I. González-Carrasco, J.L. Jiménez-Márquez and J.L. López-Cuadrado et al. Automatic detection of relationships between banking operations using machine learning. *Information Sciences* 485 (2019) 319–346.

DOI: 10.1016/j.ins.2019.02.030

Automatic detection of relationships between banking operations using machine learning

I. González-Carrasco[1], J.L. Jiménez-Márquez[2], J. L. López-Cuadrado[3], Belén Ruiz-Mezcua[4]

***Abstract* —**

In their daily business, bank branches should register their operations with several systems in order to share information with other branches and to have a central repository of records. In this way, information can be analysed and processed according to different requisites: fraud detection, accounting or legal requirements. Within this context, there is increasing use of big data and artificial intelligence techniques to improve customer experience. Our research focuses on detecting matches between bank operation records by means of applied intelligence techniques in a big data environment and business intelligence analytics. The business analytics function allows relationships to be established and comparisons to be made between variables from the bank's daily business. Finally, the results obtained show that the framework is able to detect relationships between banking operation records, starting from not homogeneous information and taking into account the large volume of data involved in the process.

***Keywords*— Machine Learning; Big Data; Pattern detection; Business Analytics; Finance**

1 Introduction

The combination of an impressive explosion of data and the rapid development of new technologies to store and process this information has transformed the way in which companies manage their businesses. After an initial period, in which big data was considered something optional for most companies, its value is now widely recognized. Big data and analytics have begun to be part of the day to day of companies. Moreover, all over the world, organizations have begun to exploit the opportunities that big data offers. However, progress has been limited in terms of quantifying the value to be obtained with the analysis of structured and unstructured data jointly to generate knowledge that supports decision making. And it is precisely the value deriving from the management of this data and its transformation into useful knowledge that is probably the biggest advantage of big data and analytics.

Data analysis has gained strategic importance for virtually any organization. It covers areas like business analytics, big data, business intelligence, and data mining, among other [35]. Business Intelligence & Analytics (BI&A) is now widely used, especially in real-world practice, to describe analytic applications. It is currently a top priority for many chief information officers and has become a strategic initiative which is now recognized by CIOs and business leaders as instrumental in driving business effectiveness and innovation. BI&A is a process that includes two primary activities: getting data in and getting data out. Getting data in, traditionally referred to as data warehousing, involves moving data from a set of source systems into an integrated data warehouse. Getting data out consists of business users and applications accessing data from the data warehouse to perform enterprise reporting, OLAP, querying, and predictive analytics [12].

Therefore, BI&A and the related field of big data have become increasingly important in both the academic and the business communities over the past two decades. At the same time, no sector, including banking or the financial sector, is immune to the impact of the digital transformation and new capabilities to use the data [2]. The management tools for large amounts of customer information allow entities to generate more individualized services that favour loyalty and process improvement and daily operations [31]. Moreover, intelligent systems are providing bankers useful tools to support their decision process and help deal with complex portfolios [16].

[1] I. González-Carrasco, Universidad Carlos III de Madrid, Computer Science Department, Leganés, Madrid, igcarras@inf.uc3m.es
[2] J.L. Jiménez-Márquez, Universidad Carlos III de Madrid, Computer Science Department, Leganés, Madrid, joseluis.jimenez.marquez@uc3m.es
[3] J. L. López-Cuadrado, Universidad Carlos III de Madrid, Computer Science Department, Leganés, Madrid, jllopez@inf.uc3m.es
[4] Belén Ruiz-Mezcua Universidad Carlos III de Madrid, Computer Science Department, Leganés, Madrid, bruiz@inf.uc3m.es

In addition, the unstoppable growth of data analytics and information base management tool opens up a new range of services for the financial sector and a great capacity for specialization and individualization of its products [32]. In this context, the use of big data to improve the customer experience as well as techniques of artificial intelligence is an increasing trend.

In their daily business, bank branches must register their operations with several systems in order to share information with other branches and to have a central repository of records. In this way, information can be analysed and processed according to different requisites. Some of the systems are local to the branch and some others are central repositories that record the same operation from different points of view. A single bank operation is recorded in different systems depending, among other parameters, on the branch and the type of operation. In addition, the recording process for the operations might not be simultaneous. For this reason, the same operation can be reflected in several different records. In the context of our paper, the number of sources for the records is very high.

In this scenario, it is important for the bank to trace an operation along the different systems in which it could be registered. But the same operation registered several times in several different systems produces inconsistencies in the data. The records can be generated by different persons or systems in different times. For example, the precision of the decimal numbers or the currency in which the operation is registered could vary among the different systems: all records refer to the same operation, but they have distinct values; even the client could be different (clientID). These inconsistencies make the work of matching up one operation among all the recording systems difficult.

Audits, legal and quality assurance requirements make it necessary to have a control among the related annotations of an operation. The work of finding all the annotations referring to a given operation is very complex and time consuming. An individual could have some heuristics in order to determine the matching, but the number of operations processed in a world-wide bank makes the matching process impossible for a human. For this reason, it is necessary to find a way to automatize this process of matching records.

For the above-mentioned reasons, the main motivation of this research is the necessity of great bank branches to analyse the huge amount of operation records generated in their worldwide activities, considering that the same operation can be registered several times by different systems using different attributes. 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 a machine learning approach capable to learn the hidden patterns that allow determining whether two records from different banking systems represent the same operation 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. Thus, a machine learning strategy is a better approach for this problem.

The aim of this paper is to introduce a framework for solving these issues based on Machine Learning (ML) techniques in a big data environment. The main contribution of the framework is the ability to manage a great number of pairs of operation records from different systems and provide a degree of similitude in order to determine whether they represent the same operation or not. The proposed framework includes several stages, in order to move from not homogeneous data to structured information and for the automatic detection of relationships between banking operations, taking into account the large volume of data involved in the process. The first stage, pre-processing, allows the unstructured information of bank branches (from different sources) and the information centralized repository (containing annotations from different bank branches) to be merged. The second stage, machine learning processing, once the training process have taken place, will match or link each annotation from bank branches with the corresponding operation in the centralized repository. The third stage of the framework, post-processing, will process all the outputs of the second stage in order to give a detailed report of all the matching records detected. Finally, the output of the third stage will feed the business intelligence analytics component in order to establish relationships and comparisons between variables of the daily business for the bank.

The remainder of this paper proceeds as follows. Section 2 outlines the relevant literature in the area of machine learning and its implications for the banking and finance sector. Section 3 discusses the main features of the framework proposed, including a usage scenario and the main components of its architecture. Section 4 describes 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 and concluding remarks.

2 Related Work

As mentioned in the previous section, machine learning techniques has been widely applied in the banking and finance sector. Next subsections show an overview of the main trends in these areas.

2.1 Machine Learning Classifiers

In machine learning, 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, Artificial Neural Networks (ANN), Support Vector Machines (SVM) and Random Forests (RFO) are some Machine Learning (ML) techniques which are currently valuable tools for researchers in many domains.

The theory and modelling of Artificial Neural Networks (ANN) have been inspired by the structure and operation of biological neural systems, in which the neurons, the cells which form the cerebral cortex of living beings, are the main element. A neuron is a microscopic structure composed of three parts, namely, the cell body, the axon and dendrites. The brain continuously receives input signals from many sources and processes them to provide the appropriate output response. The brain has billions of neurons that interconnect to form neural networks. These neural networks execute the millions of functions needed to sustain normal life. ANN are an information processing paradigm that is inspired by the biological nervous system. It is also considered a mathematical model, composed of a large number of elements or processing units also called neurons. These neurons work together in order to solve specific problems. Similar to its structure, a neural network is a system that connects neurons through a network and distributes them in different levels to produce an output stimulus.

There are many ANN types classified according to characteristics such as topology, learning strategy or the type of input received. Due to their computational power, generalization capacity and dynamics properties, ANNs have been successfully used in solving complex problems in various fields such as medical diagnosis, forecasting foreign exchange rates or speech recognition, pattern recognition and computer vision.

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 [48]. 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 [47]. The performance of a support vector machine (SVM) depends highly on the selection of the kernel function type and relevant parameters [49]. SVM classifiers have been used for image denoising, multi-class sentiment classification, or even for online suicide prevention.

RFO are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges a.s. to a limit as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them [3]. RFO classifiers have been used for Account classification in online social networks, image classification or feature extraction.

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 [37].

2.2 Machine learning and banking

The banking sector has used computational resources and infrastructure since the beginning of the computer science era in the late 1950s. Since then, data storage and processing have been at the core of every banking company world-wide. Moreover, at present, being at the top of digital banking services can play a key role in any banking company's success. Young and even older customers can decide to continue with their bank or choose a different one based on the digital services and account security that a given bank is able to provide.

Banking companies also make use of computer programs for many internal services (accounting, human resources, the stock market, etc.) and for serving as an interface for other banking companies as well as government institutions. Even though such programs have had the ability to process huge amounts of information, there was scant or non-existent ability to obtain insights or find hidden patterns in information with the existing computing resources.

Machine learning and artificial intelligence have recently become a key factor in power banking services, although for decades there were many limited machine learning applications in this domain [42]. According to [33], the banking domains where high-level techniques are applied include: credit evaluation, branches performance, e-banking, customer segmentation and retention. Nevertheless, the introduction of Bitcoin and Blockchain technologies, constitute new domains in banking, and thus the methods have to be adapted to encompass new technologies.

With the increasing use of mobile devices, new services are being developed to reach more customers. Such devices generate a vast amount of information that needs to be analysed to discover hidden patterns. However, modern banking companies not only need to face these challenges; there are also concerns regarding money laundering and mortgage fraud, where machine learning and big data technologies can help banking overcome these problems.

The areas where banking has focused much of its effort regarding machine learning are: credit, prediction, fraud and bankruptcy [33]. Credit scoring is also a very important area for banks, as it allows them to decide whether to make a loan to an individual. Koutanaei et al. developed a hybrid method of feature selection algorithms for credit scoring by applying dimensionality reduction techniques and using classifiers as Support Vector Machine (SVM) [26]. In the area of detecting and preventing bankruptcy, Carmona et al. applied XGBoost, a modern machine learning algorithm to predict bank failure by analysing annual series of 156 U.S. national commercial banks [5].

In the banking operations domain, Liébana-Cabanillas et al. propose SEM-neural network approach for predicting antecedents of m-commerce acceptance [29]. In another research, Liébana-Cabanillas et al. define SEM-neural network approach for predicting the determinants of mobile payment acceptance [30]. Hew et al. propose an ANN-based analysis to capture both linear and nonlinear relationships in a research model to understand users' resistance behaviour towards m-commerce services[22].

2.3 Machine learning and finance

The creation and evolution of new technologies in special artificial intelligence and big data have been of paramount importance in enabling the financial sector to enhance services. Finance in particular is an area of in which the modern economy and the use of Bitcoins [11] will pose new challenges. Thus, the use of machine learning is one of the key tools that banking companies will have to incorporate into their daily operations to strengthen these capacities.

In their study, Li et al. present comprehensive research about the potential areas for the use of machine learning in finance and other business activities [28]. The authors of the present paper consider that before applying a strategy of solving a problem using neural networks, it is more important to consider if the data available and the expected output could fit an ANN. The research of Li et al. summarizes different applications of artificial intelligence technologies in several domains of business administration and finance.

Regarding machine learning techniques, in the domain of finance, Heaton et al. explore the use of deep learning hierarchical models for problems in financial prediction and classification [20]. In another research, Heaton et al. propose the use of deep learning to detect and exploit interactions in the data that are, at least currently, invisible to any existing financial economic theory [21].

In financial markets, it is both important and challenging to forecast the daily direction of the stock market return. Therefore, Zhong and Enke present data mining process with ANNs to forecast the daily direction of the S&P 500 Index ETF (SPY) return based on 60 financial and economic features [50]. Moreover, Chen and Chen propose an intelligent pattern recognition model for supporting investment decisions in stock market [6]. The research of Patel et al. addresses problem of predicting direction of movement of stock and stock price index for Indian stock markets [34]. The study compares four prediction models, Artificial Neural Network (ANN), Support Vector Machine (SVM), random forest and naive-Bayes with two approaches for input to these models. Finally, Chong proposes deep learning networks for stock market analysis and prediction [7].

2.4 Discussion

As it has been stated in the previous subsections, machine learning techniques have been widely used in the finance and banking domains for data analysis process, real-time predictive analysis, simplification of time-intensive tasks and automatization of complex, manual process. Moreover, artificial intelligence, including neural networks, deep learning and machine learning,

has made numerous progress and offered new opportunity for academic research and applications in many fields, especially for business activities [28].

Furthermore, matching operations among different systems can be part of the problem of record linkage: finding the same entity in different records from one or different data sources. Christen & Gosier present a high cited review of the state of the art of data linkage and deduplication [8,9] and Gollapalli presents also a classification of the main techniques for data linkage [15]. The state of the art in this area has evolved with general approaches and other more specific ones. Moreover, finding matching records is an active research area in several computer science domains due to its difficulty [19] and the choice of relevant sets of attributes and features is application dependent [1].

Recent works focuses on machine learning techniques as a way to improve the classification process. Stonebraker & Ilyas highlight the problems of data integration in nowadays environments, and conclude that there are multiple factors in choosing the right machine learning models and exploring the large number of design choices [44]. Bahmani et al. improve the process of ML classification using knowledge based on the semantic constraints in databases and emphasizes the importance of semantic knowledge in the optimization of the matching process [1].

Sukharev et al. deal with the issue of name matching in record linkage [45]. Textual information usually is used as a key factor for the matching process. Salehian et al. also approach machine learning-based matching based on text for Restaurant menus and food data [40]. Ruggles et al. review approaches centred on Census records and remarks the great possibilities of new large-scale data infrastructures for improving the matching methods [39]. However, when information is only based on numeric values usually it is necessary the combination of several techniques [15]. Textual data also implies extra privacy issues.

In the financial area Dagade & Mali work focuses on the duplicate records on bank domain, but their approach is based on textual data [10]. Other recent works apply machine learning methods for predicting financial distress of companies as Santoso & Wibow [41], however their approach is not related to bank records. Camino et al. find suspicious activities in financial records, but the record linkage is out of their scope [4]. Kim & Giles research studies the process of finding the same entity in a set of financial records, but their approach is centred on entity finding instead of recognize the same operation in several datasets [25]. Other approaches are based on non-supervised learning. A recent work of Jurek et al. applies self-learning models to several datasets but they concluded that they not outperform supervised classification models [24].

As can be seen in this review, machine learning has been applied widely in the financial area. There are several approaches for matching records, but, as mentioned by Bahmani et al. the selection of parameters is application dependent [1]. Therefore, after reviewing the main recently research in the area the authors have not found an approach for finding the same operation among different bank systems based on numerical attributes.

