

Received July 6, 2020, accepted July 22, 2020, date of publication July 29, 2020, date of current version August 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3012836

A Machine Learning Approach to Reduce Dimensional Space in Large Datasets

RAFAEL MUÑOZ TEROL^{ID}¹, ALEJANDRO REINA REINA², SABER ZIAEI^{ID}³, AND DAVID GIL^{ID}⁴

¹Department of Software and Computing Systems, University of Alicante, 03690 Alicante, Spain

²University Institute for Computing Research, University of Alicante, 03690 Alicante, Spain

³Babol Noshirvani University of Technology, Babol 47148-71167, Iran

⁴Department of Computer Technology and Computation, University of Alicante, 03690 Alicante, Spain

Corresponding author: Rafael Muñoz Terol (rafam@dsi.ua.es)

This work was partially funded by Grant RTI2018-094283-B-C32, ECLIPSE-UA (Spanish Ministry of Education and Science), and in part by the Lucentia AGI Grant. This work was partially funded by GENDER-NET Plus Joint Call on Gender an UN Sustainable Development Goals (European Commission - Grant Agreement 741874), funded in Spain by "La Caixa" Foundation (ID 100010434) with code LCF/PR/DE18/52010001 to MTH.

ABSTRACT Large datasets computing is a research problem as well as a huge challenge due to massive amounts of data that are mined and crunched in order to successfully analyze these massive datasets because they constitute a valuable source of information over different and cross-folded domains, and therefore it represents an irreplaceable opportunity. Hence, the increasing number of environments that use data-intensive computations need more complex calculations than the ones applied to grid-based infrastructures. In this way, this paper analyzes the most commonly used algorithms regarding to this complex problem of handling large datasets whose part of research efforts are focused on reducing dimensional space. Consequently, we present a novel machine learning method that reduces dimensional space in large datasets. This approach is carried out by developing different phases: merging all datasets as a huge one, performing the Extract, Transform and Load (ETL) process, applying the Principal Component Analysis (PCA) algorithm to machine learning techniques, and finally displaying the data results by means of dashboards. The major contribution in this paper is the development of a novel architecture divided into five phases that presents an hybrid method of machine learning for reducing dimensional space in large datasets. In order to verify the correctness of our proposal, we have presented a case study with a complex dataset, specifically an epileptic seizure recognition database. The experiments carried out are very promising since they present very encouraging results to be applied to a great number of different domains.

INDEX TERMS Machine learning, data mining, large dataset, dimensionality reduction, ETL, PCA, cross-validation, dashboards.

I. INTRODUCTION

It is a fact that in matters of great importance that have financial, medical, social, or other implications, we often seek a second opinion before making a decision, sometimes a third, and sometimes many more. In doing so, we weigh the individual opinions and combine them through some thought process to reach a final decision that is presumably the most informed one [1]. Hence, research in classification environments stills being a very alive field of machine learning and pattern recognition. Thus, many classification techniques apply a unique classifier while any other ones develop a multi-classification-based approach. In this way, techniques based on classifiers combination have received a special focus in recent years

The associate editor coordinating the review of this manuscript and approving it for publication was Hao Luo ^{ID}.

and nowadays are recognized as an established pattern recognition seed. So, it is an important matter in this field how to combine the individual decisions from each classifier in some way (typically by weighted or unweighted voting) to classify new examples. Thus, by performing a wide analysis of many of the research works that have been developed in the classification and multi-classification areas during last years we can conclude that supervised learning techniques are the most frequently used by these classification systems. Then, one of the most active areas of research in supervised learning has been to study methods for constructing good ensembles of classifiers. The main discovery is that ensembles are often much more accurate than the individual classifiers that make them up [2]. Hence, ensemble-based systems have shown to produce favorable results compared to those of single-expert systems for a broad range of applications and under a

variety of scenarios. An important contribution to this matter was done by Cruz *et al.* [3]. They have shown that Multiple Classifier Systems are widely used to solve many real-world problems, such as face recognition [4], music genre classification [5], credit scoring [6], [7], class imbalance [8], recommender system [9], [10], software bug prediction [12], intrusion detection [13], [14], and for dealing with changing environments [15]–[17].

One of the main added troubles in the multi-classification problem consists of managing large datasets with a huge number of classes and features. So, it is necessary for the application of dimensionality reduction techniques and algorithms for efficiently computing them without decreasing accuracy. In this way, the following paragraphs exhibit the most commonly used algorithms for this complex task of handling large datasets related to the Big Data challenges.

A. CART ALGORITHM

The CART algorithm is based on Classification and Regression Trees by Breiman et al (1984) [18]. A CART tree is a binary decision tree that is created by dividing a node toward two child nodes frequently, starting with the root node that includes the entire learning sample. CART is a predictive model too. It helps to find a variable based on other labeled variables.

B. SUPPORT VECTOR CLUSTERING ALGORITHM

Support vector clustering (SVC) is a newly and nonparametric clustering algorithm which is based on the support vector approaches presented by Ben-Hur *et al.* [19]. SVC does not make any theory on the amount or form of the clusters in the data. Also, by using a kernel function, SVC can map data points from data space to a high dimensional feature space. With using the Support Vector Domain Description algorithm (SVDD) in the kernel's feature space, the algorithm seeks the smallest sphere that surrounds the image of the data. This sphere makes a set of contours that embed the data points when mapped back to data space. Then contours interpreted as cluster boundaries, and points enclosed by each contour are linked by SVC to the corresponding cluster. If the data is high-dimensional, a preprocessing step like using principal component analysis (PCA) can be effective, since SVC is a proper algorithm for low-dimensional data.

C. NAIVE BAYES CLASSIFIERS

Naive Bayes is a classification method based on Bayes' Theorem with a hypothesis of independence between predictors [20]. It is not a particular algorithm although a group of algorithms where all of them share a joint origin, i.e. each set of features being classified is independent. Notwithstanding their naive idea and possibly simple theories, naive Bayes classifiers have worked very efficiently in numerous complicated real-world conditions. Naive Bayes needs a small number of training data to determine the parameters required for classification. It is known as one of the prominent benefits of the NB algorithm.

D. RANDOM FOREST ALGORITHM

Random Forest creates many numbers of decision trees and merges all the decision trees to obtain the highest accuracy and further constant prediction [21]. Random Forest method performs by regarding the random feature subset selection for the splitting of a node [22]. Alternatively of finding for the optimum probable saturation points as in common decision trees, for every feature, it uses optional thresholds to gain the decision trees utmost random. This procedure does well, as the summation of more decision trees minimizes the noise impact which leads to giving further accurate results, whereas a particular decision tree can prone to noise effect. The drawback is that generated subsets of various decision trees may tend to overlap and also it is difficult to understand. The main disadvantage is taking the maximum number of trees produces ineptness in the technique and making to work slow and not fit for real-time predictions.

E. LOGISTIC REGRESSION

When the dependent variable is binary, logistic regression (LR) is the best regression analysis to handle this situation [23]. Also, logistic regression is a predictive analysis like any regression analysis. Logistic regression can be employed to represent data and to demonstrate the connection among one dependent binary variable and one or more nominal, ordinal, interval, or ratio-level independent variables. logistic regression outputs are discrete despite linear regression. In linear regression, outputs are continuous number values, but logistic regression with using the logistic sigmoid function changes its output. It yields a probability value that can then be mapped to two or more discrete classes.

F. MULTILAYER PERCEPTRON

If in the logistic regressor you insert a middle layer instead of feeding the input to the logistic regression, you create a multilayer perception (MLP). The middle layer named the hidden layer, that has a nonlinear activation function which is often sigmoid. These features discriminate MLP from a linear perception. Generally, there are three layers in MLP, an input layer, a hidden layer, and an output layer. Every node could be a neuron that uses a nonlinear activation function excluding for the input nodes [24]. For training, MLP uses backpropagation which is a supervised learning method. Moreover, MLP classifies datasets that are not linearly separable. It does this by applying a more strong and complicated architecture to learn regression and classification models for complex datasets.

