RedNIE Red Nacional de Investigadores en Economía

Redefining Regions in Space and Time: A Deep Learning Method for Spatio-Temporal Clustering

Pablo Quintana (UNCuyo)

Marcos Herrera-Gómez (CONICET/CIANECO/Universidad de Río Cuarto)

DOCUMENTO DE TRABAJO N° 368

Agosto de 2025

Citar como:
Quintana, Pablo y Marcos Herrera-Gómez (2025). Redefining Regions in Space and Time: A Deep Learning Method for Spatio-Temporal Clustering. Documento de trabajo RedNIE N°368.

Redefining Regions in Space and Time: A Deep Learning Method for Spatio-Temporal Clustering

Pablo Quintana*, Marcos Herrera-Gómez†

Abstract

Identifying regions that are both spatially contiguous and internally homogeneous remains a core challenge in spatial analysis and regional economics, especially with the increasing complexity of modern datasets. These limitations are particularly problematic when working with socioeconomic data that evolve over time. This paper presents a novel methodology for spatio-temporal regionalization—Spatial Deep Embedded Clustering (SDEC)—which integrates deep learning with spatially constrained clustering to effectively process time series data. The approach uses autoencoders to capture hidden temporal patterns and reduce dimensionality before clustering, ensuring that both spatial contiguity and temporal coherence are maintained. Through Monte Carlo simulations, we show that SDEC significantly outperforms traditional methods in capturing complex temporal patterns while preserving spatial structure. Using empirical examples, we demonstrate that the proposed framework provides a robust, scalable, and data-driven tool for researchers and policymakers working in public health, urban planning, and regional economic analysis.

JEL Codes: C23, C45, C63.

Keywords: Spatial clustering, Spatial Data Science, Spatio-temporal Classification, Territorial analysis.

*Corresponding author; Faculty of Economic Sciences - FCE UNCuyo; Centro Universitario (M5500), Mendoza, Argentina; email: pabanib@hotmail.com

†CONICET - CIANECO, National University of Rio Cuarto; Ruta 36 km. 601 (X5804BYA), Río Cuarto (Córdoba, Argentina); email: mherreragomez@conicet.gov.ar.

1 Introduction

Regional economics and spatial analysis emphasize that spatial clustering—the tendency for similar economic or social activities to group together geographically—is fundamental to the understanding of socioeconomic phenomena. This concept helps explain why industries tend to agglomerate in certain areas (Nathan and Overman, 2013), how labor markets form (Combes and Duranton, 2006), and the way cities develop spatial patterns (Portnov, 2006).

Central to these fields is the idea of a region: a set of neighboring areas that are more alike in terms of economic or social features compared to other groups (Isard, 1956; Berry, 1964). This is similar to the statistical concept of a cluster (Jain, 2010). However, a key difference is that in regional analysis, regions must be spatially contiguous, that is, the areas in a region must be connected geographically. This contiguity requirement sets regionalization apart from standard, non-spatial clustering methods (Rey et al., 2023).

Historically, traditional approaches in regional science and spatial statistics relied on techniques such as hierarchical clustering, principal components analysis, or factor analysis, often complemented by expert judgment to delineate regions (Duque et al., 2007). While these methods proved practical for policy and planning, they were limited by the available data, computational resources, and the need for manual adjustments to ensure spatial contiguity. Over the past two decades, advances in data science and machine learning have transformed regionalization: modern algorithms such as SKATER (Assunção et al., 2006), REDCAP (Guo, 2008), and the max-p-regions model (Duque et al., 2012) leverage optimization and computational power to group spatial units into regions that satisfy specific criteria.

Despite these advances, a fundamental research question remains: how can we define and identify regions that are both spatially contiguous and internally homogeneous, especially as the complexity and volume of spatial data continue to grow? Traditional regionalization methods often rely on manual parameter tuning and subjective decisions, which can introduce bias and limit reproducibility. Furthermore, these approaches typically focus on cross-sectional data, struggling to accommodate the growing availability of temporal information that characterizes modern socioeconomic datasets.

A key limitation of existing methods is their inability to effectively incorporate time series data into the regionalization process. While dedicated time series clustering techniques exist—using, for example, Euclidean distance or Dynamic Time Warping (DTW) (Sakoe and Chiba, 1978; Faloutsos et al., 1994)—they rarely integrate the spatial constraints necessary for meaningful regional analysis. Recent approaches, such as the weighted Euclidean distance method (Huang et al., 2016), offer some progress but still fall short of fully capturing the spatio-temporal complexity inherent in real-world applications.

To overcome this limitation, we propose a novel methodology that integrates unsupervised machine learning techniques to regionalize spatial units with time series attributes. Our approach acknowledges that temporal patterns in regional data often exhibit serial dependence and latent structure, which simple clustering may overlook. We employ autoencoders—advanced neural

networks for unsupervised learning—to detect these latent variables and reduce dimensionality prior to clustering. Building on the Deep Embedded Clustering (DEC) algorithm (Xie et al., 2016), we significantly extend its capabilities to incorporate spatial constraints and effectively handle time series data.

Through Monte Carlo simulations, we demonstrate that our methodology robustly addresses the complex challenges of spatio-temporal regionalization. Our results reveal that the proposed approach outperforms traditional methods in terms of both spatial coherence and the ability to capture latent temporal dynamics, even in the presence of complex, high-dimensional datasets.

The practical implications of our work are significant. First, by enabling the identification of regions that are both spatially contiguous and temporally coherent, our methodology facilitates the optimization of administrative boundaries, such as redefining local government jurisdictions or zoning districts. Second, it enables the formulation of targeted economic development policies by identifying areas where investments can be most effective, based on both spatial proximity and evolving economic conditions. Third, it advances urban and regional planning by supporting the development of infrastructure and land-use strategies that are responsive to the evolving needs of contiguous urban areas. By bridging methodological advances in machine learning with the pressing analytical needs of spatial sciences, this research contributes to more informed decision-making across a range of spatially dependent domains.

The remainder of this paper is structured as follows: Section 2 presents a review of the literature. Section 3 develops our methodological framework, presenting both traditional clustering algorithms and recent machine learning approaches. This section introduces our novel autoencoder-based algorithm and details its extension to incorporate spatial constraints. Section 4 provides a comprehensive evaluation of different methodological alternatives through Monte Carlo simulations, establishing the robustness of our approach. Section 5 illustrates the usefulness of our algorithm through two empirical applications in Argentina, highlighting the good performance of the proposal. Finally, Section 6 summarizes the main strengths of our proposal, highlighting its implications for regional analysis and describing promising directions for future research.

2 Literature review

Clustering algorithms aim to reduce the dimensionality of data by grouping n observations into K clusters. The objective is to ensure that observations within the same group are as similar as possible while remaining distinct from observations in other groups. These goals are conveyed to the algorithms through the maximization or minimization of a dissimilarity function, depending on the specific case. Similarly, regionalization methods seek to identify groups of similar data with the added requirement of spatial contiguity, meaning they aim to find geographical neighbors that exhibit similar characteristics within a region. The foundation of such work dates back to the 1970s;

however, significant advancements were not made until the arrival of the New Digital Age, which has overcome many of the earlier computational challenges.

