Expert Systems With Applications 227 (2023) 120252

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A streaming data visualization framework for supporting decision-making in the Intensive Care Unit

Miguel A. Mohedano-Munoz [b], Cristina Soguero-Ruiz [a], Inmaculada Mora-Jiménez [a], Manuel Rubio-Sánchez [a], Joaquín Álvarez-Rodríguez [c], Alberto Sanchez [a,*]

[a] Universidad Rey Juan Carlos, C/Tulipán s.n., Móstoles, 28933, Madrid, Spain
[b] U-tad - University Center for Technology and Digital Art, C/ Playa de Liencres 2 bis, Las Rozas, 28290, Madrid, Spain
[c] University Hospital of Fuenlabrada, Camino del Molino, Fuenlabrada, 28942, Madrid, Spain

ARTICLE INFO

Keywords:
Streaming data visualization
Dynamic data
Dimensionality reduction
Machine learning
Multi-drug resistant bacteria

ABSTRACT

The number of reporting activities in real time has increased over the last years. This situation has pushed the need for providing real time analysis and visualizations to support decision-making. We propose a visualization framework for exploratory data analysis of multivariate data streams that relies on dimensionality reduction and machine learning techniques for plotting the data in two dimensions. Users can demarcate regions of interest for their study, and use them to make predictions or to decide when to train a new model. The knowledge gained from these visualizations allows users to: (i) characterize the data stream scenario; (ii) track the evolution of a case of interest; and (iii) configure and raise alarms according to the user-defined regions. We illustrate the effectiveness of our proposal through a case study analyzing real-world streaming data to identify patients with multi-drug resistant bacteria when they are in a hospital intensive care unit. Our visualization framework enables the patient follow-up which can allow clinicians to support decisions about the health status evolution of a particular patient. This could provide information for deciding on a particular treatment or whether to isolate patients with a high risk of having multi-drug resistant bacteria since their presence boosts infections in intensive care units.

1. Introduction

The volume, velocity and variety of data production are increasing in recent years due to cheaper data storage devices and the inclusion of data recording technologies in everyday life (Kitchin, 2014). Some fields of application with a high rate of data generation, such as analysis of network traffic, clinical, or sensor data, have to deal with processing data streams obtained continuously. This requires specific streaming architectures to produce and consume the data almost in real-time. Specifically, this data has to be processed sequentially, usually considering time windows, to perform different data analysis tasks. These tasks can range from simple operations, such as variable filtering or correlation computing, to more advanced data mining techniques (Ikonomovska, Loskovska, & Gjorgjevik, 2007). Retrieving information from streaming processes is difficult due to its changing nature, which conditions the entire analysis pipeline (Mansmann, Fischer, & Keim, 2012).

Information visualization allows analysts to combine their domain knowledge with their ability to visually gain insights about relationships and underlying patterns in the data. Visualization in data mining processes involves domain experts in a more immersive way, allowing experts to easily handle noisy datasets. Moreover, it is acknowledged that user interaction is often essential for visual analytics (Stadler, Donlon, Siewert, Franken, & Lewis, 2016) since it facilitates and speeds up tasks related to data exploration, decision-making, or drawing conclusions. However, visualizing data streams adds difficulties to the analysis process, such as the computing architecture to support it, how to manage and understand the gradual changes that occur continuously in data, and the possible associated loss of context (Krstajić & Keim, 2013).

Raw data are usually high-dimensional and therefore susceptible to the curse of dimensionality when considering Machine Learning (ML) algorithms (Bellman, 1957). One of the strategies to tackle the curse of dimensionality is to consider dimensionality reduction (DR) algorithms (Friedman, Hastie, & Tibshirani, 2001). These methods apply linear or nonlinear transformations to map data into lower-dimensional spaces (two or three, for visualization purposes). Some supervised DR methods, like Linear Discriminant Analysis (LDA) (McLachlan, 2004) or Large Margin Nearest Neighbours (LMNN) (Domeniconi, Gunopulos, &

* Corresponding author.
E-mail addresses: miguel.munoz@u-tad.com (M.A. Mohedano-Munoz), cristina.soguero@urjc.es (C. Soguero-Ruiz), inmaculada.mora@urjc.es (I. Mora-Jiménez), manuel.rubio@urjc.es (M. Rubio-Sánchez), joaquin.alvarez@salud.madrid.org (J. Álvarez-Rodríguez), alberto.sanchez@urjc.es (A. Sanchez).

https://doi.org/10.1016/j.eswa.2023.120252
Received 10 November 2022; Received in revised form 11 April 2023; Accepted 22 April 2023
Available online 29 April 2023

Peng, 2005), map the data in order to separate different classes in a low-dimensional space. Other unsupervised methods like t-distributed Stochastic Neighbor embedding (t-SNE) (Maaten & Hinton, 2008) or Uniform Manifold Approximation and Projection (UMAP) (McInnes, Healy, Saul, & Großberger, 2018), can be employed as an approach to clustering tasks as they try to group similar data in the lower-dimensional space.

In this paper, we propose a visualization framework for exploratory data analysis based on DR and ML techniques. Our framework allows users to analyze multivariate data streams and preserve the so-called "mental map" (Eades, Lai, Misue, & Sugiyama, 1991) that allows them to mentally organize and integrate new information into their existing knowledge structures. The visualization interface enables users to define regions of interest and to monitor the temporal evolution of cases by considering a new record each time an event occurs. Additionally, the interface can also be used to raise alerts when the evolution of the new records tends towards a risk condition. When new records are received (in each time window) they are mapped using the current DR transformation, similar to an out-of-sample forecast. The current DR transformation is obtained by considering historical records (the training set). Note that the DR transformation is obtained using training data collected at some point in time. Therefore, it may not represent a future context adequately. In practice, users can visualize the new records and decide whether to use them to compute a new DR transformation.

The review presented by Dasgupta, Arendt, Franklin, Wong, and Cook (2018) organizes the different kinds of visualizations for streaming data into three categories, according to their goals: (i) active monitoring of the evolution process, (ii) follow-up of an event, and (iii) release situational awareness. The review assesses 22 proposals, but only one fits into the three defined categories. Our proposal also belongs to the three categories due to the possibilities of: (i) monitoring current and historical scenarios, (ii) following-up the progress of a specific case, and (iii) raising alarms if a case transitions into a risky state. For this purpose, we combine the use of both information visualization and ML techniques.

As a case study, we have applied the proposed framework in the intensive care unit (ICU) of the University Hospital of Fuenlabrada, a public hospital in the area of Madrid (Spain). Monitoring both the health status evolution of a particular patient and also the global health condition is especially relevant in ICU settings. Patients admitted in this clinical unit have a critical health status and it is essential to follow them up in order to act as soon as possible and therefore avoid a non-reversible worsening. Therefore, the health-status real-time visualization of a patient's condition is a very relevant tool for clinicians. In collaboration with the University Hospital of Fuenlabrada, we analyzed the Electronic Health Records (EHR) of patients with multi-drug resistant (MDR) bacteria in the ICU. The provided dataset, which corresponds to real-world data over a period of one year, feeds our framework as a data stream. The main purpose of the proposed framework is the early detection of MDR bacteria, which could provide crucial clinical information for deciding on a particular treatment or whether to isolate patients with a high risk of having MDR. Towards that end, the visualization tool enables clinicians: (i) to characterize the EHR data stream scenario by plotting the data in two dimensions; (ii) to follow-up and predict the evolution of the health condition of a patient of interest; and (iii) to define warning regions and configure and raise alarms according to these regions to potentially identify patients at risk of having MDR bacteria. Furthermore, clinicians can decide when to update training data to generate a new model, depending on whether they are interested in considering long or short periods of time.

The paper is structured as follows. Section 2 presents the state of the art in streaming visualization. Section 3 describes our framework and how it fits in with the different proposals on how data streaming visualization should be. Section 4 presents a case study, considering real-world data streams registered in the ICU. Finally, in Section 5 we draw conclusions and propose future lines of work.

2. Related work

Streaming visualizations can be divided in two types according to their data treatment. On the one hand, the first type implies that data is not stored in any way and users only can visualize a snapshot of the current scenario. On the other hand, the second class involves storing dynamic data and retrieving it for a data stream. We focus on this second type since we allow analysts to store the evolution of data to train ML models. Visualizations for dynamic data have to address some issues related to the way humans detect changes and movements of the analyzed elements over time in graphical representations. Specifically, these visualizations have to try to preserve the context of the data and their mental map as they are obtained from new data records. This includes managing the volume and speed of the stream and being able to identify changes in the represented mappings (Dasgupta et al., 2018; Endert, Pike, & Cook, 2014).

Most of the user-friendly alternatives to create data visualizations do not provide native integration with streaming data architectures, such as Tableau (Inseok & Hyejung, 2017), or it requires adherence to a payment plan, as it occurs with Qlik Replicate (QlikTech International, 2022). This has prompted the data visualization community to create ad-hoc graphical interfaces to deal with data streams. Specifically, Dasgupta et al. (2018) present visualization solutions for streaming data. Ten of these visualization proposals (Alsakran, Chen, Zhao, Yang, & Luo, 2011; Bach et al., 2016; Forbes, Höllerer, & Legrady, 2010; Gotz & Stavropoulos, 2014; Li & Baciu, 2014; Moere, 2004; Satyanarayan, Russell, Hoffswell, & Heer, 2015; Steiger et al., 2014; Wong, Foote, Adams, Cowley, & Thomas, 2003; Xie & Qiu, 2007) use scatter plots as a type of mapping, as in our proposal. Some of them are focused on data management and performance, leaving the implementation of the visualization display to the users (Forbes et al., 2010; Satyanarayan et al., 2015). From the rest of proposals, we want to emphasize specifically one. Steiger et al. (2014) define a visualization system to analyze anomalies on sensor data streaming. The tool uses a DR algorithm to present the obtained data and determine the area of influence of the clusters from a training stage. As a complement, they also offer a calendar view. This proposal is the closest to our tool in the reviewed literature. The main point in common is the use of linear DR projections to define different areas of anomalies by using the k-Means clustering algorithm. We differ in the use of other linear and non-linear DR methods in our process, such as LDA, and UMAP, resulting in linked mappings that can increase the knowledge obtained by looking at the information from different points of view.

