IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 31, 2020, accepted September 14, 2020, date of publication September 18, 2020,
date of current version September 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3024558

Automatic Learning Framework for Pharmaceutical Record Matching

JOSÉ LUIS LÓPEZ-CUADRADO[1], ISRAEL GONZÁLEZ-CARRASCO[1], JESÚS LEONARDO LÓPEZ-HERNÁNDEZ[1], PALOMA MARTÍNEZ-FERNÁNDEZ[1], AND JOSÉ LUIS MARTÍNEZ-FERNÁNDEZ[1,2]

[1]Computer Science Department, Universidad Carlos III de Madrid, 28911 Leganés, Spain
[2]MeaningCloud LLC, New York, NY 11106, USA

Corresponding author: Israel González-Carrasco (igcarras@inf.uc3m.es)

This work was supported by the Research Program of the Ministry of Economy and Competitiveness, Government of Spain, through the DeepEMR Project, under Grant TIN2017-87548-C2-1-R.

ABSTRACT Pharmaceutical manufacturers need to analyse a vast number of products in their daily activities. Many times, the same product can be registered several times by different systems using different attributes, and these companies require accurate and quality information regarding their products since these products are drugs. The central hypothesis of this research work is that machine learning can be applied to this domain to efficiently merge different data sources and match the records related to the same product. No human is able to do this in a reasonable way because the number of records to be matched is extremely high. This article presents a framework for pharmaceutical record matching based on machine learning techniques in a big data environment. The proposed framework aims to explode the well-known rules for the matching of records from different databases for training machine learning models. Then the trained models are evaluated by predicting matches with records that do not follow these known rules. Finally, the production environment is simulated by generating a huge amount of combinations of records and predicting the matches. The obtained results show that, despite the good results obtained with the training datasets, in the production environment, the average accuracy of the best model is around 85%. That shows that matches which do not follow the known rules can be predicted and, considering that there is not a human way to process this amount of data, the results are promising.

INDEX TERMS Big data, data integration, machine learning, pattern detection, medicine.

I. INTRODUCTION

With the move from traditional databases (DB) inside a company to scenarios where new services demand the capability of sharing data intra/inter organizations, efficient data integration approaches are required. This prevalent problem of data integration from heterogeneous sources is a challenge in many fields such as biology, medicine, government among others. There are several reasons that make difficult integrating and sharing data such as different software systems talking to each other (for instance, relational database management systems that are based on SQL standard but with differences that need to make compatible), and semantic heterogeneity consisting on having several semantic representations of an universe of discourse that should be integrated in a common schema. More in detail, if data have to be ready for use in decision making, two approaches can be followed in retrieving and combining data from multiple sources: creating a new DB (datawarehouse) or accessing original sources without building a new DB. Regardless the approach, similar tasks have to be performed on data to fix or remove data in a DB because of wrong entered data, incomplete data (e.g., adding a *ZIP code* to an *address*) improperly formatted (e.g., distinct date formats, scales of measurement units, etc.), semantic heterogeneity (attributes that refer to the same concept, e.g. *class*/*category* or *address*/*location*, granularity as matching of an attribute to two or more attributes, e.g. *price* in a table and base *price+tax rate* in another table are synonyms). Sometimes, different syntactic conventions are applied (e.g., *Street*, *St.*, *Str.*) and usually there are differences about how real-world values and objects are represented (data heterogeneity), e.g. multiple references to the same entity

The associate editor coordinating the review of this manuscript and approving it for publication was Shaojun Wang.

(*USA*, *United States of America*, *United States*, *US*, *U.S.*). For a detailed description of problems and techniques to solve semantic and data heterogeneity in integration see [1].

Semantic and data heterogeneity are problems related to the process of matching names and values of attributes that are different in different databases but refer to the same concept. For example, in DB1 there can be a field called 'Customer name' with a value like 'Company A Co.' and DB2 can have an attribute 'Wholesaler' with a value like 'Company B Co.'. In the matching process of the DBs, both fields should be the same, as wholesalers are also customers. ML techniques can identify patterns in the attributes suggesting that both fields should be considered as referring to the same concept, 'Customer'. Matching and reconciling data among different DB and files is a time-consuming manual task and ML approaches are required. The work presented in this article focuses on solving the matching problem through ML techniques.

Pharmaceutical domain is one of these scenarios where new integrations approaches to match records from different sources are required. Moreover, regulatory authorities and pharmaceutical manufacturers play a role in public safety. Taking this into account pharmaceutical manufacturers have made great efforts focused on responsibilities for providing accurate and quality information regarding drugs [2].

For the above-mentioned reasons, the main motivation of this research is the necessity of great pharmaceutical manufacturers to analyse the huge number of products generated in their worldwide activities, considering that the same product can be registered several times by different systems using different attributes.

There is a business need for these integration requirements. Big pharmaceutical business growth is partially achieved through the acquisition or merging of smaller (or sometimes not so smaller) pharmaceutical companies. After the acquisition, the need to integrate previously isolated systems and data arise. New products are included in the portfolio of the new company, business processes need to be aligned (for example, to manage product supply in an integrated manner) requiring the merging of different data inputs. Building a new system from scratch is a fantasy, the only road to take is allowing the integration of previously isolated systems. Regarding data, the aforementioned need to map different semantic models arise, and it is a hard work to do. The hypothesis of this research work is that deep learning can be the solution to efficiently merge different data sources using reduced domain knowledge.

No human is capable to do this in a reasonable way because the number of records to be matched is extremely high. For this reason, it is required an automatic learning approach, based on Machine Learning (ML) techniques, capable to learn the hidden patterns that allow determining whether two records from different systems represent the same product or not. A rule-based approach is possible for given systems and operations, however new rules should be defined in case of new types of records or new types of operations. In a general case, when two data sources must be merged, humans can provide some mapping rules that cover up to the 70% of cases. This data can be used to build a training set to develop a ML model. This model should allow for a higher matching accuracy in an automatic way. Thus, a machine learning strategy is a better approach for this problem.

Furthermore, this article takes the work carried out by the authors in [3], which demonstrated the validity of a ML-based framework for matching heterogeneous records of bank operations. The main breakthroughs of the framework are: (i) classify records with high accuracy (ii) identify the most relevant variables for prediction (iii) detect common structures among the records.

In summary, the main goal of this article is to introduce a framework for solving data integration based on automatic learning using ML techniques in a big data environment. The proposed framework includes several steps, in order to move from not homogeneous data to structured information and for the automatic detection of relationships between pharmaceutical products, taking into account the large volume of data and different data sources involved in the process. The first step, pre-processing, allows merging the unstructured information of pharmaceutical products from different sources. The second step explores different ML approaches in the problem proposed. The last step is the production environment. This step matches or links each product from one source with the corresponding product in other sources. Finally, the post-processing analyses all the outputs of the second stage in order to give a detailed report of all the matching records detected.

The rest of the document is organized as follows: Section 2 contains a review of works related to ML, data mining and record matching in medicine and drug context. Section 3 discusses the main features of the framework proposed, including a usage scenario and the main components of its architecture. Section 4 describes the assessment of this tool. This section also includes a description of the sample, the method used. along with test results and a final discussion. Finally, the paper ends with a discussion on research findings, limitations, concluding remarks and future work.

II. RELATED WORK

A. MACHINE LEARNING CLASSIFIERS

Classification is a supervised learning approach in which the classifier learns from the data input given to it and then uses this learning to classify new observation. In particular, Multilayer Perceptron (MLP), Deep Neural Networks (DNN), Logistic-Regression (LoG), Support Vector Machines (SVM), K-Nearest Neighbours (KNN), Random Forests (RFO) are some ML techniques which are currently valuable tools for researchers and companies in many domains for solving complex problems in various fields such as process control [4], vehicle driving [5], weather forecasting [6], medical diagnosis [7], forecasting foreign exchange rates [8] or speech recognition, pattern recognition and computer vision [9].

MLP is a feed forward neural network and maps the inputs to a fitting set of outputs. MLP is made up of several layers of nodes inside a graph, so that each layer is entirely linked to the next one with a nonlinear activation function, excluding the input nodes. MLP employs a supervised learning technique called back propagation for training purposes and a nonlinear activation function [10], [11]. In the medical field, MLP have been widely explored for diagnosis, prediction or decision support [12].

DNN are neural network architectures formed by many layers [13]. DNN can represent functions with higher complexity if the numbers of layers and units in a single layer are increased [9], [14]. Nowadays, DNN is a trending technique and is employed in many areas such as drug-drug interaction [15], [16], medical predictions [17] or pharmaceutical sales forecasting [18].

LoG is a widely used statistical direct probability model and has been utilized in numerous landslide susceptibility assessments, providing accurate and reliable results in a rather simple manner. Based on its learning mechanism it is characterized as a discriminative model which estimates the probability for a given feature (x) and the label (y) directly from the training data by minimizing error [19]. Numerous papers can be found through the scientific literature that take advantage of their ability to sufficiently assess data, including the logistic regression approach [20].

SVM are universal classifiers and are widely utilized both for the classification of patterns as well as nonlinear regression. The main idea behind a SVM is to construct a hyperplane as a decision dimension which maximizes the margin of separation between the positive and negative examples in a data set [21]. This induction principle is based on the fact that the error coefficient of the test data, that is, the coefficient of the generalization error, is limited by the sum of the coefficient of the training error, and this term depends on the Vapnik-Chervonenkis dimension [22]. The performance of a support vector machine (SVM) depends highly on the selection of the kernel function type and relevant parameters [23]. SVM classifiers have been used for image denoising [24], multi-class sentiment classification [25], medical diagnosis [26], or even for online suicide prevention [27].

SVM-L is a linear classifier based on SVM the approach. The SVM-L classifier is implemented specifically for massive levels of data and features. SVM-L have been used for both feature ranking and classification in different domains [28]–[30].

KNN is a popular classification method in data mining and statistics because of its simple implementation and significant classification performance. KNN classifier is a type of instance-based learning or non-generalizing learning: it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the nearest neighbours of each point: a query point is assigned to the data class which has the most representatives within the nearest neighbours of the point. However, it is impractical for traditional KNN methods to assign a fixed k value (even though set by experts) to all test samples [31], [32]. KNNs have been used for classification tasks as visual recognition [33], text categorization [34] and medicine [35].

RFO classifier is an ensemble classifier that produces multiple decision trees, using a randomly selected subset of training samples and variables. Over the last two decades the use of the RFO classifier has received increasing attention due to the excellent classification results obtained and the speed of processing [36]. RFO algorithm is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size, but the samples are drawn with replacement in this case. The Number of Trees (NT) in the RFO algorithm for supervised learning has to be set by the user. It is unclear whether NT parameter should simply be set to the largest computationally manageable value or whether a smaller NT parameter may be sufficient or in some cases even better [37], [38]. RFO classifiers have been used for account classification in online social networks [39], image classification [40] or feature extraction [41].