3 Solution Proposed

The related work section has shown the relations between machine learning, big data and finance sector. As it has been stated, despite the number of works in these areas, the concrete approach for the selection of parameters should be application dependent [1]. For this reason, a new framework for matching the same operation registered in different bank records from different systems is proposed. In the next subsections the main problem and the proposed framework is presented.

3.1 Description of the problem

As previously described, a single bank operation is recorded in different systems depending, among other parameters, on the branch and the type of operation. The recording process of the operations might not be simultaneous. It could be generated by different persons or systems at different times. Let's suppose several departments with different applications where each one process the same operations from different points of view (e.g. risk management, accounting, credit management, etc.) generating records of the same operation in different databases, and each system records the operation without taking into account the other ones (i.e. each system does not use unique identifiers for the operation, uses different currency or represents the amount of the operation using different precision). For this reason, the same operation can be reflected in several different records. Such annotations can also be different in key attributes such as nominal values. For example, one annotation can be valued in euros and another in dollars: both annotations refer to the same operation, but they have distinct values. In the context of this paper, the number of sources of annotations is very high. Banks require having a strict control of all the related records for one operation and finding all the records related to a given operation is very complex and time-consuming. Furthermore, the heuristics used by a person to determine similarity among different operations is difficult to represent due to the vast number

of sources and types of operations. In such a context, it is necessary to find a way to automatize the process of matching the annotations.

3.2 Framework description

Our research focuses on detecting matches between bank records by means of applied intelligence techniques in a big data environment and business intelligence analytics. This business analytics function allows relationships to be established and comparisons to be made between variables of a bank's daily business.

Based on this hypothesis, our study describes the designed framework based on machine learning and business analytics. Our proposed framework includes several stages in order to move from a not homogeneous structure to a common structure for all the information and for the automatic detection of relationships between banking records and for making comparisons between variables.

The different components and stages of the framework are shown in Figure 1. The framework is fed from different sources, systems from international bank branches and a centralized repository (data lake). For example, each bank branch has different systems that register the daily operation. Each record generated for each system of each branch should be matched with the operations stored in the data lake, taking into account that the same operation is stored several times depending on the systems involved. Moreover, the proposed framework has two different running scenarios or environments. In the first one, the train-test process of the machine learning component will be carried out. The aim of this process is to compare different combinations of patterns (set of parameters of the different records), ML classifiers and learning algorithms for this domain, following the breakthroughs introduced in the research conducted by Gonzalez-Carrasco et al. [18]. After this process, a benchmark and analysis of the performance will be done to detect the best combination of *pattern+architecture+algorithm* (called the neural model). Secondly, and once the training process have been done, the framework will be applied in a production environment for matching or linking each record from bank branches with the corresponding operation in the centralized repository. The best combination found in the train-test process will be included in this production step. Also, the business analytics component will receive the outputs of the framework in order to establish relationships and make comparisons between variables of the bank's daily business.

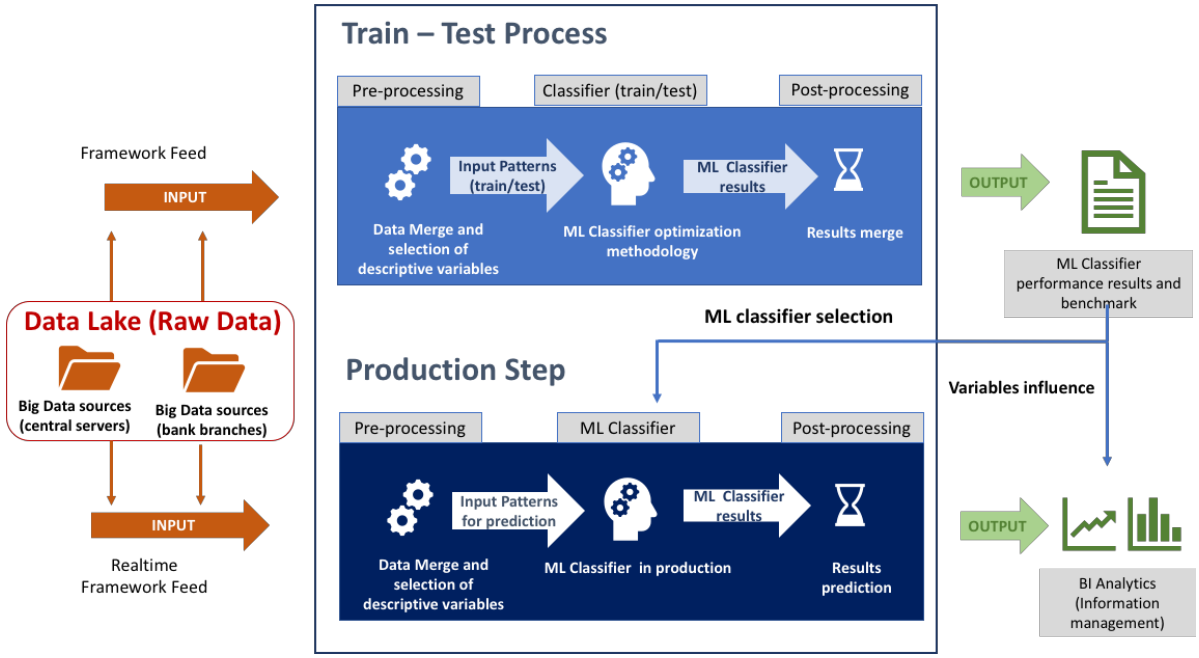

Figure 1. Framework for detecting relationships in bank operations

The first stage, pre-processing, allows unstructured information of bank branches (from different sources) and information from a centralized repository (containing operations from different bank branches) to be merged. All operations are stored in a

data lake, a repository that contains raw information of the records from all bank sources. In the training scenario, this stage is based on knowledge regarding matching of the operations. The information of the same operation in different records is merged in a single line of text in csv format and form the "positive cases": for example, two "known records" that represent the same operation are merged and labelled as positive in order to train the machine learning models. These csv lines contain a field "Found" with the value "true". Negative cases are generated by merging records from one source with records that are not matches from other sources (it is known they represent distinct operations). Each combination is merged into a single line of text in csv format ("negative cases"), and they contain a field "Found" with the value "False". In this way, the framework has a number of positive and negative cases to be used in the training and test phase of the ML classifier. In the production step, the system will receive candidate records that can be found in the record data lake. Given a candidate operation, a set of possible records are retrieved from the data lake. These possible records are selected according to a set of criteria such as the date of the operation, branch, etc., in order to restrict the number of possible combinations. Once the possible records are selected, the candidate operation is merged with each of the possible records in order to be compared by the ML classifier. As a result, the ML classifier provides the value true if both records match, or false otherwise.

The second stage, machine learning processing based on classifiers (ANNs and Random Forest), follows the breakthroughs introduced in research conducted by Gonzalez-Carrasco et al. [18]. The 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. Therefore, in the train-test process the authors apply knowledge obtained in previous research (using an optimization methodology for ML classifiers). This knowledge has been applied in order to guide the search for the best neural model in the given problem and hence improve performance of this task both in time and accuracy. To homogenize the large number of alternatives, they have been grouped into three elements, following the premise "neural model = pattern + architecture + algorithm". For each term of the equation, different techniques and methods will be analysed in order to improve the final performance. For the production step, the best combination found in the train-test process will be used.

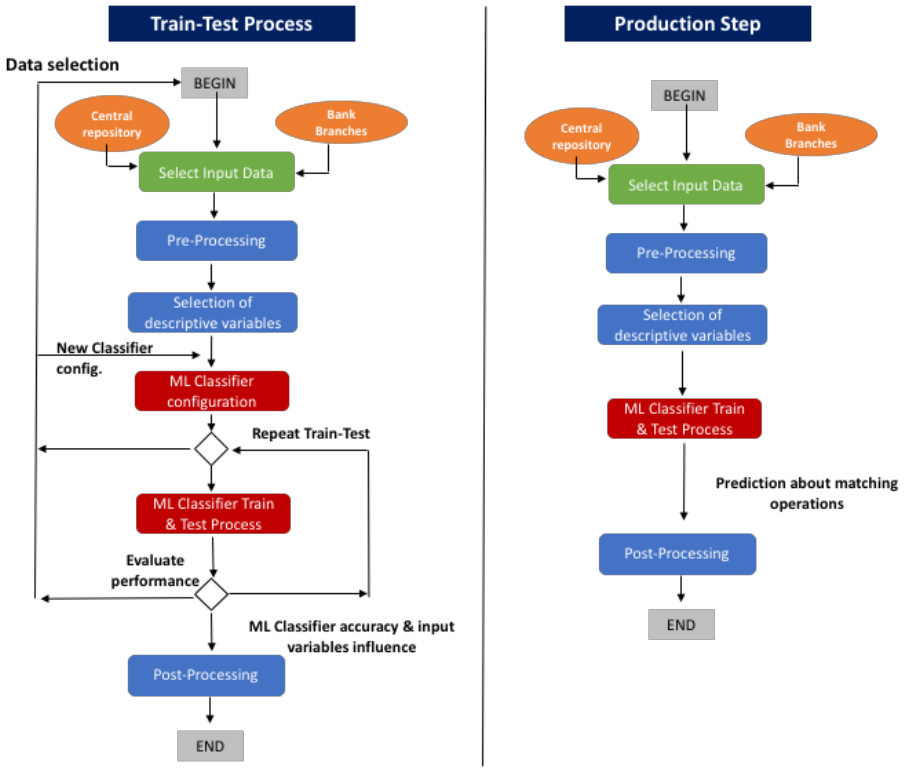

Figure 2. Framework steps for train-test and productions scenarios

The third stage of the framework is the post-processing. In the train-test process, this stage will combine all the partial results obtained and will process all the outputs of the second stage in order to give a detailed report of all the matching operations detected. With this information, a ML classifier performance analysis will be done to choose the best combination *of pattern + architecture + algorithm* to be included in the production step. Finally, in this production step, the output of the third stage

will feed the business intelligence analytics component in order to establish relationships and make comparisons between variables of the daily business for the bank.

In order to connect and automatize all the steps and stages of the framework, a pipeline has been defined using python. Moreover, the framework feed and the export process are made by webservices for connecting with the data lake (input) and the BI&A component (output). The data is structured with comma-separate values and JSON format.

3.3 Framework steps

As described in the previous sections, the framework is structured in three stages (preprocessing, machine learning and post-processing) and two scenarios (train-test and production) in order to manage and process the information. The first scenario set up the machine learning training with known data (it is known the operations that matches), while the second scenario receives new operations (with no information about whether they match or not) and decides about them. The input of the framework is a big set of operation records. All operation records are stored in a data lake that centralises all the bank information. As explained, one operation is registered in different ways in the data lake (several systems for each branch or type of operation for example): for this reason, an operation generates n records in the data lake. Locating all those records is a great challenge for humans due to the enormous amount of data stored in the data lake. In order to set up the proposed framework, a testing dataset has been provided by the bank. On the one hand, operation data about all bank branches has been provided in records obtained by an internal central system called CERE (CEntral REpository). On the other hand, records from the local systems of each branch have been provided: all these records have the same format but are generated by each branch and sent separately to the data lake. In addition, information about the correspondence of the operations has been provided in order to train and test the framework. Figure 2 shows the sources described as well as the subsequent steps for processing these operational records in both scenarios of the framework (train-test and production).

3.3.1 Stage 1. Preprocessing.

The first stage is in charge of preparing the data from the data lake that will be processed by the framework. The pre-processing stage is divided in four different steps: (1) “select candidate operations”, (2) “select possible records”, (3) “generate & merge“ and “filter”.
First of all, the main terms used in the description of the process will be defined. A candidate operation is the operation to be found in the data lake (step 1). The candidate operation is represented by a record (candidate record) that describes its attributes in a given system. In the context of this paper, the record of a candidate operation is in CERE format. The ML classifier will compare this candidate record with records from other sources in the data lake. The records to be compared come from the data lake and are called possible records. They are selected according to experts’ criteria in order to avoid unnecessary comparisons. For example, all possible records must have the same date as the candidate record (step 2).

Table 1. **ML classifiers included in the framework for train-test process**

Acronym	Classifier	Description	
RBF	Radial Basis Function	Radial basis function (RBF) network is nonlinear hybrid networks with a single hidden layer of processing elements (PEs). This layer uses gaussian transfer functions, rather than the standard sigmoidal functions employed by MLPs. The centres and widths of the gaussians are set by unsupervised learning rules, and supervised learning is applied to the output layer. These networks tend to learn much faster than MLPs. RBF networks have a very strong mathematical foundation rooted in regularization theory for solving ill-conditioned problems.	- Hidden layer and neurons: 0-0. - Training algorithm: Levenberg–Marquardt. - Competitive Rule: ConscienceFull - Activation function (output): -*f*tanhiper
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	- Hidden layer and neurons: 0-0. - Training algorithm: Kernel Adatron.

		are near the decision surface since they provide the most information about the classification.	
MLP	MultiLayer 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).	- Hidden layer and neurons: Follow the rules exposed in [18] . - Training algorithm: Extended BackPropagation. - Activation function (input-hidden-output): $f$sigmoid-$f$sigmoid-$f$linear. - Learning parameters: Genetic algorithm in each topology: μ and η for input.
RFO	Random Forest	Decision tree based classifiers like Random Forest (RFO) operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes. The random forest machine learner, is a meta-learner; Meaning consisting of many individual learners (trees). The random forest combined multiple random trees that votes on a particular outcome. The individual random tree growth process is repeated to develop multiple random trees machine learners. The out of bag data sets are used for evaluating the corrective ness of each trees as well as the entire forest. Each out-of-bag dataset is put down each tree to get a classification. The average misclassification over all trees is known as the out-of-bag error estimate.	- Number of parameters: 0. - Number of iterations: 100. - Bagsize percent: 100.
LSVC	Linear SVC	Scalable Linear Support Vector Machine for classification. It has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples. LSVC supports both dense and sparse input and the multiclass support is handled according to a one-vs-the-rest scheme.	- Hidden layer and neurons: 0-0. - Training algorithm: Kernel Linear.
BAY	Bayes Network	Bayes Network learning using various search algorithms and quality measures. This Bayes network learning algorithm uses a hill climbing algorithm restricted by an order on the variables (K2 algorithm). Also, this Bayes network uses select estimator algorithm for finding the conditional probability tables of the Bayes Network Simple Estimator is used for estimating the conditional probability tables of a Bayes network once the structure has been learned. Estimates probabilities directly from data.	- Estimator: Simple Estimator (alpha 0,5). - Search Algorithm: K2

The most important features of the classifiers included in this study are also detailed in Table 1. Two complementary stopping criteria were used during the network training for all the alternatives: reaching a specific number of epochs, and early stopping. Therefore, with the set of data available, a part has been assigned to the network's training (train and validation subset), whereas others have been assigned to test the results obtained (test subset). Moreover, common features of ML classifiers are shown in Table 2.