There are straightforward alternatives for combining traditional clustering methods with spatial techniques. One such approach is implemented in the GeoDa program, as described by Anselin et al. (2010). In this method, geographic coordinates are aggregated as additional attributes of the spatial units, allowing the loss function to consider this information as any other attribute. Another similar alternative, proposed by authors like Webster and Burrough (1972), Wise et al. (1997),Haining et al. (2000), and more recently, Yuan et al. (2015), Cheruvelil et al. (2017), Chavent et al. (2018), the spatial information is treated with a different weighting scheme. Here, the spatial information is typically represented by a contiguity matrix, denoted as $\mathbf{W}$, which contains the spatial weights. However, it is important to note that in both strategies, the inclusion of spatial information does not guarantee that the resulting groups will be spatially adjacent.

To ensure strict contiguity in the clustering process, a spatial weight matrix must be utilized, grouping only those units that belong to the same neighborhood. Openshaw (1977) introduced the automatic zoning problem (AZP), which implements a clustering methodology that considers spatial restrictions in a strict form. The problem consists of progressively assigning areas, into K contiguous regions. The solution is achieved through an iterative process that begins with an initial partition, P_p^0. During this process, areas are interchanged between neighboring regions based on a global target function that indicates whether these changes result in improvements. However, this methodology requires substantial computational resources and is prone to local solutions that may prevent reaching the optimal outcome. As a solution to this limitation, Openshaw and Rao (1995) proposed utilizing local search mechanisms, such as tabu search and simulated annealing.

The previous methodology achieves very good results, but it requires the number of regions to be either known or arbitrarily defined by the researcher. To address this limitation, Duque et al. (2011) and Duque et al. (2012) increased the complexity of the problem by incorporating the choice of the number of groups into the algorithm. This approach, called Max-p, identifies the optimal number of groups that meets a threshold established based on a population variable (for example, the number of persons per region). Once a suitable partition is found, the groups are refined in a manner similar to the AZP method. This notable feature of this methodology, that is the number of regions is determined endogenously through an intriguing algorithm, has had undergone different adaptations, as can see in the works of Li et al. (2014), She et al. (2017), and Ye et al. (2018).

By following hierarchical methodologies, it is also possible to adapt the methods to solve the problem of spatial constraints. The methodology called Spatial Constrains into an Agglomerative Hierarchical Clustering (SCHC), developed by Murtagh (1985), Gordon (1996), among others, constructs hierarchical trees constrained by a graph derived from the contiguity matrix. In more recent years, Assunção et al. (2006) propose an spatial proceed called SKATER (Spatial 'K'luster Analysis by Tree Edge Removal). The method consists of using the Minimum Spanning Tree (MST) to prune the relationships between spatial units in a way that improves the target function. The MST consists of connecting all units with a graph that passes through each node once. Therefore,

the graph contains n nodes and $n-1$ edges. For this, each time an edge is pruned, the tree is divided into two. This process will continue until the desired number of regions is found.

Another hierarchical method is called REDCAP (Regionalization with Dynamically Constrained Agglomerative Clustering and Partitioning), proposed by Guo (2008). This methodology consists of complementing hierarchical methods with a differential treatment for spatial contiguity. In fact, it combines six methodologies: four ways to link the units and two ways to handle spatial contiguity. One key advantage of using these methodologies is their lower demand for computational resources compared to approaches like AZP or Max-p

On the other hand, time series clustering presents distinct challenges, primarily centered around how the distance between two sequences is measured. When we work with time series clustering, what is usually sought is that the curves coincide in time, that is, within the same group we have similar behavior in time. The more popular metrics are: the Euclidean distance used in Faloutsos et al. (1994) or the Dynamic Time Warping (DTW) used by Sakoe and Chiba (1978). The latter is not really a distance, but it is very useful for measuring the disguise of two temporary series, although it is a more complicated process from computational processing. Adaptations of clustering algorithms have been used in the literature to account for temporal groupings. These approaches generally aim to transform the space and capture the shapes of time series. For instance, Huang et al. (2016) proposes smoothing sequences before clustering them using the *k-means* logic. Similarly, Paparrizos and Gravano (2015) focuses on highlighting the shapes of time series prior to clustering.

3 Methodological proposal

The algorithm presented in this section aims to determine regions where variables have followed the same trends, and showed a similar enhancement but differentiated from other regions. The impositions of contiguity restriction favor the visualization and can facilitate the application of specific measures. These measures will apply through regions defined by their characteristics, in contrast to the definitions of regions based on political limits, common in regional studies.

The problem addressed arises when dealing with areas or spatial units, which are the basic units of analysis, containing a series of variables that represent temporal sequences. Within the framework of this article, we will call attributes to these variables, regions to a set of areas with similar characteristics belonging to the same neighborhood and territory to the set of all areas.

Formally, let $A = A_1, A_2, \ldots, A_n$ be the set of all areas, where $n = |A|$.

The attributes of the i-th area are represented by a matrix $\mathbf{X}_i$, composed of p attributes and a maximum of T periods, such that $\mathbf{X}_i = \{\mathbf{x}_{i1}, \ldots, \mathbf{x}_{ip}\}$ where $\mathbf{x}_{il} = \{x_1, x_2, \ldots, x_T\}$ is a vector that captures the temporal sequence of possible attribute values of $l \in \{1, \ldots, p\}$.

To establish relationships between areas, we define a dissimilarity function $d : A \times A \to \mathbb{R}^+ \cup \{0\}$ based on the attribute matrix $\mathbf{X}$. This function $d_{ij} = d(A_i, A_j)$ must satisfy $d_{ij} \geq 0$, $d_{ij} = d_{ji}$, and $d_{ii} = 0$, $\forall i, j = 1, 2, \ldots, n$.

Let $G = (V, E)$ be the graph or network associated with A, where node $v_i \in V$ corresponds to area $A_i \in A$ and edge $\{v_i, v_j\} \in E$, if and only if areas A_i and A_j belong to the same neighborhood $\mathcal{V}$. Note that graph G, in this case, is similar to matrix $\mathbf{W}$, defined in the previous section, which relates with a 1 if A_i and A_j are neighboring areas and with 0 otherwise.

Let $P_K = \{R_1, R_2, \ldots, R_K\}$ be a partition of areas A into K regions with $1 \leq p \leq n$. We denote Π as the set of all possible partitions of A.

3.1 Characteristics Space

Time series clustering pursues different objectives compared to traditional feature clustering. In this context, the focus lies in analyzing the shapes of time series and grouping them based on their similarities. Specifically, we aim to identify series that exhibit simultaneous increases, reach peaks and valleys at similar times, share analogous slopes, and display other distinctive features indicating a relationship among the series. For spatio-temporal data, the nature of the relationship is additionally influenced by the graph G. Thus, it is not sufficient for the series to demonstrate similar behaviors; they must also exhibit a spatial neighborhood connection.

Temporal series, in this context, represent the evolution of a characteristic across different areas over a given period. This means that an attribute $\mathbf{x}_{il}$ of area i is a vector of dimension T, as it has a value for each analyzed period. Therefore, the characteristic space corresponds to an area-specific data matrix, $\mathbf{X}_i$, of dimensions $T \times p$. To enable comparisions with other areas, a vectorization function is aplied as follows: $\boldsymbol{\mathcal{X}}_i = \text{vec}(\mathbf{X}_i) = (x_{i11}, \ldots, x_{i1T}, x_{i21}, \ldots, x_{i2T}, , \ldots, x_{ip1}, \ldots, x_{ipT})$. It is worth noting that this methodology does not require all variables to share the same number of time periods, where T_l denotes the number of periods for characteristic l. However, it is essential for a given characteristic to maintain the same number of periods across all areas.

