A Machine Learning-Based Approach for Economics-Tailored Applications: The Spanish Case Study

Zakaria Abdelmoiz Dahi[1,2(✉)], Gabriel Luque[1], and Enrique Alba[1]

[1] ITIS Software, Edificio Ada Byron, University of Malaga, Málaga, Spain
zakaria.dahi@uma.es, {gabriel,eat}@lcc.uma.es
[2] Department of Fundamental Computer Science and Its Applications, Fac. NTIC, University of Constantine 2, Constantine, Algeria
zakaria.dahi@univ-constantine2.dz

Abstract. The continuous evolution of economy hinders the decision-making process in this field. The former requires sophisticated techniques, and thus, manual and empirical methods are becoming increasingly obsolete. In this paper, we propose a computationally-supported approach that performs an economic profiling of cities based on their economic features and also a prediction of the future evolution of these economical metrics. Our contributions include (I) a data-ingestion module to extract, transform and load data, (II) a profiling module that achieves an unsupervised classification via a new distance-based cellular genetic algorithm and K-means and (III) a prediction unit based on long short-term memory artificial neural networks. Our proposal is tested on Spain, analysing all its 52 cities, where we use 33 types of real-world economic data that have been recorded monthly for fifteen years. All data has been obtained from the Spanish National Institute of Statistics. Our experiments show that the 52 cities could be clustered into only three economic profiles. This decrease in the complexity of entities to be considered allows managers at several levels and countries to take faster and more accurate decisions by dealing with few profiles rather than treating each city apart. Also, we found that each profile contains repetitive similarity patterns that are not only determined by economics but also indirectly ruled by the cities' geo and demographic situations. Results also showed our prototype's promising economic predictions.

Keywords: Economy · Machine learning · Metaheuristics

1 Introduction

Economy is a key domain that can be affected by numerous unpredictable factors requiring rapid and adequate measures, whereas the world's fast evolution increases the complexity of economical decisions (e.g. budget allowance). Thus, to take such decisions, the use of exhaustive and empirical methods is

J. L. Jiménez Laredo et al. (Eds.): EvoApplications 2022, LNCS 13224, pp. 567–583, 2022.
https://doi.org/10.1007/978-3-031-02462-7_36

becoming inefficient, while Artificial Intelligence (AI) appears as a promising alternative [1], especially that nowadays, valuable, raw and diverse information is continuously recorded. The question is: *how AI and raw data can empower computationally-affordable AI-assisted decision-making systems in economy?*

As a tentative answer to this question, we make a realistic assumption that the decision-taking complexity in economy comes principally from two problems: `(I)` the number of entities and constraints to be considered (e.g. companies, budget, etc.) and `(II)` the long-term planning. Therefore, as a solution to the first issue, we believe that reducing the complexity of one (or both) of entities/constraints will ease making faster appropriate decisions. To do so and considering that the "sparse, raw and non-annotated/unlabeled" nature of economic data hinders the use of supervised and semi-supervised learning, the unsupervised one through K-means clustering appears as a reputed computationally-cheap but still efficient solution. The K-means' initial centroids quality is one of the main factors that determines its efficiency [11]. Such task can be modeled as an optimisation problem, where Evolutionary Computation (EC) has already proven its solving efficiency for suchlike problems [3]. Moving to the second problem, we think that providing relevant predictions can help managers take more accurate decisions. Bearing in mind this, Long Short-Term Memory Artificial Neural Networks (LSTM ANN) have already proved their efficacy [6].

Our contribution consists in proposing a prototype of a nation-wise AI-assisted profiling and prediction approach based on both unsupervised and supervised Machine Learning (ML). The unsupervised part is done via K-means, where initial centroids are selected using a newly-proposed Distance-based cellular Genetic Algorithm (DcGA) that considers the clustering properties. Our proposal creates groups of one country's cities having each similar economic features (e.g. employment situation) [12]. On the other hand, the supervised ML unit consists of an LSTM ANN. This will help managers to make adequate long-run decisions by considering a few economic profiles rather than treating each city apart. We propose an entire profiling prototype, so our contributions range from an Extract-Transform-Load mechanism (ETL), including data acquisition, pre- and post-processing (e.g. feature selection) to the EC-ML process that results in an economic profiling and prediction via ML. Our approach has been tested on Spain, studying all its 52 cities, where each city has data recorded monthly for a period of 15 years (2003–2017). The economic profiling is made using 33 types of real-world economic information obtained from the Spanish National Institute of Statistics (Instituto Nacional de Estadística INE) (https://www.ine.es/). When going through the literature of ML in economics (e.g. [8]), we believe that our work is the first to combine these functionalities together in such a way.

The remainder of the paper is structured as follows. Section 2 introduce our approach, while Sect. 3 presents our experiments. Finally, Sect. 4 concludes the paper.

2 The Proposed Approach

Our work presents both *methodological* and *practical* contributions. First, we have performed several experiments to design each component of both our profiling and prediction units (e.g. feature selection, data series reduction, the hybrid clustering algorithms, etc.). Second, we present here the practical implementation of our proposal, including data acquisition, data formatting and pre-processing, data clustering pre-processing, profiling and prediction (see Fig. 1).

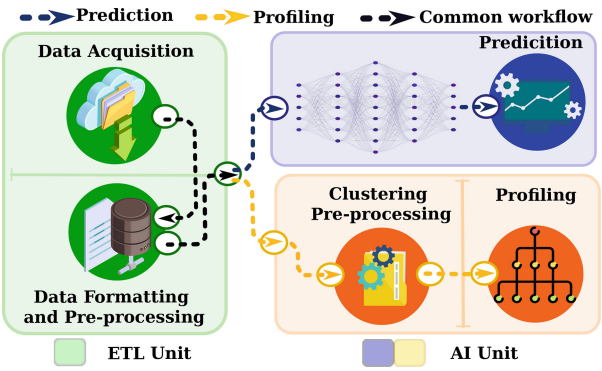

Fig. 1. The proposed economics-tailored approach

2.1 Data Acquisition, Formatting and Pre-processing

We take the ETL architecture given in [7] as a reference, although we neglect its in-depth technicality considering that our ETL purpose, architecture and data size are different. We focus on `I)` extracting raw data, `II)` making the data usable (i.e. cleaning and formatting), and `III)` loading the data to the database. To ensure our proposal's applicability to other datasets, we used real-world economic data obtained from INE, which is an official Spanish organism in charge of national and EU statistics-related duties. The manual downloading of INE data is not suitable for us since getting and updating data is time-consuming. This hinders automating the whole approach and prevents non-experts from using it. Therefore, we have implemented our own code that uses the open INE API (available at: https://www.ine.es/), to automatically download data.