Table 2. **Common features for classifiers in machine learning step**

Feature	**Description**
Inputs	V_{inp} (depends on the input data file)
Outputs	V_{out} (match or no match): 2 neurons.
Patterns distribution (train-test-validation)	33 %/33 %/33 % with hold-out cross validation
Cost function	MSE simple
Weight update	*Batch*
Weight initialization	Haykin heuristic
Convergence criteria	*Epochs* [10000] and early stopping

As the ML classifier will determine the similarity between the operation records from two sources, the pre-processing phase takes the candidate records from the central source and generates positive and negative samples to train the ML classifier by merging them with records from a single branch of the data lake (step 3). The pre-processing process must be analysed from two different points of view: training and production.

In the training environment, the pre-processing phase generates positive and negative cases to be used for training the ML classifier, using operations provided by the bank. In the production environment, the pre-processing phase processes the information in the same way as is processed by the classifier in order to determine whether the case is positive or negative.

Given a candidate operation, a positive case is a combination of a candidate record with a possible record of the data lake that represents the same operation (i.e. the same operation registered in two different systems). On the other hand, a negative case is a combination of a candidate record with a possible record that represents a different operation.

In the training-test process, the correct correspondence among records is known in order to train and test the model. Each record (both candidate and possible) is represented as a line of a csv text file. Thus, in the training phase, the pre-processing phase will take a number of candidate operation records to be found and which will be merged with their corresponding possible records in another branch, labelling each merged line as positive ("Found" = "true"). Negative training cases are generated by combining the possible records that are not matches with the candidates with said candidates and are labelled as negative ("Found" = "false").

The raw training set is formed by combining both positive and negative cases (step 3). The number of negative cases is higher than the positive ones because the number of combinations of candidate operations with no matched possible operations is very high.

Once positive and negative cases are generated, it is necessary to analyse the relevance of the fields of the records involved in the comparison (step 4). Thus, fields with a high number of null values are discarded. There are alternatives to deal with missing values in machine learning but the decision to discard them was due to the fact that the fields implied in these null values are not easy to replace (i.e. account numbers). Also, some classifiers such as MultiLayer Perceptron (MLP) or Support Vector Machines (SVM), only accept numeric values as input data. However, some fields such as the counterpart are string values. Some approaches translate the strings into numeric values using keys of Wordnet for example [38]. In this case, relevant fields based on string values are translated into numerical values by means of hash functions, because they are nonsense combinations of characters so that they cannot be found in a given list of words. In this way, a numerical input can be used for classifiers like MLP or SVM. In the same way, date values are translated into numerical values by using the convection of counting the number of days from January 1,1900.

3.3.2 Stage 2. Machine Learning.

Once the data has been prepared in the previous stage, Stage 2 involves the machine learning paradigm based on ANN algorithms and has two different running scenarios or environments. In the first one, the train-test process of the machine earning component will be carried out. The aim of this process is to compare different combinations of patterns, ANN topologies and learning algorithms for this domain, following the breakthroughs introduced in research conducted by Gonzalez-Carrasco, Garcia-Crespo, Ruiz-Mezcua, Lopez-Cuadrado, & Colomo-Palacios [18]. After this process, a benchmark and analysis of the performance will be carried out to detect the best neural model (combination of *pattern+architecture+algorithm*). Secondly, and once the training process have been done, the framework will be applied in a production environment to match or link each record from bank branches with the corresponding operation in the centralized repository. The best combination found in the train-test process will be included in this production step.

The strength of an ML classifier is reflected in its capacity to recognize complex patterns in the real world which can represent noise or uncertainty due to its intrinsic nature. The different classifiers included in this step are shown in Table 1.

The calculation of the uncertainty associated with each pattern is probabilistic. The outputs of the neural model, within this classification environment, estimate the probabilities that an input pattern belongs to one class or another. In this domain, each output is dichotomous, with values YES (matching), or NO (not matching), internally converting itself to the binary values [0, 1]. For extreme values, the network behaviour is obvious, however when values are close to the border decision, e.g. 0.5, some uncertainty arises about its classification in any of the classes. To solve these situations, the probabilistic Bayes theorem has

been applied where C_k indicates the class C for k output (output neuron of the classifier) and x indicates each of the different patterns.

$$P(C_k \mid\mid x) = \frac{p(x \mid\mid C_k)P(C_k)}{\sum_{k=1}^{k} p(x \mid\mid C_k)P(C_k)} \quad (1)$$

being the decision rule for each pattern x the following:

assign x to C_i class if $P(C_i \mid\mid x) > P(C_j \mid\mid x)$ for all $j \neq I$ (2)

As additional information, during the training and test phase, the correct classification percentage is obtained:

$$E\% = \frac{100}{NP} \sum_{j=0}^{P} \sum_{i=0}^{N} \frac{\mid\mid dy_{ij} - dd_{ij} \mid\mid}{ddi_j} \quad (3)$$

Where:

- P number of output neurons.
- N number of patterns.
- y_{ij} denormalized output obtained for the pattern i in the output j.
- d_{ij} real denormalized output for the pattern i in the output j

Nevertheless, if the parameters calculated in each scenario enable determination of the functioning and performance of a concrete ANN, it is common to check various alternative models with similar results, which does not allow the better choice to be determined. To facilitate this task, two statistical estimators have been included which indicate the goodness of fit of an estimated statistical model. These indicators, measures of quality based on information are Akaike's Information Criterion (AIC) and Minimum Description Length (MDL) [17,18] The use of these parameters in the field of ML classifiers allows the optimal neural model for a given problem to be selected from a number of candidates.

AIC is used to measure the trade-off between training performance and network size. The goal is to minimize this term to produce a network with the best generalization:

$$AIC(k) = Nln(MSE) + 2k \quad (4)$$

Where:

- k is the number of weights of the network.
- N is the number of observations in the training set.
- MSE is the average quadratic error obtained.

The AIC indicator has been used by researchers for different aims, e.g. for optimizing ML classifiers, and even for design committees of networks.

MDL criterion is similar to the AIC in that it tries to combine the model's error with the number of degrees of freedom to determine the level of generalization. The goal is to minimize this term:

$$MDL(k) = Nln(MSE) + 0{,}5\ln(N) \quad (5)$$

Where:

- k is the number of weights of the network.

- N is the number of examples of the training set.
- MSE is the average quadratic error obtained

The MDL indicator has enabled ML classifier optimization and selection of relevant input parameters [23].

In the production step, the combination of candidate records with possible records will be evaluated and classified as true or false. The value true means that both records match, meanwhile the value false indicates that both records represent different operations. For each class (true and false) the values of Precision, Recall and F1 will be calculated [46]. Next formulae were applied:

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \quad (6)$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives} \quad (7)$$

$$F1 = 2\frac{Precision \cdot Recall}{Precision + Recall} \quad (8)$$

Where:

- Precision represents the percentage of the correct classified cases (true positives) among the ones classified by the system (true and false positives).
- Recall represents the percentage of the correct classified records provided by the system (true positives) among the number of real correct ones (true positives and false negatives).
- 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.

Moreover, in Table 3 different heuristics such as the Pruning and Sensitivity analysis techniques, have been included to quantify variable importance in the prediction made by the neural model. This stage ensures an optimal generalization ability, since it is responsible for obtaining the smallest ML classifier architecture possible. The conclusions obtained using these techniques will allow the classifier complexity to be reduced, thereby shortening the train-test process, as well as also feeding the BI&A component for decision making.

Table 3. **Techniques for analysing variables influence**

Technique	Acronym	Description	References
Sensitivity About the Mean	SAM	Shows the determination of the influence of each of the inputs in the output which the network obtains	See [17,18] for more details.
Indicator of Variable Criticality	IVC	Represents in the range [0, 1] the number of times which a variable k was outside the range when the prediction has failed, and the total number of times that a variable k has been outside the range.	
Backward Stepwise Method	SWM	Eliminates a variable from the input vector sequentially and analyses the effect on the output of the network	

Sensitivity About the Mean (SAM) metric permits the determination of the influence of each of the inputs in the output which the network obtains. The change applied to the inputs is generated by adding a random value, obtained on the basis of a variance determined by the user, for each example. Then the corresponding output is calculated for each example. This process is repeated a number of times, determining the result for each of the different inputs and in order to obtain the sensitivity of an input k by means of the following equation:

$$S_k = \frac{\sum_{P=1}^{p} \sum_{i=1}^{0} (y_{ip} - \bar{y}_{ip})^2}{\sigma_k^2} \quad (9)$$

Where:

- $\bar{y}_{ip}$ is the *i*th output obtained with the fixed weights for the *i*th pattern p.
- o is the number of output neurons.
- P is the number of patterns.
- $\sigma^2{}_k$ is the variance of the alteration introduced.

This indicator is calculated once the network has been trained, and effectively measures how a modification to the inputs affects the output based on the dataset available. The data with greater sensitivity hold greater importance, and thus should be maintained in the neural model. On the contrary, those with little sensitivity can be taken account for their elimination, which permits an improvement of the training given that the size of the network is reduced. It also allows a decrease in the cost of obtaining the data, with an insignificant effect on the performance of the network.

IVC represents, in the range [0, 1] the number of times which a variable k was outside the range when the prediction has failed, and the total number of times that a variable k has been outside the range. Outside the range is understood as a value which has not been utilized during the training, and thus, is new for the ANN. The equation used for this indicator for a variable k is the following:

$$C_{k,n} = \frac{R_{k,n}}{F_{k,n}} \quad (10)$$

Where:

- $R_{k,n}$ indicates the number of times that the variable k has gone out of range in n test.
- $F_{k,n}$ represents the number of times which the output has failed when the variable k has yielded atypical values in n test.

SWM method is an observation of the change in the error value when an adding (forward) or an elimination (backward) step of the input variables is operated. In this case, the elimination of the input occurs when the network is trained and the connection weights corresponding to the input variable studied are also eliminated. The variable that gives the largest MSE when eliminated is the most important. A classification of the variables can thus be made [14].

In the train-test process, the procedure followed for each simulation performed within this process is outlined below:

1. Select one ML classifier alternative.
2. Select descriptive variables.
3. Configure ML classifier: topology and learning algorithm (see Table 2).
4. Apply the resampling and distribution method to the trial-test set (see Table 2).
5. Train and test the corresponding ML classifier using the repeated 20×10 cv method (k-fold cross validation repeated 20 times with weights being randomly initialized and 10 folds).
6. Analyse the quality criteria (correct classification percentage, AIC and MDL values).
7. Perform the sensitivity analysis (see Table 3)

For the production step, when the train-test process has finished, the procedure is the following:

1. Choose the winner classifier from train-test process.
2. Select descriptive variables (using feedback from train-test process).
3. Configure ANN: topology and learning algorithm (see Table 2).
4. Apply the resampling and distribution method to the trial-test set (see Table 2).
5. Perform the accuracy analysis (precision, recall & F1 measure).

3.3.3 Stage 3. Post-processing.

Once the stage 2 has obtained the results for the different ANN classifiers the third stage of the framework, post-processing, starts. This stage is necessary in order to incorporate the knowledge obtained with the framework into the BI&A component. Post processing will combine all the partial results obtained and process all the outputs of the second stage in order to give a detailed report of all the matching operations detected.

In the train-test process, all the results obtained in Stage 2 will be processed. Stage 2 gives as output the accuracy of all configurations of ANN. With this information, a ML classifier performance analysis will be done to choose the best combination of *pattern* + *architecture* + *algorithm* to be included in the production step. Moreover, as output of Stage 2, this stage receives the influence of input variables in the output prediction of the classifier to determine dependences and redundancy in the input data (see Table 3).

All the information related with dependences, redundancy, etc., extracted from SAM, IVC and SWM techniques will be structured using JSON in order to be incorporated into the BI&A component. Table 4 shows an abstraction of this JSON. Columns Experiment, Scenario and File indicate the number of the experiment performed, corresponding scenario and input file used. Column Number of patterns shows the number of records included in each input file (with positive and negative cases).

In the evaluation section, all the experiments, scenarios and input files defined in this research are explained (see Table 6). Columns SAM, IVC and SWM show the results obtained for each of these techniques in each experiment. SAM and IVC techniques obtain as output the most (↑) and least (↓) influential variables for each experiment. SWM obtains the best input vector for each experiment. SAM and IVC techniques give as output the number of variables with values bigger than 0,5 (↑) or lower than 0,2 (↓).

Table 4. **Structure defined for information related with SAM, IVC and SWM techniques**

Experiment	Scenario	File	SAM		IVC		SWM
Experiment1	*Individual 1*	*IF1*	↑	*Number of variables*	↑	*Number of variables*	*Best* V_{inp}
			↓	*Number of variables*	↓	*Number of variables*	
Experiment2	*Individual 1*	*IF2*	↑	*Number of variables*	↑	*Number of variables*	*Best* V_{inp}
			↓	*Number of variables*	↓	*Number of variables*	
…	…	…	…		…		…
Experiment21	*Pool*	*IF21*	↑	*Number of variables*	↑	*Number of variables*	*Best* V_{inp}
			↓	*Number of variables*	↓	*Number of variables*	

For the production step, this stage will receive in real time and on demand a data bucket for the prediction about matching operations. The model to be used in the production step will be the most suitable of the obtained in the training phase. As mentioned, the ML classifier compares two records in order to determine whether they represent the same operation or not. In the post-process stage, both records involved in the operation will be structured using JSON in order to be incorporated in the BI&A component. Table 5 shows an abstraction of this JSON composed by each variable ($Attribute_i$) and its influence (determined in training time for each model) for the matched records ($Record_1$ and $Record_2$) of each operation.

Table 5. **Structure defined for operations and related records.**

Operation	$Attribute_i$ *- Record 1 (Influence)*	$Attribute_j$ *- Record 1 (Influence)*	…	$Attribute_z$ *- Record 2 (Influence)*
Operation	$Attribute_i$ *- Record 1 (Influence)*	$Attribute_j$ *- Record 1 (Influence)*	…	$Attribute_z$ *– Record 2 (Influence)*
Operation	$Attribute_i$ *- Record 1 (Influence)*	$Attribute_j$ *- Record 1 (Influence)*	…	$Attribute_z$ *– Record 2 (Influence)*
…				
Operation	$Attribute_i$ *- Record 1 (Influence)*	$Attribute_j$ *- Record 1 (Influence)*	…	$Attribute_z$ *- Record 2 (Influence)*

3.3.4 Business Intelligence & Analytics.

Finally, when the results have been produced and processed in the previous stages, they can be exploded by the bank. Thus, the last step of the framework is to feed the business intelligence analytics component in order to establish relationships and comparisons between variables of the daily business for the bank. BI&A focuses on future analysis based on company information and predictive models to support decision making and improve business competitiveness. In other words, BI&A has a strong focus on the analysis of the current situation and the prediction of future events to determine the path that the company will take

In this case, the BI&A component will receive information from the framework related with the matching operations detected but also will be fed with information about the influence of each of the inputs in the output which the ML classifier obtains, the relationships and dependences between variables. The techniques for this matter have been explained in Stage 3 of the framework

Knowledge about the dependences and relationships can be analysed by managers in order to simplify banking operation records process storage or even to structure this data for improved processing.