Once $\boldsymbol{\mathcal{X}}_i$ has been obtained, it must be compared with the remaining spatial units. To do so, a dissimilarity function is required. Among the most commonly used measures is the Euclidean distance, defined as:

$$d\left(A_i, A_j\right) = \sqrt{\sum_{l=1}^{p} \sum_{t=1}^{T} \left(x_{ilt} - x_{jlt}\right)^2},$$

where x_{ilt} is the value in period t for the characteristic l of the attribute matrix $\mathbf{X}_i$. This way of working can be seen in Wei and Keogh (2006) who use it to solve time series classification problems. The Euclidean distance is one of the most intuitive ways to measure differences between areas, since it considers each value in each period as an individual characteristic. However, this measure has notable limitations: it is highly sensitive to extreme values and requires that all time series possess equal length in order to be meaningfully compared.

Alternatively, another dissimilarity function is the dynamic time warping, DTW (Sakoe and Chiba, 1978). Unlike Euclidean distance, this technique does not require that all series have the same number of periods. DTW matches time series not in a direct comparison scheme period by period, but rather identifies the closest points between different periods and then calculates their

differences. This means that, for a given value x_{ilt}, your alignment point could be $x_{il,t-r}$, where $r \in 1, \ldots, t-1$. In this way, the temporal space is deformed to allow the comparison.

Other approaches to comparing time series include the use of correlation and cross-correlation. Correlation is a useful tool for determining how sequences fluctuate as a whole, and various types of correlation can be used. For its part, cross-correlation measures the correlation between two sequences, generating a new series whose peaks can indicate similarities between the original series. For more details on these comparisons, we recommend see Alqahtani et al. (2021), which offers a comprehensive review of the literature on time series clustering and classification.

3.2 Feature extraction

When time series are large, adequate pre-processing is essential to generate an accurate representation of the data. If each time point is treated as a distinct feature, the dimensionality of the problem increases significantly. Often, many of these features are not relevant for meaningful comparison, as some variables exhibit distinctive behavior only during specific intervals—intervals that may define their similarity or dissimilarity with other series. However, if all time periods are considered equally, similarity measures—such as Euclidean distance—may yield deceptively high values due to the inclusion of numerous irrelevant periods. This can obscure the true relationships between series and reduce the effectiveness of the comparison.

Relevant feature extraction is a key strategy for mitigating the effects of high dimensionality and improving the accuracy of time series comparison. There are various methods for reducing dimensionality. A common approach involves calculating temporal averages for each variable and using these as the basis for comparison, or applying other statistical measures that serve the same purpose. Domain expertise can also be leveraged to identify the most significant features—such as periods characterized by breakouts, trends, peaks, or declines. However, the selected features must adequately represent the overall behavior of the series to ensure accurate classification and meaningful comparison.

Another approach to similarity comparison involves applying transformations to the feature space in order to identify a latent variable space. This method enables automatic feature extraction through algorithms designed to generate variables that are representative of the entire dataset. The goal is finding a function f: $\mathbb{R}^{pT} \to \mathbb{R}^h$ con $h < pT$. For this purpose, each area will be transformed their attributes in a latent variables vector $f(\boldsymbol{\mathcal{X}}_i) = \boldsymbol{h}_i$, whose dimension will be smaller than that of the original vectors. Them, the distance will be calculated on this vectors $d_{ij} = d(\boldsymbol{h}_i, \boldsymbol{h}_j)$.

One technique that operates within this methodology is Principal Component Analysis (PCA), that is a linear dimensionality reduction method, which transforms the data into a new set of variables—called principal components—based on the eigenvalues of the covariance matrix. These components are orthogonal to each other, ensuring that the resulting variables are statistically independent and capture the most significant variance in the data. Applications that use PCA are numerous, such as Singhal and Seborg (2005) and Li (2019).

In contrast, nonlinear methods offer the advantage of more effectively representing data in lower-dimensional spaces, providing greater flexibility and serving as valuable alternatives for clustering in complex contexts. When the dataset is sufficiently large—as is often the case with spatio-temporal data—nonlinear techniques such as t-Distributed Stochastic Neighbor Embedding (t-SNE) (Van der Maaten and Hinton, 2008) and AutoEncoders (AE) tend to produce more accurate and insightful results. Building on the capabilities of these algorithms, a spatially constrained clustering mechanism is proposed.

3.3 Autoencoders

Our methodological proposal operates on latent variables. Given a panel data structure, the resulting number of features can be considerably high, making it an ideal scenario for applying dimensionality reduction techniques such as AE.

In spatiotemporal problems, area-specific attributes can be represented as temporally sequenced variables, where the realization of an attribute at time t is not independent of the realization of that same attribute in the period $t - r$. This temporal dependence enables the detection of underlying structures within the series using a set of h latent variables, where $h \ll pT$. These latent variables are capable of capturing and explaining the dynamic behavior of each spatial unit over time. To achieve this identification, it is necessary to find a reduction function $f(\boldsymbol{\mathcal{X}}_i)$ able to shrink the space in a vector $\mathbf{h}_i$ of h hidden features. One of the tools that has gained prominence in the neural network literature is the AE, which we propose to utilize as the core component of our regionalization methodology.

The AE is a tool in unsupervised learning that gained prominence with the rise of deep learning and neural networks. Unlike supervised neural networks, which aim to predict a target variable, the AE uses the same space for input and output. The process begins and ends in the same space to achieve this, requiring the design of a neural network with two stages. The first stage reduces the dimensionality of the input space, creating what is commonly referred to as a bottleneck. This stage is called the "encoder." Subsequently, a second stage is established to reconstruct the original input space from the bottleneck representation. This second stage is known as the "decoder".

More formally, following the development in Vincent et al. (2010), the method of AE can be described in the following way:

1. Encoders: The goal is to map a function f_θ that transforms an input vector $\boldsymbol{\mathcal{X}}_i$ into a hidden representation $\boldsymbol{h}_i$, commonly referred to as the **encoder**. This function typically takes the form of an affine transformation followed by a nonlinear activation function:

$$f_\theta\left(\boldsymbol{\mathcal{X}}_i\right) = \boldsymbol{h}_i = s\left(\boldsymbol{\Omega}\boldsymbol{\mathcal{X}}_i + \boldsymbol{b}_i\right), \tag{1}$$

where the parameter set is $\theta = \{\boldsymbol{\Omega}, \boldsymbol{b_i}\}$. Here $\boldsymbol{\Omega}$ represents a weighting matrix of size $h \times pT$; $\boldsymbol{b_i}$ is a vector of dimension h and s is a nonlinear function, commonly referred to as the "activation function" in neural network terminology (e.g., sigmoid, tanh, ReLU).

2. Decoders: The result of the encoder stage $\boldsymbol{h}_i$ (called hidden layer, hidden variables, or latent variables) is then used for reconstructing a vector $\boldsymbol{Z}_i$ of dimension pT that should resemble the input space. For this, we use a function from $\mathbb{R}^h \to \mathbb{R}^{pt} : \boldsymbol{Z}_i = g_{\theta'}(\boldsymbol{h}_i)$, referred to as the **decoder**:

$$g_{\theta'}(\boldsymbol{h}_i) = s'\left(\boldsymbol{\Omega'}\boldsymbol{h_i} + \boldsymbol{b_i'}\right), \tag{2}$$

with parameters $\theta' = \{\boldsymbol{\Omega'}; \boldsymbol{b_i'}\}$ of appropriate size.