Some INE data/attributes can mislead the profiling or the prediction. Thus, data formatting is a key process where we organised our data as a relational database with *schema* (id: int, location_name: varchar(x), serie_name: varchar(x), year: int, period: int, value: float), and *functional dependencies* {id $\longrightarrow$ location_name, id $\longrightarrow$ serie_name, id $\longrightarrow$ year, id $\longrightarrow$ period, id $\longrightarrow$ value}. Our approach obtained 30,000 data series from which we identified 50 relevant ones. We found that: `I)` some data was not available for some/all cities, `II)` the period's range of data recording, and `III)` the recording's starting date differ

between cities. Thus, we updated our proposal to automatically remove these data inconsistencies. We obtained an homogeneous dataset of 52 Spanish cities, where each city has 33 types of economic data series concerning employment, goods, etc. that were recorded monthly from 2003 to 2017. All our intermediary and final data are available at [4].

2.2 Data-Clustering Pre-processing

The data we use is organised in 33 partitions (one partition per data series), where each partition contains 52 samples (one sample per city) with a size of 15 * 12 features each. To increase the accuracy of the economic profiling, in the following, we have performed a set of necessary pre-processing steps.

Identifying the Ideal Number of Profiles: Extracting automatically the ideal number of economic profiles K that our approach will produce is paramount to ensure a relevant profiling. Technically speaking, it is one of the principal factors influencing the K-means' clustering and stands for the number of clusters K used. Thus, first, we clustered the 33 data partitions using the DcGA-Kmeans for $K = \{1, \ldots, 10\}$. Then, for each data series, we calculate the ideal value of K using the elbow method [2]. In our study, we use the inertia $\sum_{i=1}^{K} \sum_{X \in C_i} \|X - m_i\|^2$ as a metric for selecting the elbow, where X is the data point that belongs to a given cluster C_i and m_i is the centroid of the i^{th} cluster. Finally, we consider the median of the 33 elbows as the number of clusters K to be used in our proposal. It is worth stating that more complex strategies could be used for defining K (e.g. assigning a dynamic K value). But, we used the above-introduced technique to ensure that our proposal is computationally affordable and data-independent.

Features' Reduction: Each of the 33 data partitions contains 52 samples of 180 features size each (see Fig. 2). Using samples with such a size can mislead average-based techniques such as the K-means that is sensitive to outliers. Considering these facts and to open new perspectives in feature reduction techniques, we devised a heuristic to reduce the features that will be considered during the clustering. To do so, for each of the 33 data partitions, we perform the clustering using the value K found in the previous paragraph. Then, we perform the same clustering by replacing each Z consecutive features (i.e. Z consecutive months within the same year) by their mean value, where $Z = \{2, 3, 4, 6, 12\}$. So, we will perform clusterings on data samples of sizes 90, 60, 45, 30 and 15 features instead of 180. After, for each of the 33 partitions, we compute the distance between the original clustering done using 180 features and those achieved using 90, 60, 45, 30 and 15 features. Thus, for each of the 33 data series, we will obtain 5 distances.

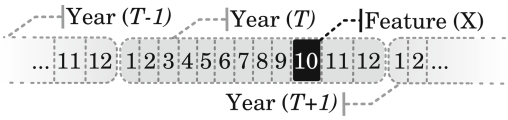

Fig. 2. Illustration of features sample

Since the distance calculation method is the same regardless to the number of features, let us consider, as an example, the distance between a clustering using 180 features and the one using 90 features. Assuming that, from previous step, both clusterings using 180 or 90 features will result in K clusters each, say the two sets $S_1 = \{A_1, \ldots, A_K\}$ and $S_2 = \{B_1, \ldots, B_K\}$, respectively. In this case, we will have $K!$ configurations of the following form: $\{\{A_1, B_a\}, \ldots, \{A_K, B_q\}\}$, where $a, \ldots, q$ are one of the $K!$ possible permutations without repetitions of K digits. For each one of the $K!$ configurations, we sum-up the distances between each pair of clusters $\{A, B\}$ (see Eq. (1)) included in that configuration. Then, the distance between the clustering using 180 features and the one using 90 features will be the minimum accumulative distances among all the $K!$ computed ones (one accumulative distance per configuration). The whole process is repeated for all the 33 data partitions. Once done, we select the number F of features (90, 60, 45, 30 or 15) that is statistically producing clusterings that are the closest (i.e. having the lowest distance) to the original clustering using 180 features and the 33 data series (see Algorithm 1).

$$\texttt{distance}(A, B) = \texttt{cardinality}((A \cup B) - (A \cap B)) \tag{1}$$

One should bear in mind that during the feature reduction step, we make the hypothesis that the original clustering produced using 180 features is *noise-free*. Thus, our feature reduction technique is designed, on purpose, to select the features that produce the closest clustering to the one with 180 features.

Algorithm 1. Feature reduction heuristic

for each of the 33 data series **do**
  for $Z \in \{2, 3, 4, 6, 12\}$ **do**
    Compute the distance between clusterings using 180 and $(180/Z)$ features.
  end for
  Extract the Z producing the smallest distance to the clustering using 180 features.
end for
Return Z that is statistically greater.

Data Series Reduction: Among the 33 economic data series, it is possible that more than two series result in the same useless clustering (i.e. economic profiles) which is computationally-penalising for our approach. Thus, we devised a heuristic to identify data series that provide different profiling from one another (see Algorithm 2). We start by forming $(\frac{33!}{(33-2)!}/2)$ pairs using the 33 data series,

then, for each data series of those pairs, we perform a clustering using K clusters and F features. For example, let us suppose that the first pair contains the 1^{st} and the 2^{nd} data series. Performing a clustering on both data series will result in K clusters each, say $S_1 = \{A_1, \ldots, A_K\}$ and $S_2 = \{B_1, \ldots, B_K\}$, respectively. The same process is repeated for the rest of the pairs. After, for each of the pairs, we compute the distance between their two clusterings S_x and S_y, where x and $y \in \{1, \ldots, 33\}$ are the indices of the series that constitutes the processed pair. The distance calculation is made as explained in Eq. (1). Once this has been achieved, we extract the list of Series considered Similar (`SS`). Data series of a given pair are said to be similar if the distance between their respective clusterings S_x and $S_y \leqslant \epsilon$. Considering our offline experiments, we have set ϵ to 26 since we admit that two clusterings are considered similar if they classify differently (i.e. in different clusters), at most, 25% of the cities (i.e. 13 cities). Knowing that each city that is classified differently in two clusterings will result in a distance of 2, in our case, 13 cities will generate a distance of 26.