G. K-NEAREST NEIGHBOR ALGORITHM

K-nearest neighbor is a non-parametric method [25]. It does not create any underlying theories concerning the distribution of information. Any possible cases store by KNN and new cases classify based on a similarity criterion like distance functions. For classifying a case, KNN takes a majority election from its neighbors, with the case being allocated to the

category utmost common between its K-nearest neighbors measured by a distance function. The neighbors are selected from a collection of objects for which the class or the object attribute value is identified. although a particular training phase is not needed, this may be considered because of the training set for the algorithm. KNN is sensitive to the local formation of the data. It considers as one of the most characteristics of the k-NN algorithm.

We must see the problem since the point of view that Big Data is the huge datasets that have polyhedral variables. Then, one of the significant characteristics of big data is Volume. Hence, concerning technological advancements, in both the number of records and attributes, big data has a huge explosion. So, it makes a plethora of challenges in data science. The first one is produced because the pretty huge volume of data in big data creates multidimensional datasets. Thus, the analysis of these high dimensional datasets is difficult for researchers. Also, depending on what sort of process is involved, high dimensional data can be gathered from diverse sources. Nowadays, researchers for convenience in big data analysis use dimension reduction. Basically, dimensionality reduction consists of decreasing the number of arbitrary variables or attributes under consideration. As an extended task from this one, high-dimensionality data reduction is considered as a pre-processing step that is very significant in various real-world applications. Currently, it is known as an important task for many purposes like machine learning, data mining, etc. For instance, suppose that you have a dataset with a number of features (columns in your database). Thus, when they are combined or merged for reducing some of the features of data attributes, it is a real challenge that it will not lose much of the vital aspects of the primary dataset. This process is known as dimensionality reduction. There are a variety of techniques applied in dimensionality reduction such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Generalized Discriminant Analysis (GDA). Moreover, based on the method used, dimensionality reduction can be linear or non-linear. In this way, the best linear approach that we used in our method is PCA.

Also, dimensionality reduction like other methods has several benefits and drawbacks. Below we summarized some of its advantages and disadvantage.

Advantages of Dimensionality Reduction:

- It can effective in data compression, so reduced storage space.
- It can decrease computation time.
- It can eliminate irrelevant features if any.
- Possibility of visualization in the representation of the space of reduced dimensions.

Disadvantages of Dimensionality Reduction:

- It can make data losses in datasets.
- It may be led to “Curse of Dimensionality”. This forces us to diminish the dimensions of our data if we want to use them for analysis.

In this paper, we have worked in the fields of dimensionality reduction in big datasets. For this purpose, our contribution

in this paper is a novel architecture that we have divided into five phases presenting a hybrid method of machine learning to reduce dimensional space in large datasets. In the first phase, we merged all datasets as a huge one. Then in the two next phases, we have applied the extract, transform, and load procedures which known as the ETL process. In phase four, As mentioned before, first we have employed machine learning classification algorithms, then we have used PCA for dimensionality reduction. Finally, in the last phase, from the previous phase, dashboards of predictive models will be made for helping to make forecasts, predictions, visualizations, extracting rules and patterns, etc.

Furthermore, with the aim of proving and evaluating our method on real-world datasets, we have developed an experimental analysis. Our experiment results show that both random forests and MLP are the fittest algorithms to optimize since they perform extremely stationary in general terms.

This paper is organized as follows: In Section II, we propose a review of some of the works related to the application of machine learning and data mining, essentially applied to the health area. Section III describes the methodology used in this paper for dealing with large datasets in order to apply hybrid techniques for reducing dimensional space. In Section IV, the study results are presented in detail. Finally, Section V details the conclusions of our research work, with the aim of generating discussion points for future work.

II. BACKGROUND

From the beginning of research in the computer science area, automatic classification models, methods, and algorithms have been applied to several domains such as biomedical engineering domains, internet of things, electronics, etc. Moreover, in the last years, different research efforts have been carried out with the aim to improve the performance of all this automatic classification portfolio over different domains.

The research work developed by Caruana *et al.* [26] applied high-performance generalized additive models with pairwise interactions (GA2Ms) are applied to real healthcare problems yielding intelligible models with state-of-the-art accuracy. Groce *et al.* [27] demonstrated a real problem in the most classification algorithms such as that some methods are able to find the arguably most difficult-to-detect faults of classifiers: cases where machine learning algorithms have high confidence in an incorrect result. The research study of Letham *et al.* [28] explored the performance of Bayesian association rules and Bayesian decision lists for estimating the risk of stroke in patients that have atrial fibrillation. Zhang *et al.* [29] developed a straightforward warning system that analyses the input instance and predicts if a vision system is likely to produce an unreliable response. Wozniak *et al.* [30] performed a wide range study of classifiers works that combine diverse kinds of classifiers, so-called heterogeneous MCS. Homogeneous MCS, such as Random Forest (RF), is composed of classifiers of

the same kind taking as basic classifiers the ones such as Multi-Layer Perceptron (MLP), k-Nearest Neighbor (kNN), Radial Basis Function (RBF), Support Vector Machines (SVM), Probabilistic Neural Networks (PNN), and Maximum Likelihood (ML) classifiers in recent applications like remote sensing data, computer security, financial risk assessment, fraud detection, recommender systems, and medical computer-aided diagnosis.

In the multi-label classification framework, Jun *et al.* [31] made a proposal based on the classifier chains method where the label order has a strong effect on the classification performance. Lee *et al.* [32] proposed an effective memetic feature selection method based on a novel feature filter that is highly specialized in multilabel text categorization. Zhang *et al.* [33] presented a new multi-label feature selection method that categorizes labels into two groups: independent labels and dependent labels and analyzes the differences between independent labels and dependent labels by proposing a new feature relevance term, that is, the conditional mutual information between candidate features and each label has given other labels. Mencá and Janssen [34] introduced two approaches for learning such label-dependent rules. Their first solution is a bootstrapped stacking approach which can be built on top of a conventional rule learning algorithm while the second approach goes one step further by adapting the commonly used separate-and-conquer algorithm for learning multi-label rules. Nam *et al.* [35] replaced classifier chains with recurrent neural networks, a sequence-to-sequence prediction algorithm which has recently been successfully applied to sequential prediction tasks in many domains. Read *et al.* [36] detected and elaborated on connections between multi-label methods and Markovian models and study the suitability of multi-label methods for prediction in sequential data. Teyssiere [37] proposed an algorithm which is a combination of classifier chains and elastic-net regularization taking an important advantage the fact of the selection of the relevant features in an integral element of the learning process. Zhang and Zhou [38] concluded that the full understanding of label correlations, especially for scenarios with large output space, would remain as the holy grail for multi-label learning. Interestingly while not surprisingly, the best-performing algorithm for both classification and ranking metrics turns out to be the one based on ensemble learning techniques (i.e. random forest of predictive decision trees).

Subudhi *et al.* [39] presented an automated method based on a computer-aided decision system to detect the ischemic stroke using a diffusion-weighted image (DWI) sequence of MR images. The system consists of segmentation and classification of brain stroke into three types. So, different morphological and statistical features were extracted from the segmented lesions to form a feature set which was then classified with support vector machine (SVM) and random forest (RF) classifiers. Liu *et al.* [40] proposed a method for classifier fusion with contextual reliability evaluation based on inner reliability and relative reliability concepts where the inner reliability is represented by a matrix and characterizes

the probability of the object belonging to one class when it is classified into another class. Dadi *et al.* [41] presented a face recognition algorithm where Histogram of Oriented Gradient features are extracted both for the test image and also for the training images and given to the Support Vector Machine classifier. The algorithm is compared with the Eigen feature-based face recognition algorithm and together PCA are verified using 8 different datasets. The three performance curves show that the algorithm outperforms when compared with the PCA algorithm. With the aim of predicting which anatomical therapeutic chemical (ATC) class/classes a compound belongs to, Cheng *et al.* [42] developed a multi-label classifier by incorporating the information of chemical-chemical interaction, the information of the structural similarity, and the information of the fingerprinted similarity. In the cross and within project defect prediction, Zhang *et al.* [43] compared the performance of unsupervised classifiers versus supervised classifiers. So, they proposed a connectivity-based classifier (via spectral clustering) that in the cross-project setting ranked as one of the top classifiers among five widely-used supervised classifiers (random forest, naive Bayes, logistic regression, decision tree, and logistic model tree) and five unsupervised classifiers (k-means, partition around medoids, fuzzy C-means, neural-gas, and spectral clustering).