Although $\boldsymbol{Z}_i$ aims to resemble the original input $\boldsymbol{\mathcal{X}}_i$, it should not be regarded as an exact replica. Instead, within a probabilistic framework, the conditional distribution $p(X_i|Z = \boldsymbol{Z}_i)$ serves as an approximation that reflects the input data with high confidence. This relationship can be articulated in terms of the decoder's parameters θ', through the formulation: $p(X_i|Z = g_{\theta'}(\boldsymbol{h}_i))$ which indicates that the probability depends on the hidden variables. This fact leads to the establishment of the neural network's loss function, which considers both the input values $\boldsymbol{\mathcal{X}}_i$ and the reconstruction values $\mathbf{Z}_i$. The main objective of the algorithm is represented as:

$$\mathbf{L} = \min\left\{\sum_{i=1}^{n} L(\boldsymbol{\mathcal{X}}_i, \mathbf{Z}_i)\right\},$$

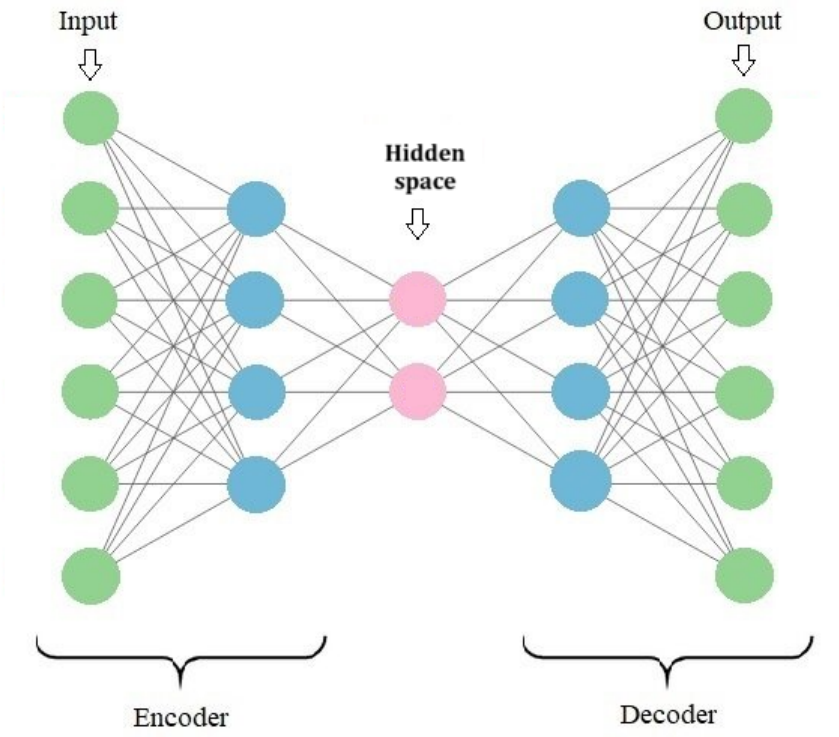
where $L(\boldsymbol{\mathcal{X}}_i, \mathbf{Z}_i)$ is the loss function for the area A_i. After minimizing the loss function and obtaining the parameter values θ and θ', the function is derived from the encoder stage, denoted as $f_\theta(\cdot)$, is reconstructed to be used for obtaining the latent variables.

The scheme presented in the equations (1) and (2) is shown without loss of generality using only one layer in Figure 1. However, in practice, neural networks often employ schemes with a greater number of layers to better capture the nonlinear relationships in the model.

In Figure 1, the general structure of the autoencoder (AE) is presented. The central part of the network depicts the latent variable space, which enables dimensionality reduction. From this stage onward, the network is designed to reconstruct the input data with the highest possible probability. Nevertheless, the resulting function tends to be difficult to interpret. The layered connections and the influence of multiple variables complicate the tracking of its internal behavior. At the same time, this complexity grants the model more flexibility when handling data aimed at grouping tasks.

As with clustering techniques, the primary objective of AE is to uncover meaningful groupings within the dataset—while model interpretability remains a secondary concern. The complexity of the regionalization problem has led to the use of these methods, as they enable simultaneous optimization of both groups and latent variables (for a more comprehensive review of the AE technique, see Goodfellow et al., 2016).

Figure 1: *AutoEncoders*



Note: Graphic representation of the autoencoders. Image extracted from López et al. (2023).

3.4 Spatial Restricted Deep Embedding Clustering

Neural networks are complex structures that establish relationships between the input and target variables through hidden layers. Deep Clustering is a method that utilizes neural networks and operates in two principal stages. In the first stage, the method is responsible for identifying and extracting the hidden latent variables that stay in the data, achieving a more simplified representation with a lower number of features, but maintaining a significant representation of structured data. In the second stage, the grouping is performed on this newly learned representation rather than the original input variables.

The procedure described above can be carried out sequentially: first, the latent variables are extracted, and then the grouping process is performed based on the learned representations (Huang et al., 2014; Tian et al., 2014). Alternatively, it can be performed jointly, allowing the algorithm to learn how to group the data and represent the latent characteristics simultaneously. This approach requires integrating the classification process into the neural network's structure (Song et al., 2013; Xie et al., 2016).

The Deep Embedding Clustering (DEC) method, introduced by Xie et al. (2016), is a clustering technique based on AE. Its results outperform many existing clustering approaches. While the

original algorithm is not designed for time series or for generating contiguous regions, it has been adapted to spatio-temporal problems. For example, Asadi and Regan (2019) applied it to group traffic data. More recently, Asadi and Regan (2021) added spatial information to the problem, expanding its application.

In what follows, we present an extension of the DEC method tailored to spatially constrained problems that also involve temporal correlations.

The method consists of three stages. The first stage works with the space of latent variables, for that, it is necessary to train a model under Autoencoders methodology that allows us to find the features more representative of the data. The temporal representation of the attributes of each area will be seen reduced to a hidden variables representation, that is $\mathbf{h}_i = (h_{i1}, h_{i2}, ..., h_{ih})$ where each h_{il} is the value of the latent variable l which is obtained by applying the encoder function $f_\theta(\boldsymbol{\mathcal{X}}_i)$. At this stage, we can see that this methodology is different from the other methodologies since now we are facing a transformed space and it is the one that will be grouped.

The second stage involves grouping the latent feature space with the K-means method. This stage provides an initial partition $P_K = \{R_1, R_2, ..., R_K\}$ where each region can be represented by its centroid $\{\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K\}$. Minimizing the distance of each area to the centroid of the region it belongs to is equivalent to the homogenization of areas within the region. For each spatial unit with which we are working, we can assign a probability of belonging to each one of the regions. This probability is calculated based on the distance between the latent vector and the centroid of the region $d\left(\mathbf{h}_i; \boldsymbol{\mu}_j\right)$. At this stage, the Euclidean distances are being interpreted in probabilistic terms. Small distances indicate a high probability that an area belongs to a particular group, while large distances suggest a low probability of belonging to that group. This form of grouping is known as a soft assignment. The t-SNE method typically uses the t-Student distribution, as shown below:

$$q_{ij} = \frac{\left(1 + \|\mathbf{h}_i - \boldsymbol{\mu}_j\|^2/\alpha\right)^{-\frac{\alpha+1}{2}}}{\sum_l \left(1 + \|\mathbf{h}_i - \boldsymbol{\mu}_l\|^2/\alpha\right)^{-\frac{\alpha+1}{2}}}, \tag{3}$$

where q_{ij} is the probability that the unit i belongs to the group j, and α represents the degrees of freedom of the distribution t-Student.