Algorithm 2. Data series reduction heuristic

Form ($\frac{33!}{(33-2)!}$ / 2) pairs of data series.
Compute the distance between the clusterings in each pair.
Extract, to `SS`, all the pairs that are considered similar.
Save, in `MS`, the data series that are not included in one of the extracted pairs.
repeat
  Rank the data series in the extracted pairs.
  Save, in `MS`, the data series with the highest occurrence and delete all the pairs in `SS` in which it is included.
until {`SS` $= \emptyset$}
Return `MS`.

Once this has been done, we save, in a list `MS` (Maintained Series), the data series not included in `SS`. Then, we rank the data series according to the number of times they are included in one of the pairs of `SS`. After, we move the data series with the highest occurrence to the `MS` list, and we delete all the pairs that it is part of in the `SS`. We repeat this process until `SS` becomes empty. At the end of this process, the list `MS` is the list of series that will be used.

2.3 Economic Profiling via Data Clustering

This section describes our proposal's AI module, where we perform a clustering using the K clusters, F features and `MS`'s data series obtained in Sect. 2.2. Once done, for each of the series in `MS`, we will obtain K clusters, where each cluster represents a subset of the 52 Spanish cities that have similar economic features, or what we call here an "*economic profile*". The clustering is done using our newly-proposed approach called `DcGA-`K`means`. It is a two-phased technique that first computes the initial clusters' centroids by solving the clustering as

an optimisation problem defined by Eq. (2). Then, using these centroids, the K-means performs the clustering. In the following, we introduce our proposal starting by the K-means then the DcGA (see Algorithm 3).

Algorithm 3. Economic profiling heuristic

```
for all economic data series do
    %% Select K initial Centroids via DcGA %%
    Random initialisation of the population.
    while maximum number of iterations not reached do
        for each grid's node do
            Selection via binary-tournament on the Von Neumann neighbourhood.
            Breeding via DPX1 crossover and distance-based mutation.
            Evaluation and synchronous replacement.
        end for
    end while
    %% Perform Clustering via K-means %%
    repeat
        Affect each data point to its closest centroid obtained using the DcGA.
        Update the centroids.
    until {No change of data points after centroid update}
    Return the list of cities in each cluster (i.e. profile).
end for
```

The *K*-Means Algorithm. The K-means was ranked the 2^{nd} of top-10 data mining algorithms considering that it has a relative simplicity ($\mathcal{O}(NKFT)$), promising efficiency and applicability to several types of data [11]. Still, other advanced clustering techniques exist, but our goal is to demonstrate our proposal's operability even using simple ones. The K-means creates non-overlapping data clusters where each group's intra-similarity is maximised, while all groups' inter-similarity is minimised. Assuming $D = \{X_1, \ldots, X_N\}$ is a set of data points to be clustered, the clustering problem can be formulated using Eq. (2), where W_X is the weight of data point X, N_i is the number of points assigned to the i^{th} cluster C_i, K is the number of clusters set by the user. First, the K-means selects K initial cluster centroids, then each data point is assigned to its closest centroid ($\sum_{X \in C_i} \frac{W_X X}{N_i}$) in terms of squared Euclidean distance. Each data points' collection will constitute a cluster. Later, each centroid is updated by considering the points that have been assigned to it. This process is repeated until no data point changes its cluster. One should keep in mind that several metrics exist for expressing the clustering distance, being the Euclidean one the most widely-used, including in our work, despite its weaknesses [11]. Our aim is to show that a K-means with simple settings can still provide a meaningful profiling.

$$\textbf{Min} \sum_{i=1}^{K} \sum_{X \in C_i} W_X \| X - \sum_{X \in C_i} \frac{W_X X}{N_i} \|^2 \tag{2}$$

The Distance-Based Cellular Genetic Algorithm. The solution to the clustering problem defined in Eq. (2) is proven to be NP-hard, which motivates the use of heuristic-based techniques to solve it [9]. Cellular genetic algorithms are simple, computationally-affordable and efficient solvers [10]. They use a grid-shaped population (e.g. 2D), where each grid's node is a possible solution to the problem being solved. The interactions between individuals are restricted to a certain neighbourhood (e.g. Von Neumann) (see Fig. 3(b)). The cGA evolves its individuals towards fitter states by applying a series of operators until a stop criterion is reached. Although it is hard to draw the basic cGA's feature since many implementations exist [3], but generally, for each grid's node, the cGA starts by performing `I)` a selection on the neighbourhood of the processed individual, `II)` crossover and mutation, `III)` the produced individual(s) is(are) evaluated using the fitness function of the problem being solved and finally, `IV)` a replacement decides on the population's components for the next iteration.

Our DcGA is based on the cGA. Each individual, say $\overrightarrow{X}$, is integer-coded and represents a possible configuration of clusters' ID for each city: $\overrightarrow{X} = \{x_1, \ldots, x_N\}$, where N is the number of cities to be clustered and $x_i \in \{1, \ldots, K\}$, where K is the number of clusters to be formed. Figure 3(b) represents a DcGA's individual configuration, where 5 cities are clustered in 4 clusters: the 1^{st}, 2^{nd}, 3^{rd}, both 4^{th} and 5^{th} cities belong to the 4^{th}, 1^{st}, 3^{rd} and 2^{nd} clusters, respectively.