Zhang *et al.* [44] applied two algorithms for fine classifying images: the first one is a contextual-based convolutional neural network with deep architecture while the second one is pixel-based multilayer perceptron with shallow structure. Masetic and Subasi [45] evaluated the effect of machine learning methods in creating the model which classifies normal and congestive heart failure on the long-term electrocardiogram (ECG) time series in order of diagnosing heart failure by the automatic ECG heartbeat classification. Finally, Lopez-Martin *et al.* [46] presented a new technique for network traffic classifier based on a combination of deep learning models that can be used for Internet of Things traffic.

Moreover, an extension of the classification and multiclassification problems is the dimensionality reduction specially when large datasets are managed. Consequently, regarding the application of PCA to the different biomedical subdomains such as the one developed in our research work, different wide classification and prediction research studies have been developed in the last years. The research work developed by Mustapha and Omer [47] proposed a new and efficient combined algorithm based on Farther Distance Based on Synthetic Minority Oversampling Techniques (FD_SMOTE) and PCA, which successfully reduces the high dimensionality and balances the minority class. Pechenizkly *et al.* [48] applied PCA-based feature transformation techniques to the classification problem by applying dimensionality reduction and extraction of new components and replacement of original features by these new components. Gárate-Escamila *et al.* [49] proposed a dimensionality reduction method and finding features of heart disease by applying a feature selection technique and demonstrated that Chi-square and principal component

analysis (CHI-PCA) with RF had the highest accuracy. Thus, the usage of PCA directly from the raw data computed lower results and would require greater dimensionality to improve the results. Wosiak [50] proposed a new PCA-based method called in-group Principal Component Analysis (igPCA) method for feature reduction in the frame of cardiac arrhythmia detection and prediction. This new PCA-based method assumed that the set of attributes can be split into subgroups of similar characteristics and then subjected to PCA. Then, it transforms the feature space into a lower dimension and gives the insight into intrinsic structure of data. Zhang *et al.* [51] proposed a new applied efficient method to the MR brain classification problem that firstly employed wavelet transform to extract features from images, followed by applying PCA to reduce the dimensions of features. Then, the reduced features were submitted to a kernel support vector machine (KSVM to assist doctors in the diagnosis. On the other hand, the wide research study developed by Van der Mateen *et al.* [52] concluded nonlinear techniques perform well on selected artificial tasks, but do not outperform the traditional PCA on real-world tasks by explaining these results by identifying the weaknesses of current nonlinear techniques, and suggests how the performance of nonlinear dimensionality reduction techniques may be improved. Hahn *et al.* [53] developed a multifactor dimensionality reduction (MDR) method for collapsing high-dimensional genetic data into a single dimension thus permitting interactions to be detected in relatively small sample sizes by integrating MDR with a cross-validation strategy for estimating the classification and prediction error of multifactor models in the biomedical domain. Keogh and Pazzani [54] worked with large time series databases in spite of the problems because of the inherent high dimensionality of the data. So, they introduced a new dimensionality reduction technique called Piecewise Aggregate Approximation and demonstrated its better performance versus other ones: The Singular Value Decomposition, the Discrete Fourier Transform, and the Discrete Wavelets Transform. The research work developed by Geng *et al.* [55] developed an improved version of the Isomap supervised nonlinear dimensionality reduction method where the neighborhood graph of the input data is constructed according to a certain kind of dissimilarity between data points, which is specially designed to integrate the class information.

Once many of classification and multi-classification methods over different biomedical and open domains have been exhibited in recent years, the next section presents our research work which also highlights the use of PCA for solving the dimensionality reduction problem as many of the different approaches that have been shown in this background section.

III. METHODOLOGY

Figure 1 exhibits the methodology developed in our research work. Each one of the five phases that represent our approach are explained in detail in the following paragraphs:

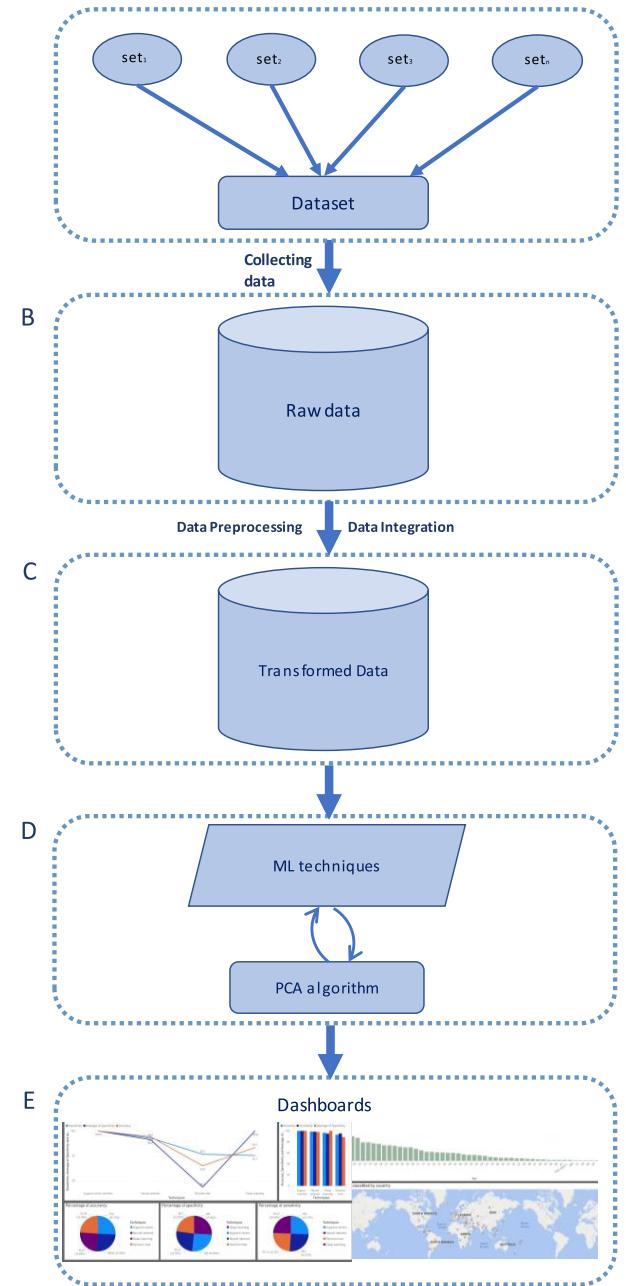


FIGURE 1. Methodology for pattern knowledge discovery. The five phases that represent this approach are explained in detail in the subsections of this section.

A. PHASE A

When we faced various datasets and tables that have many pieces of information, we would like to have all in one place: one dataset in this case. This time is where we will find the power of the merge process. “Merging” multiple datasets is the process of efficiently combining several datasets into a single big dataset. One of the main steps, as well as more complex ones in this process, is related to the alignment of the rows from each one based on common attributes or columns. It allows the addition of various data sources until a enough large dataset is composed in which conclusions can be drawn that can be extrapolated to other dataset. A case

study is usually carried out in order to validate the established approach in the proposed methodology.