Finally, ML classifiers are able to generalize behaviours based on unstructured information from previous examples. ML classifiers can be applied to a wide range of highly complex and non-linear domains because of their variety of design alternatives. Nevertheless, this variety of design alternatives can sometimes be a disadvantage: the lack of guidelines can lead the designer to make arbitrary decisions or to use brute force techniques. Some new theoretical approaches have been proposed in order to facilitate the design process, but have not been considered analytical because they cannot be applied in all cases [42].

B. MACHINE LEARNING, DATA MINING AND RECORD MATCHING

The task of matching entities names or even full records has been explored by a number of communities, including statistics, databases, and artificial intelligence. Each community has formulated the problem differently, and different techniques have been proposed for dealing with this [43]. Record matching, which identifies the records that represent the same real-world entity, is an important step for data integration. Most state-of-the-art record matching methods are supervised, which requires the user to provide training data [44].

First step of data integration is data mining. This process is defined as the automatic extraction of useful, often previously unknown information from large databases or data sets using advanced search techniques and algorithms to discover patterns and correlations in large pre-existing databases [45]. Related with this, data cleaning is a critical element for developing effective business intelligence applications. The inability to ensure data quality can negatively affect downstream data analysis and ultimately key business

decisions. A very important data cleaning operation is that of identifying records which match the same real-world entity. For example, owing to various errors in data and to differences in conventions of representing data, product names in sales records may not match exactly with records in master product catalogue tables [46]. Moreover, data quality has many dimensions one of which is accuracy. Accuracy is usually compromised by errors accidentally or intentionally introduced in a database system. These errors result in inconsistent, incomplete, or erroneous data elements [47].

ML classifiers are able to generalize behaviours based on unstructured or even inconsistent information from previous examples. ML classifiers can be applied to a wide range of highly complex and non-linear domains because of their variety of design alternatives. Nevertheless, this variety of design alternatives can sometimes be a disadvantage: the lack of guidelines can lead the designer to make arbitrary decisions or to use brute force techniques. Some new theoretical approaches have been proposed in order to facilitate the design process, but have not been considered analytical because they cannot be applied in all cases [42].

Data mining and ML have explored for the last 20 years in several areas looking for automatic tools for the analysis of large data sets [48], e.g. software development [49], biology [50], e-Commerce [51], ecology [52] or medicine [53].

Related with the context of this research, in [3] the authors propose a ML-based framework for detecting relationships between banking operation records, starting from not homogeneous information and taking into account large volume of data.

C. MACHINE LEARNING AND MEDICINE

Within the field of research of this manuscript, traditionally, statistical methods have been explored in clinical decision making by characterising patterns within data as mathematical equations; for example, linear regression suggests a 'line of best fit'. Through ML, artificial intelligence provides techniques that uncover complex associations which cannot easily be reduced to an equation. ML systems allow approaching complex problem solving just as a clinician might — by carefully weighing evidence to reach reasoned conclusions. However, unlike a single clinician, these systems can simultaneously observe and rapidly process an almost limitless number of inputs [54].

ML approaches have also been used in drug discovery for advanced application together with for data mining techniques, which require large and representative training-set compounds to learn robust decision rules [45].

Automatic monitoring of adverse drug reactions, defined as adverse patient outcomes caused by medications, is a challenging research problem that is currently receiving significant attention from the medical informatics community [2]. The rapid growth of electronically available health-related information, and the ability to process large volumes of them automatically, using Natural Language Processing (NLP) and ML algorithms, have opened new opportunities that could address some of the above-mentioned limitations.

In this context, different medicine-related areas have received attention from scientific community as pharmacovigilance, drug-drug interaction or chemo-informatics. The goal of pharmacovigilance is to detect, monitor, characterise and prevent adverse drug events with pharmaceutical products. In [55], the authors present a comprehensive structured review of recent advances in applying NLP to electronic health record narratives for pharmacovigilance. Moreover, drug-drug interaction extraction, as a typical relation extraction task in NLP has always attracted great attention. Most state-of-the-art drug-drug interaction extraction systems are based on ML learning approaches [56] with a large number of manually defined features [57] or deep neural networks [15], [16]. Regarding to the record linkage problem, there are multiple approaches based on machine learning. For example, some of them aim discover drugs [58] or relationships among medical records [59], [60]. But these kind of applications are domain-dependent [61], [62] and requires specific steps for concrete applications.

Furthermore, ML algorithms are generally developed in computer science or adjacent disciplines and find their way into chemical modelling [63]. This approach allows defining methods for building reliable, predictive models in chemoinformatics. The ML methods applied are broadly divided into clustering, classification and regression techniques [64].

III. PROPOSED SOLUTION

A. FRAMEWORK ELEMENTS

The proposed framework is based on three components or steps, depicted in Figure 1: (1) dataset generation, (2) train-test environment and (3) production environment. These three components are based on a previous successful architecture [3] that matches records in the financial environment. However, it is well known that the matching process is highly domain-dependent [61], [62]. For this reason, the steps of the architecture have been re-defined for this specific domain and its characteristics. Thus, the new framework considers domain-specific characteristics that cannot be represented in the previous framework. First of all, in the financial framework, only match and no-match cases were identified.

In the new framework, two types of match are identified: exact and non-exact. The ML algorithms of the new framework will be trained with the exact cases and the validation will be carried out with the non-exact cases in order to measure the ability of the framework of predicting non-explicit relationships among records. Secondly, in the previous financial framework, the no-match cases were generated by aleatory combinations of records. In the pharmaceutical framework, the no-match cases were provided by the company, and they were different from the records that have a direct match. For this reason, in the validation, aleatory combinations must be generated in order to simulate the real environment. That is another challenge because the models

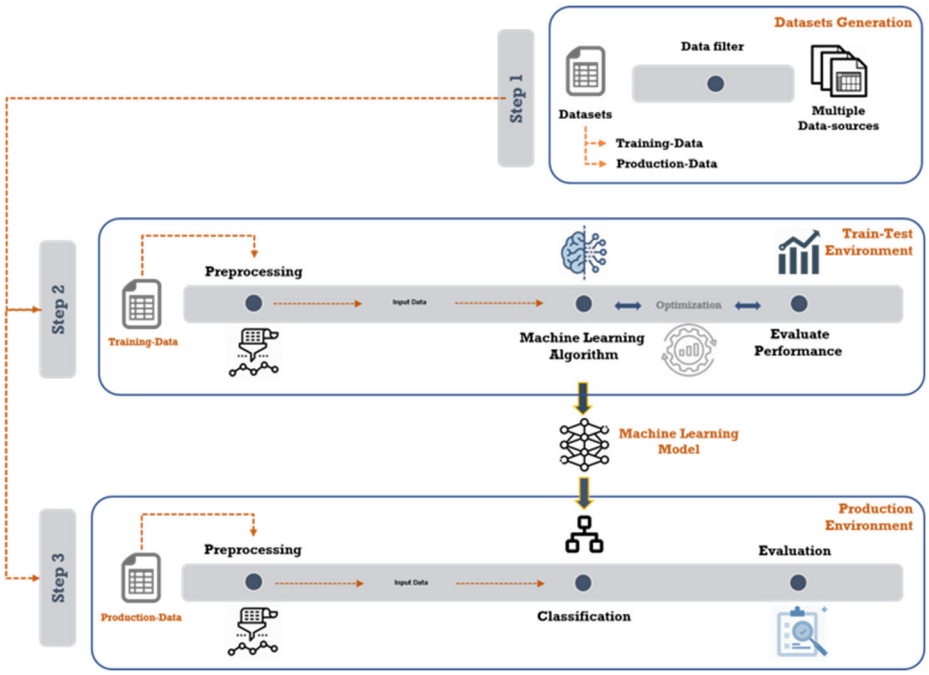

FIGURE 1. Proposed framework architecture.

will be trained with negative cases based on records that do not have a match. Regarding the data, the financial framework is based mainly on numerical attributes. In the pharmaceutical framework most of the attributes of the records are textual, and it implies a new way of pre-processing the data. Finally, the attribute selection is different in the proposed framework. In the financial framework, all the attributes of the tested databases were related to bank operations: thus, different combinations were tested in order to find the most relevant attributes. The pharmaceutical framework considers different databases that contain information from different points of about the product. This concept is different, because the financial framework looks for the same operation, and the new framework looks for a product (the record could represent different operations or points of view, but they are related to the same product).

The first step is in charge of generating the dataset. In this domain, the dataset is formed by records from databases that store pharmaceutical products. Several databases represent the product from different perspectives, depending on the company and the department that uses the data: for example, the logistic department manages different information than the financial or marketing departments. These different information requirements lead to different databases, sometimes maintained by different people in which changes are not coordinated. Therefore, the first step of the framework aims to merge the records to be compared. The records are merged in text lines, including a value one (1) if the records match, and 0 otherwise. Additionally, to the data provided, the framework can generate no-match cases by combining records from one source with others corresponding to other products in other databases.

With the merged records, the framework has a set of cases for training the ML algorithms in step 2. Several ML algorithms are evaluated, and the best one is chosen for the production environment. In step 3, the framework will receive combinations of records in order to determine whether or not they match. Each candidate record that represents a product from one database is merged with records from other databases. The candidate records from other databases can be selected through heuristics. However, it is also possible to generate brute force approaches by combining a candidate record from one database with all the records form the others. Once the candidate records are merged, the ML algorithm evaluates each combination and returns one if the records math and 0 otherwise.

1) STEP 1. DATASETS GENERATION

The dataset is generated by combining records from different sources. In the case of the pharmaceutical environment, companies have different databases that represent their products from different perspectives. As discussed in the introduction, that multiplicity of databases leads to duplication of the records.

The characteristics of these datasets are:

- Databases are independent from one to another. That means that the same product could be identified by different codes and, since each database has a concrete objective in the company, the content of each record may differ between two databases (i.e. one database may represent costs data and other logistic data).
- The records of the databases are usually curated by persons, which means that the names of the products or

other textual descriptions may contain typos, abbreviations or other kinds of errors.
- There are several ways for representing the names of drugs, as well as the active principles, and each database may use different ones (or even none of them).
- The number of attributes for each record is different in each database. Each database requires different attributes for representing the information. Furthermore, some attributes may contain null values that may affect the matching process.

Many of the problems related to the dataset can be managed by pre-processing the data. However, the proposed framework aims to avoid the pre-processing and explore the possibilities of match records with the minimum number of changes.