Finally, the BI&A component receives structured information about matching operations. This information is incorporated into the BI&A as a trace of the operation with its different records for future use. In addition, the conclusions and feedback obtained can be used to reduce resource consumption, e.g. storage space, computing time, etc.

4 Evaluation

This section shows the process of evaluation and validation performed in order to determine the contribution of the research.

4.1 Data and Experimentation

First of all, this section describes the data used in the evaluation process as well as the experiments to be executed. The framework infrastructure has been tested in big data architecture in order to assure the scalability of the framework. The data lake is a big data environment and for the experimentation included in this research a portion of the operations and records available is used.

For evaluation of the framework, different experiments have been defined and grouped into two different scenarios. In the first one, the framework is fed with information from different bank branches. The operations and records from each bank branch have different structures, so the input data for the machine learning is not homogenous and accordingly the classifier topology cannot be the same for each bank branch. Hence, the train-test and production processes are performed with operations from only one bank branch at the same time. Thus, a neural model is generated for each branch and input file. The aim of this scenario is to test the accuracy of the framework with restricted information from a single international branch.

In the second scenario, the framework will be fed with information from different bank branches at the same time. In this case, in the pre-processing, the framework will standardize and normalize all the information in order to create homogenous input data for the machine learning stage. Hence, the train-test and production processes are performed with operations from multiple sources or bank branches. In this case, a single model is generated for all branches. The aim of this scenario is to test the accuracy of the framework with global information from different international bank branches.

Moreover, for each scenario, different data files have been generated in order to include as much representative information as possible with different combinations. The descriptive variables included in the input data file will determine the classifier topology for Stage 2.

In Table 6, the list of 126 experiments performed in this research is displayed: 108 experiments for the individual scenarios and 18 experiments for the pool scenario. In the individual scenario, there are two possibilities: input files with all variables from each bank branch (individual scenario 1) and input files only with the common variables for all bank branches (individual scenario 2). In the individual scenario, each bank branch generates three different combination of individual input files. In the pool scenario, all the information from different bank branches is put together in the same input file removing the non-common variables. Again, three different combinations are generated.

The data sources extracted from the data lake represent information from International Bank Branches (IBB) and CERE (central repository). After the pre-processing stage, input files could have different dimension, depending on the steps performed in this stage. First of all, an Input File (IF) is generated for each IBB taking into account only the numerical attributes (IF1, IF4 and IF7). Numerical attributes are those whose values are a number or null. Null values represent attributes which have no value assigned. The problem of null values is that some algorithms have problems dealing with them. There are several approaches but due to the uncertainty of the impact of applying such techniques, the authors of this study have decided to train the models removing such attributes. In this way, a second set of input files is generated for each IBB (IF2, IF5, IF8) removing the attributes with at least one null value in the dataset. Finally, there are two attributes that represent the account number in both CERE and IBB. In order to test the ability of the models to match the characteristics of the operation, a final set of files has been generated omitting the account identifiers (IF3, IF6, IF9): in these cases, the operations are anonymous. At this point, the files generated tests the models for each IBB.

This process is repeated in the scenarios "individual 2" and pool but taking into account only the common variables in all the IBB. Hence, these IFs discard all the attributes that are not common to all IBBs. Despite all files have the same attributes, depending of the IBB some attributes may contain null values or not. Thus, only the attributes present in all IBBs are considered in these IFs. The criteria to generate these files is the same as that applied for each separated IBB. Doing this, different IFs have been obtained: from IF10 to IF18 for individual scenario 2 and from IF19 to IF21 for the pool scenario.

In the individual scenario 2, each IBB generates three different combinations of individual input files (in order to generate a neural model for each IBB) and in the pool scenario, three sets of IFs are generated combining the data of the IBBs in the same IF (in order to generate a common neural model for all IBBs). Thus, in the pool scenario the model is trained with records merged from all IBBs instead of only one IBB.

This point is important to determine whether the models are valid only for a single IBB or if there exists a model that is valid for all the IBBs (extrapolating the findings).

Next, each IF is used in each of the classifiers, analysing accuracy, performance and variables' influence. Finally, for each IF (individual and pool scenarios), the best classifier in the train-test experiment is also performed in the production step.

Table 6. **Breakdown of experiments performed**

Scenario	Data Source	Input file / Dimension	Number of patterns (positive/negative)	ML Classifiers
Individual / 54 experiments	IBB1 + CERE	IF1 / 55 variables	14408 records (7204 positive / 7204 negative)	RBF, SVM, MLP, RFO, LSVC and BAY
	IBB1 + CERE	IF2 / 38 variables		
	IBB1 + CERE	IF3 / 36 variables		
	IBB2 + CERE	IF4 / 55 variables	1239 records (619 positive / 620 negative)	
	IBB2 + CERE	IF5 / 35 variables		
	IBB2 + CERE	IF6 / 33 variables		
	IBB3 + CERE	IF7 / 55 variables	4418 records (2209 positive / 2209 negative)	
	IBB3 + CERE	IF8 / 41 variables		
	IBB3 + CERE	IF9 / 39 variables		
Individual (common variables) / 54 experiments	IBB1 + CERE	IF10 / 52 variables	14408 records (7204 positive / 7204 negative)	RBF, SVM, MLP, RFO, LSVC and BAY
	IBB1 + CERE	IF11 / 35 variables		
	IBB1 + CERE	IF12 / 33 variables		
	IBB2 + CERE	IF13 / 52 variables	1239 records (619 positive / 620 negative)	
	IBB2 + CERE	IF14 / 35 variables		
	IBB2 + CERE	IF15 / 33 variables		
	IBB3 + CERE	IF16 / 52 variables	4418 records (2209 positive / 2209 negative)	
	IBB3 + CERE	IF17 / 35 variables		
	IBB3 + CERE	IF18 / 33 variables		

Pool / 18 experiments	IBB1, IBB2 & IBB3 + CERE	IF19 / 52 variables	20065 records (10032 positive / 10033 negative)	RBF, SVM, MLP, RFO, LSVC and BAY
	IBB1, IBB2 & IBB3 + CERE	IF20 / 35 variables		
	IBB1, IBB2 & IBB3 + CERE	IF21 / 33 variables		

In summary, as is explained in Table 6, for each scenario there are three IF sets, depending on how the IFs have been generated:

- IF Set 1: Original files from IBB 1, 2 and 3.
- IF Set 2: Version 1.0 of files from IBB 1, 2 and 3 (without nulls).
- IF Set 3: Version 2.0 of files from IBB 1, 2 and 3 (without nulls and omitting the account identifiers).

4.2 Pre-processing

The data described in the previous section was processed according the steps described in section 3.3.1. Figure 3 depicts the different steps performed in the pre-processing stage of the framework. For the evaluation, records from CERE and three bank branches (IBB1, IBB2, IBB3) have been provided in four different csv files (cere.dat, ibb1.csv, ibb2.csv, ibb3.csv). For each IBB record, common records from CERE are identified. These records are matched with records of the IBB in order to generate a set of positive cases. In general, the correspondence between the CERE records and IBB records is one to one, but there are some cases in which one CERE record has two IBB related records. For this reason, they are separated into two different files (1-1 and 1-2). After the positive cases are generated, non-matched records of CERE are merged with the non-matched records of the IBB in order to generate the negative cases (Branchtmp file). Each positive case is tagged with an attribute "Positive=True" and each negative case is tagged with an attribute "Positive=False". Then, all records are joined into one file (branch.csv) to be used in the training-test process for the IBB. Before the train and test process, the fields are selected according to the procedure described in Stage 1 (see section 3.3). Finally, the ML classifier simulator is trained and tested with the file in order to create a neural model for the bank branch.

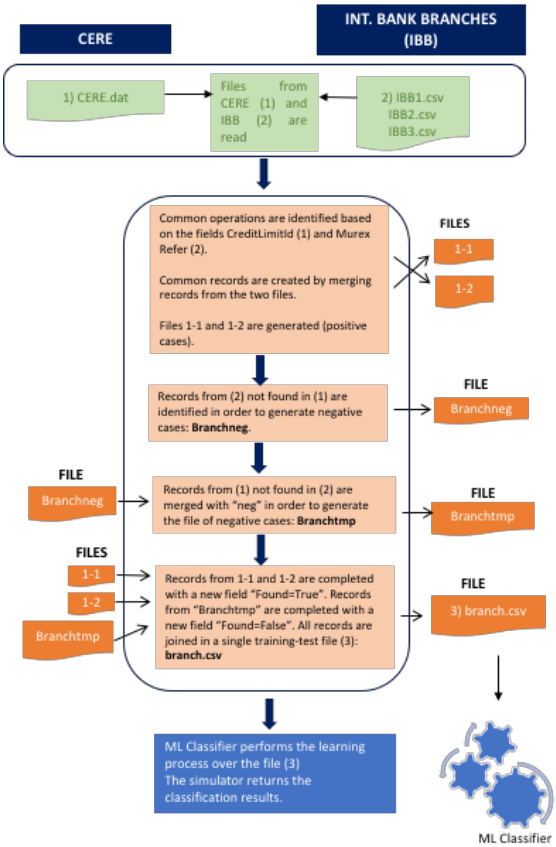

Figure 3. Sample of Pre-processing stage

4.3 Machine Learning

This section shows the evaluation process performed in the machine learning stage of the framework based on the data pre-processed in the previous stage.

4.3.1 Train-test process

During this process, the correct classification percentage is obtained for each experiment. Moreover, to corroborate the accuracy results, an analysis focused on measuring the quality of each ML classifier after applying the framework has been carried out. Two statistical criteria based on information theory, AIC and MDL, are included to compare the degrees of goodness of fit for each proposal. Table 7, Table 8 and Table 9 all show the results for these criteria and for each experiment (columns AIC and MDL) for the individual and pool scenarios.

Individual Scenario 1

In the individual scenario 1 and 2, the classifier learns with data from one IBB and CERE. In the pool scenario, the information from all IBBs (1,2 and 3) is put together in the same IF (see Table 6 for more details).

Table 7 shows accuracy results and AIC and MDL indicators for each experiment in the individual scenario 1. The best classifier is RFO with a top accuracy of 99,90% for IBB1 with IF1 and an average of 99,54% for all the experiments. In addition, the other two topologies with better accuracy, BAY **(**99,87% for IBB1 with IF3) and MLP (99,68% for IBB3 with IF7) networks, have obtained good performance with the AIC and MLP indicators. The result on bold indicates the best result for each classifier between all the experiments (IFs and IBBs).

Table 7. **Individual scenario 1. ML classifier performance (% correct classification) and AIC and MDL indicators. 20 runs×10 cv method**

Input file	Results	RBF	SVM	MLP	RFO	LSVC	BAY
IF1	Accuracy	97,63%	97,55%	99,41%	**99,90%**	97,71%	99,86%
	AIC	-5747,83	-32877,91	-47691,83	-48723,83	-28994,91	-52006,83
	MDL	-7697,32	-25714,78	-40111,32	-43853,32	-26887,78	-46000,32
IF2	Accuracy	97,21%	97,73%	98,94%	99,82%	97,69%	99,715
	AIC	-34674,91	-21590,24	-39403,91	-41704,91	-21878,24	-47221,91
	MDL	-27049,78	-9635,42	-39637,78	-34976,78	-10646,42	-44524,78
IF3	Accuracy	97,20%	97,70%	98,88%	99,80%	97,62%	**99,87%**
	AIC	-21546,24	-37580,36	-29324,24	-34141,24	-33571,36	-37816,24
	MDL	-8858,42	-26860,97	-15512,42	-12806,42	-27907,97	-20885,42
IF4	Accuracy	97,32%	**99,01%**	97,71%	99,27%	**99,21%**	99,38%
	AIC	-21108,36	-29897,91	-51407,36	-44606,36	-35566,91	-53871,36
	MDL	-22730,97	-25143,78	-45671,97	-46036,97	-29310,78	-54406,97
IF5	Accuracy	96,02%	98,94%	97,46%	98,55%	99,19%	98,11%
	AIC	-23169,15	-23054,24	-34513,15	-35582,15	-25463,24	-39850,15
	MDL	-27205,84	-8927,42	-36169,84	-34197,84	-9517,42	-38182,84
IF6	Accuracy	97,12%	98,82%	97,58%	99,31%	98,92%	99,29%
	AIC	-27189,27	-41314,36	-38149,27	-44604,27	-34458,36	-46578,27
	MDL	-28678,43	-26950,97	-38683,43	-43432,43	-26795,97	-46953,43
IF7	Accuracy	**98,73%**	98,93%	**99,68%**	99,78	98,95%	99,79%
	AIC	-105669,44	-31769,91	-110616,44	-111473,44	-27689,91	-117074,44
	MDL	-83229,86	-32234,78	-114418,86	-116302,86	-33988,78	-118442,86
IF8	Accuracy	98,53%	98,71%	99,49%	99,71%	98,64%	99,22%
	AIC	-66229,47	-27125,24	-80816,47	-81405,47	-19535,24	-85097,47
	MDL	-72189,14	-1265,42	-83531,14	-81824,14	-8042,42	-86129,14
IF9	Accuracy	97,96%	98,10%	99,66%	99,69%	98,87%	99,68%
	AIC	-79779,43	-36598,36	-81202,43	-89947,43	-40619,36	-90953,43
	MDL	-74624,92	-31462,97	-87185,92	-87758,92	-29067,97	-89829,92
Average	**Accuracy**	*97,52%*	98,63%	98,76%	**99,54%**	98,53%	99,43%

Moreover, the evolution of the train-test process in seconds for each IBB is depicted in Figure 4. These figures show the mean in seconds for all experiments per classifier and IF (20 runs). The results obtained have been split in different figures for each

IBB: IBB1 from IF1-IF3, IBB2 from I4-IF6 and IBB3 from IF7-IF10. The runtime for IF1-3 are greater than the other IFs, this is due to the fact that the IBB files have more records and operations (see Table 6 for more information).