B. PHASE B

Phases B and C develop the ETL (Extract, Transform, and Load) process [56], [57]. This process allows organizations to move data from multiple sources, reformat and clean it, and load it into another database, data mart, or data warehouse. By applying this process, the data is extracted from the information sources and loaded into the Data Warehouse or Data Mart. The data should be analyzed, cleaned, integrated, and transformed to adapt it to the warehouse design [58]–[60]. Below we explain more about this three steps regarding the ETL process.

In our method, concretely in the phase B, we perform the first task of the ETL process: the extraction task.

In the extraction task, data is extracted from the source into the scaffold space. Transformations if any are done in the scaffold space so that the efficiency of the source is not corrupted.

Furthermore, rollback will be a difficult process if damaged data copied straight from the source into the final database. The scaffold space provides a chance to verify the extracted data before it transforms into the final database.

Also, for extraction aim, there are defined two approaches:

- Complete Extraction.
- Partial Extraction with and without update notification.

Additionally, during this process, we should do some verifications such as:

- Record adaptation with the source data.
- Checking that no unpleasant data is loaded, like spam data.
- The data type should also be checked.
- Fragmented data should be removed.
- Finally, key position should also be checked.

C. PHASE C

Extracted data from the previous step is raw and not useful in its original form. Hence it requires to be cleaned, mapped, and transformed. This is the essential step, in fact, where the ETL process append values and modifies data.

In this step, you can perform customized operations or function collection on extracted data. There are several data called as pass-through data. These data do not need any more transformation.

There are many troubles in data integration because this step is one of the most complex ones. So, let us mention some of them:

- Committed hardware.
- Human error, transfer errors.
- Bugs and viruses.

Also, during this process, we should do several checkings such as:

- Filtering, data standardization, encoding handling.
- Measurement unit change.
- The data threshold should be checked.

- Data flow checking from the scaffold space to the middle tables.
- Necessary fields should not be null.
- Cleaning and exchanging both rows and columns.
- Performing any difficult data checking.

The last step regarding this ETL process is loading data into the target database. In order to improve it, the loading process should be optimized. If this process failed without losing data integrity, recovery methods should be able to be restarted from the last point.

There are three types of loading: Initial Load, Incremental Load, and Full Refresh.

Also, like the two previous ETL steps, loading process has checking techniques:

- Key field data should not be missing or blank.
- Test modelling views based on target tables.
- Mixed values and measure determination should also be checked.
- Data in the dimension and history tables as well as BI reports on the loaded fact should finally be checked.

Data processing and integration will be executed in order to accomplish a data quality process of transforming the dataset into a series of available data to apply different machine learning methods that will be carried out in the following phase.

D. PHASE D

In spite of all the previous phases (from A to C) are part of the general sequence in the process of data mining and knowledge discovery, by reaching more importance and transcendence, this work is focused on phase 3. This task consists of applying the set of ML techniques together with PCA for reducing the dimension and is explained below.

The PCA algorithm applies PCA to ML techniques in combination to continuously work with a smaller dataset. Dimensionality curse is a factor that has been studied and undertaken with different approaches throughout the literature and it is not an issue that has been solved, so it is always a challenge to work regarding this issue [61]–[64]. This fact of dimension reduction will allow us to obtain better precision and more correlation between input and output variables. It must be taken into account that these input variables appear transformed with respect to their initially generated information.

In this phase different machine learning methods are also carried out for comparing and complementing each other.

E. PHASE E

From the previous phase, dashboards of predictive models will be made in order of helping to make forecasts, predictions, visualizations, extracting rules and patterns, etc. This phase is very important for appreciating any of these relationships that are not always easy to find. In fact, visualization is crucial when working with large datasets because the intrinsic difficulty of the data complicates their study without visual tools [65]–[67].

TABLE 1. Epileptic seizure recognition data set.

Attribute Information
The response variable is y in column 179, the Explanatory variables X_1, X_2, \dots, X_{178}
y contains the category of the 178-dimensional input vector.
Specifically, y in $\{1, 2, 3, 4, 5\}$:
5 - eyes open, means when they were recording the EEG signal of the brain the patient had their eyes open
4 - eyes closed, means when they were recording the EEG signal the patient had their eyes closed
3 - Yes, they identify where the region of the tumor was in the brain and recording the EEG activity from the healthy brain area
2 - They recorder the EEG from the area where the tumor was located
1 - Recording of seizure activity

IV. CASE STUDY RESULTS

From the point of view of homogenizing this specific case, which is a particular case as can be contrasted to the general methodology explained in the previous section, the same structure of subsections will be carried out. We will test our methodology using a complex dataset from the health domain: an epileptic seizure recognition database.

A. PHASE A

This dataset is a pre-processed and re-structured/reshaped version of a very commonly used dataset in the featuring epileptic seizure detection. Thus, this dataset is obtained from the UC Irvine Machine Learning Repository [68].

The original dataset consists of 5 different folders where each one have 100 files and each file represents a single subject/person. So, each file is a recording regarding the brain activity of 23.6 second windows. The corresponding time-series are sampled into 4097 data points. Each data point is the value of the EEG recording at a different point in time. So, we have a total of 500 individuals where each one has 4097 data points with 23.5 seconds duration.

We divided and shuffled every 4097 data points into 23 chunks where each chunk contains 178 data points for 1 second, and each data point is the value of the EEG recording at a different time point. So now we have $23 \times 500 = 11500$ pieces of information (rows), each information contains 178 data points of 1 second (columns), the last column represents the label $y \{1,2,3,4,5\}$. The attributes information is included in table 1.

All subjects falling in classes 2, 3, 4, and 5 are subjects who did not have epileptic seizures. Only subjects in class 1 have epileptic seizures. Our motivation for creating this version of the data was to simplify access to the data via the creation of a .csv version of it. Although there are 5 classes where most authors have done binary classification, namely class 1 (Epileptic seizure) against the rest [69].

B. PHASE B

At this point, we work with a set of very specific elements that are in a structured database and will not need previous processes for being able to carry out the following C and D phases.

C. PHASE C

At this stage and in particular for this structured dataset neither pre-processing nor integration processes will be required. Therefore, this fact will allow us going to the next central point regarding the research presented in this work, phase D.

D. PHASE D

This phase is where the major contribution of our work lies and we have implemented it using the Python framework. We can introduce that Python is an open-source interpreted language that can be used interactively and with expressive syntax. Python allows programmers writing procedural or object-oriented code by using simple syntax both and readability, and also offers interactive interpreters that allow the code development and program running without previously compiling the code. So, Python also has the feature of both running and be embedded in different platforms and applications. Other main feature is that it has a very large number of mature libraries available that can be easily downloaded from its pypi repository and that offer support to the computer science area [70].

Another package used is the NumPy library, although, in general terms, this library is used by others (for example, the scikit-learn library is built on NumPy or the pandas library) for operating with arrays, basically NumPy is a collection where the elements occupy the same amount of bytes in memory, offering a standard representation for numerical data and implementations for mathematical calculations at a high level of programming [71].

Pandas is the main library in the environment of data analysis since this library is not only perfectly integrated with other libraries such as NumPy or scikit-learn, but it offers a standard representation for datasets by offering multiple requirements such as reading data in different formats, operations over rows or columns (for example: SQL operations, comparisons, lambda functions and statistical calculations) by allowing to perform fast analysis, cleaning data and performing advanced manipulation in an efficiently way [72].

Scikit-learn is a BSD licensed project built on Numpy, Scipy [73], and matplotlib that offers a library that allows running machine learning models on Python.

It offers a wide variety of predictive algorithms both supervised and unsupervised, grinds for performing different tasks (regression, classification, clustering) as well as the calculation of different metrics on the models, dimensionality reduction, pre-processing, visualization, model comparison such as those ones used in this paper in a simple and efficient framework for carrying out data analysis [74], [75].

Matplotlib is a library that allows the creation of different visualizations such as graphics, animated or static, and even interactive visualizations in Python [76].

