co-admitted patients

- co-admitted_patients

ICU admission information

- icu_admission_indicator
- icu_origin-[modality], with modality being any of the following cases: anaesthesia, another_hospital, cardiology, cma, dermatology, digestive, general_surgery, gynaecology, haematology, hemodynamics, icu, internal_medicine, nephrology, neurology, obstetrics, oncology, ophthalmology, other_floor, others, otorhinolaryngology, paediatrics, pneumology, psychiatry, rehabilitation, resuscitation_unit, surgery_room, traumatology, urgencies, urology
- icu_destination-[modality], with modality being one of the following cases: another_hospital, at_home, cardiology, dermatology, digestive, endocrinology, floor, general_surgery, gynaecology, haematology, internal_medicine, mortuary, nephrology, neurology, obstetrics, oncology, others, otorhinolaryngology, paediatrics, pneumology, psychiatry, resuscitation_unit, traumatology, urology, voluntary_discharge
- icu_reason_admission-[modality], with modality being one of the following cases: acute_chronic_respiratory_failure, acute_renal_failure, acute_respiratory_failure, alteration_level_of_consciousness, cardiac_insufficiency, cardio_respiratory_arrest, cardiovascular_other, diabetic_decompensation, endocrine_other, eplilepsy, gastrointestinal_bleeding, hepatic_insufficiency, hydroelectrolytic_alteration, hypovolaemia, immunosuppressed_infection, infection_other, ischemic_cardiopathy, neurological_other, neuromuscular, obstetric_pathology, other, pancreatitis, respiratory_other, scheduled_uncomplicated, scheduled_with_complications, serious_infection, severe_arrhythmia, severe_trauma, stroke, urgent_uncomplicated, urgent_with_complications, voluntary_intoxication
- icu_patient_category-[modality], with modality being one of the following cases: medical_patient, paediatrics_patient, surgical_patient
- icu_age
- icu_gender-[modality], with modality being either 0 or 1.

ICU antibiogram results

- icu_pseudomonas_aeruginosa_detected
- icu_resistance-[modality], with modality being one of the following cases: amg, car, cf4, pap, pol, qui

ICU treatment information

- icu_antibiotics-[modality], with modality being one of the following cases: amoxicilina_clavulanico_j01cr, amikacina_j01gb, ampicilina_j01ca, ampicilina_j01ca_, ampicilina_sulbactam, anfotericina_b_convencional_j02aa, anfotericina_b_lipidica_j02aa, anfotericina_b_liposomica_j02aa, anidulafungina, aztreonam_j01df, azitromicina_j01fa, caspofungina_j02a1, cefalozano_tazobactam, cefazolina_j01da, cefepime_j01dc, cefotaxima_j01da, cefoxitina_j01da, ceftazidima_j01da, ceftriaxona_j01da, cefuroxima_j01da, ciprofloxacina_j01ma, claritromicina_j01fa, clindamicina_j01ff, clotrimazol_d01ac, cloxacilina_j01cf, colistina_j01xb, cotrimoxazol_j01ee, daptomicina_j01j3, doxiciclina_j01aa, eritromicina_j01fa, ertapenem_j01dh, fluconazol_j02ac, fosfomicina_j01xx, gentamicina_j01gb, imipenem_j01dh, isavuconazol, itraconazol_j02a1a, levofloxacino_j01ma, linezolid, meropenem_j01dh, metronidazol_j01ca, metronidazol_j03ca, micafungina, naftifina_d01ae, nistatina_a01ab, norfloxacina_j01ma, ofloxacino_j01ma, paramomicina_a07aa, penicilina_sodica_j01ce, piperacilina_j01c1a, piperacilina_tazobactam_j01cr, pirimetamina_p01bd, posaconazol_j02, rifaximina_a07a, sulfadiazina_j03a, teicoplanina_j01xa, tigeciclina_j01aa, tobramicina_j01gb, vancomicina_j01xa, voriconazol_j02ac
- icu_mechanic_ventilation

References

[1] Aminov RI. A brief history of the antibiotic era: lessons learned and challenges for the future. Front Microbiol 2010;1:134.

[2] World-Health-Organization, et al. Antimicrobial resistance: global report on surveillance. World Health Organization; 2014.

[3] Hu X-Y, Logue M, Robinson N. Antimicrobial resistance is a global problem–a UK perspective. Eur J Integr Med 2020;36:101136.

[4] Fridkin SK, Gaynes RP. Antimicrobial resistance in intensive care units. Clin Chest Med 1999;20(2):303–16.