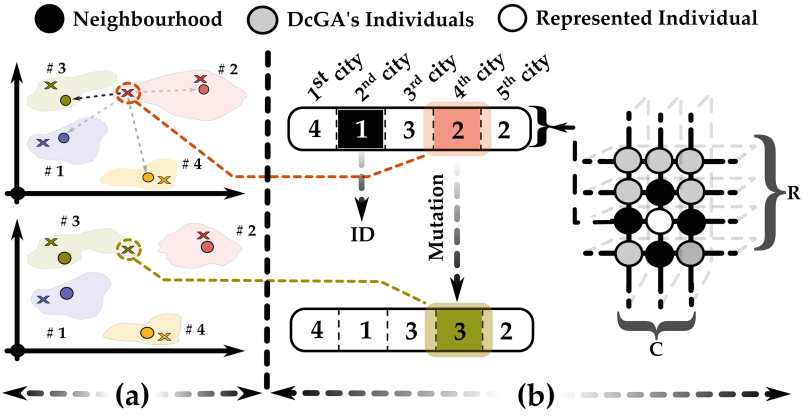

Fig. 3. DcGA's: (a) distance-based mutation and (b) solution representation

The DcGA starts by randomly initialising a grid of `R`×`C` individuals. For each individual, the DcGA selects another via a binary tournament on the Von Neumann neighbourhood of the processed node. Then, according to a probability P_c, a well-established variant of two-point crossover called `DPX1` [3] is applied on the processed individual and the selected one. The `DPX1` results in two new offspring, where the closest one (in terms of sum of absolute value of subtractions)

to the best parent is kept. Figure 4 is an example of our implementation of `DPX1`, where `Switch 1` and `2` represent the limits of the alleles' being exchanged between parents.

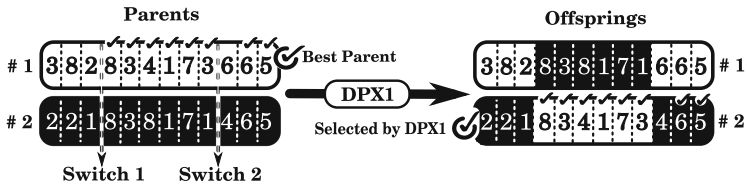

Fig. 4. The DPX1 crossover

After this, for each element of the produced offspring, our new distance-based mutation is applied according to a probability P_m. The new value of a mutated element is ruled by a dynamic probability P_d which is correlated to the widely-used Euclidean distance between the city's features and the clusters' centroids [11]. The smaller the distance the greater the probability to mutate the city cluster's ID to the one it is having a smaller distance with. Figure 3(a) illustrates how the distance-based mutation works where it represents a city that belongs to the 2^{nd} cluster and is mutated to the 3^{rd} one since it has a closer proximity with it. The mutated offspring is evaluated using the *fitness function* defined by Eq. (2) and compared to the processed individual. If it is better, it will be placed in an auxiliary population, otherwise it will be discarded. Once the synchronous replacement done, the process is performed all over again on the following individual in the grid. The whole process is repeated until all the grid's nodes are browsed. Once done, this marks the end of one DcGA iteration. The DcGA is executed again until the maximum number of iterations is reached.

It is to bear in mind that the new DcGA we propose here has been achieved after several systematic experiments studying several techniques including three types of mutation operators: distance-based (the one we propose here), opposition-based and random mutation. In our work, we emphasise our study on the mutation since previous works (e.g. [5]) have showed its major influence in clustering problems. Four metrics have been used during the comparison: `I)` sum of the distances between each cluster's samples and its centroid, `II)` sum of the distances between each cluster's centroid and its farthest sample, `III)` sum of the distances between each cluster's centroid and the nearest sample from the remaining clusters, `IV)` sum of the distances between the clusters' centroids. Considering all four metrics, our proposed distance-based cGA has been found to be the best. In addition, the present DcGA's design is made on purpose to ensure its applicability to other datasets and applications. In the following, we provide details of the mutation operators and metrics being studied.

The cGAs' Variants and Clustering Metrics. Our distance-based mutation has been initially compared against random and opposition-based mutations.

The first type of mutation assigns a randomly-chosen cluster ID to a given city when $r \leq P_m$, where r is a random number. The second mutation affects a cluster ID x_i' based on the following formula $(K - x_i + 1)$, where K is the number of profiles to be formed and x_i, x_i' represent the i^{th} city's original and mutated cluster IDs, respectively. Figure 5(a) and (b) illustrate how both mutations act on a configuration with 12 cities that need to be clustered in 4 economic profiles.

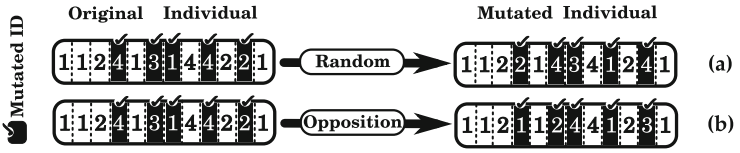

Fig. 5. Mutation: (a) random and (b) opposition-based

To analyse, from different aspects, the quality of the clustering obtained by the three cGAs' variants, we have considered three metrics besides the one described in Eq. (2). The Eqs. (3)–(5) represent, respectively, `I)` sum of the distances between each cluster's centroid and its farthest sample, `II)` sum of the distances between the clusters' centroids, `III)` sum of the distances between each cluster's centroid and the nearest sample from the remaining clusters.

$$\textbf{Min} \sum_{i=1}^{K} \underset{\forall X \in C_i}{\textbf{Max}} \; W_X \| X - \sum_{X \in C_i} \frac{W_X X}{N_i} \|^2 \tag{3}$$

$$\textbf{Max} \sum_{i=1}^{K} \sum_{j=1,\, j>i}^{K} \| \sum_{X \in C_i} \frac{W_X X}{N_i} - \sum_{X \in C_j} \frac{W_X X}{N_j} \|^2 \tag{4}$$

$$\textbf{Max} \sum_{i=1}^{K} \underset{\forall X \in C_j,\, j=1 \ldots K,\, j>i}{\textbf{Min}} \| W_X X - \sum_{X \in C_i} \frac{W_X X}{N_i} \|^2 \tag{5}$$

2.4 Economic Prediction

Our prediction module is based on LSTM ANN. Like classical multi-layer perceptron ANN, the LSTM is composed of layers of neurons. The signal propagates through the network to make the prediction. When going into the details, the LSTM are a type of recurrent ANN in which learning signal go in both sideways going in loop through the ANN. The recurrent connections add a memory to the network to harness the ordered nature of the input sequences. Indeed, instead of mapping inputs to outputs, the ANN learns a mapping functions for the inputs over time to an output which unlocks time series for the ANN. Bearing in mind this, the computational unit of an LSTM ANN is called a memory cell or block. The former is composed of weights (input, output and internal states) and gates

(Forget, input and output), which are functions that govern the information flow in the cell. Several types of LSTM ANNs exist (e.g. stacked, convolutional, bidirectional, etc.), where each one has its own features. Therefore, it is hard to draw the standard architecture of an LSTM ANN. Nonetheless, Fig. 6 illustrates the blueprint of a classical stacked LSTM ANN that includes several LSTM layers, which provides a sequence of output rather than one [6].