The tools shown in Dasgupta et al. (2018) are generally designed for their application in specific fields. For instance, most of the solutions applied are related to the analysis of social network data, where geolocation is an important feature. This is the case with ScatterBlogs2 (Bosch et al., 2013), where the authors use the superposition of Twitter records on a geographical map. Analysts train a Support-Vector Machine classifier to determine the messages that are relevant to the analysis of a topic of interest. Then they visualize through a color map the relevance of the tweets to track an event. Whisper (Cao et al., 2012) is also related to the use of social media analysis to track a process. This tool is developed to focus on data streams between user groups, such as geographic location, and a topic of interest. The rest of tools cover other areas, like cybersecurity monitoring (Fischer & Keim, 2014). The framework that we present in this paper is not restricted to a particular field and it can be applied to monitor trends, identify patterns, and make informed decisions in different application fields, such as finance, healthcare and social media. In particular, we will illustrate it by using a case study based on healthcare data for monitoring the patient health status.

Other tools and methods in the literature show the use of incremental algorithms to project new records and deal with the inherent heterogeneity of streaming data. This is the case of Neves et al. (2020),

which describes an incremental dimensionality reduction method without the need to revisit the old records. The aim of providing analysts with an efficient way to deal with the difficulties of heterogeneous data by using incremental algorithms is also present in Fujiwara et al. (2019). These tools are useful, but they differ with our proposal in that we assume a fixed set of features, and even though incremental algorithms may be useful if applied on top of the proposed method, it is beyond the scope of this study. Another research (Fekete & Primet, 2016) presents how to reduce the computational time to present data by performing progressive analysis on incremental data. This proposal is well defined, and points to future steps in our research, such as may be the use of progressive algorithms in exploratory data analysis, but does not allow applying domain expert insight and judgment as our proposal. By leveraging the streaming nature of trajectory data, the interactive visualization of hot routes in real-time, by using parallel processing on GPU, has proven to be effective (Gomes, dos Santos, Vidal, da Silva, & de Macêdo, 2018; Li, Han, Lee, & Gonzalez, 2007). The concept of Progressive Visual Analytics is discussed in more depth in Angelini, Santucci, Schumann, and Schulz (2018).

Specifically in healthcare, visualization tools can provide some advantages to support clinical decision making (Shneiderman, Plaisant, & Hesse, 2013). Among these advantages, these tools allow clinicians to obtain a proper analysis, extract clinical knowledge, characterize and understand the similarity between scenarios, visualize relationships through comparative analyses, and present risks and warnings. However, there are still some limitations that should be addressed, including the ability to stream patient datasets, link or coordinate the multiple views, group patients, and be interactive and intuitive. The review by Dunn, Burgun, Krebs, and Rance (2016) presents current tools designed to meet the needs of multidimensional clinical data visualization.

The literature on data streaming, ML and healthcare focuses on performance and robustness of the streaming architectures, rather than on visualization techniques. However, a recent review (Levy-Fix, Kuperman, & Elhadad, 2019) has compiled applications that employ heuristics, ML and/or data visualization for clinical decision support (CDS). This review groups the CDS into three classes: (i) infobuttons, which collects information from external resources such as text files of medical sources, to show results that match the search criteria; (ii) content aggregation and organization (CAO), which reorganizes clinical information recorded in the health information system to extract knowledge and support decision making; and (iii) alert CDS, which aims to generate alerts based on clinical data and ML techniques. Our proposal can fit both in CAO and Alert CDS.

Apart from some works related to the initial stages of the data analysis process in the ICU (Kotanidou, De Georgia, Kaffashi, Jacono, & Loparo, 2015; Sun et al., 2020), due to the novelty of data streaming and the challenges previously described there are just some preliminary works related to the study of ICU streams. For example, Lauritsen et al. (2020) presented an explainable early warning score system based on artificial intelligence for early detection of critical illness for early detection of acute critical illness. In the same line, an interpretable and generalizable survival model that predicts sepsis onset in the ICU 4, 6, 8, and 12 h in advance was proposed in Nemati et al. (2018). In Blount et al. (2010), a platform to collect and analyze data from a neonatal ICU, which includes a method to raise alarms for the most frequent diseases in this environment was proposed. Sow, Biem, Sun, Hu, and Ebadollahi (2010) describe a system capable of predicting the prognosis of patients on ICU environments based on physiological patient data streams. A different approach was proposed in Rejab, Nouira, and Amri (2014), where the authors created a system to tackle the storage and collection of ICU streaming data. Brich et al. (2020) present a method to perform comparative analysis among patients admitted in the ICU by using Time Curves (Bach et al., 2016). Although we believe that is an interesting perspective, we consider that the use of overlapping areas to compare patients with each others is not the most clear way to represent them. Our proposal goes beyond, implementing a more flexible system which makes it possible to consider and evaluate new DR and ML techniques. The proposed system allows users to visualize data almost in real time, and to warn, monitor, and analyze data streams.

3. Streaming visualization proposal for healthcare management

This section describes the specific requirements for visualizing healthcare streaming data and details the proposed architecture shown in Fig. 1. In this architecture the clinical records are sent to a Kafka queue for on-demand retrieval. ML and DR methods can be applied interactively. The interaction with the DR transformation/ML module helps users to refine the mapping to achieve the visualization that best fits the scenario.

3.1. Requirements

Firstly, we have taken into account the difficulties described in Dasgupta et al. (2018) to come up with our solution. Additionally, we have incorporated some requirements suggested by the clinicians to visualize healthcare data streams:

- **R1. The framework has to preserve the mental map for as long as the analyst chooses**: The mental map allows the domain experts to follow the evolution of different events. In our case study, we track medical patient records. Note that the record of an ICU patient is in continuous evolution, registering the specific health conditions at a particular moment, for example, after a microbiology test is performed. Therefore, each patient can appear with a different number of records in the training set, depending on the number of associated events and the length of the considered time interval. The inclusion of these records in the dataset is useful to characterize (both in training and testing) the patient's progress. This mental map, linked to the structure of the visual representation, is maintained as long as the DR transformation is not updated. If the scenario changes and the transformation is not accordingly updated, immediate forecasts may no longer be valid and the performance of the current model worsens. Then, it may be convenient to train a new model including the most recent records to try to adapt it to the new situation. Since this may involve changing the analysts' mental map, it is necessary to allow them to decide freely when they wish to obtain a new model (which can incorporate long periods of time).
- **R2. Define the refresh rate of the dataset according the analyst and system requirements:** The refresh rate for streaming analytics depends on the analysts' objectives and the application field. Some fields, such as urban mobility or live network monitoring to prevent cyber attacks, can need a high refresh rate than in the healthcare scenario. It is necessary to ensure that the data presentation rate does not exceed the available computing capacity or the domain expert's analysis capability. Through the user interface, users would set how often the dataset should be updated in case there were new events. This would allow users to adapt the refresh rate to the limitations in receiving, processing and analyzing the data.
- **R3. Represent the time variable in the visualizations:** The course of time should be represented in the visualization to guide users in their decisions. There are different ways to include time in the graphs, as described in Dasgupta et al. (2018). In this paper, we show the time evolution by changes in the opacity of markers that represent records of cases of interest. The incoming records are added to the previous ones in the mapping, and the opacity of the old records is modified to represent their time difference with respect to the currently represented instant. The use of different

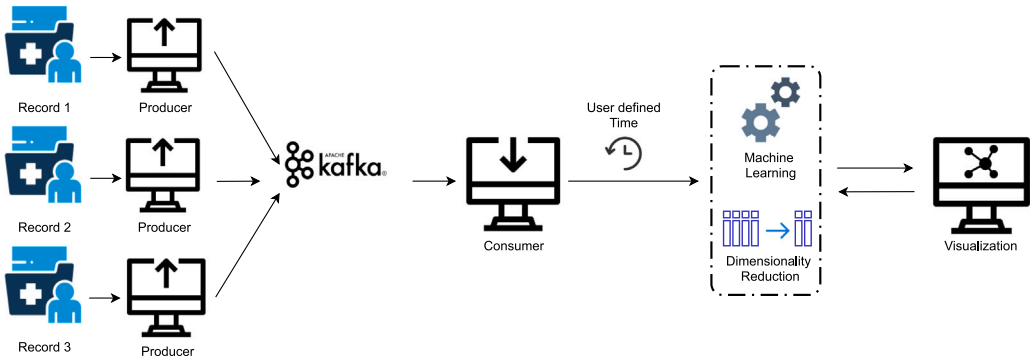

Fig. 1. Diagram of our streaming architecture.

opacity values in a multi-class scatter plot with several points may result in unexpected colors composition due to point occlusion. This can be corrected through the interactive chart by hiding particular records or by zooming in on a target area. We have also included the option to display only the records for a defined size time window and let users to move through the representation by the slider control included under the visualization as mitigation of the time representation problem.