The algorithm's loss function must be differentiable to allow weight adaptation and result improvement. The Kullback-Leibler divergence serves as the loss function, measuring the distance between two probabilities P and Q as follows:

$$L = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}, \tag{4}$$

where p_{ij} represents the probability that unit i belongs to group j. This is the target probability we aim to achieve, while q_{ij} is the probability of group membership calculated using equation (3).

We need to determine the target probability P to apply the loss function correctly. The choice of this probability is crucial for both the algorithm's performance and achieving spatial contiguity

effects. Therefore, we need to find a probability that strengthens prediction, emphasizes points assigned with high confidence, and normalizes the contribution to the loss function to prevent large groups from distorting the hidden variable space. Following Xie et al. (2016), we can consider a probability function that squares the probability q_{ij} and normalizes it with the frequency of the cluster:

$$\bar{p}_{ij} = \frac{q * ij^2/g_j}{\sum_l q_{il}^2/g_l}, \tag{5}$$

where $q_j = \sum_i q_{ij}$ is the sum of the frequencies of the cluster.

3.4.1 Proposed methodology for spatial restriction

Equation (5) does not take into account the spatial constraint, which is a key requirement. Therefore, we need to modify it. Remember that the region concept is related to a set of spatial units that presents territorial contiguity. Therefore, one of the main objectives in regionalization methods is to ensure that grouped areas are not only similar in characteristics but also spatially contiguous. This means that we must define the concept of spatial neighborhood, which leads us to the use of the contiguity matrix $\mathbf{C}$.

We define $\mathbf{C}$ as an $n \times n$ matrix where two areas A_i and A_j are considered neighbors if $c_{ij} = 1$, $c_{ji} = 1$, or $c_{ij} = c_{ji} = 1$. Although this matrix is initially constructed based on research-specific assumptions, its structure is often arbitrary. In this methodology, however, the final relationships between spatial units are shaped dynamically by the algorithm itself.

Even when two areas are geographically close, they may not be assigned to the same region if the predefined neighborhood criteria do not consider them as contiguous. As a result, spatial proximity alone does not guarantee group inclusion—the contiguity framework plays a key role in guiding regional formation.

To consider the spatial restriction, it is necessary to modify the objective probability in equation (5). Given that we seek the contiguity in the regions, the probability that a unit belongs to a region depends on whether its neighbors belong to the same region Thus, the objective probability P must consider the probability of the neighboring areas defined by the contiguity matrix for each spatial unit. In this manner, to the initial probability is added the weighted probability that the other units belong to same group, multiplying each probability by the corresponding contiguity matrix :

$$p_{ij} = \bar{p_{ij}} + \frac{\left(\sum_k q_{kj}^2 c_{ik}\right)/q_j}{\sum_l q_{il}^2/q_l}, \tag{6}$$

where q_{kj} is the probability that unit k belongs to group j, and c_{ik} is the ik element of the contiguity matrix $\mathbf{C}$ being used. Note that, if the matrix used is the contiguity, this will make only the neighbor probability be computed.

The probability p_{ij} defined in the equation (6) represents the probability that the spatial unit A_i belongs to group j. Unlike the probability established in the equation (5), the updated formulation

considers the influence of neighboring units. This means that group membership is evaluated jointly: a spatial unit has a lower probability of belonging to a certain group if its surrounding units are assigned to different groups. This feature encourages the formation of contiguous regions, which improves the interpretability of the results. The key assumption here is the existence of a joint probability influenced by spatial proximity. This aligns with the core principle of spatial regional grouping—favoring groupings that are not only statistically similar but also spatially connected.

The third stage consists in the optimization of the loss function L adapting progressively the fresh probabilities. The optimization is realized with the habitual techniques for optimizing neural networks. The most well-known method is gradient descent, but adaptations of this technique can also be used. The choice of optimization method is a hyperparameter that can be adjusted while the results are improving.

The Spatial Restricted DEC (SDEC) is still a neural network with the last layer adapting to the regionalization. Therefore, the actualizing of internal parameters is carried out through the backpropagation mechanism, i.e. the net is adjusted from the last layers to the initial layers. The centroids $\left\{\boldsymbol{\mu}_j\right\}$ of each region are the first to be adapted since the units that compose each region change with each iteration. The actualizing rule follows the opposite path to the loss function gradient. For this, in the next step, it is optimized the weights and bias necessary for building the encoders $s\left(\boldsymbol{\Omega x}+\boldsymbol{b}\right)$, where the number of layers of the encoder is another hyperparameter that must be adjusted depending on the problem.

The probability Q assigns areas to regions based on a weak form. However, in this work, we aim to develop a strong assignment methodology, which requires establishing a specific rule for this process. Xie et al. (2016) propose assigning areas to the group with the highest probability Q. To ensure robust spatial contiguity, we suggest utilizing one of the existing restricted spatial clustering methodologies from the literature. Instead of grouping the original characteristics, this approach will group the calculated probabilities Q. Our proposal involves using a hierarchical method with spatial restrictions, which will yield the final results for the regions.

4 Behavior of methodologies under simulation

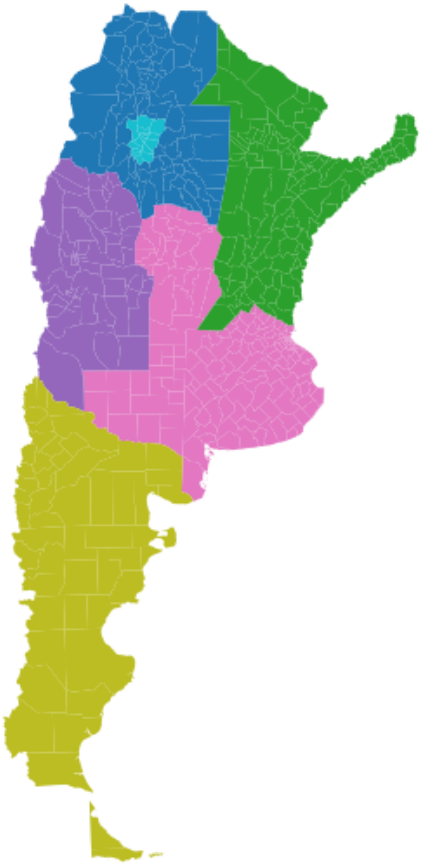
This section presents a comparative analysis between established spatially constrained clustering methodologies found in the literature and the approach proposed in this study. The goal is to assess the performance and characteristics of the proposed method in relation to existing techniques, particularly in the context of spatial grouping and regionalization. By applying these methods to representative datasets, we aim to highlight both the strengths and limitations of each approach, with a focus on how spatial constraints and latent representations are handled in practice.

Using spatial data from Argentina, we designed a simulation exercise at the departmental level. The goal is to create six fictional regions, each showing distinct behaviors across regions

but consistent characteristics within each one. This regionalization considers the evolution of two variables over one hundred time periods.