For each database to be processed, the framework will receive a file. The content of the files represents the information to be matched by the framework.

The input of the framework is a set of files in CSV format. This format is easy to generate and easy to import. The character encoding should be checked for each file in order to avoid errors in further steps. Records from one database are merged with the records of the rest of databases, generating CSV lines with all the fields. On the one hand, if the line corresponds to records of the same product, it is a ''Match case''.

Each merged line has a ''Res'' with the value ''1'' if it represents a ''match case''. ''No match cases'' are generated by merging non-related records from each database. In this way, the framework can generate a large number of no match cases.

Regarding the match cases, some records are well known and can be identified by simple rules. That represents the 70% of the cases, and they are named ''Exact matches''. Other cases do not follow a rule for the match. An expert can establish whether or not they match following heuristics or personal know-how. These cases are called ''Non-exact'' cases. Finally, some records do not match with other records. They are called ''No match''. For the training and testing phases, the ''No match'' cases are aleatorily generated by combining non-related records. One of the aims of the framework is to identify the maximum number of exact and non-exact cases.

Finally, the cases are distributed into two different sets: train and validation. The train set will be applied for generating the ML models. The validation set will be applied to simulate the production scenario..

2) STEP 2. TRAIN-TEST ENVIRONMENT

Figure 2 depicts the proposed train-test environment of the framework. Four phases form this process: (1) preprocessing, (2) attribute selection, (3) ML configuration and (4) ML train/test.

The first phase is the pre-processing of the data received. In the first phase, the data is loaded, and the datatypes of

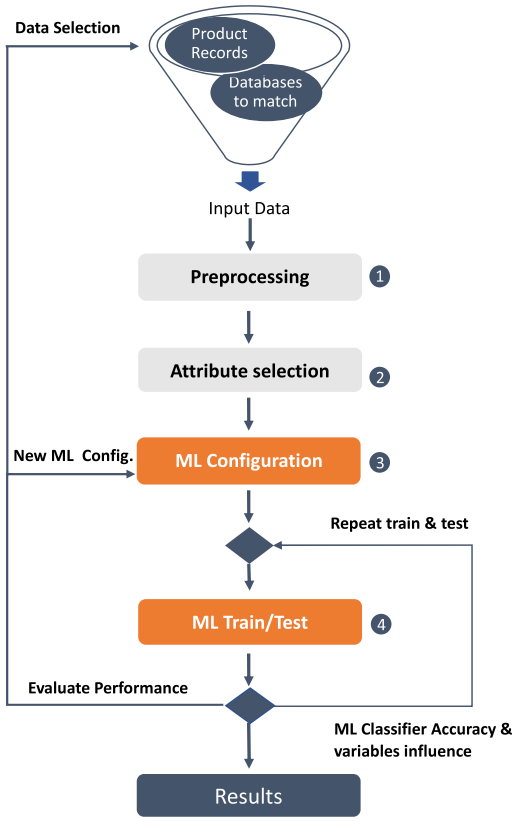

FIGURE 2. Step 2: Train & Test environment.

each column are analyzed. This second phase aims to detect fields that not contain meaningful information, or that could be problematic due to a large number of null values. Next, the existence of incomplete records is analyzed in order to fill empty values in the columns when necessary. Filling incomplete records should be verified by domain experts in order to avoid the introduction of values that change the semantic of the record. Since the information related to pharmaceutical products is mainly textual, all columns are processed as string values. Once all columns have been converted into string values, all the records are joined as sentences. These sentences are tokenized in order to create a vector that could be processed by the ML algorithms. However, when the data is processed by transforming textual values into numbers, it is necessary to fill the empty values or removing the records from the process.

After pre-processing the records received, the dataset is split into two different sub-datasets, train and test, in order to be set up the machine learning algorithms, and the attributes to be considered are decided. The train dataset has 90% of the records, while the test dataset has the other 10%.

In phase 3, ML algorithms are configured for the matching process. The machine learning algorithms considered in this phase are described in Table 1. Those algorithms have been defined and executed with the scikit-learn environment [65], [66].

TABLE 1. ML algorithms considered in the train-test environment.

Acronym	Algorithm	Description	Configuration
KNN	K-Nearest Neighbour	KNN method is a popular classification method in data mining and statistics because of its simple implementation and significant classification performance. KNN classifier is a type of instance-based learning or non-generalizing learning: it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the nearest neighbours of each point: a query point is assigned to the data class which has the most representatives within the nearest neighbours of the point [31], [32].	Algorithm= 'auto', leaf_size=30, metric= 'minkowski', n_neighbors= From 1 to 20, p=2, Weights: 'uniform'
RFO	Random Forest	RFO classifier is an ensemble classifier that produces multiple decision trees, using a randomly selected subset of training samples and variables. Over the last two decades the use of the RFO classifier has received increasing attention due to the excellent classification results obtained and the speed of processing. RFO is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size, but the samples are drawn with replacement in this case [37], [38].	Bootstrap=True, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=None, oob_score=False, random_state=None, verbose=0, warm_start=False
SVM – L	Support Vector Machine – Linear	SVM-L is a linear classifier based on the Support Vector Machine approach. The linear SVM classifier is implemented specifically for massive levels of data and features. The decision hyperplane that is calculated is used to classify samples into different categories. The selection of the error penalty factor, which expresses the tolerance to error, significantly affects the precision of the linear SVM [67].	C=1.0, class_weight=None, dual=True, fit_intercept=True, intercept_scaling=1, loss='squared_hinge', max_iter=100000, multi_class='ovr', penalty='l2', random_state=0, tol=1e-05, verbose=0
SVM	Support Vector Machine	SVMs perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. SVM is a classifier motivated by two concepts. First, transforming data into a high-dimensional space can transform complex problems (with complex decision surfaces) into simpler problems that can use linear discriminant functions. Second, SVMs are motivated by the concept of training and using only those inputs that are near the decision surface since they provide the most information about the classification [3].	C=1.0, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False
DNN	Deep Neural Network	Deep neural networks (DNNs), which employ deep architectures in NNs, can represent functions with higher complexity if the numbers of layers and units in a single layer are increased [9], [14]. In this case a model groups layers into an object with training and inference features. A Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor. A Dense layer is composed by nodes and each node is connected with all the nodes of the previous layer [68]. In the problem at hand, the input is the vector formed by the records to be compared and the output network is the result.	Model= Sequential, loss='binary_crossentropy', optimizer='adam', metrics='accuracy', Layer 1: Dense(500, input_dim=bolsa.shape[1], activation='relu') Layer 2: modelo.add(Dense(300, activation='relu')) Layer 3: modelo.add(Dense(200, activation='relu')) Layer 4: modelo.add(Dense(100, activation='relu')) Layer 5: modelo.add(Dense(50, activation='relu')) Layer 6: modelo.add(Dense(10, activation='relu')) Layer 7: modelo.add(Dense(1, activation='sigmoid'))
MLP	Multi-Layer Perceptron	MLP is one of the most widely implemented neural network topologies. For static pattern classification, the MLP with two hidden layers is a universal pattern classifier. MLPs are layered feedforward networks typically trained with static backpropagation. These networks have found their way into countless applications requiring static pattern classification. Their main advantage is that they are easy to use, and that they can approximate any input/output map. The key disadvantages are that they train slowly, and require lots of training data (typically three times more training samples than network weights) [3].	activation='relu', alpha=1e-05, batch_size='auto', beta_1=0.9, beta_2=0.999, early_stopping=True, epsilon=1e-08, hidden_layer_sizes=(70, 25), learning_rate='constant', learning_rate_init=0.002, max_iter=10000, momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5, random_state=None, shuffle=True, solver='adam', tol=0.0001, validation_fraction=0.1, verbose=False, warm_start=False
LoG	Logistic Regression	Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on. Multinomial logistic regression can model scenarios where there are more than two possible discrete outcomes. Logistic regression is a useful analysis method for classification problems, where you are trying to determine if a new sample fits best into a category. As aspects of cyber security are classification problems, such as attack detection, logistic regression is a useful analytic technique [69].	C=1.0, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, l1_ratio=None, max_iter=100, multi_class='warn', n_jobs=None, penalty='l2', random_state=None, solver='lbfgs', tol=0.0001, verbose=0, warm_start=False

Next, in phase 4, the ML algorithms are executed on the train and test datasets. First of all, each model is trained with the training dataset and validated with the test dataset. Next, the results are analyzed for each algorithm considering the behaviour and the accuracy of the model. As a result, the algorithms can be optimized in order to improve the results.

The combination of candidate records with possible records will be evaluated and classified as ''1'' or ''0''. The ''1'' means that both records match, meanwhile the value ''0'' indicates that both records represent different products.

Finally, based on the accuracy of the results obtained, the best algorithms are chosen to be executed in the production environment. The accuracy is measured as the number of records correctly classified with respect to the total amount of records. That is the accuracy of each model is measured according to the following formula:

$$Accuracy = \frac{truepositives + truenegatives}{\#R} \quad (1)$$

where true positives + true negatives is the number of records correctly classified and #R is the number of records evaluated.

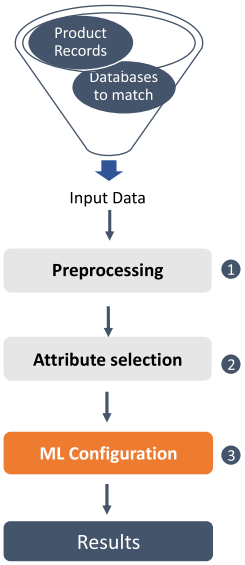

FIGURE 3. Step 3: Production environment.

3) STEP 3. PRODUCTION ENVIRONMENT

At the end of the train-test environment, the best ML algorithm is selected for the production scenario. Figure 3 depicts the process of the production environment with three phases: (1) pre-processing, (2) attribute selection and (3) ML configuration. The production environment represents the use of the framework in a real environment, in which new product records, different from the ones used for setting up the ML algorithms, should be matched. Although the process depicted in the figure looks similar to the previous one, this environment has specific characteristics.

In the production environment, the framework will receive many records to be matched against other databases. First of all, each record to be matched should be merged with all the candidate records from the rest of the databases. A candidate record is a record from one database that could match with the one being considered. At this point, in a good scenario, the candidate records could be filtered according to some expert criteria, but in many cases, brute-force combinations are necessary.

Records are merged, generating CSV lines, in the same way than in the train-test environment of the framework. In the production environment, the columns chosen from each record and the processes over the data received are the ones established in the train-test environment. Finally, all the columns are represented into string values and vectorized to be processed by the ML algorithm.