- **R4. The framework has to preserve historical data**: Preserving previous entries of the data stream can be important to try to improve ML models. The consideration of previous records makes it possible to increase the size of the training set and, potentially, to generate more accurate models and better predictions for new incoming data. Historical records can be used for training even when they are hidden in the visualizations. The newly acquired data are mapped according to the calculated model in an out-of-sample approach. In our tool we have included the option to specify the time window used to train the model. The idea is to provide a simple method so that users can focus on analyzing specific scenarios where they know interesting cases have occurred or to avoid the effects of seasonality in the data used to train the DR models.
- **R5. The framework has to enable analyst to track the evolution of a patient**: In different fields it is important to be able to track any case of interest. Our approach represents the case to be tracked using a different marker in the visualization. This allows users to quickly identify the case of interest and follow the evolution. To improve the visual follow-up of a case, we have linked consecutive projections of the same case in different time instants. Note that the possibility of visually following a case can be helpful for analysts if combined with a situational awareness system based on warning areas.
- **R6. The framework has to present situational awareness**: In the use of a streaming visualization tool as a monitoring system is important to configure alarms to detect patients who potentially transit to a state of risk, before their health status worsens. We enable analysts to set alert regions, based on their perception and domain knowledge, directly on the visualization mapping. Through these regions, e.g. it is possible to determine which patients may be at risk as their records are updated in the stream. It would be desirable that this situational awareness could be assisted by machine learning techniques but without eliminating the clinician's perception and domain knowledge as a decision-making factor.

Fig. 2 shows our interface for streaming data visualization in healthcare. Clinicians can create up to four interactive visualizations simultaneously by utilizing various dimensionality reduction methods for comparative purposes, based on a selected data source and features to be analyzed. Our prototype also provides descriptive statistics and measurements of the importance of each feature according to Sanchez, Raya, Mohedano-Munoz, and Rubio-Sánchez (2020) (see bottom-left panel) to provide CDS. Since we present a proposal with different requirements to be considered, we separate our streaming architecture and our visualization proposal. We will describe the elements related to data collection and treatment in the following subsection. In addition, we will explain visualization issues of our framework in Section 3.3.

3.2. Architecture

A typical streaming data architecture consists of a real time producer that sends the raw data, a consumer to catch the messages, a system to perform real time analysis, and a dashboard to show the data. This kind of architecture needs to present independence between the producer and the consumer, provide persistent data storage until the consumer retrieves the information, be ready for message flow peaks, and be fault tolerant (Dunning & Friedman, 2016).

We choose Apache Kafka (Kreps, Narkhede, Rao, et al., 2011) to manage our data stream. This technology allows us to manage the entry of data streams under a publisher-subscriber pattern. We linked the data production to the Kafka queue through a Python producer. This structure allows data to be added to the Kafka queue from different sources and events in the patients' medical records. The producer serializes the raw data in JSON format, and sends it to the Kafka queue at the same time as it is recorded. It also preprocesses the raw data by correcting invalid format entries. We also linked the Kafka queue with our application through a Python consumer.

The consumer incorporates the data to our main dataframe every time that the refresh event is activated. The refresh event is determined by the update of the time window defined in the user interface. Then, users can specify the parameters of the DR and ML algorithms interactively (see Fig. 2). The main limiting factor of the architecture for dealing with streaming data is the execution time of the ML and DR algorithms. This can be managed by changing the query intervals of the consumers to the Kafka queue. Based on the visualization, clinicians can decide whether to train a new model or continue using it during the next current refresh events. It also allows them to graphically demarcate regions of interest for their study, which can be used to highlight patient records within or outside this region, to use it as a warning signal.

As a dashboard, we have developed a Python application where clinicians, with the help of visualization experts, can set the visualization parameters and compute the different projections. We have implemented it using Plotly Dash (Plotly, 2021) to manage the user interactions and start the Flask (Flask, 2020) server. The sci-kit learn library (Pedregosa et al., 2011) is used as basis for the ML and DR algorithms. We have published the tool and its source code online (http://monkey.etsii.urjc.es/healthcare-streaming-analytics/dynamic-and-interactive-visual-analytics). Its pure Python architecture allows visualization experts to easily add new functionalities and adopt it to their needs.

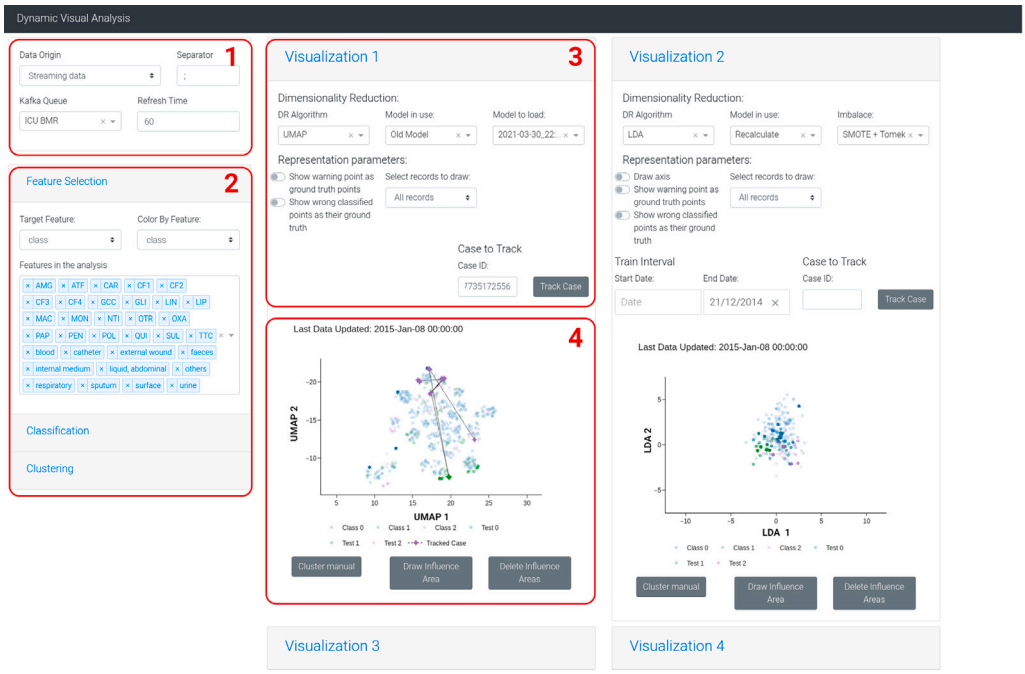

Fig. 2. Dashboard for streaming data visualization in healthcare. Domain experts can set parameters of visualization and machine learning algorithms through the user interface, as well as to create alert areas. Clinicians can also support their decision-making through descriptive statistics. Users can select the data source from the area marked as 1, selecting the features to use and defining the classification and clustering parameters from the section marked as 2. For Visualization 1, users can set its parameters in the field marked as 3 to obtain an interactive visualization in the space marked as 4. This process is similar for each of the desired visualizations (from Visualization 1 to Visualization 4).

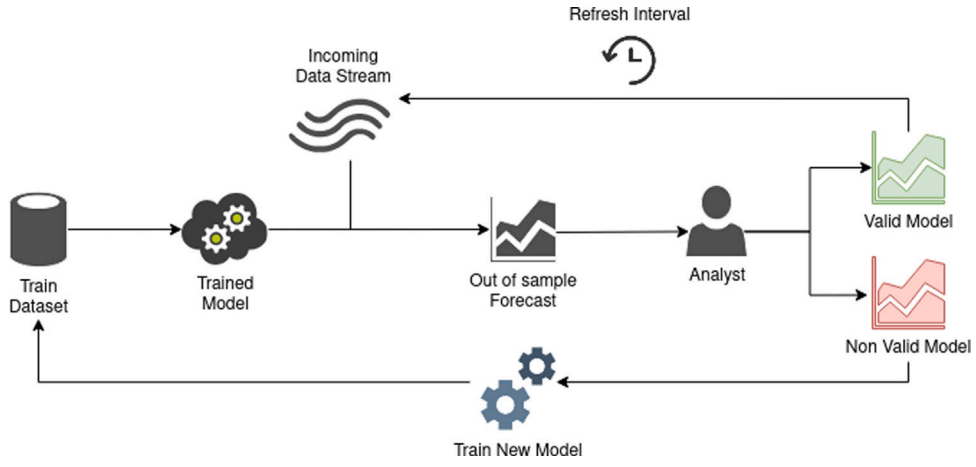

Fig. 3. Proposal for a streaming analytics pipeline. The analyst has the option to assess the validity of the previously trained models through the visualizations. If the model is considered to be valid, it will be employed to map the incoming stream. Otherwise, the analyst can train a new model.

3.3. Visualization

The target of our visualization tool is to provide support so that healthcare personnel can make predictions about current and future status of a patient. Domain experts can gain a better understanding of the current scenario and contribute to the analysis of their field of knowledge. To achieve this goal, we propose a visualization pipeline and an associated interactive prototype that follows the visual-information seeking mantra (Heer & Shneiderman, 2012; Shneiderman, 1996).

The proposed pipeline shown in Fig. 3 allows clinicians to decide which are the most suitable models from visualizations obtained after applying different DR algorithms. Analysts continue to use the selected models as long as they consider them effective to perform out-of-sample forecasts with the incoming data. When users decide that the trained model is no longer valid for the current situation, the model can be trained again with data between the dates of interest. The idea is to fit the incoming data records to this new model. In addition, the prototype allows clinicians to display only a subset of selected records of their interest.