This resource, together with other libraries, allows data to be represented graphically, helping the analyst to understand the data, as well as to make decisions step-by-step when pre-processing the data and how it affects them, as is the case

with normalization, or checking how variables behave as a function of others, among many other possibilities.

Jupyter notebook is a tool that offers IPython interactive computing, allowing users to work in different languages as python or R, offering a web interface that combines code and results in different formats such as text, math, image, video, or any other file type that can be rendered in modern browsers. This interface has been used in the research area with a significant amount of diffusion in publications and in the educational area where notebooks can be converted into different output formats such as HTML or PDF [77], [78].

This concept of multi-user environments has made it to extend for hosting notebooks in workgroups or classes, or even collaborative notebooks as is the case of Google Collaboratory (Google Colab) by allowing to run Python code into the cloud where it can be run, shared and / or modified by different users [79], [80].

Although all these tools can be installed as well as managed by the user individually, Anaconda is a BSD-licensed Python distribution which allows programmers to perform the management of the installed packages and libraries. This is an added value by being able of using an integrated environment as Anaconda. It also comes integrated with different tools such as Jupyter notebook and PyCharm IDE among others, which makes it widely used in the field of data science by allowing the installation, running and updating the different libraries in a straightforward way and managing each library interdependency with other packages [81].

Figure 2 is a PCA representation that allows us to see how they are distributed once dimensions have been reduced in the space.

In a simplified way, PCA allows representing an N-Dimensional set in main components by being able of representing the data in two or three dimensions since it is as far as human comprehension reaches. From the fourth dimension, we are not able to understand how the points are distributed in the space and each main component would include less information than its predecessor. Figure 2 shows a representation in which 22% of data variability is captured, and despite not being faithfully representative of how the original data are distributed, it allows to make an approximation in a reduced dimensional space compared to the original N-Dimensional space. In a dashboard, it would allow us to quickly see what is happening in two dimensions instead of having to explore and understand the original N-Dimensional space that is impossible to understand.

Concretely, Figure 2 shows a pattern that indicates how the dots have been concentrated in the upper part and can be seen mainly in the central part (yellow dots) while the outer ones (green and purple dots) have a greater distribution.

As Ferre concludes in [82] there is no ideal method for the selection of PCA dimensions. Throughout the literature, different methods or approaches have been discussed or followed to obtain the ideal number of main components, as is the case of [83], which uses the eigenvalue >1 criterion, or [84], which uses a cross-validation-based approach.

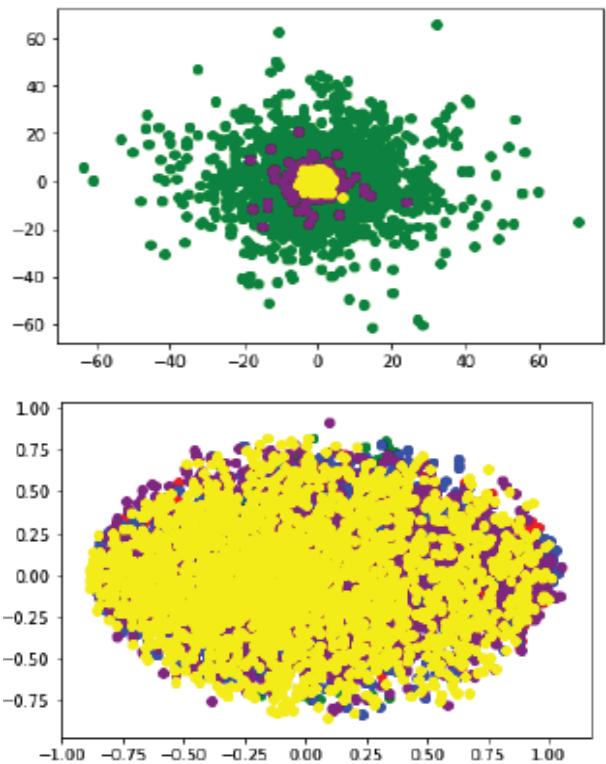


FIGURE 2. PCA representation for dimensions reduction.

For this reason, the choice of the ideal PCA dimensionality will depend on the problem being addressed. PCA allows to obtain the degree of variability captured by each main component, so a threshold of the variability to be captured can be set, for example 90%, although this value can be higher or lower. The accuracy frequently improves by using more main components and does not compensate for the temporal cost derived from such improve. In the case that we want to represent visually a space of high dimensions, we are forced to use maximum 3 components since it is up to where the humans can understand in an intuitive way.

As mentioned previously, PCA is also used in some cases where there is a high dimensionality due to the particularities of the algorithms for reducing its dimensionality. It is known that we lose accuracy, but we improve in computer time. In some cases, it is necessary to use this technique of dimensionality reduction in spite of algorithm restrictions that scale more wrongly as dimensionality grows, especially when we are faced with datasets with a high dimensional space. PCA has a method to achieve through a linear combination of variables where the first main component will be the one that has more weight, and the following main components will become less important successively in terms of variability representation. These components are constituted by a combination of the rest of variables. For example, the first main component will be represented by a percentage of all other variables (i.e. In a case that we have 3 variables and we want to reduce it to only one variable, the main component would

be formed by 50% the variable 1, 30% the variable 2, and 20% the variable 3). This approach will allow us to know which variables have more influence on each main component.

This approach can be applied to a dashboard and would allow visualizing in a graphical way a data of high dimensionality which is complex to understand into a reduced visual space that allows us to obtain information on the situation in a faster way. As it has also been mentioned before, this visualization is still complex because it reduces a high dimensional space in a two- or three-dimensional spaces but it allows a visual approach to the problem (Figure 2).

Table 2 shows the results of the different algorithms that have been applied to the dimensionally reduced spaces with different parameters to tune their metrics.

The algorithms used in this experiment that show their results in Table 2 were introduced in section I of the introduction. In summary, SVC is a novel and nonparametric clustering algorithm which is based on the support vector approaches. Logistic regression establishes the correlation among one dependent binary variable and one or more independent variables. For classifying a case, KNN takes a majority selection from its neighbors with the case being allocated to the category highest common between its K-nearest neighbors measured by a distance function. A CART tree is a binary decision tree that is created by dividing a node toward two child nodes frequently. NB is a classification algorithm based on Bayes' Theorem with a hypothesis of independence between predictors. RF creates many decision trees and merges all the decision trees to obtain the highest accuracy and, most important, further constant prediction. MLP is a classical as well as the most well-known artificial neural networks that, generally, consists of three layers, an input layer, a hidden layer, and an output layer. Every node could be a neuron that uses a nonlinear activation function, very often sigmoid, excluding for the input nodes. For training, MLP which is a supervised learning method uses backpropagation.

As for the scalers applied to the algorithms shown in Table 2, we have to start from the fact that a priori, on a highly dimensioned set it is difficult or complex to know which is the best scaler for the different applied algorithms. For example, MinMaxScaler by its nature is not a suitable scaler when we have very extreme values since the intermediate values are clustered in a very small range making the task of the algorithms complicated. Table 2 shows the permutation of the algorithms with the different scalers and the accuracy obtained for them, with the objective of choosing the scaler and the algorithm that best fits the data of the case study. The effect of the scalers can be clearly seen in the column of the SVC algorithm, that with the minmaxscaler obtains a 0.2363 of accuracy, and this value get an improvement if we use the standard scaler up to 0.56, or up to 0.65 in the case of the Quant-Normal scaler. Also, in the table, we can see as it is already known, that there are algorithms that by their nature, are not affected by the scaler, as is the case of RandomForest. As shown in table 2, RandomForest is the algorithm that offers the best results, followed

TABLE 2. Results of the different algorithms applied.