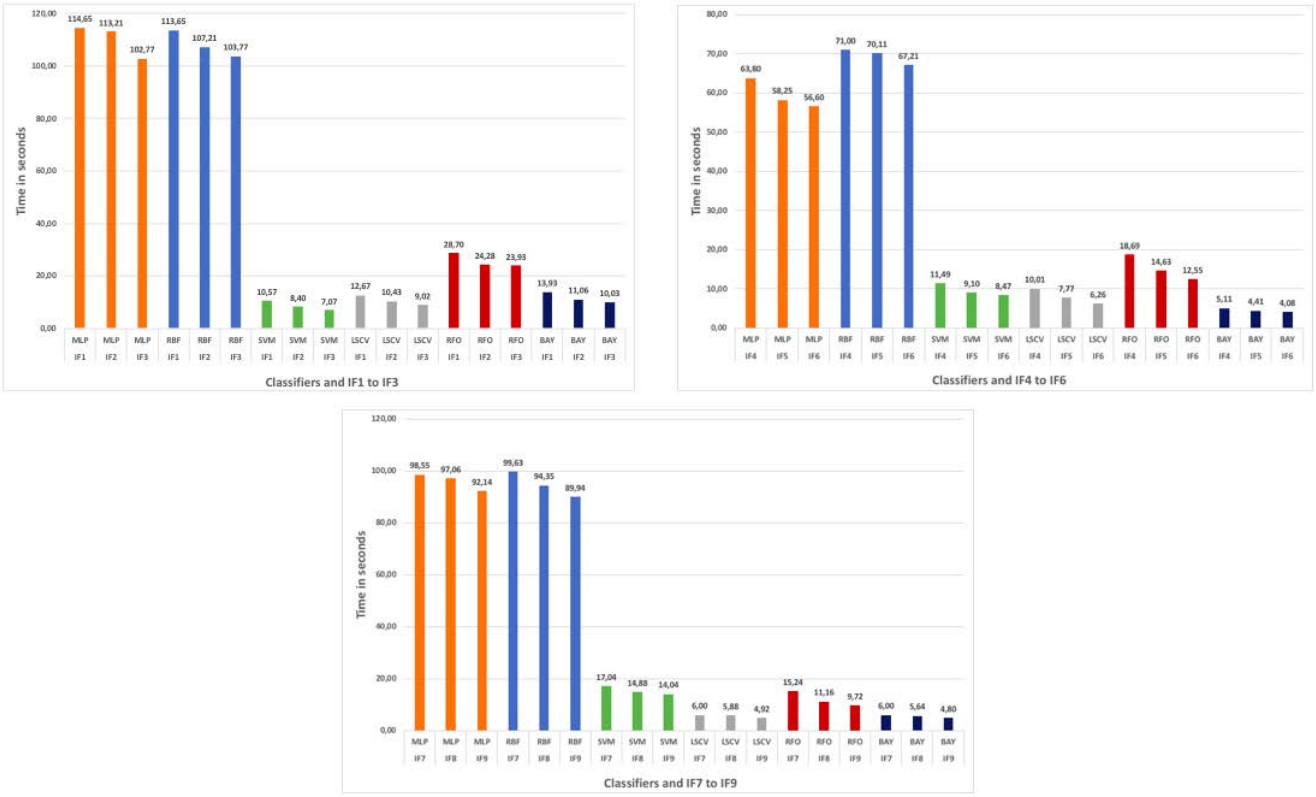

Figure 4. Evolution of training time in individual scenario 1 for all IBBs (IBB1 left-top, IBB2 right-top, IBB3 center-bottom)

In summary, in all the cases, BAY classifier obtains a great performance and a reduced runtime. MLP and RFO classifiers, although get a moderate execution time in the train-test process, they are far from LSVC or even BAY models. Finally, taking into account the results depicted in Table 7 and in Figure 4, the best classifiers are BAY (first position) and RFO (second position).

Individual Scenario 2

In the individual scenario 2, depicted in Table 8, the BAY model showed the best performance (in accuracy) and best fit for AIC and MDL statistical indicators. This allows the researcher to choose the classifier and be certain of its ability to detect future matches between records and operations. In addition, the other two topologies with better accuracy, RFO and MLP networks, have obtained good performance with the AIC and MLP indicators.

Furthermore, as is shown Table 8, for each IBB the best classifiers are the following: IBB1-RFO with IF10 (99,88%), IBB2-BAY with IF13 (99,56%) and IBB3-BAY with IF18 (99,92%). In summary, for scenario 2, the best classifier is BAY with a top accuracy of 99,92% and an average of 99,57% for all the experiments. The result on bold indicates the best result for each classifier between all the experiments (IFs and IBBs).

Table 8. **Individual scenario 2. ML classifier analysis performance (% correct classification) and AIC and MDL indicators. 20 runs×10 cv method**

Input file	Results	RBF	SVM	MLP	RFO	LSVC	BAY
IF10/ IBB1	Accuracy	97,88%	97,23%	**99,48%**	**99,88%**	97,79%	99,63%
	AIC	-8858,37	-9994,37	-12677,37	-19771,37	-9727,37	-18265,37
	MDL	-10134,65	-10161,65	-11161,65	-19744,65	-8100,65	-18135,65
IF11/ IBB1	Accuracy	97,70%	97,57%	98,70%	99,82%	97,69%	99,31%
	AIC	2486,19	-3356,815	-2201,81	-11170,82	-1060,81	-9669,81

	MDL	-6259,19	-8134,19	-9527,19	-11479,19	-8056,19	-9546,19
IF12/ IBB1	Accuracy	97,43%	97,73%	98,43%	99,62%	97,21%	99,32%
	AIC	-1731,71	-2401,28	-3001,71	-10834,71	-130,28	-8917,71
	MDL	-2323,93	-1769,07	-4923,92	-10999,93	-45,07	-9072,92
IF13/ IBB2	Accuracy	95,56%	99,10%	98,95%	98,52%	99,10%	99,56%
	AIC	-21908,36	-20908,36	-29747,36	-37436,36	-26929,36	-37086,36
	MDL	-22569,99	-20177,98	-31490,98	-38486,98	-28376,98	-38846,99
IF14/ IBB2	Accuracy	95,31%	99,19%	98,70%	99,03%	**99,17%**	99,31%
	AIC	-13406,93	-15452,92	-15950,92	-24169,92	-12837,92	-21528,92
	MDL	-11070,82	-14763,81	-13915,81	-23215,81	-14040,81	-22048,81
IF15/ IBB2	Accuracy	95,23%	**99,35%**	98,96%	99,02%	98,92%	99,29%
	AIC	-9428,81	-11885,81	-11634,81	-20360,81	-8100,81	-18272,81
	MDL	28467,65	28559,65	25871,65	15743,65	27466,65	18664,65
IF16/ IBB3	Accuracy	98,29%	99,02%	99,29%	99,89%	98,60%	99,91%
	AIC	-2311,65	-5568,64	-6814,64	-15728,64	-6637,64	-15067,64
	MDL	-5941,22	-10376,22	-10245,22	-18253,223	-8743,223	-16004,22
IF17/ IBB3	Accuracy	98,18%	99,10%	99,18%	**99,88%**	98,64%	99,89%
	AIC	-9093,41	-8706,40	-9382,40	-9929,40	-8181,40	-9248,40
	MDL	-8454,46	-9250,45	-9679,46	-9404,46	-9763,46	-9370,45
IF18/ IBB3	Accuracy	**98,34%**	99,01%	99,34%	99,85%	98,87%	**99,92%**
	AIC	1608,57	2331,56	-1373,43	-9837,43	-779,431	-7903,43
	MDL	3089,36	-1818,63	-1738,63	-9106,63	-2063,63	-7577,63
Average	**Accuracy**	*97,10%*	*98,59%*	*99,00%*	*99,50%*	*98,44%*	***99,57%***

Moreover, the evolution of the train-test process in seconds for each IBB is depicted in Figure 5. These figures show the mean in seconds for all experiments per classifier and IF (20 runs). The results obtained have been split in different figures for each IBB: IBB1 from IF10-IF12, IBB2 from IF13-IF15 and IBB3 from IF16-IF18. The runtime for IF10-12 are greater than the other IFs, this is due to the fact that the IBB files have more records and operations (see Table 6 for more information).

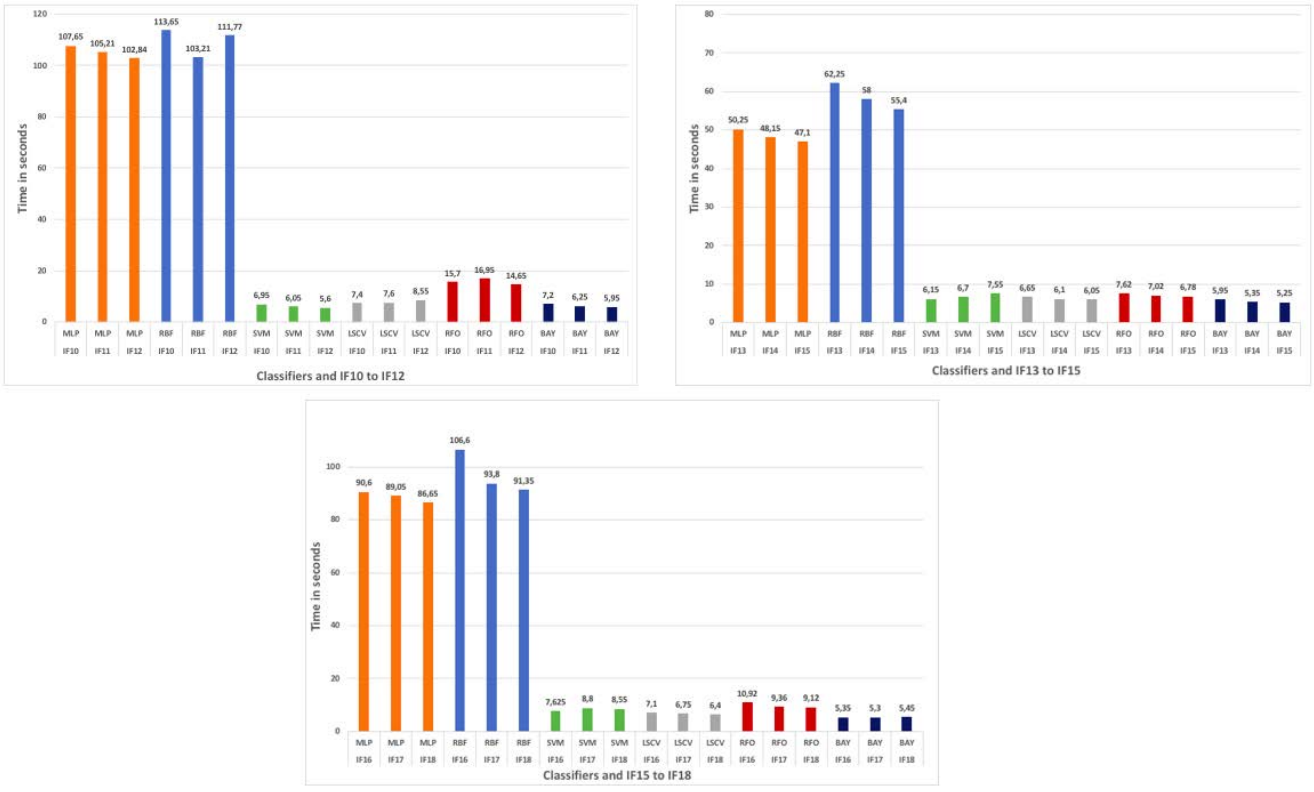

Figure 5. Evolution of training time in individual scenario 2 for all IBBs (IBB1 left-top, IBB2 right-top, IBB3 center-bottom)

Again, in all the experiments, and in the same way that happens in scenario 1, BAY classifier obtains a great performance and a reduced runtime. MLP and RFO classifiers, although get a moderate execution time in the train-test process, they are far from

LSVC or even BAY models. Finally, taking into account the results depicted in Table 8 and in Figure 5, the best classifiers are BAY (first position) and RFO (second position).

Pool Scenario

As it has been explained in section 3.1, the pool scenario is the one with least complexity in the experimentation process because there are less IFs. The idea of this scenario is to analyse if the data coming from different IBB can be fused under the same information structure (IF19, IF 20 and IF 21).

In the pool scenario, depicted in Table 9, again the BAY model for IF19 showed the best performance (99,58% in accuracy) and best fit for AIC and MDL statistical indicators. Moreover, the BAY model also has a great performance in average (99,39% in accuracy) and also for AIC and MDL indicators. This allows the researcher to choose the BAY classifier and be certain of its ability to detect future matches between records and operations. In addition, the other two topologies with better accuracy, MLP and RFO networks, have obtained good performance with the AIC and MLP indicators for all the IF of this scenario. The result on bold indicates the best result for each classifier between all the experiments (IFs and IBBs)

Table 9. **Pool scenario. ML classifier analysis performance (% correct classification) and AIC and MDL indicators. 20 runs×10 cv method**

Input file	Results	RBF	SVM	MLP	RFO	LSVC	BAY
IF19 / All IBBs	Accuracy	98,76%	**98,21%**	**99,32%**	99,18%	**98,13%**	**99,58%**
	AIC	-6412,49	29,86	-5771,98	-5952,87	-5213,87	-6981,52
	MDL	-384,56	127,88	-3298,24	-379,48	-76,42	-4384,56
IF20/ All IBBs	Accuracy	**99,11%**	97,90%	99,13%	99,12%	97,88%	99,33%
	AIC	-9251,14	-64,13	-6608,23	-6035,87	-5306,87	-6178,23
	MDL	-658,88	28,88	-3769,43	-517,48	-190,31	-3732,56
IF21 / All IBBs	Accuracy	98,8%	97,34%	98,91%	**99,21%**	97,41%	99,26%
	AIC	-9313,00	-135,13	-6874,72	-6123,87	-5351,71	-6167,67
	MDL	-1171,79	-86,12	-4022,35	-588,48	-293,42	-3424,92
Average	**Accuracy**	*97,32%*	*97,82%*	*99,12%*	*99,17%*	*97,81%*	***99,39%***

Finally, the evolution of the train-test process in seconds is depicted in Figure 6. Figure 6 shows the mean in seconds for all experiments per classifier and IF (20 runs). Again, the BAY model obtains a great performance. This point is important due to the fact that if new IBB and IF are going to be incorporated to the framework, the Stage 2 (machine learning) must be executed again. Hence, if it is possible, it is crucial to choose the ML classifier with the lowest computational time for the train-test process. In this case, MLP and RFO networks, although get a moderate execution time in the train-test process, they are far from LSVC or even BAY models.