The prototype considers different supervised and unsupervised algorithms for DR. We considered the following algorithms that allow us to map incoming data records according to the established model: Principal Component Analysis (PCA) (Jolliffe & Cadima, 2016), Multidimensional Scaling (MDS) (Cox & Cox, 2000), Locality Preserving Projection (LPP) (He, 2005), Locally Linear Embedding (LLE) (Roweis & Saul, 2000), LDA, LMNN and UMAP. Each algorithm has a different objective and can help analysts to extract knowledge from the dataset. For instance, on the one hand, the objective of the supervised algorithm LDA is to maximize the classes separability. The most relevant features for every class, and their importance (Sanchez et al.,

2018), can be characterized through the resulting linear transformation (Rubio-Sánchez, Raya, Diaz, & Sanchez, 2015). On the other hand, unsupervised algorithms, such as UMAP, can find underlying cluster structure in the data according to record similarity. The available algorithms are explained through a tool tip. We have included in the tool the possibility of oversampling and undersampling the training set to deal with the effects of apply dimensionality reduction algorithms on cases of streams with a high imbalanced rate by using the imbalanced-learn Python library (Lemaître, Nogueira, & Aridas, 2017). DR algorithms that are not sensitive to imbalanced classes are executed with the whole training set. The options considered are: Random undersampling, Tomek links (Tomek, 1976), cluster centroids, SMOTE (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) and SMOTE combined with Tomek links (Batista, Bazzan, Monard, et al., 2003). Undersampling with the Tomek links algorithm can be used before training to clean noisy records.

Clinicians can extract different information from the visualizations according to the parameters and the DR algorithms. Once they consider the obtained mapping as a reasonable one, they can interact with the dashboard (see Fig. 2). Clinicians can set points defining the vertices of a polygon to automatically delimit a region of interest, e.g., the region where most points of a certain class have been projected. From these regions they can detect whether the incoming out-of-sample forecast may have a high probability of belonging to the class of interest. As a support for this analysis, clinicians can highlight the points inside or outside the polygon, depending on the objective of the analysis. In addition, analysts can compare the perception obtained from the graphical classifier (the area built from a polygon defined in the mapping) with classification methods from ML algorithms. As a demonstration, we have implemented a k-NN classifier (Bishop, 2006) that can be applied on the mapping of all training points. In its most simple variant (voting k-NN), the class membership of a new sample is determined to be the majority class among its nearest (most similar) samples. As a consequence, its use is very intuitive in 2D even for non-experts in machine learning, which enables clinicians to directly visualize its performance in low-dimensional environments as our 2D mapping. Note that, according to Cover and Hart (1967), the asymptotic error rate of the voting k-NN rule is less than twice the Bayes error probability. We have also implemented random oversampling (Ramyachitra & Manikandan, 2014) in the classification pipeline to mitigate the effects of imbalance associated with the streaming data.

Sometimes the high number of events over time for the entire data stream scenario can interfere with the clinician's mental map and he/she may want to reduce the number of represented records. In other cases, the analyst's objective is to understand the whole scenario, which requires representing all the points involved. Therefore, we represent all the points and let the clinicians decide whether to hide certain cases (e.g. patients with a better health status) by means of a filtering option.

4. Case study

In this section we describe the use of the visualization framework for streaming data in a hospital ICU. In particular, our analysis is focused on the presence of multidrug-resistant (MDR) bacteria, since they boost the adverse impact of infections in ICUs (Martínez-Agüero et al., 2022; Mora-Jiménez, Tarancón-Rey, Álvarez-Rodríguez, & Soguero-Ruiz, 2021). In this analysis, we, as visualization experts, have provided support to clinicians. Specifically, to the head of the ICU service and his team from the University Hospital of Fuenlabrada, in the use of the framework and dimensionality reduction methods. After its use to follow up patients, the ICU clinicians positively evaluated the tool, especially its ability to interact with the visualization in order to generate warning regions (one or more) based on their knowledge and needs. They also noted as beneficial the possibility of model retraining when new data suggests the need for a better fitted model.

4.1. Multi-drug resistance bacteria fundamentals

Antibiotics have been powerful drugs to treat certain infections caused by bacteria. Since the penicillin was discovered in 1928, the use of antibiotics has saved a lot of lives. However, the misuse and overuse of antibiotics is contributing to change the bacterial environment and therefore to increase the number of bacteria resistant to current antibiotics. Indeed, MDR bacteria, i.e., bacteria which are resistant to multiple antibiotics, is one of the greatest threats to health systems in many countries around the world (World Health Organization, 2015).

In this paper we focus on streaming analytics associated with MDR bacteria in the ICU. The main reason is that ICU patients require critical medical care, and the early identification of bacteria resistant to antibiotics is essential for an effective therapy. Antimicrobial susceptibility tests are used to detect whether an antibiotic in the antibiogram is sensitive or resistant to a particular bacterium. Since these results usually require between 24 and 48 h after the bacterial culture is collected, we propose to use our visualization framework to try to better understand antibiotic pressure, making it possible to take adequate isolation and treatment for patients with high risk of having MDR bacteria. Therefore, we suggest using our framework to: (1) track the current situation in the ICU; (2) create warning regions, according to the expert knowledge, to potentially identify patients at risk of having MDR bacteria when their projections are inside the region; and (3) follow the evolution of a patient of interest.

The database used to simulate the data streaming in the current study consisted of clinical data extracted from the EHR associated with 2544 ICU patients at the University Hospital of Fuenlabrada (Spain) in the period between 2014 and 2015. Each data record contains the specific characteristics of a patient at a given time interval, and it is described by a multivariate time series (MTS). The MTS represents the families of antibiotics taken by the patient, whether the patient has mechanical ventilation or not, and the kind of bacterial culture collected. A total of 22 antibiotic families (the most usual) were extracted from the EHR and encoded as binary time series to indicate whether the patient has taken (or not) the following families of antibiotics: aminoglycosides, amphenicols, antifungals, carbapenems, first, second, third and fourth generation of cephalosporins, glycopeptide, glycylcyclines, lincosamides, macrolides, monobactams, nitroimidazoles, oxazolidinones, broad-spectrum penicilins, penicilins, polymyxins, quinolones, sulfonamides, tetracycline and non-grouped antibiotics. Regarding mechanical ventilation, a binary time series indicates if the patient is assisted with automatic ventilation. Finally, the type of bacterial culture is represented by one of the following categories: surface, respiratory, catheter, blood, external wound, faeces, internal medium, urine, sputum, liquid, abdominal or others. The result of the culture identifies whether a tested antibiotic in the antibiogram is resistant to a particular bacterium. It is interesting to remark that the antibiotics the patients intake during their stay in the ICU can be different to those tested in the culture.

According to the culture's results and taking into account that the survival time of bacteria is, on average, 15 days, the patient streaming records are classified into three different classes: class 0 identifies patients' records with negative results when testing MDR; class 1 corresponds to records with negative results to MDR bacteria, but with positive results to the test during the previous 15 days; class 2 are records with positive results to MDR bacteria. Note that, in this scenario, it is possible to find imbalance in the number of records per class.

4.2. Characterization of streaming data

We evaluate here the capabilities of the proposed framework to represent and analyze the data stream scenario (i.e., the health status of the patients at a given moment), as well as the evolution of the patients' health status in the ICU during a specific period of time.

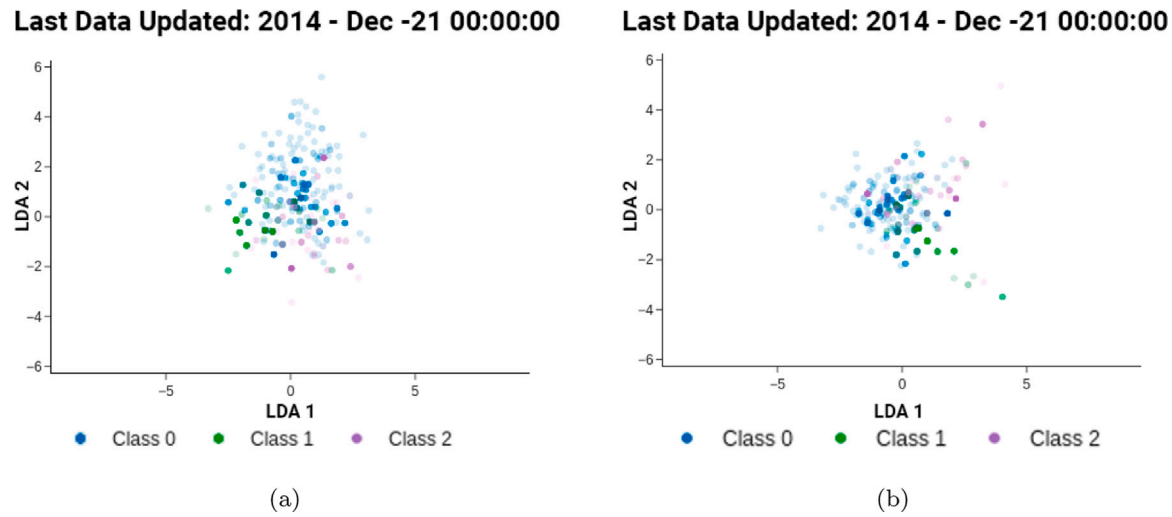

Fig. 4. Plots after applying LDA on the training dataset (from 1 September to 21 December 2014), with 524 records (380 in class 0, 78 in class 1 and 66 in class 2). To deal with imbalanced classes, the training dataset is preprocessed using: (a) SMOTE and (b) Tomek Links. The classes can be activated or deactivated by selecting them in the legend, allowing users to choose a subset of classes to be visualized.