Once the input has been prepared, the ML algorithm selected in the train-test environment is executed over the data received. This algorithm was configured and trained previously so that at this point, it is only executed for predicting the matches.

For each class (1 and 0) the values of Prec ision, Recall and F1 are calculated according to the next formulae [70]:

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \quad (2)$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives} \quad (3)$$

$$F1 = 2\frac{Precision \cdot Recall}{Precision + Recall} \quad (4)$$

The precision value represents the percentage of the correct classified cases among the ones classified by the system. The recall value represents the percentage of the correct classified records provided by the system among the number of real correct ones. Finally, F1 represents the harmonic average of precision and recall. The best value for each measure is 1, and 0 is the worst one. Thus, results near to 1 for the F1 measure are the objective of the framework.

Finally, the evaluation of the results of this phase should be done by an expert able to decide whether or not the results are acceptable.

IV. EXPERIMENTATION AND RESULTS

A. EVALUATION SET-UP

The experimentation is based on the dataset provided by a pharmaceutical enterprise. It is formed by a set of 105563 records from a financial database and 28253 records from a logistic database. Both databases represent information about pharmaceutical products from different points of view (logistic and financial). The attributes contain information relative to names, codes and identifiers, apart from specific values relative to either the financial or logistic domain. Also, the records may contain typos and errors because some data have been input by people.

The company also provided the correspondence for the 28253 records from the logistic database with the records of the financial database. The correspondence database provides three types of correspondences:

- Exact match. An exact match means that one record from the logistic database has a match on the financial database, and there is a clear rule that can be applied for representing this correspondence. Following these rules, 18435 records from the logistic database have at least

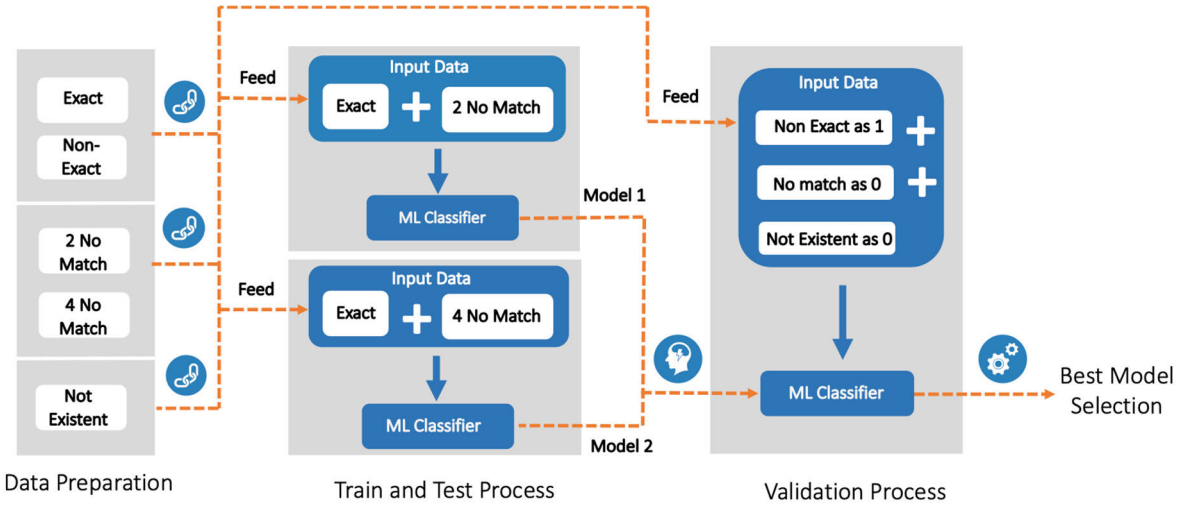

FIGURE 4. Experiments design.

one record in the financial database. The overall number of exact matches are 21867.

- Non-exact match. A non-exact match means that the record from the logistic database could match with one or several records on the financial database, but there is not a clear rule that represents this fact. That means that there are matches based on heuristics defined by the experts of the company. In this case, there are 3556 records from the logistic database with at least one possible connection with a record of the financial database. The overall number of Non-exact matches are 10691 because usually, a record from the logistic database has more than one non-exact match with the financial database (concretely an average of 3 matches).
- No-match means that one record form the logistic database does not have a correspondence with a record in the financial database. Some logistic records do not match with any record of the financial database. These records are relevant because they can be used for generating aleatory no-match cases by combining with any record of the financial database. There are 5766 records from the logistic database labelled as no-match

Figure 4 depicts the configuration of the tests performed. First of all, several datasets are created in order to train the models. Next, a first dataset formed by exact matches and no-matches is formed. One of the objectives is to determine whether a model trained with "Exact" cases can predict non-exact matches, so this first dataset is used for training the model. Two training datasets has been defined in order to determine the best train configuration: the first dataset contains all the exact matches and 2 records for each no-match case; the second one contains all the exact matches and 4 records for each no-match. The model should be able to determine both matches and no-matches, so the training datasets must contain an adequate proportion of cases. Once the models have been trained and tested, they are validated in the production environment. In the production environment, the input of the model is a number of combinations of records from the financial database with records from the logistic database, and the model should be able to determine whether or not each combination matches. As a result, the ability of each model for classifying the input as a match or no-match is evaluated. Finally, the model with the best performance is selected.

The following sections describe in detail each phase.

B. DATA PREPARATION

The first step is collecting the dataset. If a requisite expert is available, then s/he could suggest which fields (attributes, features) are the most informative. If not, then the most straightforward method is that of "brute-force," which means measuring everything available in the hope that the right (informative, relevant) features can be isolated. However, a dataset collected by the "brute-force" method is not directly suitable for induction. It contains, in most cases, noise and missing feature values and therefore requires significant preprocessing [71], [72].

Next, three different sets of data are generated:

1) Exact cases training set. Combinations of exact cases are generated by merging the tuples of the logistic database records with their corresponding exact matches in the financial database. As mentioned, 21867 cases are generated. Each exact case is labelled with the value "1". The obtained combinations are of the form:

$$< L_1, F_1, 1 >$$

where $L_1$is a record from the logistic database, F_1 is its corresponding record on the financial database and 1 means that the records match.

2) Non exact cases dataset. Combinations of non-exact cases are generated, merging the tuples of the logistic database with their corresponding non-exact matches in the financial database. They are also labelled as 1, because the aim of the framework is to identify these

cases by training with the exact ones. For this reason, they should be labelled as 1. There are 11691 non exact cases. The obtained combinations are of the same form than the exact cases:

$$< L_1, FN_1, 1 >$$

where L_1 is a record from the logistic database, FN_1 is its corresponding record on the financial database (non-exact match) and 1 means that the records match.

3) No-match cases dataset. In this case, combinations of the records labelled as no-match in the logistic database with aleatory records of the financial database are generated. For each no-match record, four combinations are generated for obtaining a no-match dataset with 23064 records. This number is close to the number of exact cases, in. order to have a balanced training dataset. The obtained combinations are of the same form than the exact and non-exact cases:

$$< LN_1, NF_1, 1 >$$

where LN_1 is a record from the logistic database labelled as no-match, NF_1 is a record from the financial database that is not related to LN_1, and 0 means that the records do not match.

4) Non-existent cases dataset. With the aim of simulating the production scenario, a fourth type of correspondence was defined. Let's suppose a record L1 from the logistic database, that matches with a record F1 from the financial database, for example:

$$< L_1, F_1, 1 >$$

Now let us combine the record L_1 with random records from the financial database, except F_1:

$$< L_1, NF_1, 0 >$$
$$< L_1, NF_3, 0 >$$
$$\ldots.$$
$$< L_1, NF_N, 0 >$$

where 0 means the records don't match.
The result is a set of n combinations of the record L1 with n records of the financial database that simulates the production environment. In that real scenario, the correspondence of the L1 record with the financial database is unknown. Then, the only way to find the matches is by combining the L records with all the records of the financial database and compare them.
Therefore, this last dataset contains combinations of the type:

$$< L_1, F_1, 1 >$$
$$< L_1, NF_1, 0 >$$
$$< L_1, NF_3, 0 >$$
$$\ldots$$
$$< L_1, NF_N, 0 >$$

The logistic database has 23 attributes. The financial database has 18 attributes, 6 of them only relative to costs: since the logistic database does not include cost information, these attributes were not considered. Therefore, the merged records have 36 attributes: 23 logistics + 12 financial + 1 extra attribute 'Res' with the value one if the record represents a match and 0 if the record represents a no-match.

The attributes were analysed in order to determine whether or not they can be translated into numerical values. Only five attributes are numerical. The other 30 attributes are considered as categorical data.

From the point of view of the data, two kinds of tests were executed. First of all, models based on categorical data were defined based on vectors. Next, the same models were trained, translating the categorical data into numerical data. In this process, five columns cannot be translated. In this transformation, many records have fields with empty or null values. As part of the data processing, records with empty attributes can be removed from the dataset or filled with a given value. Firstly, the empty data were removed from the datasets. That decision was made because the inclusion of values can affect the meaning of the records.

C. MACHINE LEARNING TECHNIQUES

This section shows the evaluation process performed in the machine learning stage of the framework for train-test and production environments.

1) TRAIN-TEST ENVIRONMENT

In the train-test environment, the machine learning models are trained from the Exact cases training set. Figure 5 describes the process of training and testing the ML models. Since the models must learn both matches and no matches, that training set is combined with cases from the no-match training dataset. Two different alternatives were evaluated: two combinations for each no-match record (2 no-match) and four combinations for each no-match record (4 no-match). The number of samples is relevant in the learning process; for this reason, these two possibilities were evaluated.

For each alternative, datasets were prepared according to the type of data of the fields. In this way each alternative was tested considering the data as categorical, that is, considering all attributes as textual, and, on the other hand, considering the data as numerical (translating the content of the attributes into numerical values).

For each experiment configuration, all machine learning techniques were tested, measuring its accuracy with the non-exact and no-match datasets. As mentioned before, the train and test dataset are composed by exact cases and no-match cases, the non-exact dataset is formed only by matches that do not follow the conventional rules of matching. The no-match dataset is formed only by no-match records.