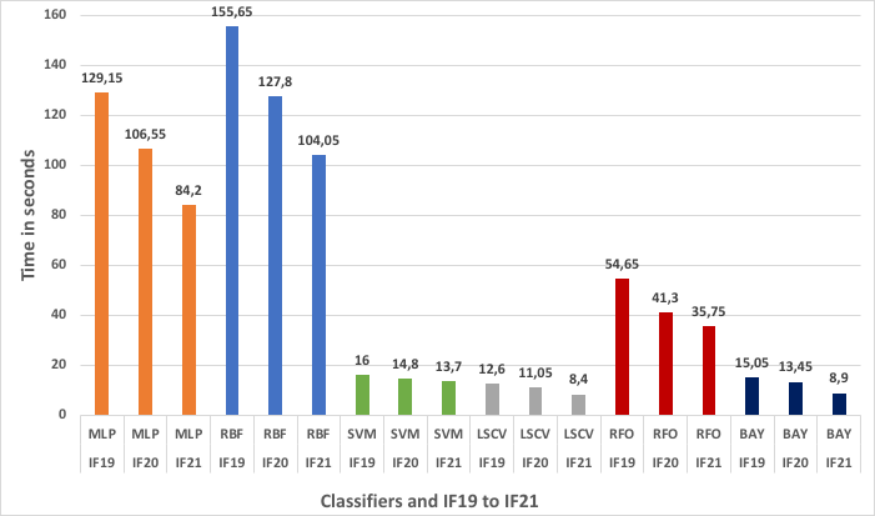

Figure 6. Evolution of training time in pool scenario for all IBBs

Statistical validation

The performance parameters calculated in each ANN for the train-test scenario, set out in Section 3.3.2, enable the functioning and performance of an ANN to be determined. However, they do not allow for the determination of which is the better choice when various alternative models with similar results appear. 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.

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 ANN, the research, development and simulation carried out by the researchers have included the use of different statistical methods for the evaluation of the results [13,18,36]. Following this trend, this research assesses and compares the different experiments proposed by statistical analysis based on the estimator t-test and its variants.

This research includes a series of estimators and statistical tests of contrast, both parametric and non-parametric, to compare the various alternatives studied in each of the phases of the improvement framework. The first serves to make assumptions on the parameters that define the population, for example normal populations and tests on the mean or standard deviation, while the latter do not refer to population parameters and are typically applied when there is no known distribution of the population or its distribution is not normal. In addition, non-parametric tests are inherently robust, that is they work relatively well even if the requirements are not met. The list of alternatives, when population is bigger than 2 (as in this research), include analysis of variance (ANOVA), multiple range test (MRT) and the Kruskal–Wallis test.

ANOVA test checks whether there is any difference between the means and the MRT test indicates the means that are significantly different from each other. If the assumptions for applying the ANOVA method are not met, non-parametric alternatives that do not use the mean as a criterion for contrast could be used. Moreover, if there are outliers or differences in the variances, the Kruskal–Wallis test will be used, comparing medians rather than means. In this case the different graphs help to judge the practical significance of the results, and enable the search for possible violations of the assumptions underlying the ANOVA.

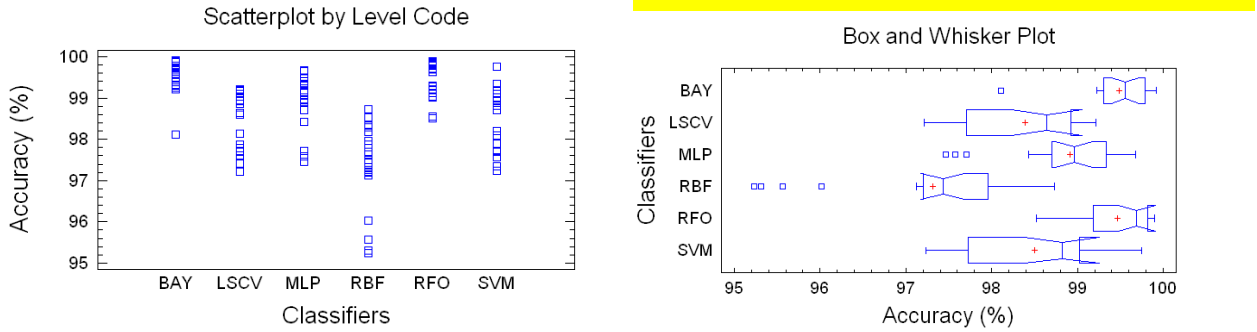

Figure 7. **Scatter and box plots using accuracy (%) of the classifiers (train-test scenario)**

Therefore, as first step, a visual analysis of the results obtained is show. For this analysis, in the train-test scenario, the accuracy (%) obtained by the classifiers in all the experiments has been used. Figure 7 includes the scatter and box plots associated with the results. The first describes the behaviour of all the samples obtained for each classifier through a cloud of points. The usefulness of the second is that it offers, by simple visual inspection, a rough idea of the central tendency (through the median), dispersion (through the interquartile) of the symmetry of the distribution (through the symmetry of the graph) and possible outliers in each classifier. The rectangular part of the plot extends from the lower quartile to the upper quartile, covering the centre half of each sample. The centre lines within each box show the location of the sample medians. The plus signs indicate the location of the sample means. The whiskers extend from the box to the minimum and maximum values in each sample, except for any outside or far outside points, which will be plotted separately. In this case, there are outside points and far outside points. 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.

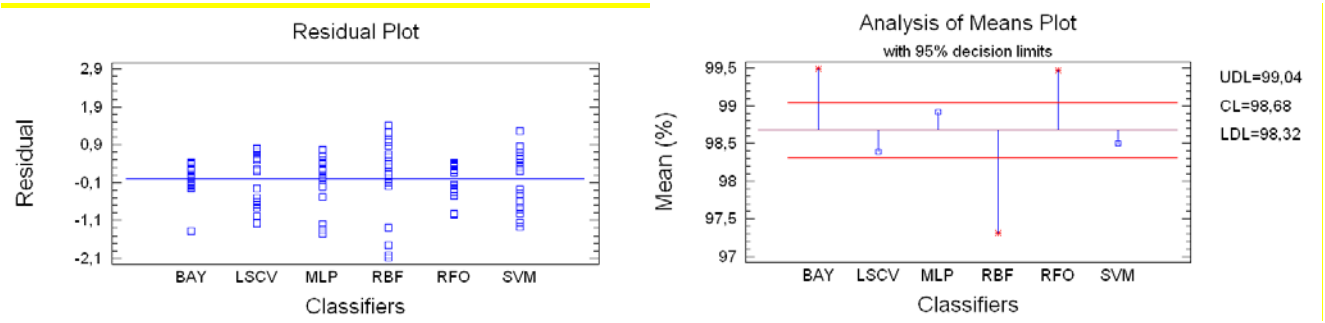

Figure 8**. Residual and analysis of means plot (train-test scenario)**

Figure 8 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 RBF). This second plot shows the mean of each of the six samples. Also shown is the grand mean and the 95% decision limits. The samples which fall outside the decision limits, except LSCV, MLP and SVM, 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 [18]. The three statistics displayed in Table 10 test the null hypothesis that the standard deviations of the results within each of the six 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 10**. Variance check (train-test scenario)**

Contrast	**Value**	**p-value**
Cochran's C test	0,3653	0,0016514

Bartlett's test	1,2055	0,0005219
Levene's test	2,7803	0,0205915

Bartlett's test is a general contrast, which means that the populations are normal and that they are independent samples, of equal size or not. The Cochran contrast is useful when the sample variance is much greater than the rest or when the number of alternatives to analyse is higher than 12. The Levene test does not require normality in the distributions. In the three cases, if the resulting p-value is less than the critical value (typically 0.05), it is unlikely that the differences found in the variations of the sample have been produced based on random sampling. Thus, the null hypothesis of equal variances is rejected and the conclusion is that there is a difference between the variances of the population.

Once it has been determined that there is a statistically significant difference between the variances, the Kruskal–Wallis test is the most suitable method for comparing populations whose distributions are not normal [27] . This is a non-parametric method, derived from the F-test to check the equality of medians of a group of populations. The reason for using the median is that it is robust, that is less sensitive to outliers, while the mean is more sensitive. If the distribution is normal, mean and median coincide but if there is a discrepancy between the two, the median is preferable. Thus, in the absence of normality, the mean contrasts are not relevant, and those on the median are preferable.

The null hypothesis of the Kruskal–Wallis test is:

- H_0: The t medians are all equal.
- H_1: At least one of the medians is different.

Table 11. **Kruskal Wallis test for train-test scenario (accuracy %)**

Contrast	**Average rank (%)**
BAY	99,381
RFO	96,023
MLP	68,619
SVM	51,761
LSCV	45,309
RBF	19,904
Statistical = 74,6647	P-value = 0,0

In Table 11 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 six alternatives. The data of all levels are first combined and sorted from lowest to highest and then the average rank for the data in each level is calculated. 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 7 the width of the notches indicates the approximate confidence interval of 95.0

As is depicted in Table 11, the BAY and RFO classifiers present a homogeneous behavior and the distributions of the results are significantly different from all the rest. Moreover, the average percentage of accuracy of BAY is higher than RFO in 0,18% and its average rank at the end of the Kruskal-Wallis test is higher in 3.3572%.

Therefore, it can be concluded that considering all the results (accuracy %) obtained in the experiments of the train-test scenario BAY classifier obtains a great performance and a reduced runtime.

Analysis of variables' influence

One of the factors which most highly influences the performance of a ML classifier is the complexity of its architecture or topology. In order to overcome this issue, there are various alternatives, such as the pruning of the network or elimination of the weights associated with the connections. In this research, two different wide-spread strategies were chosen to be applied in a combined way. One is oriented to studying the relationship between the training and generalization error in terms of the complexity of the model, the SWM method, and the second is aimed at estimating the sensitivity of the classifier for training and test values, the IVC and SAM techniques. SAM and IVC techniques give as output the number of variables with values bigger than 0,5 (↑) or lower than 0,2 (↓).

The results obtained are shown in Table 12, Table 13 and Table 14. The study for this step has been divided in different scenarios (individual 1, individual 2 and pool) and for each table details about IF, IBB and number of variables have been included. Columns SAM, IVC and SWM depict the results obtained for each technique.

As summary and considering the results obtained for all the experiments and scenarios for SAM, IVC and SWM techniques, the most important V_{inp} variables are the following: COPT (counterpart), ACCN (account number), SUMCHAR (sum all charges). These variables should always appear as a component of the input vector to ensure higher accuracy. Nevertheless, when ACCN variable is not available due to pre-processing stage (e.g. IF7-9, IF16-18 and IF21), the results obtained has correct performance as well.

The least important V_{inp} variables are the following: OPPT (type of interest), BRAR (base rate), DRRC (diff rate code), DRAR (differential rate). Therefore, their non-inclusion will simplify the final architecture as well as reduce the network training time.

Finally, the pruning process provides valuable information not only for the ML classifier area, but also for different stakeholders. For example, it provides relevant variables related to bank operations and records that should be taken into account in IBBs and for CERE information. Accordingly, these results of the SAM, IVC and SWM techniques are forwarded to the BI&A component. This information helps in determining the influence of input variables in the output prediction of the ANN, allowing dependences and redundancy to be detected in the input data and even for information to be structured more properly.

Table 12. **Individual scenario 1. Summary of variables' influence and best input vector**

File / IBB / Variables	SAM		IVC		SWM
IF1 / IBB1/ 55 variables	↑	6 variables	↑	7 variables	21 variables
	↓	9 variables	↓	8 variables	
IF2/ IBB1 / 38 variables	↑	5 variables	↑	6 variables	18 variables
	↓	8 variables	↓	5 variables	
IF3 / IBB1/ 36 variables	↑	3 variables	↑	4 variables	17 variables
	↓	4 variables	↓	5 variables	
IF4/ IBB2 / 55 variables	↑	6 variables	↑	8 variables	25 variables
	↓	8 variables	↓	9 variables	
IF5 / IBB2 / 35 variables	↑	5 variables	↑	6 variables	22 variables
	↓	8 variables	↓	7 variables	
IF6/ IBB2 / 33 variables	↑	3 variables	↑	5 variables	21 variables
	↓	4 variables	↓	6 variables	
IF7/ IBB3 / 55 variables	↑	7 variables	↑	7 variables	18 variables
	↓	11 variables	↓	10 variables	
IF8/ IBB3/ 41 variables	↑	5 variables	↑	6 variables	16 variables
	↓	8 variables	↓	9 variables	
IF9/ IBB3 / 39 variables	↑	3 variables	↑	4 variables	15 variables
	↙	7 variables	↓	7 variables	

Table 13. **Individual scenario 2 (common variables). Summary of variables' influence and best input vector**

File / IBB / Variables	SAM		IVC		SWM
IF10 / IBB1/ 52 variables	↑	6 variables	↑	7 variables	20 variables
	↓	8 variables	↓	7 variables	
IF11/ IBB1 / 35 variables	↑	5 variables	↑	5 variables	17 variables
	↓	7 variables	↓	4 variables	
IF12 / IBB1/ 33 variables	↑	3 variables	↑	4 variables	17 variables
	↓	4 variables	↓	5 variables	
IF13 / IBB2 / 52 variables	↑	5 variables	↑	8 variables	23 variables
	↓	8 variables	↓	8 variables	
IF14 / IBB2 / 35 variables	↑	3 variables	↑	4 variables	20 variables
	↓	8 variables	↓	7 variables	
IF15/ IBB2 / 33 variables	↑	3 variables	↑	5 variables	17 variables

	↓	4 variables	↓	6 variables	
IF16 / IBB3 / 52 variables	↑	7 variables	↑	7 variables	17variables
	↓	9 variables	↓	10 variables	
IF17 / IBB3/ 35 variables	↑	6 variables	↑	6 variables	14 variables
	↓	7 variables	↓	9 variables	
IF18 / IBB3 / 33 variables	↑	5 variables	↑	4 variables	14 variables
	↓	5 variables	↓	6 variables	

Table 14. **Pool scenario. Summary of variables' influence and best input vector**

File / IBB / Variables	SAM		IVC		SWM
IF19 / all / 52 variables	↑	6 variables	↑	7 variables	19 variables
	↓	9 variables	↓	8 variables	
IF20 / all / 35 variables	↑	5 variables	↑	6 variables	16 variables
	↓	8 variables	↓	5 variables	
IF21 / all / 33 variables	↑	3 variables	↑	4 variables	15 variables
	↓	4 variables	↓	5 variables	

4.3.2 Production step

Once the previous steps have finished, in the production step, records from CERE and branches are received on demand and encapsulated in data buckets. In the data bucket, several pairs of records are incorporated from the data lake in order to determine if the records match or not. So, in this case, a pair represents a record from CERE and a candidate record from a branch.

In the production step, records from CERE and branches are received in runtime. In this case, they represent a record from CERE and a candidate record from a branch. The records received are pre-processed by merging their attributes and removing those established during the training phase. Merged records are the input of the classifier in order to determine whether they represent the same operation. Thus, the ML classifier produce a true or false value. The process is repeated for all records to be analysed in order to determine the related records from each system.