Fig. 6. Architecture of a stacked LSTM ANN

The original data used by our prediction module is a 180×1 value. This corresponds to the information recorded monthly during 15 years (2003–2017). Then, this sample is normalised to a range of [0,1] using the min-max technique. Once this is done, we set our time-series-like data in samples of the form $\{(t-n),\ ...,\ t\}$, by shifting one data series value each time. This produces a 176×5 prediction value. We use 70% of the original data (123×5 values) for training our model and the remaining 30% (53×5 values) as testing data. Technically speaking, our model trains using the values recorded from January 2003 to July 2013, while our tests are done on the data recorded from August 2013 to December 2017. Our implementation of LSTM is composed of three layers of 32, 24 and 12 neurons. As a machine learning algorithm, we use `Adam` version of stochastic gradient descent and the mean squared error as a loss function. We train our model for 50 epochs using a batch size equal to 32.

3 Experimental Results and Analysis

In this section, we present the results obtained by both the profiling and prediction modules of our proposal.

3.1 The *-CGAs' Variants Comparison

Based on offline tuning, the three cGA's variants have been run with a grid of 40×10 nodes, $P_c = 0.5$ and $P_m = 0.2$. The experiments have been repeated over 30 executions, where each has a stopping criterion of 50 iterations. The latter has been chosen since beyond it, we found that the algorithm(s) evolution decreases (or stagnates). Table 1 presents a summary results obtained when comparing the cGA's variants using the distance-based, random and opposition-based mutations. The results consists in how many times a given cGA variant has achieved the best results, in each economic data series, according to the three metrics explained in Sect. 2.3 and by applying a clustering using 1–10 clusters. The ranking takes into account the average of the 30 values obtained in each

metric and for each K clusters, where the best scores are highlighted in bold and light-gray. One can note that in 32 out of the 33 data series, our proposed DcGA is the one obtaining the best results. Thus, the former solver is the only one being investigated in the rest of the experiments. For further detailed results, one can access to [4].

Table 1. The scores of cGAs using: distance, opposition and random-based mutations

cGAs \ Series	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
Distance	**22**	**22**	**22**	**22**	**22**	**23**	**22**	**22**	**23**	**22**	**22**	**22**	**22**	**22**	**22**	**22**	**24**	**22**	**22**	**22**	**22**	**22**	**22**	**24**	**25**	**24**	**25**	**22**	**22**	**22**	**22**	**22**	**22**
Opposition	20	8	21	14	8	12	13	12	16	7	14	19	19	19	11	8	8	13	10	13	4	15	13	18	17	17	8	12	11	12	16	20	18
Random	6	18	5	12	18	13	13	14	9	19	12	7	7	7	15	18	16	13	16	13	**22**	11	11	5	7	6	18	14	15	14	10	6	8

Figure 7 represents the fitness evolution of the cGA's variants when profiling the data series of *men unemployment* using metric n° 1 and 6 profiles. On the other hand, Fig. 8 illustrates the cGAs' fitness evolution when profiling *women unemployment* using metric n° 1 and 10 profiles. The figures have been created using randomly-chosen executions and both show the superiority of our proposal, as well as its continuous and smooth convergence compared to others.

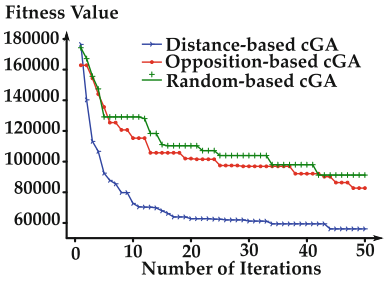

Fig. 7. Men unemployment

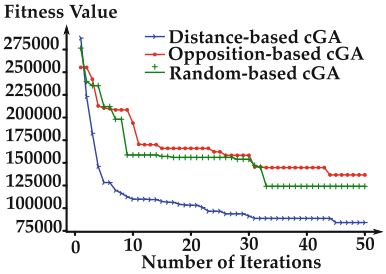

Fig. 8. Women unemployment

It is also worth stating that all our results have been confirmed using both Kruskal-Wallis and a post-hoc tests with a significance level of 0.05.

3.2 Economic Profiling

The implementation and database manipulation have been done using Python 3.8.5 and SQLite 3.27.2, respectively. The two first steps (i.e. data collection, formatting and pre-processing) can be seen directly in the database that we have obtained [4]. Thus, in the following, we will present the results of the remaining phases.

Based on our experiments, we found that using our elbow-based analysis, the ideal number of clusters K to be applied is 3 [4]. Using our feature reduction heuristic [4], we found that when considering 2 months (90 features), in 16 times, it is generating the lowest distance between its clustering and the one obtained using the original 180 features, 3 months (60 features) is producing it 13 times, 4 months (45 features) is giving the lowest distance 7 times, 12 months (15 features) produces it 14 times and the 6 months (30 features) is producing the lowest distance 17 times which is the best one. Thus, we set F to 30 features.

Afterwards, as explained in Sect. 2.2, we reduce the number of economic data series to be considered during the economic profiling. Experiments extracted the following `MS` list: Activity Percentage of {1} Men and {2} Women. {3} Men Unemployment Percentage. Also, Employment Percentage of {4} Men (I), {5} Women (I), {6} Men (II) and {7} Women (II). Index. of {8} Monthly Variation, {9} Aliments and non-alcoholic beverages, {10} Alcoholic beverages and tobacco, {11} Dress and footwear, {12} Health, {13} Transportation, {14} Communications, {15} Leisure and culture, {16} Teaching, {17} Restaurants and Hotels and {18} Other goods and Services. Total companies {19} Without employees, with a workforce {20} From 3 to 5, {21} From 6 to 9, {22} From 10 to 19, {23} From 20 to 49, {24} From 50 to 99, {25} From 100 to 199, {26} From 200 to 499, {27} From 500 to 999, {28} From 1000 to 4999. As it can be seen, the data series reduction has considered only 28 data series instead of the original 33 economic pieces of information that we have selected in Sect. 2.1.