A screenshot of the framework dashboard was presented in Fig. 2, where we show the ICU scenario between 2014-09-01 and 2015-01-08. Each point in the scatter plot corresponds to a patient record, labeled according to the class the record belongs to. We identify class 0 with blue, class 1 with green and class 2 with purple. The class records can be activated (displayed) or deactivated (not displayed) by clicking in the corresponding class directly in the legend. The markers are different for the train and the test sets. If a record is employed to train the model, then it is represented using a circle. If the record is part of the test set or is an out-of-sample record, it is represented through an hexagram or six-pointed start. Furthermore, we use three opacity levels, with values between 1.0 and 0.15 to identify the data records: (1) the high level, *opacity value = 1.0*, implies that the record is linked to the last MDR test of a current ICU patient; (2) medium level, *opacity value = 0.35*, represents previous records for a current ICU patient; and (3) lower level, *opacity value = 0.15*, identifies records mapped in the visualization associated with patients that are no longer in the ICU. The refresh interval for streaming data is established in the user interface (60 s in the screenshot shown in Fig. 2, see the edit box "refresh time" expressed in seconds in the area marked as 1). A step represents the data obtained during the last refresh interval. As the time interval to be represented is very wide (more than three months), we have simulated the data stream by increasing the data presentation rate in several orders of magnitude. This way, data collected during one day are presented every 60 s, allowing us to obtain a larger volume of data more quickly. In any case, the infrastructure allows us to do it in real time.

The mental map is created based on the mapping of the patients' records during the validity period of the trained model. This period can be established according to clinical criteria. Note that, when the clinicians consider the model is no longer valid, it should be advisable to train it again with a more recent data stream. This could lead to a different mapping, which may be somewhat inconsistent with the previous mental map. As an example of different mappings, we show in Fig. 4 the LDA projection in the period from September 2014 to December 2014, but two strategies to handle imbalanced classes are considered. The results obtained when considering SMOTE are represented in Fig. 4(a), whereas those with Tomek Links are in Fig. 4(b). We decided to employ the Tomek Links method as reference to handle imbalanced classes for the rest of LDA projections in this case study.

The clinicians analyzed the data stream scenario in the ICU with our help once the model was trained. Note that if the current scenario changes and the new patients' records are not adequately represented by the model, it is possible to train it again including more recent records to try to properly capture the new situation of the ICU.

4.3. Definition of warning regions

Once the clinicians got an overall perspective of the current scenario, they can draw one or more polygons on the projected space to establish each one as a region of interest. Each region, which can be easily modified at any time, can be considered as a warning area associated with potential MDR events. Warning regions can be set according to the occurrence of cases in the mapping, by defining a polygon encompassing the selected points, and can include details of the data to be displayed, on demand. We have included an alarm system based on the movements of cases in/out the warning regions.

Fig. 5(a) shows an example of the region defined by the points (streaming records) selected after training the model. It is possible to raise a warning over the data records when their projection is included or excluded in the warning region, depending on the purpose of the analysis. In this case, the clinicians decided to raise a warning for records within the warning region, which mostly is associated with class 2 (see Fig. 5(b), with triangle markers for records in the warning area).

Since the model was trained with the purpose of monitoring and gaining insights of new incoming records, the clinicians decided to visualize just the records associated with the step linked to the date of December 21, 2014, as in Fig. 5(c). Note that there are two records inside the polygon: one record classified as class 2 (purple triangle) and another classified as class 0 (blue triangle), although the region was supposed to be associated with class 2.

This may help experts to follow what is happening at any given time in the ICU, making it easier to track the patient records not considered for training the model.

4.4. Patient evolution tracking

Health information systems provide access to patients' records, enabling clinicians to follow them up as time evolves in their ICU stay. Though raw data are available in the bedside monitors for each patient individually, a general dashboard of the events associated with all ICU patients is appropriate to get an overview of the ICU environment, especially when considering the spread of adverse pathogens. The visualization of these records in the same display may help the clinicians to predict in advance bacterial multi-resistance analytic results. The delay in a proper diagnosis of multi-resistance bacteria and in the appropriate treatment presents a risk both to the particular patient and to the rest of patients in the ICU. Just take into consideration that, in this

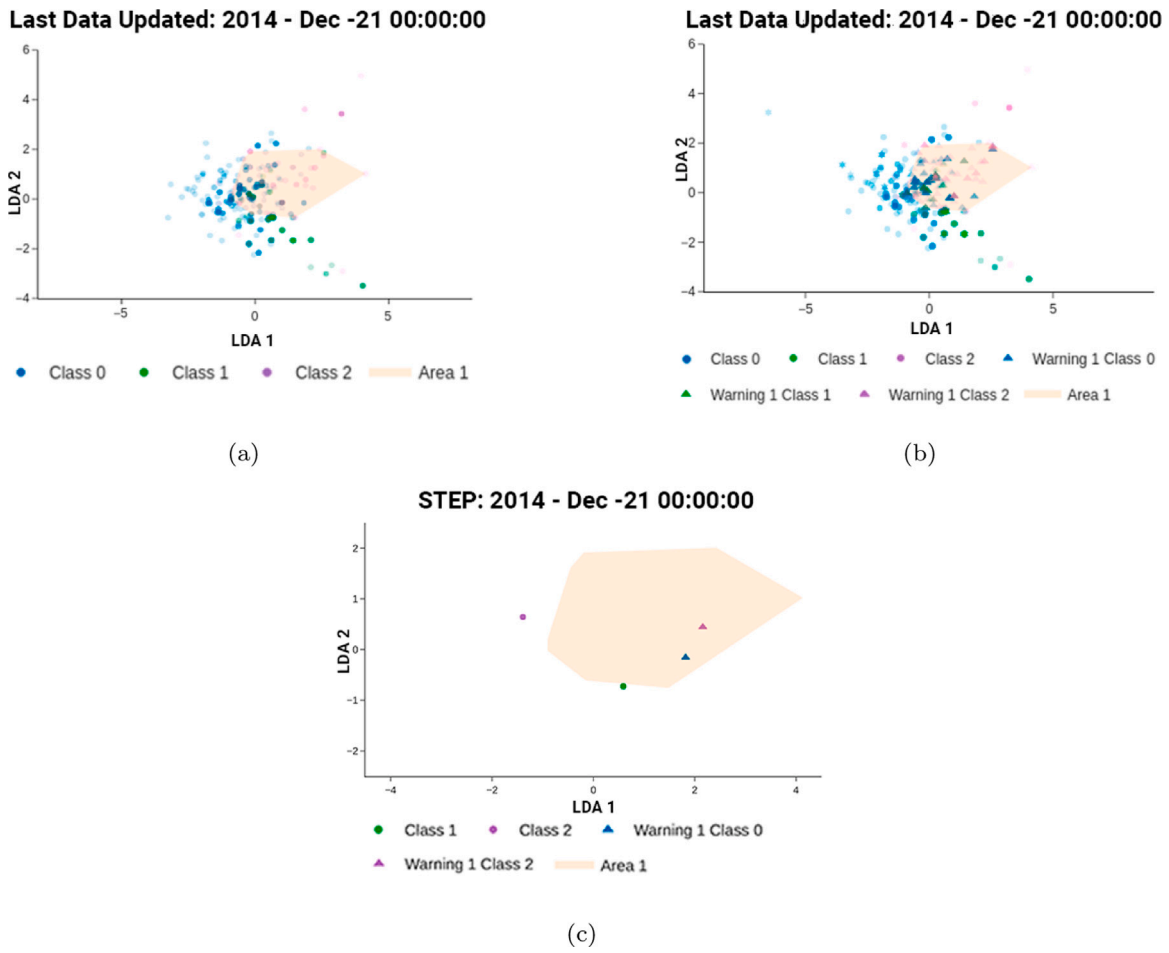

Fig. 5. Warning region defined in accordance with the accumulation of high-risk records (class 2). (a) Training records and warning region; (b) identification of high-risk records, represented as triangle markers with the color associated with the ground truth class; (c) simplification of (b) just representing records associated with data collected in the ICU on December 21.

particular application, the culture results are usually provided between 24 and 48 h after taking the biological sample. In this scenario, the possibility of early MDR predictions would enable the activation of safety measures, such as patient isolation. It is important to remark that, to make visualizations independent of the time the culture's result is provided, a record is labeled with the corresponding ground truth in all figures presented in this paper.

Fig. 6 shows the following-up of a patient of interest (marked with the star diamond symbol) throughout his/her ICU stay from December 22, 2014 to January 9, 2015. Records of patients in the ICU during this period of time are also represented with hexagrams. The model was trained with data stream records until December 21, 2014, using the supervised LDA technique for DR. Panels from (a) to (f) represent the follow-up of different data stream records for the same patient of interest. Note that every panel in this figure is associated with a different date, and just records registered that specific date are represented except for the records of the patient who is being tracked (all records are visualized). In this figure, the first mapped record (marked with a star diamond symbol) of the patient of interest is on December 22, 2014, one day after training the model using LDA as DR (see Fig. 6(a)). Note that the record of interest in panel (a) is inside the warning area. The second record is on December 23, 2014. Though there is no specific screenshot for this date, we visualize that it is inside the warning area of high risk. It is labeled as class 2 two days after. The third record happened on December 25, 2014 (see Fig. 6(b), including the mapping of the record on December 23). Note that the record mapping of the patient who is being tracked is outside of the warning region. Two new bacterial cultures, with positive results, were tested for this patient: on December 27 and 30, 2014. Records associated with these results are represented in purple in Fig. 6(c). The last record of this patient is on January 5, 2015 (see Fig. 6(d)). Since the patient was discharged from the ICU on January 8, the markers related with his/her records are still depicted in Fig. 6(e) and are not displayed on January 9 (see Fig. 6(f)). This visualization in real time allows to identify patients who present high-risk of having MDR bacteria and, consequently, enable taking appropriate actions as for example, to isolate the patient.