The models were trained based on the exact cases with combinations of no-match records. Then the models were evaluated using the non-exact cases and another set of

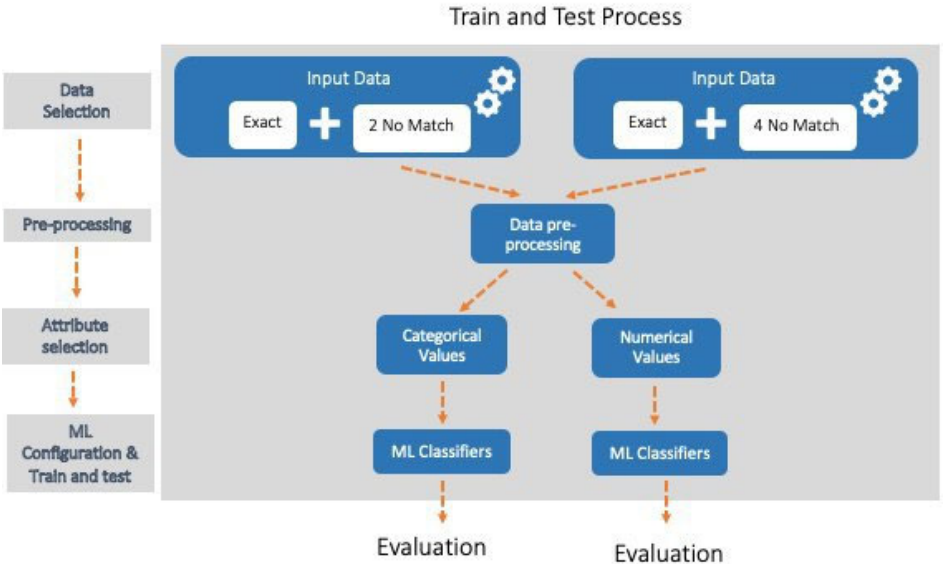

FIGURE 5. Detail of the train and test processes in the experiments.

TABLE 2. Results for 2 no-match, Numerical (30 cols.), 10 runs.

Alg.	Accuracy				
	Train	Test	Val.	NE	NM
KNN	0.99	0.98	0.91	0.99	0.78
RFO	0.99	0.99	0.94	0.98	0.88
SVM-L	0.73	0.73	0.65	0.24	0.76
SVM	0.91	0.91	0.91	0.99	0.76
DNN	0.81	0.71	0.41	0.63	0.48
MLP	0.92	0.92	0.57	0.29	0.91
LoG	0.92	0.92	0.85	0.46	0.97

Alg.: Algorithm. Val.: Validation. NE: non-exact, NM: no-match

TABLE 3. Results for 2 no-match. Categorical (35 cols.). 10 runs.

Alg.	Accuracy				
	Train	Test	Val.	NE	NM
KNN	0.99	0.96	0.80	0.89	0.67
RFO	0.99	0.97	0.67	0.99	0.71
SVM-L	0.99	0.96	0.66	0.97	0.99
SVM	0.70	0.70	0.62	0.99	0.46
DNN	0.98	0.61	0.49	0.99	0.44
MLP	0.99	0.96	0.55	0.16	0.99
LoG	0.88	0.88	0.51	0.22	0.91

Val.: Validation. NE: non-exact. NM: no-match.

TABLE 4. Results for 4 no-match, Numerical (30 cols.), 10 runs.

Alg.	Accuracy				
	Train	Test	Val.	NE	NM
KNN	0.99	0.98	0.89	0.99	0.83
RFO	0.99	0.99	0.93	0.96	0.91
SVM-L	0.76	0.76	0.59	0.16	0.88
SVM	0.94	0.94	0.92	0.97	0.87
DNN	0.69	0.64	0.17	0.89	0.62
MLP	0.94	0.94	0.67	0.43	0.82
LoG	0.92	0.91	0.80	0.24	0.99

Val.: Validation. NE: non-exact, NM: no-match.

TABLE 5. Results for 4 no-match, Categorical (35 cols.), 10 runs.

Alg.	Accuracy				
	Train	Test	Val.	NE	NM
KNN	0.99	0.98	0.83	0.77	0.88
RFO	0.99	0.98	0.66	0.99	0.52
SVM-L	0.99	0.98	0.57	0.97	0.99
SVM	0.76	0.76	0.58	0.79	0.84
DNN	0.94	0.56	0.27	0.99	0.94
MLP	0.99	0.98	0.64	0.12	0.99
LoG	0.87	0.86	0.56	0.35	0.99

Val.: Validation. NE: non-exact, NM: no-match.

no-match cases. In this way, the ability of the models for recognizing the non-exact and no-match cases can be measured.

Table 2 and Table 3 shows the results for the models trained with the exact cases 2 no-match dataset. Table 2 includes the results obtained using numerical data. In this case RFO obtains the best results: other classifiers obtain better results testing the non-exact dataset or the no-match dataset, but RFO obtains the best combination of both results. Table 3 shows the results based on categorical data. In this table, SVM-L obtains the best results. It is noticeable that the validation results are below 80% of accuracy.

Table 4 and Table 5 show the results for the 4 no-match approach. Table 4 shows the results obtained with the numerical approach. In this case, the best combination is achieved again by the RFO classifier. Finally, Table 5 presents the results for the categorical approach. In this case, the SVM-L approach obtains the best combination in the prediction of NE and NM but has the worst Validation accuracy.

2) PRODUCTION ENVIRONMENT

In the production environment, for each record of the logistic database with non-exact matches in the financial database, two new random combinations are generated with records of the financial database (different than the ones that match). That means two differences with respect to the train-test scenario. The first difference is that the matches do not follow

TABLE 6. Results for 4 no-match, Categorical (35 cols.), filling empty values, 10 runs.

Alg.	Accuracy			
	Train	Test	Val.	Prod.
KNN	0.98	0.99	0.96	0.46
RFO	0.99	0.93	0.93	0.74
SVM-L	0.95	0.98	0.98	0.47
SVM	0.99	0.99	0.99	0.85
DNN	0.80	0.80	0.53	0.58
MLP	0.99	0.99	0.99	0.50
LoG	0.99	0.97	0.97	0.47

Val.: Validation. Prod. Production dataset.

TABLE 7. Precision and Recall for 4 no-match, Categorical (35 cols.), filling empty values, 10 runs.

Alg.	Accuracy			
	RES	Avg. Precision	Avg. Recall.	Avg. F1
RFO	0	0.72	0.86	0.79
	1	0.77	0.58	0.66
SVM	0	0.90	0.88	0.89
	1	0.76	0.79	0.77

RES: classification result. Avg. Precision: average precision of the experiments. Avg. Recall: average recall of the experiments. Avg. F1: average F1 measure of the experiments.

the same rules than the training dataset (now they are non-exact matches). The second difference is that the no-match cases are obtained combining the candidate record with random records from the financial database (see data preparation section). In this way, a simulation of the production environment is generated (Production dataset). In the production environment, each candidate record from the logistic database must be combined with different records from the financial database in order to determinate whether they match or not.

The trained models in the train-test environment evaluate the production dataset. As a result, the accuracy of the system in production is determined. The first experiments showed that the results with the production dataset were worse than the obtained in the train-test ones. Since the categorical data may contain empty values not considered in the numerical approach, a new pre-processing approach was included, filling the empty values in the categorial approach. This experiment was limited to the 4 no-match approach because the results were better in the no-match identification than the 2 no-match approach (notice that the production environment will deal with a huge number of no-match cases). Table 6 shows the results obtained in this experiment. The Prod. column represents the results of the trained ML models for the production dataset. As can be seen, with this new approach the models for Train, Test and Val achieve reasonable results. The models improved their accuracy and SVM and RFO achieved promising results in the production dataset.

Table 7 shows the precision and recall values for the selected ML algorithms. As can be seen, the SVM algorithm is able to identify both 0 (no-match) and 1 (match). In the case of the matches, SVM is able to identify the 79% of the matches that does not follow the pattern of the exact ones. Furthermore, SVM also identify around the 88% of the no-matches. This is a relevant fact, because in the production environment the number of no-matches will be extremely high with respect to the number of matches. In the case of RFO the accuracy is tricky because the model is identifying around the 86% of the no-matches. As mentioned, the number of no-matches is larger than the number of matches, so the accuracy of the model is high. However, the precision of RFO with the no-matches is 72% which means that the model is classifying as no-match many of the records that actually are matches. Taking a look to the recall of RFO for matches, the fact that the model is missing many matches can be confirmed: only a 58% of the matches are detected.

D. STATISTICAL VALIDATION

In any empirical scientific work, when repeating an experiment in conditions which are indistinguishable to the researcher, it is very common for the results to show some variability; this is known as experimental error. Therefore, in any scientific experimental study it is crucial to compare and evaluate the characteristics of the different sets of samples and the results obtained. In the field of machine learning, the research, development and simulation carried out by the researchers have included the use of different statistical methods for the evaluation of the results [73]. Following this trend, this research assesses and compares the different experiments proposed by statistical analysis based on the estimator t-test and its variants.

Hence, in this section, and following the ideas exposed in [74], a statistical validation has been performed for the determination of which is the better choice among comparable models (with similar results). To facilitate this task, different analyses have been included to evaluate and compare the generalization ability of neural models designed from the statistical point of view.

For this analysis in the production scenario the accuracy has been used. Figure 6 includes the scatter and box plots associated with the results for the production scenario. In this case, outliers can be found in MLP SVM classifier. The chart also includes a notch to the median, which indicates the approximate width of the confidence interval of 95%. In the case that two notches for any pair of medians overlap, there is no statistically significant difference between the medians at the 95% confidence level.

Figure 7 includes the residual plot and analysis of means (ANOM) plot. The first plot shows the residuals versus each classifier. The residuals are equal to the observed values of correct classifications percentage minus the mean percentage for the group from which they come. This plot checks that the variability within each classifier is approximately the same (except for DNN because there are some important residuals). This second plot shows the mean of each of the five samples. Also shown is the grand mean and the 95% decision limits. Analysis of means plot include the Upper Decision Limit (UDL), Centre Limit (CL) and Lower Decision Limit (LDL) The samples which fall outside the decision

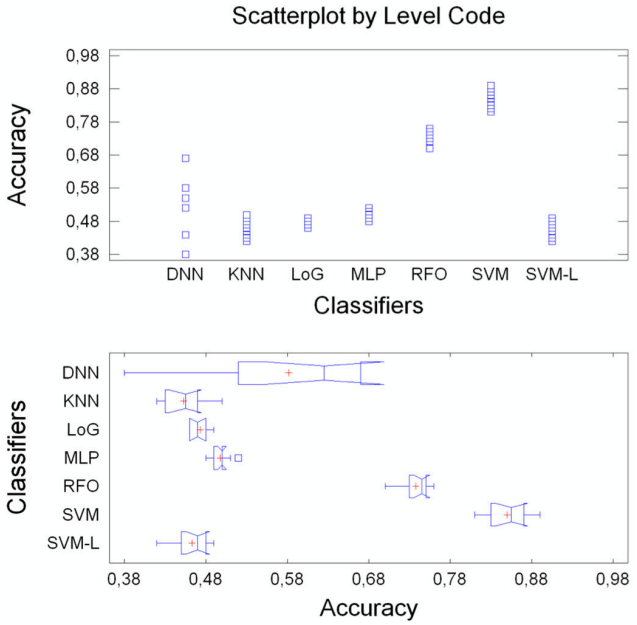

FIGURE 6. Scatter and box plots using accuracy of the classifiers (production scenario).