[5] Revuelta-Zamorano P, Sánchez A, Rojo-Álvarez JL, Álvarez-Rodríguez J, Ramos-López J, Soguero-Ruiz C. Prediction of healthcare associated infections in an intensive care unit using machine learning and big data tools. In: XIV mediterranean conference on medical and biological engineering and computing 2016. Springer; 2016, p. 840–5.

[6] Brusselaers N, Vogelaers D, Blot S. The rising problem of antimicrobial resistance in the intensive care unit. Ann Intensive Care 2011;1(1):1–7.

[7] Joshi S. Hospital antibiogram: a necessity. Indian J Med Microbiol 2010;28(4):277–80.

[8] Maurer FP, Christner M, Hentschke M, Rohde H. Advances in rapid identification and susceptibility testing of bacteria in the clinical microbiology laboratory: implications for patient care and antimicrobial stewardship programs. Infect Dis Rep 2017;9(1):18–27.

[9] Beaudoin M, Kabanza F, Nault V, Valiquette L. Evaluation of a machine learning capability for a clinical decision support system to enhance antimicrobial stewardship programs. Artif Intell Med 2016;68:29–36.

[10] Jimenez F, Palma J, Sanchez G, Marin D, Palacios MF, López ML. Feature selection based multivariate time series forecasting: An application to antibiotic resistance outbreaks prediction. Artif Intell Med 2020;104:101818.

[11] Khaledi A, Weimann A, Schniederjans M, Asgari E, Kuo T-H, Oliver A, et al. Predicting antimicrobial resistance in pseudomonas aeruginosa with machine learning-enabled molecular diagnostics. EMBO Mol Med 2020;12(3):e10264.

[12] Chowdhury AS, Call DR, Broschat SL. PARGT: A software tool for predicting antimicrobial resistance in bacteria. Sci Rep 2020;10(1):1–7.

[13] Liu Z, Deng D, Lu H, Sun J, Lv L, Li S, et al. Evaluation of machine learning models for predicting antimicrobial resistance of actinobacillus pleuropneumoniae from whole genome sequences. Front Microbiol 2020;11:48.

[14] Tlachac M, Rundensteiner EA, Barton K, Troppy S, Beaulac K, Doron S. Predicting future antibiotic susceptibility using regression-based methods on longitudinal massachusetts antibiogram data. In: Proceedings of the 11th international joint conference on biomedical engineering systems and technologies (BIOSTEC 2018) - HEALTHINF, Vol. 5. 2018, p. 103–14.

[15] Hernàndez-Carnerero À, Sànchez-Marrè M, Mora-Jiménez I, Soguero-Ruiz C, Martínez-Agüero S, Álvarez-Rodríguez J. Modelling temporal relationships in pseudomonas aeruginosa antimicrobial resistance prediction in intensive care unit. In: Proceedings of the first international AAI4H, advances in artificial intelligence for healthcare workshop: co-Located with the 24th european conference on artificial intelligence (ECAI 2020): Santiago de Compostela, Spain, September 4, 2020. 2020, p. 60–7, http://ceur-ws.org/Vol-2820/AAI4H-12.pdf.

[16] Hernàndez-Carnerero À, Sànchez-Marrè M, Mora-Jiménez I, Soguero-Ruiz C, Martínez-Agüero S, Álvarez-Rodríguez J. Antimicrobial resistance prediction in intensive care unit for pseudomonas aeruginosa using temporal data-driven models. Int J Interact Multimed Artif Intell 2021;6(5):119–33.

[17] Tsymbal A, Pechenizkiy M, Cunningham P, Puuronen S. Handling local concept drift with dynamic integration of classifiers: Domain of antibiotic resistance in nosocomial infections. In: 19th IEEE symposium on computer-based medical systems (CBMS'06). IEEE; 2006, p. 679–84.

[18] Eickelberg G, Sanchez-Pinto LN, Luo Y. Predictive modeling of bacterial infections and antibiotic therapy needs in critically ill adults. J Biomed Inform 2020;109:103540.

[19] Martínez-Agüero S, Mora-Jiménez I, Lérida-García J, Álvarez-Rodríguez J, Soguero-Ruiz C. Machine learning techniques to identify antimicrobial resistance in the intensive care unit. Entropy 2019;21(6):603.

[20] Lewin-Epstein O, Baruch S, Hadany L, Stein GY, Obolski U. Predicting antibiotic resistance in hospitalized patients by applying machine learning to electronic medical records. Clin Infect Dis 2021;72(11):e848–55.