Finally, the Bayes classifier (BAY) has been selected due to the fact that it has the best performance in the train-test process. However, the production scenario has been tested with all of the tested classifiers in order to determine their behaviour in the production scenario.

In order to evaluate the results, according to the analysis of Sokolova & Lapalme [43] and the guidelines of Christen & Goiser [9] it is recommended that the quality be measured using the precision-recall or F-measure graphs rather than single numerical values [9,43]. Therefore, a Precision, Recall and F1 study has been performed, as shown in Table 15 and Table 16 and Table 17. Results obtained show optimal results for precision and recall for the RFO and BAY classifiers.

Table 15. **Individual scenario 1. Precision, Recall & F1 measure.**

Input file	Results	RBF	SVM	MLP	RFO	LSVC	BAY
IF1	Precision	0,96	0,982	0,992	**0,999**	0,982	**0,999**
	Recall	0,971	0,982	0,998	**0,998**	0,985	**0,998**
	F1	0,971	0,981	0,995	**0,998**	0,98	**0,998**
IF2	Precision	0,96	0,98	0,983	**0,998**	0,98	0,995
	Recall	0,971	0,98	0,991	**0,998**	0,98	0,998
	F1	0,971	0,98	0,987	**0,997**	0,98	0,997
IF3	Precision	0,96	0,971	0,99	0,998	0,98	**0,999**
	Recall	0,971	0,975	0,988	0,998	0,98	**0,998**
	F1	0,971	0,975	0,994	0,997	0,98	**0,998**
IF4	Precision	0,96	0,99	0,978	**0,993**	0,99	0,991
	Recall	0,993	0,99	0,976	**0,993**	0,99	0,993
	F1	0,976	0,99	0,976	**0,993**	0,99	0,991
IF5	Precision	0,88	0,878	0,973	**0,983**	0,988	0,98
	Recall	0,998	0,98	0,975	**0,988**	0,985	0,979
	F1	0,935	0,98	0,975	**0,985**	0,985	0,935

IF6	Precision	0,945	0,98	0,975	0,993	0,982	**0,992**
	Recall	0,998	0,978	0,976	0,993	0,98	**0,998**
	F1	0,97	0,978	0,976	0,996	0,98	**0,998**
IF7	Precision	0,988	0,987	0,995	**0,998**	0,988	**0,998**
	Recall	0,987	0,987	0,998	**0,998**	0,987	**0,998**
	F1	0,985	0,985	0,997	**0,998**	0,985	**0,998**
IF8	Precision	0,982	0,99	0,993	**0,997**	0,99	0,993
	Recall	0,985	0,99	0,997	**0,997**	0,99	0,991
	F1	0,985	0,99	0,995	**0,998**	0,99	0,992
IF9	Precision	0,987	0,99	0,995	**0,998**	0,99	0,998
	Recall	0,988	0,99	0,998	**0,998**	0,99	0,998
	F1	0,985	0,99	0,998	**0,998**	0,99	0,997

In the individual scenario 1, depicted in Table 15, the best Precision and Recall values corresponds with RFO and BAY classifier. The average F1 value is 0,995 for RFO in contrast to the 0,988 value for BAY. As mentioned, due to the best performance showed in the tests, the BAY classifier is the best option, because the differences in the results are minimal with respect to the difference of performance.

Table 16. **Individual scenario 2. Precision, Recall & F1 measure.**

Input file	Results	RBF	SVM	MLP	RFO	LSVC	BAY
IF10	Precision	0,953	0,975	0,996	**0,998**	0,997	0,996
	Recall	0,955	0,975	0,994	**0,999**	0,997	0,996
	F1	0,945	0,975	0,995	**0,999**	0,997	0,996
IF11	Precision	0,953	0,976	0,994	**0,998**	0,997	0,992
	Recall	0,955	0,976	0,98	**0,999**	0,997	0,994
	F1	0,944	0,976	0,987	**0,999**	0,997	0,993
IF12	Precision	0,953	0,978	0,991	**0,999**	0,996	0,992
	Recall	0,952	0,978	0,978	**0,999**	0,996	0,994
	F1	0,943	0,978	0,984	**0,999**	0,996	0,993
IF13	Precision	0,918	0,991	0,99	**0,992**	0,990	0,993
	Recall	0,956	0,991	0,99	**0,993**	0,990	0,992
	F1	0,957	0,991	0,99	**0,994**	0,990	0,991
IF14	Precision	0,914	0,991	0,987	0,991	0,991	**0,992**
	Recall	0,953	0,991	0,987	0,991	0,991	**0,992**
	F1	0,955	0,991	0,987	0,99	0,991	**0,992**
IF15	Precision	0,913	0,993	0,99	0,99	0,989	**0,994**
	Recall	0,952	0,993	0,99	0,99	0,989	**0,994**
	F1	0,955	0,993	0,99	0,99	0,989	**0,994**
IF16	Precision	0,983	0,990	0,989	**0,992**	0,986	0,99
	Recall	0,984	0,990	0,987	**0,992**	0,986	0,99
	F1	0,994	0,990	0,993	**0,991**	0,986	0,989
IF17	Precision	0,983	0,990	0,987	**0,990**	0,986	0,99
	Recall	0,984	0,990	0,987	**0,990**	0,986	0,900
	F1	0,993	0,990	0,992	**0,990**	0,986	0,989
IF18	Precision	0,983	0,990	0,99	**0,990**	0,988	0,989
	Recall	0,993	0,990	0,987	**0,990**	0,988	0,989
	F1	0,993	0,990	0,993	**0,990**	0,988	0,989

In the individual scenario 2, as shown in Table 16, once again the RFO classifier shows the best results. It is remarkable that all classifier obtains excellent results, above 98%. The average F1 value is 0,993 for RFO in contrast to the **0,991** value for BAY.

Table 17. **Pool scenario. Precision, Recall & F1 measure.**

Input file	Results	RBF	SVM	MLP	RFO	LSVC	BAY
IF19	Precision	0,980	0,981	0,993	0,997	0,981	**0,998**
	Recall	0,980	0,981	0,997	0,997	0,981	**0,998**
	F1	0,980	0,981	0,995	0,997	0,981	**0,998**
IF20	Precision	0,970	0,970	0,995	**0,994**	0,970	0,990
	Recall	0,970	0,970	0,988	**0,996**	0,970	0,993
	F1	0,970	0,970	0,991	**0,995**	0,970	0,991
IF21	Precision	0,970	0,970	0,989	0,993	0,971	**0,995**

	Recall	0,970	0,970	0,979	0,994	0,971	**0,995**
	F1	0,970	0,970	0,984	0,994	0,971	**0,995**

Finally, in the pool scenario, as shown in Table 17 once again the RFO and BAY classifiers show the best results. The average F1 value is 0,995 for RFO in contrast to the 0,991 value for BAY.

Statistical validation

The performance parameters calculated in each ANN for the production scenario, set out in Section 3.3.2, enable the functioning and performance of an ANN to be determined. However, they do not allow for the determination of which is the better choice when various alternative models with similar results appear. 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.

Again, as first step, a visual analysis of the results obtained is show. For this analysis, in the train-test scenario, the F1 parameter obtained in the Precision, Recall & F1 test has been used. Figure 9 includes the scatter and box plots associated with the results for the production scenario. In this case, there are only one outlier point. 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.

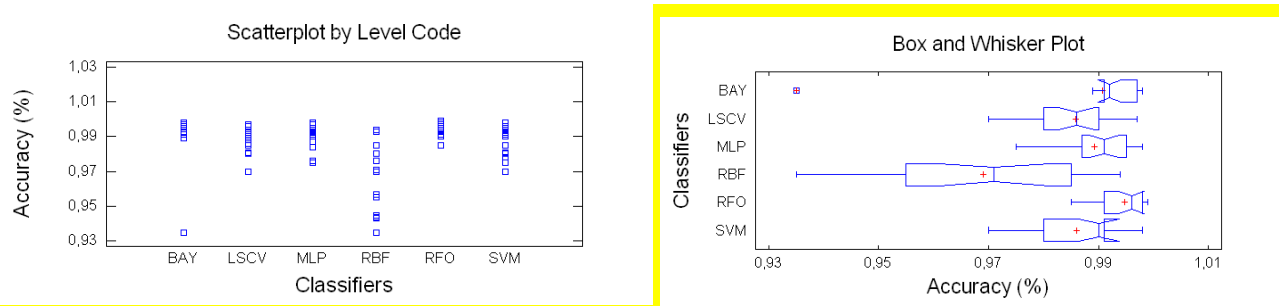

Figure 9. **Scatter and box plots using accuracy (%) of the classifiers (production scenario)**

Figure 10 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 RBF because there is one outlier). This second plot shows the mean of each of the six samples. Also shown is the grand mean and the 95% decision limits. The samples which fall outside the decision limits, except RFO, are significantly different from the grand mean.

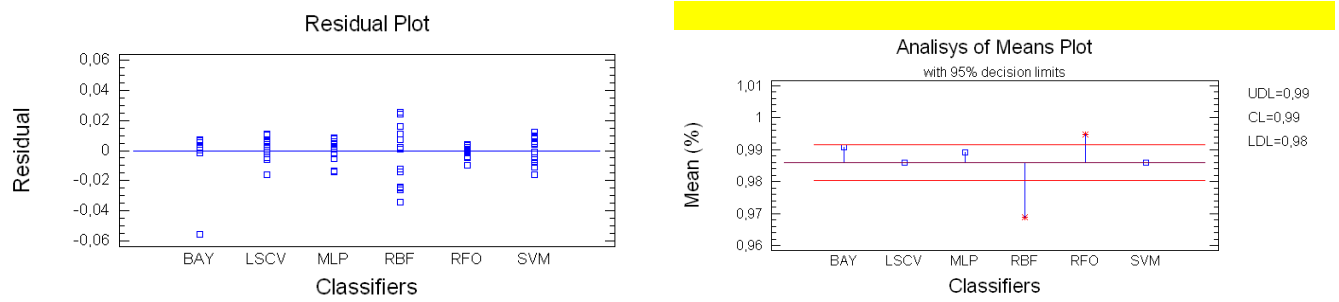

Figure 10**. Residual and analysis of means plot (production scenario)**

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 18 test the null hypothesis that the standard deviations of the results within each of the six 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 18**. Variance check (production scenario)**

Contrast	Value	p-value
Cochran's C test	0,464166	0,00000265
Bartlett's test	1,53614	1,07955E-9
Levene's test	4,72092	0,00056832

In Table 19 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 six 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 9 the width of the notches indicates the approximate confidence interval of 95.0 As is depicted in Table 19, the BAY and RFO classifiers present a homogeneous behaviour and the distributions of the results are significantly different from all the rest. Moreover, the average value F1 of RFO is higher than BAY in 0,004%.

Table 19. **Kruskal Wallis test for production scenario (F1 parameter)**

Contrast	Average rank
RFO	96,595
BAY	82,619
MLP	68,000
LSCV	52,119
SVM	55,190
RBF	26,476
Statistical = 48,2269	P-value = 3,19262E-9

Therefore, it can be concluded that considering all the results (F1 parameter) obtained in the experiments of the production scenario BAY and RFO classifiers obtain a great performance and a reduced runtime.

Table 20. Comparison with other approaches.

Work	Best F1 measure
Jurek et al. (2017) [24]	0,96
Kim & Giles (2016) [25]	0,9744
Best RFO	0,998
Best BAY	0,998

Table 20 shows the best results of the proposed framework with the best results of other approaches. Jurek et al. [24] apply ensemble learning classifiers over four datasets widely used (but not in the financial domain) and obtain a F1 measure of 0,96 in the best scenario. Kim & Giles [25] apply random forest for linking entities in a financial dataset, and obtain a F1 measure of 0,9744 in the best scenario. Despite the datasets are not comparable, results show that the proposed framework obtain very promising results. The proposed framework achieves an F1 measure of 0,998 in both RFO and BAY. The results of Kim & Giles are based on different approach because they match records by text and the proposed framework matches operations based on numerical attributes, but the comparison shows that the results obtained by the proposed framework are in the line of financial approaches.

4.4 Post-processing and BI&A

Once the machine learning stage has been executed, the BI&A step is evaluated. In this case, the BI&A component will receive information from the framework related with the matching operations detected but also will be fed with information about the influence of each of the inputs in the output which the classifier obtains, the relationships and the dependences between variables. The techniques for this matter have been explained in Stage 3 of the framework (see section 3.3.3 for more details).

For each pair of records classified as positive by the ANN, the framework will generate a JSON line with the data related to the operation. This information will be shown to stakeholders (bank staff and even managers) in order to validate the result.

Non-matched records are discarded (negative cases) because the number of negative combinations is too high to be checked by a human, taking into account as well the precision achieved by the framework in the best configuration analysed. Managers are provided with the matched records in order to validate whether they are correctly matched or not: the main point is that the number of matched records provided by the framework is processable by humans. Even more, the system can be configured with a threshold in order to validate only records with fewer than a given value (i.e. records with a 99% of probability can be directly approved and only show managers records with a given error probability). The human intervention can be avoided but the framework allows this option in order to manage cases in which the accuracy is critical and the responsibility cannot be in hands of the machine.

Thus, all attributes ($Attribute_i$) whose records ($Record_1$ and $Record_2$) have been matched in the classifier are presented in a JSON format in order to be sent to the BI&A component. Non-relevant attributes are discarded in the preprocess stage and are not shown. In addition, each of the attributes is shown along with its degree of influence in the final decision. This fact provides the user with additional information in order to analyse the results and for further decision making. The JSON data set is composed by three list of attributes with the information relative to the influence degree of each variable for each sensibility technique (SAM, IVC and SWM), and a list of matched operations with their corresponding records and attributes.

4.5 Discussion

As shown in previous sections, the individual scenarios 1 and 2 show very promising results. In the individual scenario 1, the system achieved a best performance with 99,90% of accuracy for BAY classifier and IBB1-IF2. The rest of the IBBs present results near to this value. It implies that the proposed models fit with the problem and are able to classify the provided operations. In the individual scenario 2, in which only common attributes of the records for all IBBs are considered, results are quite similar. In this case, the system achieved a performance of 99,92% with BAY classifier and IBB3-IF18.

The third scenario (pool), which uses data combined from the three IBBs, present also stable results. The best performance obtained is 99,58% for BAY classifier and IF19, similar to the obtained in the rest of scenarios. This fact shows that the system framework is able to scale when new IBBs are added to the model.

Looking at the sensibility analysis, the relevant attributes are similar for both scenarios for all branches. It explains the similar result as well as it shows that the inclusion of new branches will predictably provide the same results with these attributes. Thus, the model could scale to more IBB that use these attributes in their records.