TABLE 8. Variance check (approaches A and B).

Contrast	Value	p-value
Cochran's C test	0.82130	0.0
Bartlett's test	3.59537	0.0
Levene's test	9,19335	3,1701E-7

limits, all classifiers except DNN, are significantly different from the grand mean.

Next, to verify that the population variances are equal a series of widespread statistical tests of equality have been included: Bartlett contrast, Cochran C contrast and the Levene test. The three statistics displayed in Table 8 test the null hypothesis that the standard deviations of the results within each of the seven levels of classifiers are the same. Since the smaller of the p-values is lower than or equal to 0.05, there is statistically significant difference amongst the standard deviations at the 95.0% confidence level. So, the assumptions for applying the ANOVA are not accomplished and Kruskal-Wallis test will be performed.

TABLE 9. Kruskal Wallis test for production scenario.

Classifier	Average Rank
DNN	38.0
KNN	14.05
LoG	21,6
MLP	35.65
RFO	55.5
SVM	65.5
SVM-L	18.2
Statistical = 54.7519	P-value= 5.20207E-10

In Table 9 the results of the Kruskal Wallis test are shown to test if a group of data comes from the same population. In this case, the null hypothesis of equality of the medians is checked for the percentage of success in each of the seven alternatives. Since the p-value is less than 0.05, there is great statistical evidence against the model (the results obtained by all the techniques are similar). To determine which medians are significantly different from each other, in the box and whisker plot of Figure 1 the width of the notches indicates the approximate confidence interval of 95.0 As is depicted in Table 9, the SVM and RFO classifiers present a homogeneous behaviour and the distributions of the results are significantly different from all the rest. Moreover, the average accuracy of SVM is higher than RFO. Therefore, it can be concluded that considering all the results) obtained in the experiments of the production scenario SVM and RFO classifiers obtain a consistence performance from a statistical point of view.

TABLE 10. Comparison of F1 results for other approaches.

Work Best F1 measure	Work Best F1 measure
Gonzalez-Carrasco et al. (2019) [3]	0.998
Jurek et al. (2017) [75]	0.96
Kim & Giles (2016) [76]	0.9744
Proposed SVM	0.85

Table 10 shows a comparison between the best F1 result obtained by the proposed framework and the best F1 result of other approaches. Jurek *et al.* [75] applies different classifiers over four textual datasets not related to the pharmaceutical domain, obtaining a F1 measure of 0,96. The work of Kim and Giles [76] is based on a financial dataset and obtain a F1 measure of 0,9774 with Random Forest in the best scenario. The authors of the proposed framework presented a previous work based on the financial domain [3], reporting a F1 measure of 0,998. In this case the dataset of the previous work was based on numerical data of banking operations.

Despite the differences, Hand and Kristen (2018) stated that the F1 measure is relative to each system and are not directly comparable because it depends on the relative importance given to precision and recall depending upon the number of predicted matches and the techniques applied [62]. In this sense, the proposed work is based on a set of specific premises that make it different from other research works. Based on this premises the results can be improved in future, but we consider it is a promising start.

E. ANALYSIS AND DISCUSSION

As shown in the train-test environment, the tested models obtain promising results. In general, results for the 4 no-match dataset are better than the 2 no-match. All models obtain good results in the train and test scenarios, and these results decrease in the validation. That anticipates the problems found in the production environment when new cases that do not follow the usual rules must be evaluated. However, training with the 4 no-match approach, KNN, RFO and SVM obtain promising results with the non-exact dataset, and all the algorithms perform well in general with the no-match dataset.

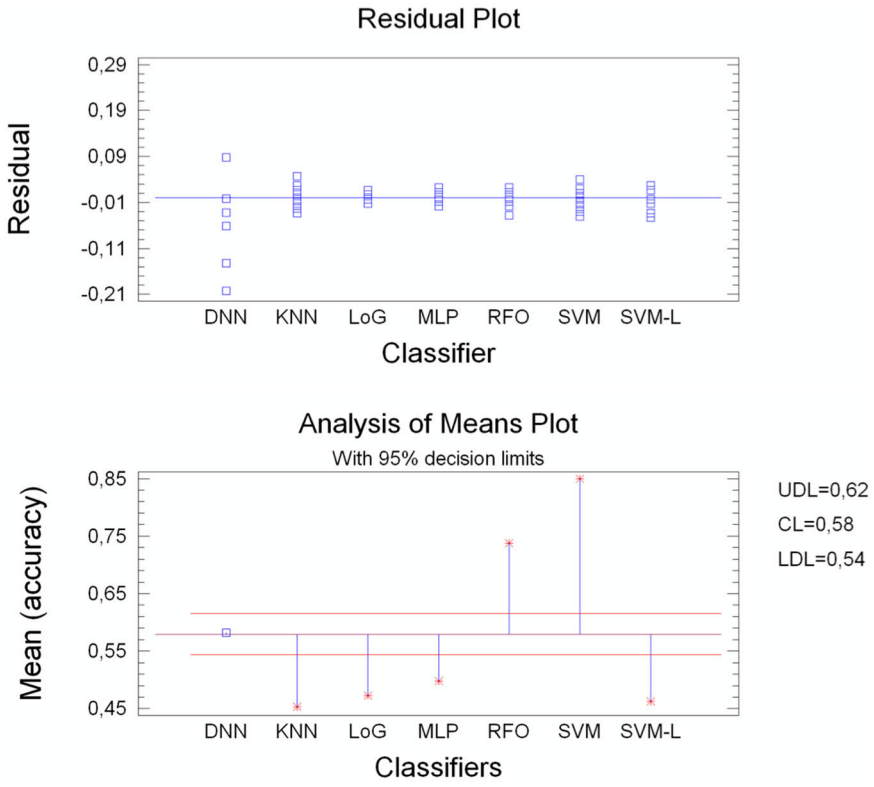

FIGURE 7. Residual and analysis of means plot (production scenario).

Although time measures have not been considered, the runtime performance of the classification with numerical data is better than the one with categorical data.

It is remarkable the considerable difference between the results of the train-test environment and the production environment when the records with empty values are processed. Despite the results of the validation with non-exact and no-match show reasonable results, when the models have to predict the result among multiple combinations of the same record from the logistic database with random records from the financial database, they tend to classify all of them as 1 or 0. It is noticeable that the models are trained only with exact data and no-match cases. For this reason, when all records were 1 in the non-exact dataset or 0 in the no-match dataset, the models performed well in the training scenario. That means that the models had problems for generalising the results.

However, when the empty and null values are filled with default values, the performance of the algorithms varies, and two classifiers obtain reasonable results in the categorical approach: Random Forest achieves an accuracy of 0.75 in the production environment with categorical data, and SVM achieves 0.85 in the same conditions. Analysing the precision and recall values for these models both of them identify the no-match cases better than the match cases, however SVM obtains better performance predicting the matches. That means that including more information in the records, the models can improve their ability to recognise the new cases.

Of course, this value is far from the 1 obtained in the train-test environment. However, considering that the models are trained with exact cases and validated with non-exact cases and that there is not a direct rule to infer the non-exact cases if the model can identify around the 84% of these values then the results improves the actual situation in the company.

V. CONCLUSION AND FUTURE LINES

The main motivation of this research was the necessity of great pharmaceutical manufacturers to analyse a huge number of products related to their worldwide activities, considering that the same product can be registered several times by different systems using different attributes. The task of finding the records and match the products cannot be done by a human in a reasonable way, because the number of records to be matched is extremely high. Humans can provide some mapping rules that cover up to the 70% of cases, but the other 30% has not clear rules for its identification. For this reason, this article proposes a ML approach trained with this 70% of cases, based on the previous experience of the authors in bank operations [3].

The proposed framework is structured into two different environments. The train-test environment set up different ML models in order to find the most suitable for the problem at hand. In the train-test environment, the ML models are trained with cases that can be determined using the rules of human experts, and they are validated against the cases that follow not clear rules. Then the best models are selected

to be tested in the production environment. The production environment simulates the case in which the pharmaceutical enterprise has a set of records to be evaluated. In this case, the only way of proceeding is to combine each candidate record from one database with all the possible alternatives from the other databases. The models trained in the train-test environment are executed in the production environment to determine whether or not the results are suitable.

When examining the results, the train-test environment shows promising results in models trained with numerical data and a balanced number of exact and no-match cases. In this environment, KNN obtains better results. However, when the models were tested in the production environment, the results were different, and the accuracy of the models was not adequate. After analysing the results, the pre-processing of the records was adapted, completing the empty values in the categorical approach. In this case, the results of the production environment improved: SVM obtains an accuracy of 84% with an equilibrium between the detection of matches and no-matches. Meanwhile, RFO achieves an accuracy around 74%, but the detection of match cases is worse than the SVM. Considering that the models are tested with records that cannot be identified by human heuristics and only 70% of records can be identified by exact rules, the results obtained by the framework with the SVM classifier are acceptable.

Thus, the proposed framework can determine whether or not two records represent the same product when the matching cannot be determined through direct rules, with a reasonable degree of accuracy.

Finally, future research will test the framework with other pre-processing approaches and different configurations to improve accuracy. The inclusion of Heterogenous Distance Metrics such as Heterogeneous Value Difference Metric (HVDM) will be considered. Also, new databases should be included in order to test the ability of the framework for generalising the matching process. This approach will be twofold. On the one hand, the framework will be tested by matching the logistic database with other databases in order to test the ability of the framework for detecting matches with different information. On the other hand, the framework will be tested by matching different databases on the domain in order to test the ability of the framework for detecting the products from different sources.