	SVC	LR	KNN	CART	NB	RF	MLP
MinMaxScaler	0.2363	0.2615	0.4768	0.4777	0.4374	0.7030	0.4440
StandardScaler	0.5630	0.2502	0.4787	0.4774	0.4374	0.7027	0.6796
MaxAbsScaler	0.3009	0.2603	0.4771	0.4803	0.4374	0.7022	0.6892
RobustScaler	0.6343	0.2495	0.4794	0.4790	0.4374	0.7015	0.6797
Quant-Normal	0.6591	0.2556	0.4296	0.4803	0.4516	0.7010	0.6684
Quant-Uniform	0.4977	0.2428	0.4056	0.4797	0.4397	0.7038	0.6680
PowerTransf-yeoJhonson	0.5623	0.3119	0.4790	0.4823	0.4377	0.7045	0.6897
Normalizer	0.2657	0.2357	0.5286	0.3689	0.2312	0.5845	0.5950

by MLP with the PowerTransf-yeo-Jhonson scaler and with the MaxAbsScaler.

We can observe that the Radom Forest algorithm obtains the best accuracy with 70.3%.

These results indicate that both random forests mainly, but also MLP behave very stable in general terms and are the best algorithms to optimize. Moreover, as Figure 2 exhibits, we can see the importance of the selection of one scaler or another for the algorithms due mainly to the taxonomy of the data.

Although random forest is not affected by a scaler due to the very nature of the algorithm, we can see more clearly how it affects the SVC results. The standard parameters of the scikit-learn library are used for the experiments. This selection of parameters is justified because they are the best initial parameters, offering good results in a stable way on different datasets. In addition, scikit-learn incorporates changes in both parameters and algorithms, improving the implementation of the algorithm, as is the case of MLP and the incorporation of adam optimizer or updating a standard parameter from one version to another and the justification of the change as is the case of the SVC algorithm [85]–[87]. This approach allows us to obtain the algorithm/scaler set that works best and on which to invest effort since the tuning of these parameters is a fine and complex adjustment. This approach requires an understanding of both the algorithm and the different metrics and what are the problems that the algorithm is having when classifying. In the case of using some automatic technique for parameter optimization, it should be considered that these are very expensive especially when the parameters of the algorithm are continuous, so this method allows us to dedicate resources on the tuple algorithm/scaler that is working better.

E. PHASE E

The visualization phase, with multiple types of dashboards to facilitate the understanding of correlations between data is an area that is continuously growing and completely current.

In this context, as shown in figure 3, we will use the confusion matrix which we have applied certain graphical specific packages, provides a rich visual representation with a wealth of detail. The use of the confusion matrix is a key element for statistical classification within the machine learning field and mainly within the supervised learning algorithms [88].

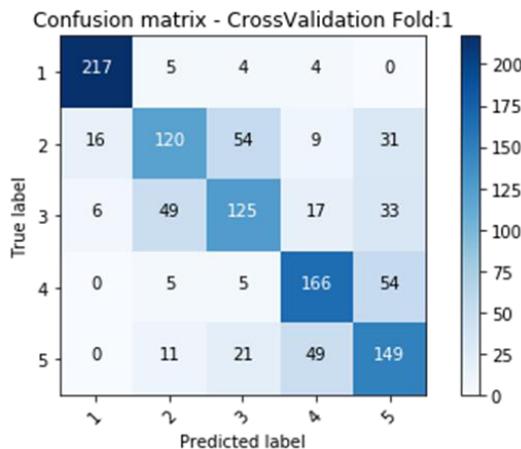


FIGURE 3. Confusion matrix. This figure represents the confusion matrix in a particular way since, as can be seen, the blue tone intensifies as the precision increases.

We can appreciate in the confusion matrix that the classification obtains better accuracy with the classes 1, 4, and 5, in this order. Furthermore, it is illustrated that classes two and three produce more misclassification, besides the accuracy is high, in general. The visual representation of the confusion matrix, where the blue tone intensifies as the precision increases helps to understand the correctness in the classification of every class.

Besides the accuracy, it allows obtaining directly from their values of the elements of the matrix, the sensitivity and specificity measures [89], [90].

V. CONCLUSION

The dimensionality reduction problem is responsible for the process of reducing the dimension of a large feature set into a combined reduced-feature set that makes up a large sphere in the n-dimensional space. Hence the dimensionality reduction problem presents advantages like computational efficiency and redundancy removing and other disadvantages such as data-losing and feature-losing in datasets.

We have worked in the fields of dimensionality reduction in large datasets. So, our machine learning method for reducing dimensional space in large datasets merges all datasets as a huge one in the first stage. Then, we used PCA for dimensionality reduction, that solves the problems encountered in previous proposals, after applying the ETL process. Hence, our novel method highlights the use of the Python framework where standard representation for numerical data and implementations for mathematical calculations at a high level of programming are performed in an efficient environment. Besides, we have applied our research in the real environment Epileptic Seizure Recognition Data Set provided by UCI Machine Learning Repository.

The experimental results show that Random Forest outcomes better than the rest of the algorithms with this complex dataset obtaining an accuracy of 70.3%. We can also

appreciate that MLP behaves very stable in general terms and together with Random Forest are the best algorithms to be optimized. In this regard, as shown in figure 2, we can realize how important is the correct selection of a particular scaler or another for the algorithms due mainly to the taxonomy of the data.

The intrinsic complexity of the dataset tested in this manuscript suggests excellent conditions for adaptation to other health care scenarios, where the complexity of biological systems will also be adapted to our generic methodology.

Although the results obtained are very encouraging, the greatest achievement in the authors' opinion is the possibility of future lines. At this point and due to the complexity of the data currently generated, with characteristics of variability and other aspects besides the volume, such as velocity, veracity of the big data, where a new world has opened up to continue in this fascinating area of data science research.

REFERENCES

- [1] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits Syst. Mag.*, vol. 6, no. 3, pp. 21–44, 3rd Quart., 2006.
- [2] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. Int. Workshop Multiple Classifier Syst.*. Berlin, Germany: Springer, 2000.
- [3] R. M. O. Cruz, R. Sabourin, and G. D. C. Cavalcanti, "Dynamic classifier selection: Recent advances and perspectives," *Inf. Fusion*, vol. 41, pp. 195–216, May 2018.
- [4] S. Bashbaghi, E. Granger, R. Sabourin, and G.-A. Bilodeau, "Dynamic selection of exemplar-SVMs for watch-list screening through domain adaptation," in *Proc. 6th Int. Conf. Pattern Recognit. Appl. Methods*, 2017, pp. 738–745.
- [5] P. R. L. de Almeida, E. J. da Silva Junior, T. M. Celinski, A. de Souza Britto, L. E. S. de Oliveira, and A. L. Koerich, "Music genre classification using dynamic selection of ensemble of classifiers," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2012, pp. 2700–2705.
- [6] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen, "Benchmarking state-of-the-art classification algorithms for credit scoring," *J. Oper. Res. Soc.*, vol. 54, no. 6, pp. 627–635, Jun. 2003.
- [7] H. Xiao, Z. Xiao, and Y. Wang, "Ensemble classification based on supervised clustering for credit scoring," *Appl. Soft Comput.*, vol. 43, pp. 73–86, Jun. 2016.
- [8] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 463–484, Jul. 2012.
- [9] C. Porcel, A. Tejeda-Lorente, M. A. Martínez, and E. Herrera-Viedma, "A hybrid recommender system for the selective dissemination of research resources in a technology transfer office," *Inf. Sci.*, vol. 184, no. 1, pp. 1–19, Feb. 2012.
- [10] M. Jaher, A. Tötscher, and R. Legenstein, "Combining predictions for accurate recommender systems," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 693–701.
- [11] D. Di Nucci, F. Palomba, R. Oliveto, and A. De Lucia, "Dynamic selection of classifiers in bug prediction: An adaptive method," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 1, no. 3, pp. 202–212, Jun. 2017.
- [12] A. Panichella, R. Oliveto, and A. De Lucia, "Cross-project defect prediction models: L'Union fait la force," in *Proc. Softw. Evol. Week-IEEE Conf. Softw. Maintenance, Reeng., Reverse Eng. (CSMR-WCRE)*, Feb. 2014, pp. 164–173.
- [13] G. Giacinto, F. Roli, and L. Didaci, "Fusion of multiple classifiers for intrusion detection in computer networks," *Pattern Recognit. Lett.*, vol. 24, no. 12, pp. 1795–1803, Aug. 2003.
- [14] G. Giacinto, R. Perdisci, M. Del Rio, and F. Roli, "Intrusion detection in computer networks by a modular ensemble of one-class classifiers," *Inf. Fusion*, vol. 9, no. 1, pp. 69–82, Jan. 2008.
- [15] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Wohniak, "Ensemble learning for data stream analysis: A survey," *Inf. Fusion*, vol. 37, pp. 132–156, Sep. 2017.