[21] Martínez-Agüero S, Mora-Jiménez I, García-Marqués A, Álvarez-Rodríguez J, Soguero-Ruiz C. Applying LSTM networks to predict multi-drug resistance using binary multivariate clinical sequences. In: Proceedings of starting AI researchers‘ symposium (STAIRS) at the 24th european conference on artificial intelligence (ECAI 2020). 2020.

[22] Mora-Jiménez I, Tarancón-Rey J, Álvarez-Rodríguez J, Soguero-Ruiz C. Artificial intelligence to get insights of multi-drug resistance risk factors during the first 48 hours from ICU admission. Antibiotics 2021;10(3):239.

[23] Escudero-Arnanz Ó, Rodríguez-Álvarez J, Mikalsen KØ, Jenssen R, Soguero-Ruiz C. On the use of time series kernel and dimensionality reduction to identify the acquisition of antimicrobial multidrug resistance in the intensive care unit. 2021, arXiv preprint arXiv:2107.10398.

[24] Pascual-Sánchez L, Mora-Jiménez I, Martínez-Agüero S, Álvarez-Rodríguez J, Soguero-Ruiz C. Predicting multidrug resistance using temporal clinical data and machine learning methods. In: 2021 IEEE international conference on bioinformatics and biomedicine. BIBM, IEEE; 2021, p. 2826–33.

[25] Ismail Fawaz H, Forestier G, Weber J, Idoumghar L, Muller P-A. Deep learning for time series classification: a review. Data Min Knowl Discov 2019;33(4):917–63.

[26] Zhao B, Lu H, Chen S, Liu J, Wu D. Convolutional neural networks for time series classification. J Syst Eng Electron 2017;28(1):162–9.

[27] Bagnall A, Janacek G. A run length transformation for discriminating between auto regressive time series. J Classification 2014;31(2):154–78.

[28] Kini BV, Sekhar CC. Large margin mixture of AR models for time series classification. Appl Soft Comput 2013;13(1):361–71.

[29] Bagnall A, Lines J, Bostrom A, Large J, Keogh E. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min Knowl Discov 2017;31(3):606–60.

[30] Li H. On-line and dynamic time warping for time series data mining. Int J Mach Learn Cybern 2015;6(1):145–53.

[31] Lines J, Taylor S, Bagnall A. Time series classification with HIVE-cote: The hierarchical vote collective of transformation-based ensembles. ACM Trans Knowl Discov Data 2018;12(5).

[32] Shallue CJ, Vanderburg A. Identifying exoplanets with deep learning: A five-planet resonant chain around kepler-80 and an eighth planet around kepler-90. Astron J 2018;155(2):94.

[33] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80.

[34] Guo A, Beheshti R, Khan YM, Langabeer JR, Foraker RE. Predicting cardiovascular health trajectories in time-series electronic health records with LSTM models. BMC Med Inform Decis Mak 2021;21(1):1–10.

[35] Sikora A, Zahra F. Nosocomial infections. 2021.

[36] Saderi H, Owlia P. Detection of multidrug resistant (MDR) and extremely drug resistant (XDR) p. aeruginosa isolated from patients in tehran, Iran. Iran J Pathol 2015;10(4):265.

[37] Tam VH, Chang K-T, Abdelraouf K, Brioso CG, Ameka M, McCaskey LA, et al. Prevalence, resistance mechanisms, and susceptibility of multidrug-resistant bloodstream isolates of pseudomonas aeruginosa. Antimicrob Agents Chemother 2010;54(3):1160–4.

[38] Diamantidis N, Giakoumakis EA. Don't care values in induction. Artif Intell Med 1996;8(5):505–14.

[39] Li J, Cheng K, Wang S, Morstatter F, Trevino RP, Tang J, Liu H. Feature selection: A data perspective. ACM Comput Surv 2017;50(6):1–45.

[40] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 1994;5(2):157–66.

[41] Gers FA, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM. Neural Comput 2000;12(10):2451–71.

[42] Agarap AF. Deep learning using rectified linear units (relu). 2018, arXiv preprint arXiv:1803.08375.

[43] Talathi SS, Vartak A. Improving performance of recurrent neural network with relu nonlinearity. 2015, arXiv preprint arXiv:1511.03771.

[44] Baldi P, Sadowski PJ. Understanding dropout. Adv Neural Inf Process Syst 2013;26:2814–22.

[45] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016, p. 117–9, ch. 5.

[46] Kamalov F, Smail L, Gurrib I. Stock price forecast with deep learning. In: 2020 international conference on decision aid sciences and application. DASA, IEEE; 2020, p. 1098–102.

[47] Fawcett T. An introduction to ROC analysis. Pattern Recognit Lett 2006;27(8):861–74.