Now, we present the results of the economic profiling described in Sect. 2.3. This has been achieved using $K = 3$ clusters, which corresponds to the number of economic profiles. The number of features has been reduced to 30 features which designates two data values per year. The economic profiling has been done for the 52 Spanish cities and this according to 28 data series. Table 2 presents the results of this phase, where we show in a different colour each economic profile: orange for the 1^{st} profile, yellow for the 2^{nd} one and green for the 3^{rd}. Considering the large amount of data contained in each data series, it is impossible to present further correlations between the original data and the clustering achieved in this section or the centroids coordinates for each data series. For further experimentations on all the presented steps, one can use our profiling approach's code and also our experimentation results available at [4].

As it can be seen in Table 2, our approach could classify the 52 Spanish cities into 3 profiles and this for each of the 28 data series. This substantial decrease in the cases facilitates taking decisions and dealing with abnormal situations (e.g. worldwide pandemic). Indeed, for instance, this allows a smart governance by allowing rulers to establish at most 3 economic strategies (e.g. budget allowance) instead of 52 strategies (one for each city). This shows another keen advantage of our proposal which is that the number of profiles can be set in compliance with the decision-makers' needs and application purposes. Also, even if the profiling is based on economic data, we found that some cities are indirectly clustered considering a common geographical (e.g. overseas cities: Melilla, Ceuta, Santa Cruz de Tenerife), administrative (e.g. Comunidad de Andalucía: Córdoba, Cádiz, Sevilla, Málaga, etc.) or demographical (e.g. Madrid and Barcelona) situation (Fig. 9).

Table 2. Results of the economic profiling

City \ Data Series N°	1	2	3	5	6	8	9	10	11	12	13	14	15	16	17	18	19	20	21	23	24	25	26	27	28	29	30	31
A Coruña	1	3	2	2	3	2	2	2	2	1	1	2	1	2	1	3	1	2	2	3	1	3	2	2	3	1	2	2
Albacete	1	3	3	2	3	2	2	3	1	2	1	1	1	1	1	1	3	2	3	1	3	2	3	1	2	3	1	1
Alicante/Alacant	2	3	3	2	3	2	2	1	1	2	1	3	1	3	1	3	3	2	2	3	1	3	2	2	3	1	1	1
Almería	3	1	1	2	3	2	2	3	1	2	2	2	3	1	3	2	2	2	3	1	3	2	3	1	2	3	1	1
Araba/Álava	3	1	2	3	1	3	3	2	2	2	1	2	1	2	2	3	3	1	3	1	3	2	3	1	2	3	1	1
Asturias	1	2	3	1	2	1	1	2	1	1	1	1	1	2	1	3	1	2	2	3	1	3	2	2	3	1	2	2
Ávila	1	2	3	1	2	1	1	3	1	2	2	2	3	2	1	1	3	1	3	1	3	2	3	1	2	3	1	1
Badajoz	2	2	1	1	2	1	1	1	1	2	2	3	3	3	3	2	2	2	3	1	3	2	3	1	2	3	1	1
Barcelona	3	1	3	3	1	3	3	1	2	1	1	2	1	2	2	1	1	1	1	2	2	1	1	3	1	2	3	3
Bizkaia	2	3	2	2	3	2	2	2	2	1	1	2	1	2	2	3	1	1	2	3	1	3	2	2	3	1	2	2
Burgos	2	3	2	3	3	3	2	2	2	1	1	2	3	1	1	1	3	2	3	1	3	2	3	1	2	3	1	1
Cáceres	1	2	3	1	2	1	1	2	1	2	2	1	1	2	3	3	2	2	3	1	3	1	3	1	2	3	1	1
Cádiz	2	3	1	1	2	1	1	1	1	2	2	1	1	3	3	1	3	2	2	3	1	3	2	2	3	3	1	1
Cantabria	2	3	2	2	3	2	2	2	1	1	1	1	3	1	2	3	3	1	3	1	3	2	3	1	2	3	1	1
Castellón/Castelló	3	1	3	2	3	2	2	1	2	2	2	3	3	3	1	2	2	2	3	1	3	2	3	2	3	1	2	1
Ceuta	3	2	3	2	2	2	1	1	3	3	2	1	2	1	1	3	2	3	3	1	3	2	3	1	2	3	1	1
Ciudad Real	2	2	3	2	2	2	1	3	1	2	1	1	1	2	3	2	2	2	3	1	3	2	3	1	2	3	1	1
Córdoba	2	3	1	1	2	1	1	1	1	1	1	2	1	2	1	2	1	2	3	1	3	2	3	1	2	3	1	1
Cuenca	2	2	3	2	2	2	1	3	1	2	2	1	3	1	3	2	3	2	3	1	3	2	3	1	2	3	1	1
Gipuzkoa	2	3	2	3	1	3	3	2	2	1	2	2	1	1	2	3	1	2	2	1	3	3	2	2	3	1	2	1
Girona	3	1	3	3	1	3	3	3	2	1	1	2	1	3	2	1	1	1	3	3	1	3	2	2	3	3	1	1
Granada	2	3	1	1	2	1	1	1	1	2	2	3	1	1	3	2	3	2	2	1	3	2	3	1	2	3	1	1
Guadalajara	3	1	2	3	1	3	3	3	1	2	1	1	3	2	2	2	2	3	3	1	3	2	3	1	2	3	1	1
Huelva	2	3	1	1	2	1	1	1	1	2	2	1	1	2	3	2	2	2	3	1	3	2	3	1	2	3	1	1
Huesca	2	3	2	3	3	3	2	3	2	1	1	2	1	1	3	2	3	1	3	1	3	2	3	1	2	3	1	1
Illes Balears	3	1	3	3	1	3	3	1	1	1	1	3	2	1	2	3	3	1	2	3	1	3	2	2	3	1	2	2
Jaén	2	2	1	1	2	1	1	1	1	1	1	1	1	1	3	2	2	3	3	1	3	2	3	1	2	3	1	1
La Rioja	3	3	2	3	3	3	2	1	1	1	1	1	1	2	2	2	2	1	3	1	3	2	3	1	2	3	1	1
Las Palmas	3	1	1	2	3	2	2	1	3	3	3	1	2	2	3	1	2	3	2	3	1	3	2	2	3	1	2	1
León	1	2	3	1	2	1	1	3	2	2	1	1	3	2	3	1	1	2	3	1	3	2	3	1	2	3	1	1
Lleida	3	1	2	3	1	3	3	3	2	1	1	1	1	1	2	1	3	1	3	1	3	2	3	1	2	3	1	1
Lugo	1	2	2	1	3	1	2	2	1	1	1	1	1	3	2	2	1	3	3	1	3	2	3	1	2	3	1	1
Madrid	3	1	2	3	1	3	3	1	1	2	1	3	1	2	2	2	3	1	1	2	2	1	1	3	1	2	3	3
Málaga	2	3	1	1	2	1	1	1	1	2	2	1	1	1	3	2	3	1	2	3	1	3	2	2	3	1	1	1
Melilla	3	2	3	2	2	2	1	2	3	3	2	3	2	3	1	2	2	2	3	1	3	2	3	1	2	3	1	1
Murcia	3	3	3	3	3	3	2	1	1	2	1	3	1	2	1	2	2	2	2	3	1	3	2	2	3	1	2	2
Navarra	3	1	2	3	1	3	3	2	1	1	1	2	1	3	2	1	2	1	3	1	3	2	2	2	3	1	2	2
Ourense	1	2	3	1	2	1	1	1	1	1	2	1	1	2	1	2	3	3	3	1	3	2	3	1	2	3	1	1
Palencia	2	2	2	2	2	2	1	2	1	2	2	1	3	3	1	1	1	2	3	1	3	2	3	1	2	3	1	1
Pontevedra	2	3	3	2	3	2	2	3	1	1	1	2	1	2	1	3	1	2	2	3	1	3	2	2	3	1	1	1
Salamanca	1	2	2	2	2	2	1	2	1	2	2	3	3	2	3	2	1	2	3	1	3	2	3	1	2	3	1	1
Santa Cruz de Tenerife	3	1	1	2	3	2	2	1	3	3	3	3	2	1	1	3	2	3	2	3	1	3	2	2	3	1	1	1
Segovia	2	3	2	3	3	3	2	2	2	1	1	2	3	2	2	1	1	1	3	1	3	2	3	1	2	3	1	1
Sevilla	3	3	1	2	2	2	1	1	1	2	2	1	1	3	1	2	3	1	2	3	1	3	2	2	3	1	2	2
Soria	2	3	2	3	3	3	2	2	2	1	1	2	3	1	1	2	1	2	3	1	3	1	3	1	2	3	1	1
Tarragona	3	1	3	3	1	3	3	1	1	1	2	2	3	1	3	1	1	1	3	1	3	2	3	1	3	3	1	1
Teruel	1	2	2	2	3	2	2	3	2	2	2	2	1	2	3	3	1	2	3	1	3	2	3	1	2	3	1	1
Toledo	3	3	3	3	2	3	1	3	1	2	2	1	3	1	1	1	2	3	3	1	3	2	3	1	2	3	1	1
Valencia/València	3	1	3	2	3	2	2	1	2	2	1	3	1	3	2	3	1	2	2	3	1	3	2	2	3	1	2	2
Valladolid	2	3	2	3	3	3	2	3	1	2	1	2	3	1	1	2	1	2	3	1	3	2	3	1	2	3	1	1
Zamora	1	2	3	1	2	1	1	2	2	2	1	2	1	1	1	2	3	2	3	1	3	2	3	1	2	3	1	1
Zaragoza	3	3	2	3	3	3	2	1	1	2	1	1	1	1	1	2	3	1	2	3	1	3	2	2	3	1	2	2