The temporal behavior in each region follows a negative quadratic function, with differences based on their maximum values, time spans, and the points where they intersect the x-axis—indicating the beginning and end of the observed phenomenon. Each spatial unit follows the curve associated with its assigned region. The target regionalization used in the simulation is shown in the Figure 2.

Figure 2: Target map



Source: Own elaboration based on simulated data

We propose two different scenarios with added random noise. In the first case, we add a random component to each unit drawn from a uniform distribution centered at zero with a variability range of 0.5 (values from -0.25 to 0.25). In the second case, the random component is similar but with a range of 1 (values from -0.5 to 0.5). Since the simulation aims to represent proportional behavior (values in the interval [0,1]), we add restrictions in both cases to prevent negative values.

The methodologies used for comparison are: (i) hierarchical method with spatial restriction calculated from Python's scikit-learn library; (ii) SKATER, (iii) REDCAP, and (iv) SCHC, all of

which were implemented using the rgeoda package in R. Each method was configured to produce six regions and used its default parameter settings.

Our SDEC clustering methodology was implemented in Python using the Keras and scikit-learn libraries. The initial configuration of SDEC begins with six autoencoders and six regions, using hierarchical clustering with spatial restriction as the spatial method. All cases use a k-nearest neighbor matrix with k equal to six. It is important to note that methods like SDEC involve a wide range of configurable parameters—such as the number of neurons, number of layers, number of clusters, the spatial weighting matrix, the network architecture, and other neural network settings. These parameters are selected with the aim of minimizing the loss function. However, given the vast number of possible combinations, it's impractical to exhaustively explore every variant. In both the simulations and the empirical case presented in the next section, we consider two characteristic variables with T time periods each. The variables are separated so that latent variables are extracted for each time series individually in the first layers. This means the neural network adapts the weights of each time series separately in the initial layers before combining them to form the output of the intended latent variables. The number of encoders adapts to the desired number of groups, meaning $k = h$, and all tests were conducted this way.

4.1 Results of the simulations

To evaluate the methodologies, we use four performance metrics commonly found in the literature. Since the data are simulated, we have the advantage of being able to use the so-called extrinsic metrics, which rely on the real information about how the data are grouped. In empirical cases, these metrics cannot be used, and alternative metrics must be employed to evaluate the algorithms.

In Table 1, the results proposed by each of the methodologies are presented according to four metrics: adjusted mutual information, mutual information, homogeneity, and rand score. All of these metrics are calculated using the Python library Scikit-Learn (Pedregosa et al., 2011). In each case, a higher value reflects a better fit of the method to the expected grouping structure.

SDEC consistently outperforms the other algorithms presented, achieving results very close to one for adjusted mutual information, homogeneity, and rand score—where one represents the maximum possible value. Additionally, SDEC also attains higher values in mutual information compared to other methodologies.

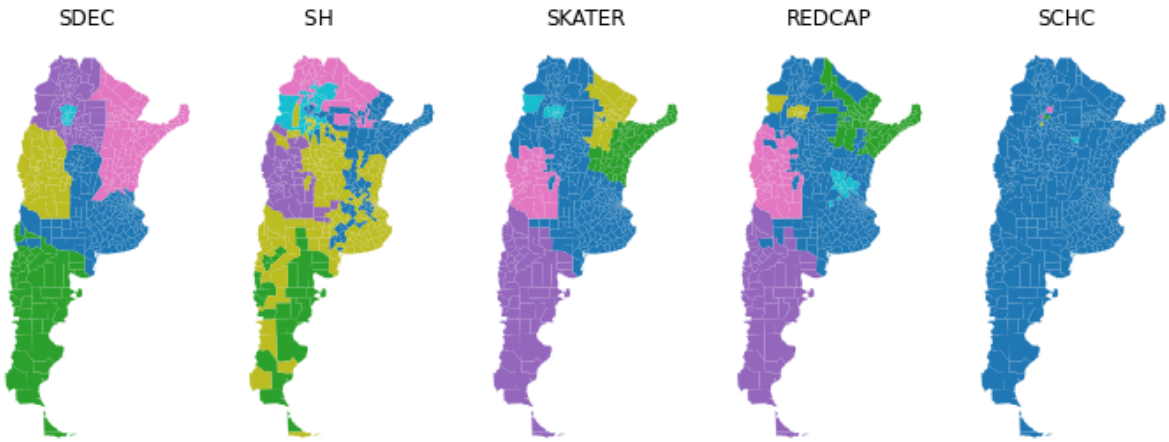
Table 1: Extrinsic metrics for the first case

	SDEC	SH	SKATER	REDCAP	SCHC
Adj Mutual Inf	**0,9622**	0,5711	0,6300	0,6064	0,0264
Mutual Inf	**1,5225**	0,9011	0,9124	0,8648	0,0377
Homog	**0,9597**	0,5680	0,5751	0,5451	0,0238
Rand score	**0,9890**	0,7635	0,7621	0,7530	0,2559

Note: performance metrics for each of the regionalization methodologies. SDEC: spatial deep embedding clustering, SH: spatial hierarchical, SKATER: spatial 'K'luster analysis by tree edge removal, REDCAP: regionalization with dynamically constrained agglomerative clustering and partitioning, SCHC: Spatial Constraints into an Agglomerative Hierarchical Clustering.

The results of the five methodologies can be seen in Figure 3. The map obtained by the SDEC algorithm is very similar to the target map shown in Figure 2. Both these graphics and the calculated metrics demonstrate that, for this simulation, SDEC successfully discovers the underlying patterns in the data—a task that proves more challenging for other methodologies.

Figure 3: Results over the first simulation



Note: Results of the regionalization with each technique of spatial clustering.

In the second scenario, where the data exhibit greater variance, the performance of all methodologies declines as expected. Nevertheless, they continue to follow the same relative pattern of results observed in the previous case. Again, SDEC performs better than the others across all calculated metrics (Table 2). Figure 4 shows the map structures generated by each algorithm. In this scenario, the resemblance to the original map is less evident compared to the previous case. However, the SDEC algorithm still manages to reveal a clearer spatial pattern. Despite the increased noise introduced in the simulation, the performance metrics validate

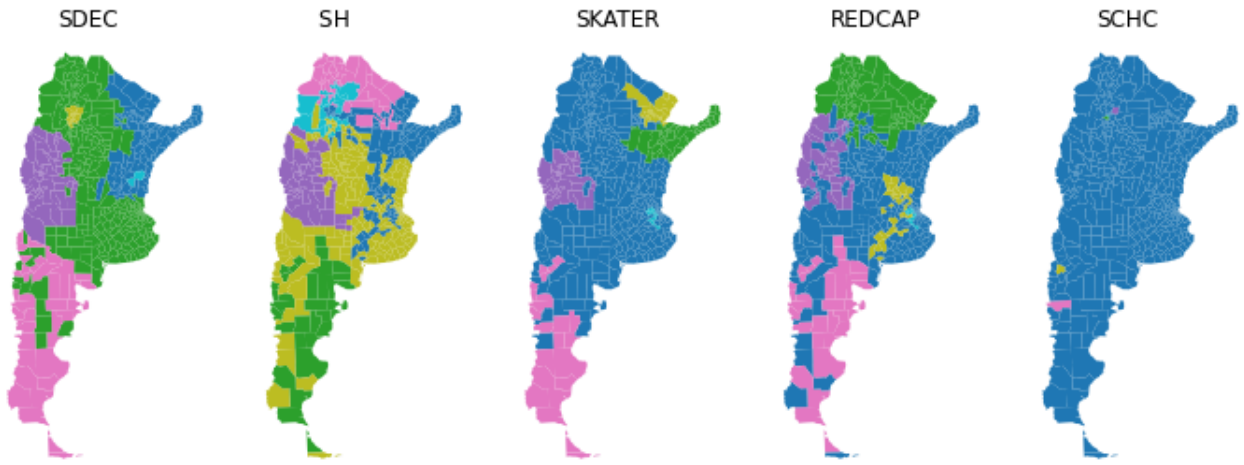
the method's reliability, indicating that it continues to produce satisfactory results under more challenging conditions.