- [16] L. I. Kuncheva, "Classifier ensembles for changing environments," in *Proc. Int. Workshop Multiple Classifier Syst.* Berlin, Germany: Springer, 2004.
- [17] R. Polikar, L. Upda, S. S. Upda, and V. Honavar, "Learn++: An incremental learning algorithm for supervised neural networks," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 31, no. 4, pp. 497–508, Nov. 2001.
- [18] L. Breiman, J. H. Friedman, R. Olshen, and C. J. Stone, *Classification and Regression Trees*. Pacific Grove, CA, USA: Wadsworth & Brooks/Cole Advanced Books & Software, 1984.
- [19] A. Ben-Hur, "Support vector clustering," *Scholarpedia*, vol. 3, no. 6, p. 5187, 2001.
- [20] D. D. Lewis, *Naive(Bayes)at Forty: The Independence Assumption in Information Retrieval* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 1398, 1998, pp. 4–15.
- [21] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
- [22] J. Breckling, Ed., *The Analysis of Directional Time Series: Applications to Wind Speed and Direction* (Lecture Notes in Statistics), vol. 61. Berlin, Germany: Springer, 1989.
- [23] J. Tolles and W. J. Meurer, "Logistic regression: Relating patient characteristics to outcomes," *Jama*, vol. 316, no. 5, pp. 533–534, 2016.
- [24] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2009.
- [25] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, "Efficient kNN classification algorithm for big data," *Neurocomputing*, vol. 195, pp. 143–148, Jun. 2016.
- [26] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 1721–1730.
- [27] A. Groce, T. Kulesza, C. Zhang, S. Shamasunder, M. Burnett, W.-K. Wong, S. Stumpf, S. Das, A. Shinsel, F. Bice, and K. McIntosh, "You are the only possible oracle: Effective test selection for end users of interactive machine learning systems," *IEEE Trans. Softw. Eng.*, vol. 40, no. 3, pp. 307–323, Mar. 2014.
- [28] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan, "Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model," *Ann. Appl. Statist.*, vol. 9, no. 3, pp. 1350–1371, Sep. 2015.
- [29] P. Zhang, J. Wang, A. Farhadi, M. Hebert, and D. Parikh, "Predicting failures of vision systems," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3566–3573.
- [30] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Inf. Fusion*, vol. 16, pp. 3–17, Mar. 2014.
- [31] X. Jun, Y. Lu, Z. Lei, and D. Guolun, "Conditional entropy based classifier chains for multi-label classification," *Neurocomputing*, vol. 335, pp. 185–194, Mar. 2019.
- [32] J. Lee, I. Yu, J. Park, and D.-W. Kim, "Memetic feature selection for multilabel text categorization using label frequency difference," *Inf. Sci.*, vol. 485, pp. 263–280, Jun. 2019.
- [33] P. Zhang, G. Liu, and W. Gao, "Distinguishing two types of labels for multi-label feature selection," *Pattern Recognit.*, vol. 95, pp. 72–82, Nov. 2019.
- [34] E. L. Mencía and F. Janssen, "Learning rules for multi-label classification: A stacking and a separate-and-conquer approach," *Mach. Learn.*, vol. 105, no. 1, pp. 77–126, Oct. 2016.
- [35] J. Nam, E. L. Mencía, H. J. Kim, and J. Fürnkranz, "Maximizing subset accuracy with recurrent neural networks in multi-label classification," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5413–5423.
- [36] J. Read, L. Martino, and J. Hollmén, "Multi-label methods for prediction with sequential data," *Pattern Recognit.*, vol. 63, pp. 45–55, Mar. 2017.
- [37] P. Teisseire, "CCnet: Joint multi-label classification and feature selection using classifier chains and elastic net regularization," *Neurocomputing*, vol. 235, pp. 98–111, Apr. 2017.
- [38] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.
- [39] A. Subudhi, M. Dash, and S. Sabut, "Automated segmentation and classification of brain stroke using expectation-maximization and random forest classifier," *Biocyber. Biomed. Eng.*, vol. 40, no. 1, pp. 277–289, Jan. 2020.
- [40] Z. Liu, Q. Pan, J. Dezert, J.-W. Han, and Y. He, "Classifier fusion with contextual reliability evaluation," *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1605–1618, May 2018.
- [41] H. S. Dadi and G. K. M. Pillutla, "Improved face recognition rate using HOG features and SVM classifier," *IOSR J. Electron. Commun. Eng.*, vol. 11, no. 04, pp. 34–44, Apr. 2016.
- [42] X. Cheng, S.-G. Zhao, X. Xiao, K.-C. Chou, "iATC-mISF: A multi-label classifier for predicting the classes of anatomical therapeutic chemicals," *Bioinformatics*, vol. 33, no. 3, pp. 341–346, 2017.
- [43] F. Zhang, Q. Zheng, Y. Zou, and A. E. Hassan, "Cross-project defect prediction using a connectivity-based unsupervised classifier," in *Proc. Int. Conf. Softw. Eng.*, May 2016, pp. 309–320.
- [44] C. Zhang, X. Pan, H. Li, A. Gardiner, I. Sargent, J. Hare, and P. M. Atkinson, "A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 140, pp. 133–144, Jun. 2018.
- [45] Z. Masetic and A. Subasi, "Congestive heart failure detection using random forest classifier," *Comput. Methods Programs Biomed.*, vol. 130, pp. 54–64, Jul. 2016.
- [46] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for Internet of Things," *IEEE Access*, vol. 5, pp. 18042–18050, Sep. 2017.
- [47] N. Mustafa, R. A. Memon, J.-P. Li, and M. Z. Omer, "A classification model for imbalanced medical data based on PCA and farther distance based synthetic minority oversampling technique," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 1, pp. 61–67, 2017.
- [48] M. Pechenizkiy, A. Tsymbal, and S. Puuronen, "PCA-based feature transformation for classification: Issues in medical diagnostics," in *Proc. 17th IEEE Symp. Comput.-Based Med. Syst.*, Jun. 2004, pp. 535–540.
- [49] A. K. Gárate-Escamilia, A. H. El Hassani, and E. Andrès, *Classification Models for Heart Disease Prediction Using Feature Selection and PCA*. Amsterdam, The Netherlands: Elsevier, 2020.
- [50] A. W.-O. Physics. (2019). *Principal Component Analysis Based on data Characteristics for Dimensionality Reduction of ECG Recordings in Arrhythmia Classification*. [Online]. Available: <https://www.degruyter.com>
- [51] Y. D. Zhang and L. Wu, "An MR brain images classifier via principal component analysis and kernel support vector machine," in *Proc. Prog. Electromagn. Res.*, vol. 130, 2012, pp. 369–388.
- [52] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: A comparative review," *J. Mach. Learn. Res.*, vol. 10, nos. 66–71, p. 13, 2009.
- [53] L. W. Hahn, M. D. Ritchie, and J. H. Moore, "Multifactor dimensionality reduction software for detecting gene-gene and gene-environment interactions," *Bioinformatics*, vol. 19, no. 3, pp. 376–382, Feb. 2003.
- [54] E. J. Keogh and M. J. Pazzani, "A simple dimensionality reduction technique for fast similarity search in large time series databases," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, Apr. 2000, pp. 122–133.
- [55] X. Geng, D.-C. Zhan, and Z.-H. Zhou, "Supervised nonlinear dimensionality reduction for visualization and classification," *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1098–1107, Dec. 2005.
- [56] K. Kakish and T. A. Kraft. (2012). *ETL Evolution for Real-Time Data Warehousing*. [Online]. Available: <https://www.researchgate.net>
- [57] R. Kimball and J. Caserta, *The Data Warehouse ETL Toolkit*. 2004.
- [58] B. Devlin and L. Cote, *Data Warehouse: From Architecture to Implementation*. 1996.
- [59] R. Kimball, M. Ross, W. Thornthwaite, J. Mundy, and B. Becker, *The Data Warehouse Lifecycle Toolkit*. 2008.
- [60] W. H. Inmon. (1996). *The Data Warehouse and Data Mining*. [Online]. Available: <https://www.go.gale.com>
- [61] J. H. Friedman, "On bias, variance, 0/1-loss, and the curse-of-dimensionality," *Data Mining Knowl. Discovery*, vol. 1, no. 1, pp. 55–77, 1997.
- [62] M. Verleysen and D. François, "The curse of dimensionality in data mining and time series prediction," in *Proc. Int. Work-Conf. Artif. Neural Netw.* Berlin, Germany: Springer, Jun. 2005, pp. 758–770.
- [63] M. A. Bessa, R. Bostanabad, Z. Liu, A. Hu, D. W. Apley, C. Brinson, W. Chen, and W. K. Liu, "A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality," *Comput. Methods Appl. Mech. Eng.*, vol. 320, pp. 633–667, Jun. 2017.
- [64] F. Bach, "Breaking the curse of dimensionality with convex neural networks," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 629–681, 2017.
- [65] O. Yigitbasoglu and O. Velcu, *A Review of Dashboards in Performance Management: Implications for Design and Research*. Amsterdam, The Netherlands: Elsevier, 2012.