REFERENCES

[1] A. Doan, A. Halevy, and Z. Ives, *Principles of Data Integration*. Waltham, MA, USA: Morgan Kaufmann, 2012.
[2] A. Sarker, R. Ginn, A. Nikfarjam, K. O'Connor, K. Smith, S. Jayaraman, T. Upadhaya, and G. Gonzalez, "Utilizing social media data for pharmacovigilance: A review," *J. Biomed. Informat.*, vol. 54, pp. 202–212, Apr. 2015, doi: 10.1016/j.jbi.2015.02.004.
[3] I. González-Carrasco, J. L. Jiménez-Márquez, J. L. López-Cuadrado, and B. Ruiz-Mezcua, "Automatic detection of relationships between banking operations using machine learning," *Inf. Sci.*, vol. 485, pp. 319–346, Jun. 2019, doi: 10.1016/j.ins.2019.02.030.
[4] J. Li, C. Wu, and H. Wu, "Wavelet neural network process control technology in the application of aluminum electolysis," in *Electrical Power Systems and Computers*. Berlin, Germany: Springer, 2011, pp. 937–941.
[5] X. Peng, H. Zhe, G. Guifang, X. Gang, C. Binggang, and L. Zengliang, "Driving and control of torque for direct-wheel-driven electric vehicle with motors in serial," *Expert Syst. Appl.*, vol. 38, no. 1, pp. 80–86, Jan. 2011, doi: 10.1016/j.eswa.2010.06.017.
[6] I. Gómez and M. P. Martín, "Prototyping an artificial neural network for burned area mapping on a regional scale in mediterranean areas using MODIS images," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 13, no. 5, pp. 741–752, Oct. 2011.
[7] M. S. Bascil and F. Temurtas, "A study on hepatitis disease diagnosis using multilayer neural network with Levenberg Marquardt training algorithm," *J. Med. Syst.*, vol. 35, no. 3, pp. 433–436, Jun. 2011.
[8] P. Tenti, "Forecasting foreign exchange rates using recurrent neural networks," in *Artificial Intelligence Applications on Wall Street*. Evanston, IL, USA: Routledge, 2017, pp. 567–580.
[9] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.
[10] P. Naraei, A. Abhari, and A. Sadeghian, "Application of multilayer perceptron neural networks and support vector machines in classification of healthcare data," in *Proc. Future Technol. Conf. (FTC)*, Dec. 2016, pp. 848–852, doi: 10.1109/FTC.2016.7821702.
[11] D. Hush and B. G. Home, "Progress in supervised neural networks," *IEEE Signal Process. Mag.*, vol. 10, no. 1, pp. 8–39, Jan. 1993. [Online]. Available: https://ieeexplore.ieee.org
[12] J. S. Sonawane and D. R. Patil, "Prediction of heart disease using multilayer perceptron neural network," in *Proc. Int. Conf. Inf. Commun. Embedded Syst. (ICICES)*, Feb. 2014, pp. 1–6, doi: 10.1109/ICICES.2014.7033860.
[13] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, Eds., *Deep Learning*, 4th ed. San Mateo, CA, USA: Morgan Kaufmann, 2017, ch. 10, pp. 417–466.
[14] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning—A new frontier in artificial intelligence research [research frontier]," *IEEE Comput. Intell. Mag.*, vol. 5, no. 4, pp. 13–18, Nov. 2010, doi: 10.1109/MCI.2010.938364.
[15] V. Suárez-Paniagua, R. M. R. Zavala, I. Segura-Bedmar, and P. Martínez, "A two-stage deep learning approach for extracting entities and relationships from medical texts," *J. Biomed. Informat.*, vol. 99, Nov. 2019, Art. no. 103285.
[16] V. Suárez-Paniagua, I. Segura-Bedmar, and P. Martínez, "Exploring convolutional neural networks for drug–drug interaction extraction," *Database*, vol. 2017, pp. 1–15, Jan. 2017.
[17] T. Sakellaropoulos, K. Vougas, S. Narang, F. Koinis, A. Kotsinas, A. Polyzos, T. J. Moss, S. Piha-Paul, H. Zhou, E. Kardala, and E. Damianidou, "A deep learning framework for predicting response to therapy in cancer," *Cell Rep.*, vol. 29, no. 11, pp. 3367–3373.e4, 2019, doi: 10.1016/j.celrep.2019.11.017.
[18] O. Chang, I. Naranjo, C. Green, D. Criollo, J. Guerron, and G. Mosquera. (2017). *A Deep Learning Algorithm to Forecast Sales of Pharmaceutical Products*. Accessed: Sep. 18, 2020. [Online]. Available: http://www.academia.edu/download/54874858/A_Deep_Learning_Algorithm_to_Forecast_Sales_of_Pharmaceutical_Products_A_.pdf
[19] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and Naive Bayes," in *Proc. 14th Conf. Adv. Neural Inf. Process. Syst.*, vol. 2. Cambridge, MA, USA: MIT Press, 2002, pp. 841–848.
[20] P. Tsangaratos and I. Ilia, "Comparison of a logistic regression and Naïve Bayes classifier in landslide susceptibility assessments: The influence of models complexity and training dataset size," *Catena*, vol. 145, pp. 164–179, Oct. 2016, doi: 10.1016/j.catena.2016.06.004.
[21] Y. Xu, L. Wang, and P. Zhong, "A rough margin-based ν-twin support vector machine," *Neural Comput. Appl.*, vol. 21, no. 6, pp. 1–11, 2011.
[22] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 1995.
[23] S. Yin and J. Yin, "Tuning kernel parameters for SVM based on expected square distance ratio," *Inf. Sci.*, vols. 370–371, pp. 92–102, Nov. 2016, doi: 10.1016/j.ins.2016.07.047.
[24] X.-Y. Wang, H.-Y. Yang, Y. Zhang, and Z.-K. Fu, "Image denoising using SVM classification in nonsubsampled contourlet transform domain," *Inf. Sci.*, vol. 246, pp. 155–176, Oct. 2013, doi: 10.1016/j.ins.2013.05.028.
[25] Y. Liu, J.-W. Bi, and Z.-P. Fan, "A method for multi-class sentiment classification based on an improved one-vs-one (OVO) strategy and the support vector machine (SVM) algorithm," *Inf. Sci.*, vols. 394–395, pp. 38–52, Jul. 2017, doi: 10.1016/j.ins.2017.02.016.

[26] I. Babaoğlu, O. Findik, and M. Bayrak, "Effects of principle component analysis on assessment of coronary artery diseases using support vector machine," *Expert Syst. Appl.*, vol. 37, no. 3, pp. 2182–2185, Mar. 2010, doi: 10.1016/j.eswa.2009.07.055.

[27] B. Desmet and V. Hoste, "Online suicide prevention through optimised text classification," *Inf. Sci.*, vols. 439–440, pp. 61–78, May 2018, doi: 10.1016/j.ins.2018.02.014.

[28] D. Hardin, I. Tsamardinos, and C. F. Aliferis, "A theoretical characterization of linear SVM-based feature selection," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, pp. 377–384, doi: 10.1145/1015330.1015421.

[29] X. Cao, C. Wu, P. Yan, and X. Li, "Linear SVM classification using boosting HOG features for vehicle detection in low-altitude airborne videos," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 2421–2424, doi: 10.1109/ICIP.2011.6116132.

[30] Y.-W. Chang and C. J. Lin, "Feature ranking using linear SVM," *Featur. Rank. Using Linear SVM*, vol. 3, pp. 53–64, Dec. 2008.

[31] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, "Learning k for kNN classification," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 3, pp. 1–19, Apr. 2017, doi: 10.1145/2990508.

[32] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1774–1785, May 2018, doi: 10.1109/TNNLS.2017.2673241.

[33] Q. Liu and C. Liu, "A novel locally linear KNN method with applications to visual recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 9, pp. 2010–2021, Sep. 2017.

[34] S. Jiang, G. Pang, M. Wu, and L. Kuang, "An improved K-nearest-neighbor algorithm for text categorization," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 1503–1509, Jan. 2012, doi: 10.1016/j.eswa.2011.08.040.

[35] H.-L. Chen, C.-C. Huang, X.-G. Yu, X. Xu, X. Sun, G. Wang, and S.-J. Wang, "An efficient diagnosis system for detection of Parkinson's disease using fuzzy k-nearest neighbor approach," *Expert Syst. Appl.*, vol. 40, no. 1, pp. 263–271, Jan. 2013, doi: 10.1016/j.eswa.2012.07.014.

[36] M. Belgiu and L. Dragu, "Random forest in remote sensing: A review of applications and future directions," *ISPRS J. Photogramm. Remote Sens.*, vol. 114, pp. 24–31, Apr. 2016, doi: 10.1016/j.isprsjprs.2016.01.011.

[37] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random forests," in *Ensemble Machine Learning: Methods and Applications*, vol. 45. Boston, MA, USA: Springer, 2012, pp. 157–175.

[38] P. Probst and A. L. Boulesteix, "To tune or not to tune the number of trees in random forest," *J. Mach. Learn. Res.*, vol. 18, pp. 1–8, Apr. 2018.

[39] R. A. Igawa, S. Barbon, K. C. S. Paulo, G. S. Kido, R. C. Guido, M. L. Proenca, and I. N. D. Silva, "Account classification in online social networks with LBCA and wavelets," *Inf. Sci.*, vol. 332, pp. 72–83, Mar. 2016, doi: 10.1016/j.ins.2015.10.039.

[40] C. Zhang, J. Cheng, Y. Zhang, J. Liu, C. Liang, J. Pang, Q. Huang, and Q. Tian, "Image classification using boosted local features with random orientation and location selection," *Inf. Sci.*, vol. 310, pp. 118–129, Jul. 2015, doi: 10.1016/j.ins.2015.03.011.

[41] H. Deng, G. Runger, E. Tuv, and M. Vladimir, "A time series forest for classification and feature extraction," *Inf. Sci.*, vol. 239, pp. 142–153, Aug. 2013, doi: 10.1016/j.ins.2013.02.030.

[42] K. L. Priddy and P. E. Keller, *Artificial Neural Networks: An Introduction*, vol. 68. Bellingham, WA, USA: SPIE, 2005.

[43] W. Cohen, P. Ravikumar, and S. Fienberg, "A comparison of string metrics for matching names and records," in *Proc. KDD Workshop Data Cleaning Object Consolidation*, 2003, pp. 73–78.

[44] W. Su, J. Wang, and F. H. Lochovsky, "Record matching over query results from multiple Web databases," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 4, pp. 578–589, Apr. 2010, doi: 10.1109/TKDE.2009.90.

[45] A. Lavecchia, "Machine-learning approaches in drug discovery: Methods and applications," *Drug Discovery Today*, vol. 20, no. 3, pp. 318–331, Mar. 2015, doi: 10.1016/j.drudis.2014.10.012.

[46] S. Chaudhuri, B.-C. Chen, V. Ganti, and R. Kaushik, "Example-driven design of efficient record matching queries," in *Proc. 33rd Int. Conf. Very Large Data Bases (VLDB)*. Vienna, Austria: VLDB Endowment, 2007, pp. 327–338.

[47] V. Verykios, A. Elmagarmid, and E. Houstis, "Automating the approximate record-matching process," *Inf. Sci*, vol. 126, nos. 1–4, pp. 83–98, 2000.