Table 2: Extrinsic metrics for second case

	SDEC	SH	SKATER	REDCAP	SCHC
Adj Mutual Inf	**0,7248**	0,4481	0,4041	0,3729	0,0169
Mutual Inf	**1,0171**	0,7020	0,5359	0,5441	0,0289
Homog	**0,6411**	0,4425	0,3378	0,3430	0,0182
Rand score	**0,8086**	0,7584	0,5793	0,6660	0,2522

Note: performance metrics for each of the regionalization methodologies. SDEC: spatial deep embedding clustering, SH: spatial hierarchical, SKATER: spatial 'K'luster analysis by tree edge removal, REDCAP: regionalization with dynamically constrained agglomerative clustering and partitioning, SCHC: Spatial Constraints into an Agglomerative Hierarchical Clustering.

Figure 4: Results of the second simulation



Note: Results of the regionalization with each technique of spatial clustering.

Based on the simulation results, we conclude that using latent variables to capture the essential features of the underlying phenomena is a robust and effective strategy. Training autoencoders and then increasing the chance of each unit being assigned to a group helps the SDEC algorithm remove much of the random noise that does not carry useful information. This makes it easier for the method to identify the spatiotemporal patterns behind how the data behaves across space and time.

5 Empirical applications

In this section, we present two applications of the proposed methodology to illustrate its performance under different conditions and its potential for analyzing complex spatiotemporal patterns.

The first application focuses on the spread and impact of COVID-19 across departments in Argentina. Using fatality data relative to population from March 2020 to July 2021, we explore how the SDEC algorithm captures spatial heterogeneity and reveals meaningful groupings based on the evolution of the pandemic.

The second application analyzes the evolution of wages and employment levels in Argentina. Here, we examine how changes in these economic indicators over time can be grouped into regions with similar patterns, shedding light on underlying socioeconomic dynamics across the national territory.

As is common in clustering methodologies, in both cases to avoid distortions caused by features that stand out only due to their magnitude, we standardized and adjusted the variables in the database similarly to the simulated datasets section. This allowed us to use the same SDEC neuronal network that was built for the simulations.

To address the empirical limitations of clustering methodologies regarding the optimal number of groups, researchers have developed techniques such as the elbow plot. We employ this technique in this section to determine the appropriate number of groups.

5.1 Regionalization the impact of COVID-19 in Argentina

SARS-CoV-2 (COVID-19), which emerged in China in late 2019, causes a severe respiratory illness and spreads rapidly from person to person (Li et al., 2020). Due to its high transmission rate, the virus quickly spread beyond China's borders and evolved into a global pandemic. This fast expansion led to serious challenges from both health and socioeconomic perspectives. In response, many countries—including Argentina—enforced mandatory quarantines as a key strategy to control the virus.

The person-to-person spread of COVID-19 highlights the importance of using spatial models to study the pandemic. These models help reveal how the virus moves through populations and how physical and geographic proximity influence transmission. Because viral diffusion often follows spatial patterns, this approach is especially useful for understanding the pandemic's dynamics. For this reason, the SDEC methodology offers valuable information on the spatial differences in how infections and deaths evolved across Argentina's different regions. To implement SDEC, we analyzed the proportion of both variables relative to the total population for each department from March 2020 to July 2021.

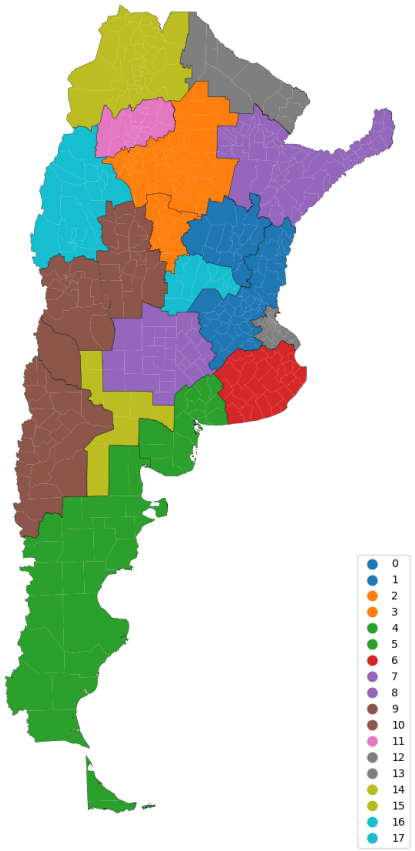
In the first stage of SDEC, the system trains an autoencoder to obtain encodings. The neural network initially processes both variables (infections and fatalities) separately before concatenating them to determine the encoders. It then implements a decoder for the entire set to

obtain the input matrix as previously explained. Processing the variables separately at first allows the algorithm to capture the temporal effects of each variable before considering their relationship.

Once the autoencoder training is complete, the encoding variables are used for initial clustering. In this case, we used the k-means methodology with the default parameters from the Scikit-Learn library. Finally, the initial clustering is optimized using the spatial restricted hierarchical methodology with a k-nearest neighbors matrix.

Figure 5 shows the regionalization results. An interesting feature of SDEC is that the regions created have similar numbers of areas, and the encoded variables help eliminate noise from potential outliers. As expected, the resulting spatial configuration confirms that the regions preserve geographic contiguity, which strengthens the interpretability of spatial patterns.

Figure 5: SDEC results for the evolution of the COVID19 in Argentina.



Note: Own elaboration based on data from the SDEC clustering.

Table 3 reports the performance metrics of the algorithm for the seventeen clusters generated. The RBTSS (Ratio of Between-Group to Total Sum of Squares) reflects the proportion of between-cluster variance relative to total variance. The first value corresponds to the calculation based on the original variables, while the second derives from the encoder variables constructed by

the model. The difference between both is substantial: the first yields a value of 0.29, whereas the second reaches 0.98. This discrepancy arises because the encoder variables are specifically generated by the algorithm to maximize differentiation between clusters based on the most relevant features. In any case, the value of 0.29 obtained from the original variables represents an acceptable result, considering the number and heterogeneity of variables involved.

Additionally, we assessed the reconstruction loss using cosine similarity. Although the resulting value (0.15) is not particularly high, it should be noted that the primary objective of the autoencoder within the SDEC framework is not precise data reconstruction. Its main function is to generate encoder variables that facilitate optimal cluster differentiation. In this context, that objective is clearly achieved: the RBTSS value of 0.98, calculated over the encoders, indicates that the codification process effectively captures the main structural features of each region.