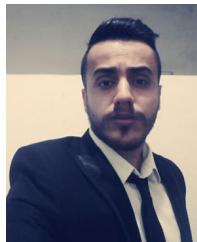
- [66] W. W. Eckerson, *Performance Dashboards: Measuring, Monitoring, and Managing Your Business*. Hoboken, NJ, USA: Wiley, 2010.
- [67] K. Verbert, S. Govaerts, E. Duval, J. L. Santos, F. Van Assche, G. Parra, and J. Klerkx, "Learning dashboards: an overview and future research opportunities," *Pers. Ubiquitous Comput.*, vol. 18, no. 6, pp. 1499–1514, 2014.
- [68] UCI Machine Learning Repository. Accessed: Jun. 9, 2020. [Online]. Available: <https://archive.ics.uci.edu/ml/index.php>
- [69] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *APS*, vol. 64, no. 6, p. 8, 2001.
- [70] P. F. Dubois, "Guest editor's introduction: Python: Batteries included," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 7–9, May-2007.
- [71] S. van der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy array: A structure for efficient numerical computation," *Comput. Sci. Eng.*, vol. 13, no. 2, pp. 22–30, Mar. 2011.
- [72] W. McKinney, "Data structures for statistical computing in Python," in *Proc. 9th Python Sci. Conf.*, vol. 445, 2010, pp. 51–56.
- [73] P. Virtanen et al. (2020). *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*. [Online]. Available: <https://www.nature.com>
- [74] Scikit-Learn: Machine Learning in Python—Scikit-Learn 0.22.2 Documentation. Accessed: May 6, 2020. [Online]. Available: <https://scikit-learn.org/stable/>
- [75] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [76] N. Ari. (2014). *Matplotlib in Python*. [Online]. Available: www.ieeexplore.ieee.org
- [77] M. Ragan-Kelley, F. Perez, B. Granger, T. Kluyver, P. Ivanov, J. Frederic, and M. Bussonnier. (2014). *The Jupyter/IPython Architecture: A Unified View of Computational Research, From Interactive Exploration to Communication and Publication*. [Online]. Available: <https://adsabs.harvard.edu>
- [78] H. Nguyen, D. A. Case, and A. S. Rose, "NGLview—interactive molecular graphics for Jupyter notebooks," *Bioinformatics*, vol. 34, no. 7, pp. 1241–1242, 2018. [Online]. Available: <https://academic.oup.com>
- [79] T. Carneiro, R. V. M. Da Nóbrega, T. Nepomuceno, G.-B. Bian, V. H. C. De Albuquerque, and P. P. R. Filho. (2018). *Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications*. [Online]. Available: <https://ieeexplore.ieee.org>
- [80] Colaboratory. Accessed: May 6, 2020. [Online]. Available: <https://colab.research.google.com/notebooks/intro.ipynb>
- [81] Individual Edition|Anaconda. Accessed: May 6, 2020. [Online]. Available: <https://www.anaconda.com/products/individual>
- [82] L. Ferré, "Selection of components in principal component analysis: A comparison of methods," *Comput. Statist. Data Anal.*, vol. 19, no. 6, pp. 669–682, Jun. 1995.
- [83] J. Josse and F. Husson, "Selecting the number of components in principal component analysis using cross-validation approximations," *Comput. Statist. Data Anal.*, vol. 56, no. 6, pp. 1869–1879, Jun. 2012.
- [84] Y. Sung, S. M. Choi, H. Ahn, and Y.-A. Song, "Dimensions of luxury brand personality: Scale development and validation," *Psychol. Marketing*, vol. 32, no. 1, pp. 121–132, Jan. 2015.
- [85] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [86] sklearn.neural_network.MLPClassifier—Scikit-Learn 0.23.0 Documentation. Accessed: May 14, 2020. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier
- [87] sklearn.svm.SVC—Scikit-Learn 0.23.0 Documentation. Accessed: May 14, 2020. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC>
- [88] S. Visa, B. Ramsay, A. Ralescu, and E. Van Der Knaap, "Confusion matrix-based feature selection," in *Proc. CEUR Workshop*, 2011, pp. 120–127.
- [89] A. K. Akobeng, "Understanding diagnostic tests 1: Sensitivity, specificity and predictive values," *Acta Paediatrica*, vol. 96, no. 3, pp. 338–341, Mar. 2007.
- [90] H. Brenner and O. Gefeller, "Variation of sensitivity, specificity, likelihood ratios and predictive values with disease prevalence," *Statist. Med.*, vol. 16, no. 9, pp. 981–991, May 1997.



RAFAEL MUÑOZ TEROL received the master's and Ph.D. degrees in computer science from the University of Alicante, in 2002 and 2009, respectively. Since 2002, he has been a Lecturer with the Lucentia Research Group, Department of Software and Computing Systems, University of Alicante. His research interests include big data analytics, predictive data mining, business intelligence systems, big data, and key performance indicators.



ALEJANDRO REINA REINA received the degree in computer science, in 2018. He is currently pursuing the Ph.D. degree in computer science with the University of Alicante. Since 2018, he has been with the Department of Languages and Information System, University of Alicante, as a Researcher in the IoT. His main research interests include big data, data analytics, machine learning, analytics models, the IoT, and e-health area.



SABER ZIAE received the bachelor's degree in computer software engineering from the Lamei Gorgani Institute of Higher Education, in 2016, and the master's degree in computer software engineering from the Babol Noshirvani University of Technology, in 2019.

He currently works as a Freelance Researcher in the area of data mining, machine learning, big data analytics, and recommender systems.



DAVID GIL is currently an Assistant Teacher with the Department of Computing Technology and Data Processing, University of Alicante. He has participated in numerous national and international projects. He has participated with many conferences. His work has been published in international journals and conferences, with more than 50 published articles. He has agreements with private companies and public organizations, related to his research topics. His main research interests include artificial intelligence applications, data mining, open data, big data, and decision support systems in medical and cognitive sciences.