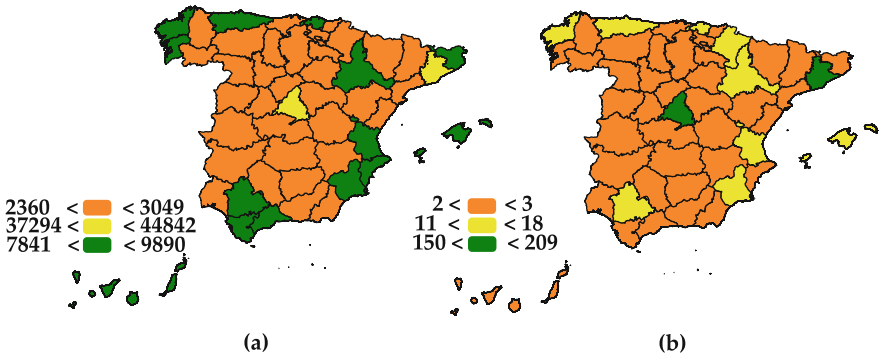

Fig. 9. Visual profiling (a) companies with 3–5 and (b) 1000–4999 employees

Figure 10(a) and (b) represent a parallel-coordinates' illustration of the profiles' centroids obtained for two extreme data series representing the companies with `I)` 3 to 5 and `II)` 1000 to 4999 employees, respectively. Data normalisation hs been done using the min-max technique. The features displayed are those selected as described in Sect. 2.2. As it can be seen, our approach could extract distinct centroids, although for hard instances, our approach could provide better profiling by fine-tuning our AI module (e.g. number of runs, iterations, operators, etc.), applying other ML algorithms, etc.

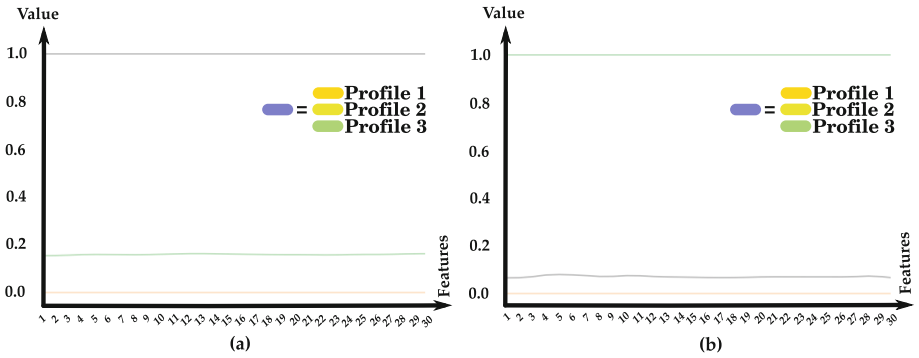

Fig. 10. Parallel coordinates of (a) companies with 3–5 and (b) 1000–4999 employees

3.3 Economic Prediction

Since it is impossible to present the predictions made for all the 52 Spanish cities, we provide here the predictions made for the capital city Madrid for both companies with 3–5 and 1000–4999 employees. Furthermore, one can use our source code to perform further predictions on the rest of the Spanish cities or data series. As it can be seen in Fig. 11(a) and (b), our model is able of predicting

values that are close to the real ones. This tends to affirm its applicability for forecasting future unknown evolution of a given economical metric. We believe that due to the type of data being used, it is hard to obtain exact prediction. The latter can still be achieved by enhancing the LSTM model being used. Also, the data pre-processing step can be elaborated in order to cope with this shortfall.