[48] Jović A, Brkić K, Bogunović N. A review of feature selection methods with applications. In: 2015 38th international convention on information and communication technology, electronics and microelectronics. MIPRO, IEEE; 2015, p. 1200–5.

[49] Hooker S, Erhan D, Kindermans P-J, Kim B. A benchmark for interpretability methods in deep neural networks. In: Advances in neural information processing systems, Vol. 32. Curran Associates, Inc.; 2019.

[50] Montavon G, Samek W, Müller K-R. Methods for interpreting and understanding deep neural networks. Digit Signal Process 2018;73:1–15.

[51] Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. In: International conference on machine learning. PMLR; 2017, p. 3319–28.

[52] Cerliani M. Feature importance with time series and recurrent neural network. 2021.

[53] Räsänen O, Pohjalainen J. Random subset feature selection in automatic recognition of developmental disorders, affective states, and level of conflict from speech. In: Proceedings of the annual conference of the international speech communication association, INTERSPEECH. 2013, p. 210–4.

[54] Padmaja DL, Vishnuvardhan B. Comparative study of feature subset selection methods for dimensionality reduction on scientific data. In: 2016 IEEE 6th international conference on advanced computing. IACC, IEEE; 2016, p. 31–4.

[55] Breiman L. Random forests. Mach Learn 2001;45(1):5–32.

[56] Breiman L. Statistical modeling: The two cultures (with comments and a rejoinder by the author). Statist Sci 2001;16(3):199–231.

[57] Huang N, Lu G, Xu D. A permutation importance-based feature selection method for short-term electricity load forecasting using random forest. Energies 2016;9(10):767.

[58] Fisher A, Rudin C, Dominici F. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. J Mach Learn Res 2019;20(177):1–81.

[59] Gómez-Ramírez J, Ávila-Villanueva M, Fernández-Blázquez MÁ. Selecting the most important self-assessed features for predicting conversion to mild cognitive impairment with random forest and permutation-based methods. Sci Rep 2020;10(1):1–15.

[60] Xie J, Xu B, Chuang Z. Horizontal and vertical ensemble with deep representation for classification. 2013, arXiv:1306.2759.

[61] Kolář M, Urbánek K, Látal T. Antibiotic selective pressure and development of bacterial resistance. Int J Antimicrob Ag 2001;17(5):357–63.

[62] Skalet AH, Cevallos V, Ayele B, Gebre T, Zhou Z, Jorgensen JH, et al. Antibiotic selection pressure and macrolide resistance in nasopharyngeal streptococcus pneumoniae: a cluster-randomized clinical trial. PLoS Med 2010;7(12):e1000377.

[63] Figuera C, Mora-Jiménez I, Guerrero-Curieses A, Rojo-Alvarez J, Everss E, Wilby M, Ramos-López J. Nonparametric model comparison and uncertainty evaluation for signal strength indoor location. IEEE Trans Mob Comput 2009;8(9):1250–64.

[64] Soguero-Ruiz C, Gimeno-Blanes F-J, Mora-Jiménez I, Martínez-Ruiz MP, Rojo-Álvarez J-L. On the differential benchmarking of promotional efficiency with machine learning modeling (i): Principles and statistical comparison. Expert Syst Appl 2012;39(17):12772–83.

[65] Efron B. The bootstrap and modern statistics. J Amer Statist Assoc 2000;95(452):1293–6.

[66] Martínez-Agüero S, Soguero-Ruiz C, Alonso-Moral JM, Mora-Jiménez I, Álvarez-Rodríguez J, Marques AG. Interpretable clinical time-series modeling with intelligent feature selection for early prediction of antimicrobial multidrug resistance. Future Gener Comput Syst 2022;133:68–83.

[67] Chesneau C, Hebiri M. Some theoretical results on the grouped variables lasso. Math Methods Statist 2008;17(4):317–26.

[68] Widmer G, Kubat M. Learning in the presence of concept drift and hidden contexts. Mach Learn 1996;23(1):69–101.

[69] Han M, Liu X. Feature selection techniques with class separability for multivariate time series. Neurocomputing 2013;110:29–34.

[70] Fang L, Zhao H, Wang P, Yu M, Yan J, Cheng W, et al. Feature selection method based on mutual information and class separability for dimension reduction in multidimensional time series for clinical data. Biomed Signal Process Control 2015;21:82–9.

[71] Sun Y, Li J, Liu J, Chow C, Sun B, Wang R. Using causal discovery for feature selection in multivariate numerical time series. Mach Learn 2015;101(1):377–95.