To evaluate the influence of the DR method, Fig. 7 shows the following-up of the same patient mapped in Fig. 6 when using UMAP (unsupervised method) instead of LDA. Remark here that, for each panel, we display both records considered to train the model as records of test ICU patients during the specified dates. Owing to the high the number of records, larger-sized dots are depicted to represent mapping areas with markers overlapping. Firstly, in Fig. 7(a), we represent the records after training the model. As previously indicated, the patient of interest is admitted in the ICU on December 22, 2014, and his/her record is represented by a purple diamond star (see Fig. 7(b)). This patient was tested on December 23 and 25. Note that these patient records are mapped on two very different areas, with the most recent record labeled as class 1 (transition class) and located very distant to the record mapped on December 23. To complement this explanation, remark that the result of the bacterial culture on December 25 was sensitive, and therefore it was assigned to class 1 (see Fig. 7(c)). The next culture test of the tracked patient was performed on December 27, being multi-drug resistant to some bacteria. As a consequence, two purple stars associated with records for this patient on December 27 and 30, respectively, are represented in Fig. 7(d). The last record

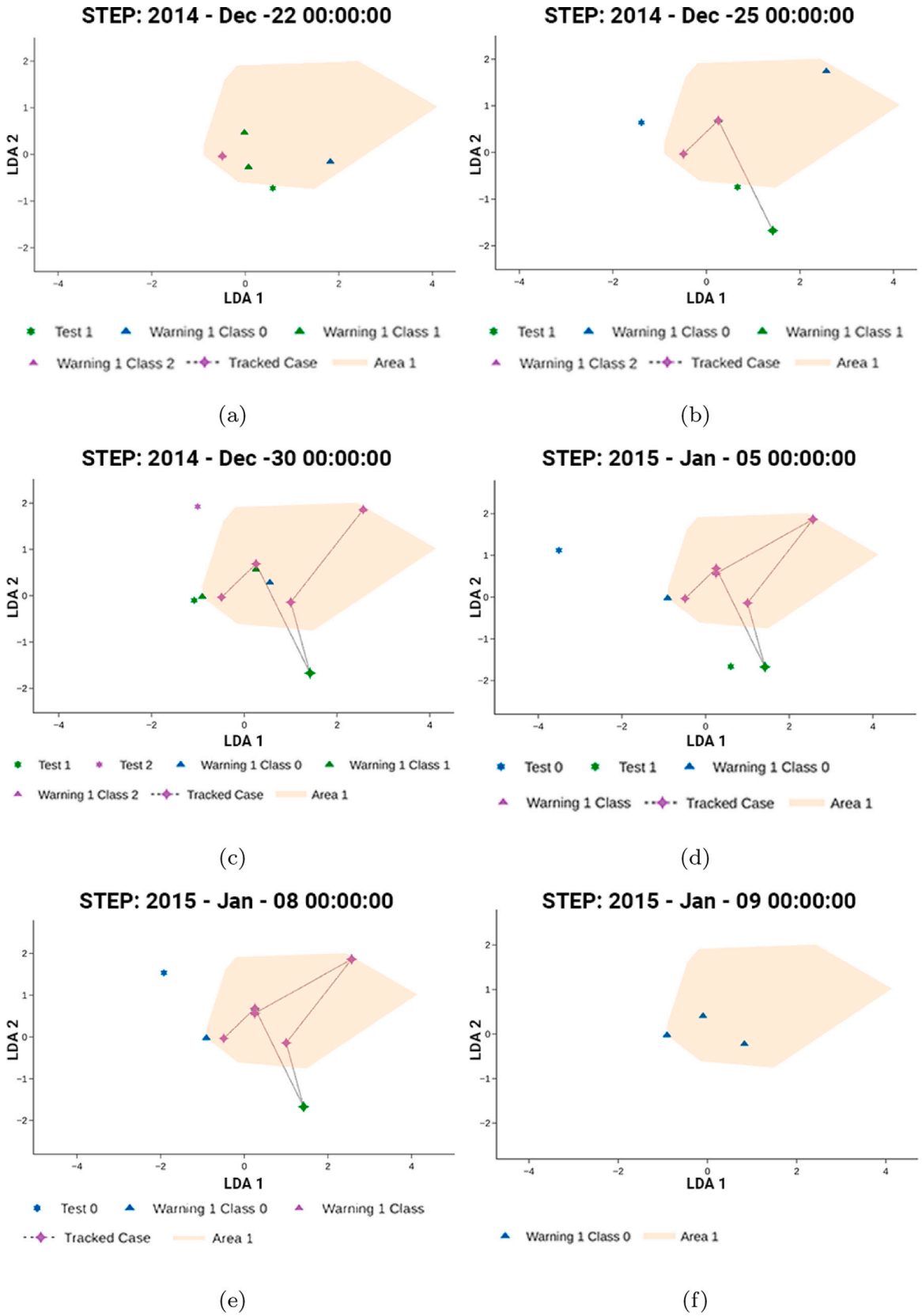

Fig. 6. Warning regions in LDA plots for identifying records with high risk due to MDR bacteria (class 2), and following-up of the records of a specific patient (marked with a star diamond symbol) in the ICU. Records of hospitalized ICU patients during the considered time period (from December 22, 2014 to January 9, 2015 at different steps) are also represented.

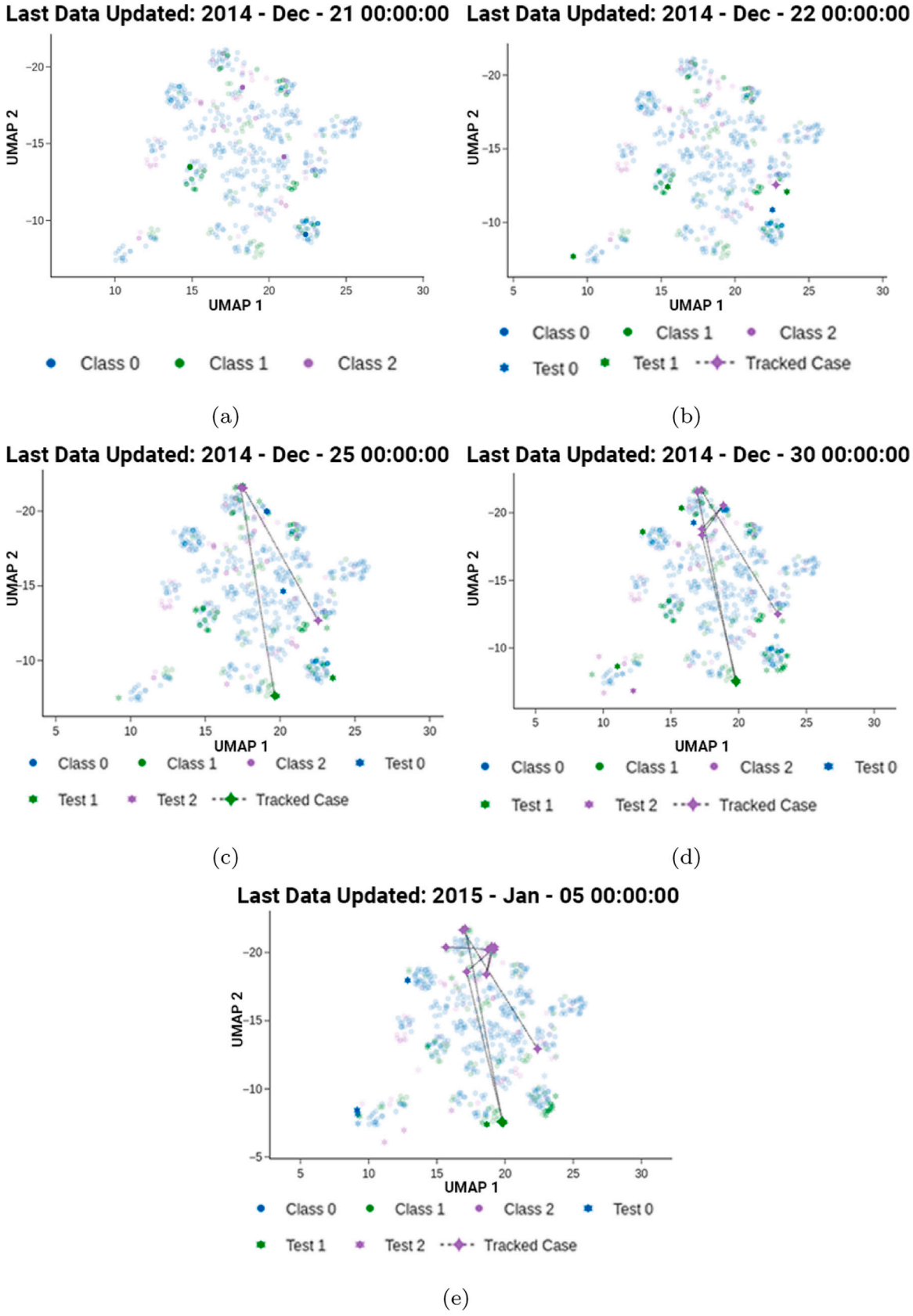

Fig. 7. Following-up of a patient of interest (marked with a star diamond symbol) throughout his/her ICU stay when using UMAP mappings. Records of patients in the ICU during the considered time period (from December 21, 2014 to January 5, 2015) are represented with hexagrams, as they are out of the training set.