[48] H. Mannila, "Data mining: Machine learning, statistics, and databases," in *Proc. 8th Int. Conf. Sci. Stat. Data Base Manage.*, 1996, pp. 2–8, doi: 10.1109/SSDM.1996.505910.

[49] S. M. Ghaffarian and H. R. Shahriari, "Software vulnerability analysis and discovery using machine-learning and data-mining techniques: A survey," *ACM Comput. Surv.*, vol. 50, no. 4, pp. 1–36, Nov. 2017, doi: 10.1145/3092566.

[50] D. Chicco, "Ten quick tips for machine learning in computational biology," *BioData Mining*, vol. 10, no. 1, p. 35, Dec. 2017, doi: 10.1186/s13040-017-0155-3.

[51] Z. Zeng, H.-K. Rao, and A.-P. Liu, "Research on personalized referral service and big data mining for e-commerce with machine learning," in *Proc. 4th Int. Conf. Comput. Technol. Appl. (ICCTA)*. May 2018, pp. 35–38, [Online]. Available: http://ieeexplore.ieee.org, doi: 10.1109/CATA.2018.8398652.

[52] C. Bellinger, M. S. Jabbar, O. Zaïane, and A. Osornio-Vargas, "A systematic review of data mining and machine learning for air pollution epidemiology," *BMC Public Health*, vol. 17, no. 1, pp. 1–10, 2017. [Online]. Available: https://bmcpublichealth.biomedcentral.com

[53] N. Lavrac, "Selected techniques for data mining in medicine," *Artif. Intell. Med.*, vol. 16, no. 1, pp. 3–23, May 1999, doi: 10.1016/S0933-3657(98)00062-1.

[54] V. H. Buch, I. Ahmed, and M. Maruthappu, "Artificial intelligence in medicine: Current trends and future possibilities," *Brit. J. Gen. Pract.*, vol. 68, no. 668, pp. 143–144, Mar. 2018, doi: 10.3399/bjgp18X695213.

[55] Y. Luo, W. K. Thompson, T. M. Herr, Z. Zeng, M. A. Berendsen, S. R. Jonnalagadda, M. B. Carson, and J. Starren, "Natural language processing for EHR-based pharmacovigilance: A structured review," *Drug Saf.*, vol. 40, no. 11, pp. 1075–1089, Nov-2017, doi: 10.1007/s40264-017-0558-6.

[56] I. Segura-Bedmar, P. Martínez, and C. de Pablo-Sánchez, "Using a shallow linguistic kernel for drug–drug interaction extraction," *J. Biomed. Informat.*, vol. 44, no. 5, pp. 789–804, Oct. 2011.

[57] S. Liu, B. Tang, Q. Chen, and X. Wang, "Drug-drug interaction extraction via convolutional neural networks," *Comput. Math. Methods*, vol. 2016, Jan. 2016 Art. no. 6918381, doi: 10.1155/2016/6918381.

[58] S. Ekins, A. C. Puhl, K. M. Zorn, T. R. Lane, D. P. Russo, J. J. Klein, A. J. Hickey, and A. M. Clark, "Exploiting machine learning for end-to-end drug discovery and development," *Nature Mater.*, vol. 18, no. 5, p. 435, 2019.

[59] C. Redfield, A. Tlimat, Y. Halpern, D. W. Schoenfeld, E. Ullman, D. A. Sontag, L. A. Nathanson, and S. Horng, "Derivation and validation of a machine learning record linkage algorithm between emergency medical services and the emergency department," *J. Amer. Med. Inform. Assoc.*, vol. 27, no. 1, pp. 147–153, Jan. 2020.

[60] Y. Siegert, X. Jiang, V. Krieg, and S. Bartholomaus, "Classification-based record linkage with pseudonymized data for epidemiological cancer registries," *IEEE Trans. Multimedia*, vol. 18, no. 10, pp. 1929–1941, Oct. 2016.

[61] Z. Bahmani, L. Bertossi, and N. Vasiloglou, "ERBlox Combining matching dependencies with machine learning for entity resolution," *Int. J. Approx. Reasoning*, vol. 83, pp. 118–141, Apr. 2017, doi: 10.1016/j.ijar.2017.01.003.

[62] D. Hand and P. Christen, "A note on using the F-measure for evaluating record linkage algorithms," *Statist. Comput.*, vol. 28, no. 3, pp. 539–547, May 2018, doi: 10.1007/s11222-017-9746-6.

[63] J. B. O. Mitchell, "Machine learning methods in chemoinformatics," *Wiley Interdiscipl. Rev., Comput. Mol. Sci.*, vol. 4, no. 5, pp. 468–481, Sep. 2014, doi: 10.1002/wcms.1183.

[64] M. Karthikeyan, R. Vyas, M. Karthikeyan, and R. Vyas, "Machine learning methods in chemoinformatics for drug discovery," in *Practical Chemoinformatics*. New Delhi, India: Springer, 2014, pp. 133–194.

[65] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, and R. Layton, "API design for machine learning software: Experiences from the scikit-learn project," in *Proc. ECML PKDD Workshop, Lang. Data Mining Mach. Learn.*, 2013, pp. 108–122.

[66] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011. [Online]. Available: https://dl.acm.org/doi/10.5555/1953048.2078195

[67] W. Yang, Y. Si, D. Wang, and B. Guo, "Automatic recognition of arrhythmia based on principal component analysis network and linear support vector machine," *Comput. Biol. Med.*, vol. 101, pp. 22–32, Oct. 2018, doi: 10.1016/j.compbiomed.2018.08.003.

[68] N. Ketkar, "Introduction to keras," in *Deep Learning With Python*. Berkeley, CA, USA: Apress, 2017, pp. 97–111.
[69] T. W. Edgar and D. O. Manz, "Exploratory study," in *Research Methods for Cyber Security*. Amsterdam, The Netherlands: Elsevier, 2017, pp. 95–130.
[70] K. M. Ting, "Precision and recall," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Boston, MA, USA: Springer, 2010, p. 781.
[71] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: A review of classification and combining techniques," *Artif. Intell. Rev.*, vol. 26, no. 3, pp. 159–190, Nov. 2006, doi: 10.1007/s10462-007-9052-3.
[72] S. Zhang, C. Zhang, and Q. Yang, "Data preparation for data mining," *Appl. Artif. Intell.*, vol. 17, nos. 5–6, pp. 375–381, May 2003, doi: 10.1080/713827180.
[73] R. Pita, E. Mendonça, S. Reis, M. Barreto, and S. Denaxas, "A machine learning trainable model to assess the accuracy of probabilistic record linkage," in *Proc. Int. Conf. Big Data Analytics Knowl. Discovery*, 2017, pp. 214–227.
[74] I. Gonzalez-Carrasco, A. Garcia-Crespo, B. Ruiz-Mezcua, J. L. Lopez-Cuadrado, and R. Colomo-Palacios, "Towards a framework for multiple artificial neural network topologies validation by means of statistics," *Expert Syst.*, vol. 31, no. 1, pp. 20–36, Feb. 2014.
[75] A. Jurek, J. Hong, Y. Chi, and W. Liu, "A novel ensemble learning approach to unsupervised record linkage," *Inf. Syst.*, vol. 71, pp. 40–54, Nov. 2017, doi: 10.1016/j.is.2017.06.006.
[76] K. Kim and C. L. Giles, "Financial entity record linkage with random forests," in *Proc. 2nd Int. Workshop Data Sci. Macro-Modeling (DSMM)*, 2016, pp. 1–2, doi: 10.1145/2951894.2951908.

JOSÉ LUIS LÓPEZ-CUADRADO received the Ph.D. degree in computer science from the Universidad Carlos III de Madrid, in 2009. He is currently a Visiting Professor with the Computer Science Department, Universidad Carlos III de Madrid. His research interests include software engineering, recommender systems, and accessibility. He is a coauthor of several papers in international journals, books, and conferences, and he also collaborates as a Reviewer in several international journals.

ISRAEL GONZÁLEZ-CARRASCO received the Ph.D. degree in computer science from the Universidad Carlos III de Madrid, in 2010. He is currently a Visiting Professor with the Computer Science Department, Universidad Carlos III de Madrid, where he is also the Assistant Manager. He is also a coauthor of several papers in international journals (indexed in ISI-JCR) and international conferences. His research interests include soft computing, software engineering, and accessibility. He has been involved in different national and international projects. He is also a member of the Editorial Reviewer Board of international journals (indexed in ISI-JCR) and the Organizing Committee at international conferences.

JESÚS LEONARDO LÓPEZ-HERNÁNDEZ received the bachelor's and master's degrees in computer systems from the Instituto Tecnológico de Orizaba (ITO), Veracruz, Mexico, in 2009 and 2011, respectively. He is currently pursuing the Ph.D. degree in computer science with the Universidad Carlos III de Madrid (UC3M), Spain. From 2013 to 2017, he was a Professor with the Universidad Tecnológica del Centro de Veracruz (UTCV), Mexico. His main research interests include brain computer interface, signals processing, accessibility, and software engineering.

PALOMA MARTÍNEZ-FERNÁNDEZ received the degree in computer science and the Ph.D. degree in computer science from the Universidad Politécnica de Madrid, Spain. She is currently the Head of the human language and accessibility technologies (HULAT) with the Computer Science and Engineering Department, Universidad Carlos III de Madrid. Her research interests include human language technologies, with the focus on information extraction in the biomedical domain, and web accessibility. She is a coauthor of more than 40 articles in indexed journals and more than 100 international conference contributions. She has been a Principal Investigator and participated in over 40 national and international research projects. She is also a member of the Spanish Society for Natural Language Processing (SEPLN) and the Dynamization Network for Activities on Natural Language Processing Technologies. She is also a Collaborator of the Spanish Center of Captioning and Audiodescription (CESyA).

JOSÉ LUIS MARTÍNEZ-FERNÁNDEZ received the Executive M.B.A. degree from the IE Business School and the Ph.D. degree in telecommunications from the Technical University of Madrid. He is currently a Co-Founder and a Stakeholder of MeaningCloud, an SME that develops services and software to improve information access by applying language and data analysis technologies. He has been working in the field of information access since he finished his grade, gathering experience in areas such as information retrieval, information extraction natural language processing, business intelligence, semantic technology, and big data. During these years, he has taken part (first as an Engineer and a Researcher, and then as the Manager) in many projects involving information access technology to satisfy customer needs. He also enjoys teaching and, since 2002, he is a part-time Professor with the Computer Science Department, Carlos III University of Madrid. He teaches subjects related to database management and development, information systems integration, and data structures and algorithms.