Looking at the runtime, the less complex files are those that require less runtime. Thus, when the number of attributes is reduced the runtime is also reduced. It is remarkable that the omission of relevant attributes like the account number has not reduced significantly the quality of the results: it implies that the model learns the characteristic of the operation far away than elements that could help to identify the operations. At this point, it is important to consider that runtime is only comparable among similar IFs, because different IFs have different size and structure.

All results are homogeneous for all the classifiers, obtaining good results for all of them. However, taking into account the performance values of each classifier, the BAY classifier shows promising results combining accuracy and performance. In summary, taking into account the results obtained in the train-test stage (in terms of accuracy and MDL-AIC indicators) and the runtime of each experiment, the classifier chosen for the production step is the BAY. Running time is an important issue considering the future scalability of the framework for including new IBBs or new operation types.

In the production scenario, RFO and BAY show the best Precision & Recall values, but all classifiers show stable results with regard to the obtained in the train & test scenario. It indicates that the models of the train & test process can be extrapolated to the production step with good results, as shown in the result tables.

The quality process (MDL-AIC indicators and Kruskal-Wallis test) performed in both scenarios, train-test and production, ensures the validity of the results from a statistical standpoint, thereby reducing the appearance of experimental errors or the appearance of possible randomness. This process has demonstrated that the results obtained with the RFO and BAY classifiers, as well as being higher on average, are significantly different and show no homogeneity with other classifiers. Moreover, considering the running time of the classifiers, the BAY is the best choice, so this allows the researchers to incline towards this classifier with no doubt about its fitness for this problem.

5 Conclusions and Future Lines

As mentioned before, the main motivation of this research was the necessity of great bank branches to analyse the huge amount of operation records generated in their worldwide activities, considering that the same operation can be registered several times by different systems using different attributes. In their daily business, bank branches register their operations with several systems in order to share information with the other branches and to have a central repository of records. In this way, the information can be analysed and processed according to different requirements. In the problem tackled in this research, some systems are local, related with International Bank Branches or IBB, and other systems are related with the central repository or CERE. These different systems record the same operation with different structure and even different information. In addition, the recording process of the operations might not be simultaneous. For this reason, the same operation registered several times in several different systems produces inconsistencies in the data. These inconsistencies make the work of matching one operation among all the recording systems difficult. An individual could have knowledge about the matching criteria, but the number of operations processed in a world-wide bank, as well as the heterogeneity of the data sources (different branches, different software systems…) makes the matching process impossible for a single human being in a reasonable amount of time. For this reason, such work needs to be automatized.

In this scenario, it is important for the bank to trace an operation among the different systems in which it could be registered. As such, the aim of this research was to define a framework to help with this problem, based on machine learning techniques in a big data environment, moving from unstructured to structured information, and for the automatic detection of relationships between banking operations. The output of the framework feeds the business intelligence analytics component in order to establish relationships and make comparisons between variables of the bank's daily business. Knowledge about these dependences and relationships is analysed by bank managers in order to simplify the banking operation records process storage or even to structure this data for better processing. Also, the conclusions and feedback obtained can be used to reduce resource consumption, e.g., storage space, computing time, etc.

The data pre-processing allows operation attributes not relevant for the classifiers to be discarded, as well as removing inconsistent data from the different data sources: for example, some attributes are used or not depending on the branch; even in the same branch some attributes could be used or not. All these attributes are identified and filtered in order to take into account only the attributes which are used consistently among the branches. The introduction of a machine learning stage in the framework has allowed us to compare different ML classifier configurations, optimizing them and selecting the most accurate one: in this case the BAY. The results obtained show 99.58% accuracy in predicting the matching operations, which is a high indicator of success for estimations. Moreover, the results obtained indicate that the different ML classifiers learn correctly. These results lead us to believe that the proposed framework can be a valuable tool for managers, banks and the different stakeholders involved in this process. Results obtained in the production step show stable behaviour regarding the train & test stage. All classifiers show promising Precision & Recall values in all scenarios and, considering the performance of the train & test scenario, BAY classifier is the selected one. Thanks to the post-processing stage, the most relevant variables for this prediction have been identified. These variables should always appear as a component of the input vector to ensure higher accuracy. Hence, this stage provides valuable information about the structure of records for a same operation, discarding superfluous or unnecessary variables for the matching process. Moreover, the post-processing stage automatizes, in real time, the prediction about matching operations, identifying the different records related to the same operation and discarding the records that do not match.

From an analytical point of view, even though the machine learning stage is a black box for manager and stakeholders, the output of the framework allows managers and stakeholders to have a clearer idea about the common structure among records from different IBBs and CERE systems.

Thus, this research provides a framework to manage a great number of pairs of operation records from different systems and provide a degree of similitude in order to determine whether they represent the same operation or not, as well as additional information about the relevance of the attributes of each operation. All information can be exploded in a BI&A stage in order to support the decision-making processes of the bank.

Finally, future lines of research can be summarized as:

- Test the scalability if the framework by including new IBBs and types of operations in order to determine the capability of the framework to be generalized for all possible alternatives. In this case, train process must be performed again for incorporating new knowledge about these new types of operations to the ML classifiers. In addition, the number

of records and branches is very high for an international bank (even taking into account operations of a single day). For this reason, future research will introduce new data from the data lake, incorporating new IBBs considering big data analytics, in order to deal with the problem in an efficient way.

- Add new ML classifiers in the framework. In this case, train process must be performed again for each new ML classifier. However, knowledge extracted from the current ML classifiers of the framework can be considered, so the configuration process of the classifiers can be reduced in terms of complexity and running time.

- Information about relevant variables can be extracted from the current framework, so complexity of pre-process and machine learning stages can be reduced in the future applying this knowledge. Relevant variables should always appear as a component of the input vector to ensure higher accuracy. Hence, this point provides valuable information about the structure of records for a same operation, discarding superfluous or unnecessary variables for the matching process.

Acknowledgments

This work was supported by the Research Program of the Ministry of Economy and Competitiveness - Government of Spain, (DeepEMR project TIN2017-87548-C2-1-R)

References

[1] Z. Bahmani, L. Bertossi, N. Vasiloglou, ERBlox: Combining matching dependencies with machine learning for entity resolution, Int. J. Approx. Reason. 83 (2017) 118–141.
[2] B. Bátiz-Lazo, D. Wood, An historical appraisal of information technology in commercial banking, Electron. Mark. 12 (2002) 192–205.
[3] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.
[4] R.D. Camino, R. State, L. Montero, P. Valtchev, Finding suspicious activities in financial transactions and distributed ledgers, in: IEEE Int. Conf. Data Min. Work. ICDMW, 2017.
[5] P. Carmona, F. Climent, A. Momparler, Predicting failure in the U.S. banking sector: An extreme gradient boosting approach, Int. Rev. Econ. Financ. (2018) 1–20.
[6] T. Chen, F. Chen, An intelligent pattern recognition model for supporting investment decisions in stock market, Inf. Sci. (Ny). 346–347 (2016) 261–274.
[7] E. Chong, C. Han, F.C. Park, Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies, Expert Syst. Appl. 83 (2017) 187–205.
[8] P. Christen, A survey of indexing techniques for scalable record linkage and deduplication, IEEE Trans. Knowl. Data Eng. (2012).
[9] P. Christen, K. Goiser, Quality and complexity measures for data linkage and deduplication, Stud. Comput. Intell. (2007).
[10] A. Dagade, M. Mali, De-duplication framework to reduce the record linkage problem, in: 2017 Int. Conf. Comput. Commun. Informatics, ICCCI 2017, 2017.
[11] C. Dirican, The Impacts of Robotics, Artificial Intelligence On Business and Economics, Procedia - Soc. Behav. Sci. 195 (2015) 564–573.
[12] A. Gandomi, M. Haider, Beyond the hype: Big data concepts, methods, and analytics, Int. J. Inf. Manage. 35 (2015) 137–144.
[13] A. García-Crespo, J.L. López-Cuadrado, R. Colomo-Palacios, I. González-Carrasco, B. Ruiz-Mezcua, Sem-Fit: A semantic based expert system to provide recommendations in the tourism domain, Expert Syst. Appl. 38 (2011).
[14] M. Gevrey, I. Dimopoulos, S. Lek, Review and comparison of methods to study the contribution of variables in artificial neural network models, Ecol. Modell. 160 (2003) 249–264.
[15] M. Gollapalli, Literature Review Of Attribute Level And Structure Level Data Linkage Techniques, ArXiv Prepr. ArXiv1510.02395. (2015).
[16] I. Gonzalez-Carrasco, R. Colomo-Palacios, J.L. Lopez-Cuadrado, Á. García-Crespo, B. Ruiz-Mezcua, PB-ADVISOR: A private banking multi-investment portfolio advisor, Inf. Sci. (Ny). 206 (2012) 63–82.
[17] I. Gonzalez-Carrasco, A. Garcia-Crespo, B. Ruiz-Mezcua, J.L. Lopez-Cuadrado, An optimization methodology for machine learning strategies and regression problems in ballistic impact scenarios, Appl. Intell. 36 (2012) 424–441.
[18] I. Gonzalez-Carrasco, A. Garcia-Crespo, B. Ruiz-Mezcua, J.L. Lopez-Cuadrado, R. Colomo-Palacios, Towards a framework for multiple artificial neural network topologies validation by means of statistics, Expert Syst. 31 (2014) 20–36.
[19] D. Hand, P. Christen, A note on using the F-measure for evaluating record linkage algorithms, Stat. Comput. (2018).
[20] J.B. Heaton, N.G. Polson, J.H. Witte, Deep Learning in Finance, CoRR. abs/1602.0 (2016).
[21] J.B. Heaton, N.G. Polson, J.H. Witte, Deep learning for finance: deep portfolios, Appl. Stoch. Model. Bus. Ind. 33 (2017) 3–12.
[22] J.-J. Hew, L.-Y. Leong, G.W.-H. Tan, K.-B. Ooi, V.-H. Lee, The age of mobile social commerce: An Artificial Neural Network analysis on its resistances, Technol. Forecast. Soc. Change. (2017).
[23] S. Hussain, A. Al Alili, A pruning approach to optimize synaptic connections and select relevant input parameters for neural network modelling of solar radiation, Appl. Soft Comput. 52 (2017) 898–908.
[24] A. Jurek, J. Hong, Y. Chi, W. Liu, A novel ensemble learning approach to unsupervised record linkage, Inf. Syst. (2017).
[25] K. Kim, C.L. Giles, Financial Entity Record Linkage with Random Forests, in: Proc. Second Int. Work. Data Sci. Macro-Modeling - DSMM'16, 2016.
[26] F.N. Koutanaei, H. Sajedi, M. Khanbabaei, A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring, J. Retail. Consum. Serv. 27 (2015) 11–23.
[27] W.H. Kruskal, W.A. Wallis, Use of ranks in one-criterion variance analysis, J. Am. Stat. Assoc. 47 (1952) 583–621.
[28] Y. Li, W. Jiang, L. Yang, T. Wu, On neural networks and learning systems for business computing, Neurocomputing. 275 (2018) 1150–1159.
[29] F. Liébana-Cabanillas, V. Marinković, Z. Kalinić, A SEM-neural network approach for predicting antecedents of m-commerce acceptance, Int. J.

Inf. Manage. 37 (2017) 14–24.
[30] F. Liébana-Cabanillas, V. Marinkovic, I. Ramos de Luna, Z. Kalinic, Predicting the determinants of mobile payment acceptance: A hybrid SEM-neural network approach, Technol. Forecast. Soc. Change. 129 (2018) 117–130.
[31] C. Matt, T. Hess, A. Benlian, Digital transformation strategies, Bus. Inf. Syst. Eng. 57 (2015) 339–343.
[32] A. McAfee, E. Brynjolfsson, T.H. Davenport, D.J. Patil, D. Barton, Big data: the management revolution, Harv. Bus. Rev. 90 (2012) 60–68.
[33] S. Moro, P. Cortez, P. Rita, Business intelligence in banking: A literature analysis from 2002 to 2013 using text mining and latent Dirichlet allocation, Expert Syst. Appl. 42 (2015) 1314–1324.
[34] J. Patel, S. Shah, P. Thakkar, K. Kotecha, Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques, Expert Syst. Appl. 42 (2015) 259–268.
[35] G. Peters, R. Weber, dynXcube – Categorizing dynamic data analysis, Inf. Sci. (Ny). 463–464 (2018) 21–32.
[36] R. Pita, E. Mendonça, S. Reis, M. Barreto, S. Denaxas, A machine learning trainable model to assess the accuracy of probabilistic record linkage, in: Int. Conf. Big Data Anal. Knowl. Discov., 2017: pp. 214–227.
[37] K.L. Priddy, P.E. Keller, Artificial neural networks: an introduction, SPIE press, 2005.
[38] M. Rubiolo, M.L. Caliusco, G. Stegmayer, M. Gareli, M. Coronel, Knowledge Source Discovery: An Experience Using Ontologies, WordNet and Artificial Neural Networks, in: J.D. Velásquez, S.A. Ríos, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intell. Inf. Eng. Syst., Springer Berlin Heidelberg, Berlin, Heidelberg, 2009: pp. 66–73.
[39] S. Ruggles, C.A. Fitch, E. Roberts, Historical census record linkage, Annu. Rev. Sociol. (2018).
[40] H. Salehian, P. Howell, C. Lee, Matching Restaurant Menus to Crowdsourced Food Data, in: Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD '17, 2017.
[41] N. Santoso, W. Wibowo, Comparative Study of Kernel Function for Support Vector Machine on Financial Dataset, Int. J. Soft Comput. 13 (2018) 129–133.
[42] P. Sarlin, K. Björk, Neurocomputing Machine learning in finance — Guest editorial, Neurocomputing. 264 (2017) 1.
[43] Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manag. 45 (2009) 427–437.
[44] M. Stonebraker, I.F. Ilyas, Data Integration: The Current Status and the Way Forward., IEEE Data Eng. Bull. 41 (2018) 3–9.
[45] J. Sukharev, L. Zhukov, A. Popescul, Parallel Corpus Approach for Name Matching in Record Linkage, in: 2014 IEEE Int. Conf. Data Min., 2014.
[46] K.M. Ting, Precision and Recall, in: C. Sammut, G.I. Webb (Eds.), Encycl. Mach. Learn., Springer US, Boston, MA, 2010: p. 781.
[47] V. Vapnik, The Nature of Statistical Learning Theory, 1995.
[48] Y. Xu, L. Wang, P. Zhong, A rough margin-based ν-twin support vector machine, Neural Comput. Appl. 21 (2011) 1–11.
[49] S. Yin, J. Yin, Tuning kernel parameters for SVM based on expected square distance ratio, Inf. Sci. (Ny). 370–371 (2016) 92–102.
[50] X. Zhong, D. Enke, Forecasting daily stock market return using dimensionality reduction, Expert Syst. Appl. 67 (2017) 126–139.