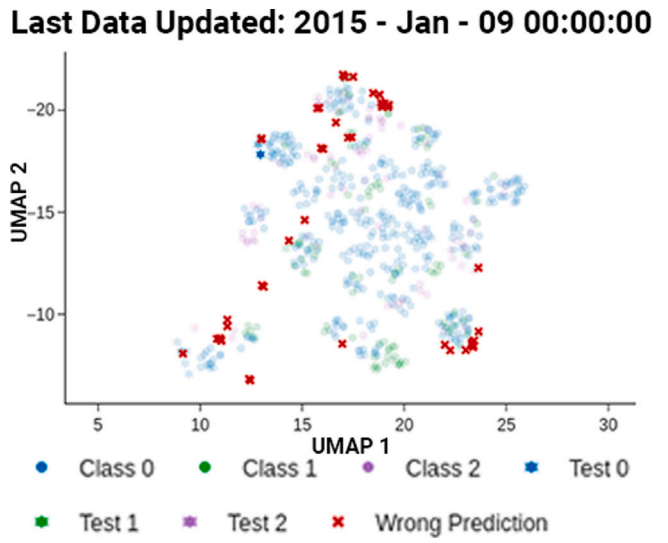

Fig. 8. Visual representation of the application of k-NN on the UMAP visualization shown in Fig. 7 some days later. Circles indicate records employed as training set of classes 0, 1 and 2. Hexagrams represent test records of classes 0, 1 and 2. Both are consistent with the colormap employed along the case study. Red xcross markers indicate misclassifications.

of this patient is on January 5 and is still class 2 (see Fig. 7(e)). Note that, once the patient is considered as class 2, next records are placed in close areas since the clinical condition does not change. The representation of the whole data stream scenario can support the patient's risk assessment by analyzing the mapping of the records closest to the record of interest.

4.5. Classification with a machine learning algorithm

Once the UMAP visualization was generated, we decided to design a classifier trained with a ML algorithm. The tool currently allows users to apply the k-NN classifier considering the mapped records after DR to make it possible to intuitively visualize its performance directly on the 2D mapping. In this case study, the training set is composed of the original records registered between September 1, 2014 and December 21, 2014. To palliate the effects of the class imbalance in the training set, we perform a preprocessing stage of random oversampling (Ramyachitra & Manikandan, 2014).

The classifier performance was evaluated on a test set not considered for training the classifier. In particular, we chose all patients between December 22, 2014 and January 9, 2015, i.e., 15 ICU patients providing 97 records. Following the heuristic proposed in Dasarathy (1991) and widely used in the literature, we chose $k \approx \sqrt{n}$ by default, where n is the number of cases in the original training set. In this case, $k = 22$. Quantitative results about classification (both in absolute and relative numbers) are presented in Table 1, while visualization of misclassifications is displayed with red xcross marker in Fig. 8. Note that the best accuracy is obtained for class 2 (63.63%), which is the class of greatest interest for the clinicians. The worst performance corresponds to the classification of records labeled in class 1 (47.61%). This result seems reasonable since class 1 can be considered as a transition class between negative and positive MDR results.

The early identification and the tracking of patients with high risk to be multi-drug resistant in real time may provide useful knowledge for the patient, for the ICU, and for the healthcare system in general.

5. Conclusions and future work

This paper proposes the analysis of multidimensional data streams through visualization as a support to decision-making. Data streams present some challenges, such as the changing underlying dynamics in the data over time, and no control about class imbalance. Our approach is a framework oriented to fields with continuous data generation. It is based on a publish-subscriber architecture to incorporate the data and an interactive dashboard where DR and ML algorithms are applied.

The data analysis pipeline is subject to user decisions. It is also up to users to decide whether to train a new model to preserve their mental map. We have shown that the methodology developed can be useful for identifying and tracking a possible worsening. It can also be useful for extracting a global and historical perspective on different settings.

In the particular case study presented in this paper, we analyzed MDR bacteria in ICU patients. With our prototype, it is possible to identify in the visualization the warning regions for records associated with patients with multi-resistant bacteria. These regions can be used to generate alerts for new records projected through out-of-sample approaches. Even with the presence of false positives in this area, clinicians tracked patients during their ICU stay and exploit the temporal evolution of the records.

As a future work, our approach can be adapted to solve challenging problems. If the problem requires large-scale datasets, it may be interesting to migrate the management of ML algorithms and visualization to other alternatives that rely on GPU-based paradigms. Similarly, we plan to include other classification machine learning methods beyond k-NN, such as decision trees, support vector machines or neural networks (Bishop, 2006) that can complement our visual analytics approach. Also of interest is the consideration of A-tSNE (Pezzotti et al., 2017), an implementation of t-SNE for progressive visual analysis. The use of A-tSNE may be interesting as an alternative to UMAP. In this line of work, it also would be interesting to migrate the dimensionality reduction algorithms included in the framework to incremental algorithms in order to deal with the heterogeneous nature of streaming data.

Table 1
Confusion matrix of the test set when applying k-NN on the UMAP visualization. Rows indicate the ground truth label of the record, and columns the label predicted with k-NN.

	Predicted class			Total
	Class 0	Class 1	Class 2	
Class 0	17 **(51.51%)**	6 (18.18%)	10 (30.30%)	33 (100%)
Class 1	12 (28.57%)	20 **(47.61%)**	10 (23.80%)	42 (100%)
Class 2	3 (13.63%)	5 (22.72%)	14 **(63.63%)**	22 (100%)

CRediT authorship contribution statement

Miguel A. Mohedano-Munoz: Software, Writing – original draft, Validation, Investigation. **Cristina Soguero-Ruiz:** Conceptualization, Writing – original draft, Methodology. **Inmaculada Mora-Jiménez:** Methodology, Writing – review & editing. **Manuel Rubio-Sánchez:** Formal analysis, Writing – review & editing, Visualization. **Joaquín Álvarez-Rodríguez:** Resources, Data curation, Validation. **Alberto Sanchez:** Conceptualization, Writing – original draft, Methodology, Visualization, Supervision, Funding acquisition.

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.

Data availability

Published source code online. Access to the data can be provided upon official request if approved by the Committee of Ethics of the University Hospital of Fuenlabrada.

Funding

This research was funded by the Spanish Research Agency, grant numbers PID2021-122392OB-I00, PID2019-106623RB-C41/AEI/10.13039/501100011033 and PID2019-107768RA-I00; and by Universidad Rey Juan Carlos (URJC) and Community of Madrid, Spain, grant number 2020-66.

References

Alsakran, J., Chen, Y., Zhao, Y., Yang, J., & Luo, D. (2011). STREAMIT: Dynamic visualization and interactive exploration of text streams. In *2011 IEEE pacific visualization symposium* (pp. 131–138).

Angelini, M., Santucci, G., Schumann, H., & Schulz, H.-J. (2018). A review and characterization of progressive visual analytics. *Informatics*, *5*(3), 31.

Bach, B., Shi, C., Heulot, N., Madhyastha, T., Grabowski, T., & Dragicevic, P. (2016). Time curves: Folding time to visualize patterns of temporal evolution in data. *IEEE Transactions on Visualization and Computer Graphics*, *22*(1), 559–568.

Batista, G. E., Bazzan, A. L., Monard, M. C., et al. (2003). Balancing training data for automated annotation of keywords: a case study. In *WOB* (pp. 10–18).

Bellman, R. (1957). *Dynamic programming*. Courier Corporation.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Blount, M., Ebling, M. R., Eklund, J. M., James, A. G., McGregor, C., Percival, N., et al. (2010). Real-time analysis for intensive care: development and deployment of the artemis analytic system. *IEEE Engineering in Medicine and Biology Magazine*, *29*(2), 110–118.

Bosch, H., Thom, D., Heimerl, F., Püttmann, E., Koch, S., Krüger, R., et al. (2013). ScatterBlogs2: Real-time monitoring of microblog messages through user-guided filtering. *IEEE Transactions on Visualization and Computer Graphics*, *19*(12), 2022–2031.

Brich, N., Schulz, C., Peter, J., Klingert, W., Schenk, M., Weiskopf, D., et al. (2020). Visual analysis of multivariate intensive care surveillance data. In *VCBM* (pp. 71–83).

Cao, N., Lin, Y., Sun, X., Lazer, D., Liu, S., & Qu, H. (2012). Whisper: Tracing the spatiotemporal process of information diffusion in real time. *IEEE Transactions on Visualization and Computer Graphics*, *18*(12), 2649–2658.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*, 21–27.

Cox, T. F., & Cox, M. (2000). *Multidimensional scaling, second edition* (2 ed.). Chapman and Hall/CRC.

Dasarathy, B. V. (1991). *Nearest neighbor (NN) norms: NN pattern classification techniques*. Los Alamitos, CA: IEEE Computer Society Press.

Dasgupta, A., Arendt, D. L., Franklin, L. R., Wong, P. C., & Cook, K. A. (2018). Human factors in streaming data analysis: Challenges and opportunities for information visualization. *Computer Graphics Forum*, *37*(1), 254–272.

Domeniconi, C., Gunopulos, D., & Peng, J. (2005). Large margin nearest neighbor classifiers. *IEEE Transactions on Neural Networks*, *16*(4), 899–909.

Dunn, W., Jr., Burgun, A., Krebs, M.-O., & Rance, B. (2016). Exploring and visualizing multidimensional data in translational research platforms. *Briefings in Bioinformatics*, *18*(6), 1044–1056.

Dunning, T., & Friedman, E. (2016). *Streaming architecture: new designs using apache kafka and mapr streams*. O'Reilly Media, Inc..

Eades, P., Lai, W., Misue, K., & Sugiyama, K. (1991). *Preserving the mental map of a diagram*: *Technical Report IIAS-RR-91-16E*, Fujitsu Laboratories.