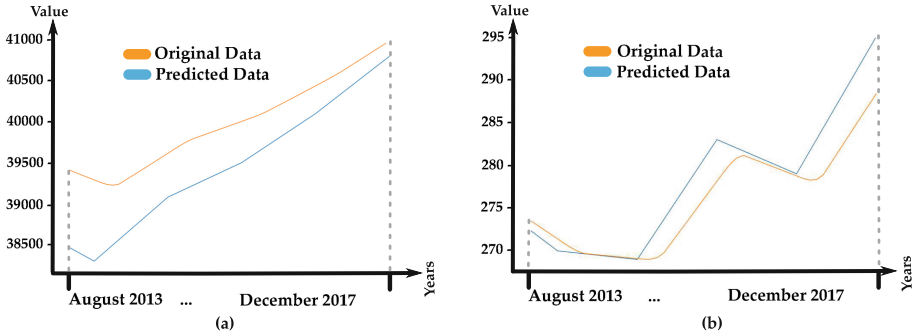

Fig. 11. Prediction of the city of Madrid for companies with (a) 3–5 and (b) 1000–4999 employees

4 Conclusion

We have proposed an automatic nation-wise economic profiling and prediction approach including: `I)` a data ETL to extract, process and load raw data and `II)` an AI module that performs a clustering by combining K-means and a newly-proposed DcGA and `III)` a prediction unit based on LSTM ANN. The contributions of our work range from the identification, collection, pre- and post-processing of new and real-world data to the data-clustering pre-processing and automatic profiling and prediction. Our approach has been applied on Spain, studying all its 52 cities, 33 types of economic data series recorded monthly from 2003 to 2017. Our proposal classified all cities into 3 economic profiles, which decreases the complexity and number of cases that decision-makers have to consider. We also found repetitive similarity patterns due to cities' common-hidden administrative or demo/geographic situation. As a perspective, we intend to design more flexible and user-oriented ML and EC techniques for economic profiling and apply them on larger countries (e.g. China). Also, considering the temporal nature of the economic data, we seek to design a more elaborated profiling by applying multivariate time-series clustering techniques and compare the obtained profiling with the one of classical clustering methods.

Acknowledgements. This work resulted from a stay at the University of Malaga (Spain) using the grant "*stays of researchers with prestigious recognition*" and is also partially funded by the Universidad de Málaga, Consejería de Economía y Conocimiento de la Junta de Andalucía and FEDER under grant number UMA18-FEDERJA-003 (PRECOG); under grant PID 2020-116727RB-I00 (HUmove) funded

by MCIN/AEI/10.13039/501100011033; and TAILOR ICT-48 Network (No. 952215) funded by EU Horizon 2020 research and innovation programme. Furthermore, the views expressed are purely those of the writer and may not in any circumstances be regarded as stating an official position of the European Commission.

References

1. Bonaccorso, G.: Mastering Machine Learning Algorithms: Expert Techniques for Implementing Popular Machine Learning Algorithms, Fine-Tuning Your Models, and Understanding How They Work, 2nd edn. Packt Publishing, Birmingham (2020)
2. Chakrabarty, N., Rana, S., Chowdhury, S., Maitra, R.: RBM based joke recommendation system and joke reader segmentation. In: Deka, B., Maji, P., Mitra, S., Bhattacharyya, D.K., Bora, P.K., Pal, S.K. (eds.) PReMI 2019. LNCS, vol. 11942, pp. 229–239. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34872-4_26
3. Dahi, Z.A., Alba, E.: The grid-to-neighbourhood relationship in cellular GAs: from design to solving complex problems. Soft. Comput. **24**(5), 3569–3589 (2019). https://doi.org/10.1007/s00500-019-04125-w
4. Dahi, Z.A., Luque, G., Alba, E.: Database, source code and results of the proposed machine learning-based approach for economics-tailored applications. https://github.com/Zakaria-Dahi/ML-Economis.git. Accessed 09 Feb 2022
5. El-Shorbagy, M.A., Ayoub, A.Y., Mousa, A.A., El-Desoky, I.M.: An enhanced genetic algorithm with new mutation for cluster analysis. Comput. Stat. **34**(3), 1355–1392 (2019). https://doi.org/10.1007/s00180-019-00871-5
6. Istiake Sunny, M.A., Maswood, M.M.S., Alharbi, A.G.: Deep learning-based stock price prediction using LSTM and bi-directional LSTM model. In: 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), pp. 87–92 (2020)
7. Kimball, R., Caserta, J.: The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming and Delivering Data. Wiley, Hoboken (2011)
8. Ozden, E., Guleryuz, D.: Optimized machine learning algorithms for investigating the relationship between economic development and human capital. Comput. Econ. (2021)
9. Pérez-Ortega, J., Almanza-Ortega, N.N., Vega-Villalobos, A., Pazos-Rangel, R., Zavala-Díaz, C., Martínez-Rebollar, A.: The K-means algorithm evolution. In: Sud, K., Erdogmus, P., Kadry, S. (eds.) Introduction to Data Science and Machine Learning (chap. 5). IntechOpen, Rijeka (2020)
10. Whitley, L.D.: Cellular genetic algorithms. In: Proceedings of the 5th International Conference on Genetic Algorithms, p. 658. Morgan Kaufmann Publishers Inc., San Francisco (1993)
11. Wu, J.: Advances in K-Means Clustering: A Data Mining Thinking. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29807-3
12. Yamarone, R.: The Trader's Guide to Key Economic Indicators. Bloomberg Financial Series, 3rd edn. Wiley, Hoboken (2012)