Endert, A., Pike, W. A., & Cook, K. (2014). From streaming data to streaming insights: The impact of data velocities on mental models. In *Proceedings of the 2014 workshop on human centered big data research* (pp. 24–26). New York, NY, USA: Association for Computing Machinery.

Fekete, J., & Primet, R. (2016). Progressive analytics: A computation paradigm for exploratory data analysis. CoRR, abs/1607.05162.

Fischer, F., & Keim, D. A. (2014). NStreamAware: Real-time visual analytics for data streams to enhance situational awareness. In *Proceedings of the eleventh workshop on visualization for cyber security* (pp. 65–72). ACM.

Flask (2020). Flask: User guide. https://flask.palletsprojects.com/en/1.1.x/. Accessed: 2020-10-21.

Forbes, A., Höllerer, T., & Legrady, G. (2010). Behaviorism: a framework for dynamic data visualization. *IEEE Transactions on Visualization and Computer Graphics*, *16*(6), 1164–1171.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning, Vol. 1*. Springer Series in Statistics.

Fujiwara, T., Chou, J.-K., Shilpika, S., Xu, P., Ren, L., & Ma, K.-L. (2019). An incremental dimensionality reduction method for visualizing streaming multidimensional data. *IEEE Transactions on Visualization and Computer Graphics*, *26*(1), 418–428.

Gomes, G. A. M., dos Santos, E. M., Vidal, C. A., da Silva, T. L. C., & de Macêdo, J. A. F. (2018). Real-time discovery of hot routes on trajectory data streams using interactive visualization based on GPU. *Computers & Graphics*, *76*, 129–141.

Gotz, D., & Stavropoulos, H. (2014). DecisionFlow: Visual analytics for high-dimensional temporal event sequence data. *IEEE Transactions on Visualization and Computer Graphics*, *20*(12), 1783–1792.

He, X. (2005). *Locality preserving projections* (Ph.D. thesis), Chicago, IL, USA: Faculty of the Division of the Physical Sciences, University of Chicago.

Heer, J., & Shneiderman, B. (2012). Interactive dynamics for visual analysis. *Communications of the ACM*, *55*(4), 45–54.

Ikonomovska, E., Loskovska, S., & Gjorgjevik, D. (2007). A survey of stream data mining. In *Proceedings of 8th national conference* (pp. 19–21).

Inseok, K., & Hyejung, C. (2017). Interactive visualization of healthcare data using tableau. *Healthcare Informatics Research*, *23*(4), 349–354.

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, *374*(2065).

Kitchin, R. (2014). *The data revolution: big data, open data, data infrastructures and their consequences*. Sage Publications Ltd.

Kotanidou, A., De Georgia, M. A., Kaffashi, F., Jacono, F. J., & Loparo, K. A. (2015). Information technology in critical care: Review of monitoring and data acquisition systems for patient care and research. *The Scientific World Journal*, *2015*.

Kreps, J., Narkhede, N., Rao, J., et al. (2011). Kafka: A distributed messaging system for log processing. In *Proceedings of the NetDB, Vol. 11* (pp. 1–7).

Krstajić, M., & Keim, D. A. (2013). Visualization of streaming data: Observing change and context in information visualization techniques. In *2013 IEEE international conference on big data* (pp. 41–47).

Lauritsen, S. M., Kristensen, M., Olsen, M. V., Larsen, M. S., Lauritsen, K. M., Jø rgensen, M. J., et al. (2020). Explainable artificial intelligence model to predict acute critical illness from electronic health records. *Nature communications*, *11*(1), 1–11.

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, *18*(17), 1–5.

Levy-Fix, G., Kuperman, G. J., & Elhadad, N. (2019). Machine learning and visualization in clinical decision support: Current state and future directions. arXiv preprint arXiv:1906.02664.

Li, C., & Baciu, G. (2014). VALID: A web framework for visual analytics of large streaming data. In *2014 IEEE 13th international conference on trust, security and privacy in computing and communications* (pp. 686–692).

Li, X., Han, J., Lee, J.-G., & Gonzalez, H. (2007). Traffic density-based discovery of hot routes in road networks. In *Proceedings of the 10th international conference on advances in spatial and temporal databases* (pp. 441–459). Berlin, Heidelberg: Springer-Verlag.

Maaten, L. v. d., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(Nov), 2579–2605.

Mansmann, F., Fischer, F., & Keim, D. A. (2012). Dynamic visual analytics—Facing the real-time challenge. In *Expanding the frontiers of visual analytics and visualization* (pp. 69–80). London: Springer London.

Martínez-Agüero, S., Soguero-Ruiz, C., Alonso-Moral, J. M., Mora-Jiménez, I., Álvarez-Rodríguez, J., & Marques, A. G. (2022). Interpretable clinical time-series modeling with intelligent feature selection for early prediction of antimicrobial multidrug resistance. *Future Generation Computer Systems*, *133*, 68–83.

McInnes, L., Healy, J., Saul, N., & Großberger, L. (2018). UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software*, *3*(29), 861.

McLachlan, G. J. (2004). *Wiley series in probability and mathematical statistics. probability and mathematical statistics*, *Discriminant analysis and statistical pattern recognition*. Wiley-Interscience.

Moere, A. V. (2004). Time-varying data visualization using information flocking boids. In *IEEE symposium on information visualization* (pp. 97–104).

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

Nemati, S., Holder, A., Razmi, F., Stanley, M. D., Clifford, G. D., & Buchman, T. G. (2018). An interpretable machine learning model for accurate prediction of sepsis in the ICU. *Critical Care Medicine*, *46*(4), 547.

Neves, T. T., Martins, R. M., Coimbra, D. B., Kucher, K., Kerren, A., & Paulovich, F. V. (2020). Xtreaming: an incremental multidimensional projection technique and its application to streaming data. arXiv preprint arXiv:2003.09017.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pezzotti, N., Lelieveldt, B. P. F., Maaten, L. v. d., Höllt, T., Eisemann, E., & Vilanova, A. (2017). Approximated and user steerable tSNE for progressive visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, *23*(7), 1739–1752.

Plotly (2021). Dash: User guide. https://dash.plotly.com/. Accessed: 2020-10-21.

QlikTech International, A. B. (2022). Qlik replicate setup and user guide. URL https://help.qlik.com/en-US/replicate/Content/Replicate/6.5/PDF/Setup_User_Guide.pdf.

Ramyachitra, D., & Manikandan, P. (2014). Imbalanced dataset classification and solutions: a review. *International Journal of Computing and Business Research (IJCBR)*, *5*(4), 1–29.

Rejab, F. B., Nouira, K., & Amri, B. (2014). Physiological data stream from monitoring system in intensive care unit.. In *International work-conference on bioinformatics and biomedical engineering* (pp. 1717–1728).

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*(5500), 2323–2326.

Rubio-Sánchez, M., Raya, L., Diaz, F., & Sanchez, A. (2015). A comparative study between RadViz and star coordinates. *IEEE Transactions on Visualization and Computer Graphics*, *22*(1), 619–628.

Sanchez, A., Raya, L., Mohedano-Munoz, M. A., & Rubio-Sánchez, M. (2020). Feature selection based on star coordinates plots associated with eigenvalue problems. *The Visual Computer*, 14.

Sanchez, A., Soguero-Ruiz, C., Mora-Jimenez, I., Rivas-Flores, F. J., Lehmann, D. J., & Rubio-Sánchez, M. (2018). Scaled radial axes for interactive visual feature selection: A case study for analyzing chronic conditions. *Expert Systems with Applications*, *100*, 182–196.

Satyanarayan, A., Russell, R., Hoffswell, J., & Heer, J. (2015). Reactive vega: A streaming dataflow architecture for declarative interactive visualization. *IEEE Transactions on Visualization and Computer Graphics*, *22*.

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. of the IEEE symposium on visual languages* (pp. 336–343). IEEE.

Shneiderman, B., Plaisant, C., & Hesse, B. W. (2013). Improving healthcare with interactive visualization. *Computer*, *46*(5), 58–66.

Sow, D., Biem, A., Sun, J., Hu, J., & Ebadollahi, S. (2010). Real-time prognosis of ICU physiological data streams. In *2010 annual international conference of the IEEE engineering in medicine and biology* (pp. 6785–6788). IEEE.

Stadler, J., Donlon, K., Siewert, J., Franken, T., & Lewis, N. (2016). Improving the efficiency and ease of healthcare analysis through use of data visualization dashboards. *Big Data*, *4*, 129–135.

Steiger, M., Bernard, J., Mittelstädt, S., Lücke-Tieke, H., Keim, D., May, T., et al. (2014). Visual analysis of time-series similarities for anomaly detection in sensor networks. *Computer Graphics Forum*, *33*(3), 401–410.

Sun, Y., Guo, F., Kaffashi, F., Jacono, F. J., DeGeorgia, M., & Loparo, K. A. (2020). INSMA: An integrated system for multimodal data acquisition and analysis in the intensive care unit. *Journal of Biomedical Informatics*, *106*.

Tomek, I. (1976). Two modifications of CNN. *IEEE Transactions on Systems, Man, and Cybernetics*, *6*, 769–772.

Wong, P. C., Foote, H., Adams, D., Cowley, W., & Thomas, J. (2003). Dynamic visualization of transient data streams. In *IEEE symposium on information visualization 2003 (IEEE Cat. No.03TH8714)* (pp. 97–104).

World Health Organization (2015). *Global action plan on antimicrobial resistance*: *Technical Report*, World Health Organization.

Xie, J., & Qiu, Z. (2007). The effect of imbalanced data sets on LDA: A theoretical and empirical analysis. *Pattern Recognition*, *40*(2), 557–562.