The remaining metrics—Silhouette Coefficient, Calinski-Harabasz Index, and Davies-Bouldin Index—are standard intrinsic measures for evaluating clustering quality, and in this case, were all computed based on the encoder variables. The results for all three metrics indicate satisfactory model performance. The Silhouette Coefficient, which ranges between -1 and 1 (with values closer to 1 indicating better-defined clusters), presents a favorable outcome. The Calinski-Harabasz Index, for which higher values indicate better clustering, and the Davies-Bouldin Index, where lower values reflect better performance, further corroborate the robustness of the clustering solution.

Table 3: Performance metrics of the model.

Metric	Category	Value	Description
RBTSS (Original)	Similarity	0.29	Baseline reconstruction quality
RBTSS (Encoders)	Similarity	0.98	Enhanced reconstruction post-encoding
Reconstruction loss	Error	0.15	Loss value from autoencoder reconstruction
Coefficient Silhouette	Clustering	0.65	Cohesion and separation of clusters
Calinski-Harabasz Index	Clustering	1180.79	Cluster dispersion ratio
Davies-Bouldin Index	Clustering	1.20	Intra-cluster similarity vs separation

Note: Values reflect reconstruction accuracy and clustering effectiveness.

Evaluation and interpretation of the results

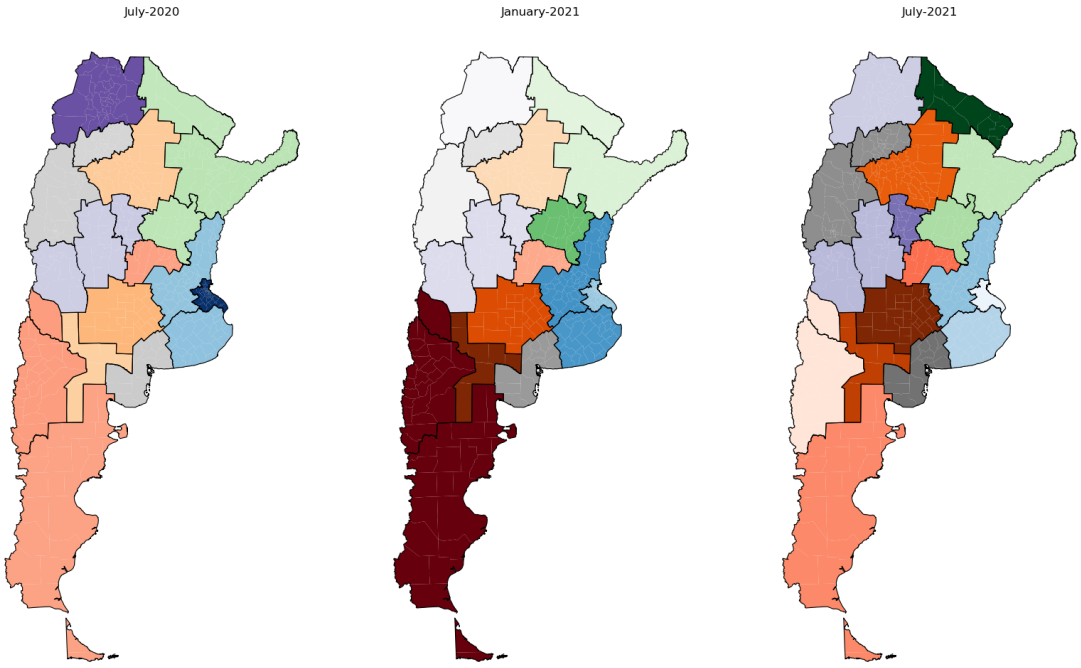
Regarding the empirical results (Figure 5), it is interesting to note how some detected clusters coincide with provincial boundaries, highlighting a homogeneous behavior among departments belonging to the same province. For example, northern provinces such as Formosa, Jujuy, and Misiones showed consistent intra-provincial patterns, with all their departments belonging to the same cluster. A similar trend is observed in southern provinces such as Santa Cruz and Tierra del Fuego. However, many clusters include departments from different provinces, suggesting that inter-provincial mobility controls were not effective in disrupting the localized dynamics

of COVID-19. This outcome is particularly evident in the province of Buenos Aires, where five distinct clusters can be observed. A similar pattern appears in the province of Córdoba, where four clusters contain departments from within the province.

The interpretation of these findings lies in understanding the significance of this regional differentiation. It implies that departments within the same region exhibit similar behavior (spatial homogeneity), while displaying significant differences from departments in adjacent regions (spatial heterogeneity). This is one of the key insights detected by the methodology. Although the spatial contiguity constraint groups departments together, the separation criterion ensures that two neighboring regions must, by definition, behave differently.

The previously displayed map served only to delineate the constructed regions. To effectively visualize the behavioral differences between them, new graphics are presented. In the Figure 6, three specific months for the COVID-19 case (indicated in the subtitle). The SDEC regions are outlined in black on the maps, and distinct color palettes are assigned to each region—some palettes repeat because of the large number of clusters. Each region is visualized using a color scale that reflects the intensity of the analyzed variable across the three selected periods. Lighter colors denote lower values, while deeper and more vivid colors signify higher intensities, with saturation peaking at the maximum end of the scale.

Figure 6: Evolution of contagious for COVID-19 in Argentina.



Note: Own elaboration based on data from the SDEC clustering.

This figure shows that regions exhibit different inter-regional dynamics. In the case of infections for COVID-19, the high values of contagious beginning in the region center, around Buenos Aires and then the more intensity of variable locate in the south of the country. For last the more critical regions was the north regions of Argentina.

5.2 Regionalization of salary and workers in Argentina

The second empirical case analyzes the registered workers and their wages across Argentina's departments. Argentina has a large and diverse territory, which creates considerable variation and heterogeneity in economic activities across regions.

Using the same approach applied in the simulations and the COVID-19 case, we selected two variables recorded over multiple time periods. The first variable measures average wage relative to the department's population, and the second reflects the number of workers relative to the population for each department too. These indicators vary across 506 departments, although some lack data due to their remote locations. For this application, we use monthly data, treating each month as a separate input variable for clustering. The period of analysis spans from January 2014 to November 2023.

To handle the wide variability and dynamic behavior of the data, it is important to stabilize the variable trends. Compared to the COVID-19 case data, which showed more stable and predictable patterns, economic indicators are more complex and sensitive to local conditions. For example, in smaller departments with agriculture-based economies, seasonal worker inflows for certain activities can strongly affect local statistics. Meanwhile, in larger and more diverse cities, such changes tend to have less impact on overall trends.

To smooth out these fluctuations, we use a moving average over twelve periods and express the data in logarithmic form. This helps reduce high variability and sudden changes, making the patterns easier to interpret and compare across regions.

Table 4 like in earlier case shows the performance metrics of the algorithm for the twenty clusters created. In this case, the metrics showed lower performance compared to the COVID example presented earlier. The RBTSS calculated on the original variables, with a value of 0.13, indicates that the algorithm struggled to capture variance through clustering. However, the same ratio computed on the encoder variables still shows a strong result, reaching 0.9. This again highlights that the objective of generating variables capable of differentiating clusters was successfully achieved, even though the reconstruction performance, with a value of 0.18, was lower. Nevertheless, it should be noted that these reconstruction values still indicate that the algorithm is moving in the right direction on this aspect as well.

Regarding the other metrics, it can be stated that the results are acceptable. The Silhouette Coefficient, with a value of 0.37, is reasonable for real-world problems involving a large number of groups and high heterogeneity, as in this case. Similarly, the remaining indicators, although not yielding particularly strong results, still fall within acceptable ranges for problems of this nature.

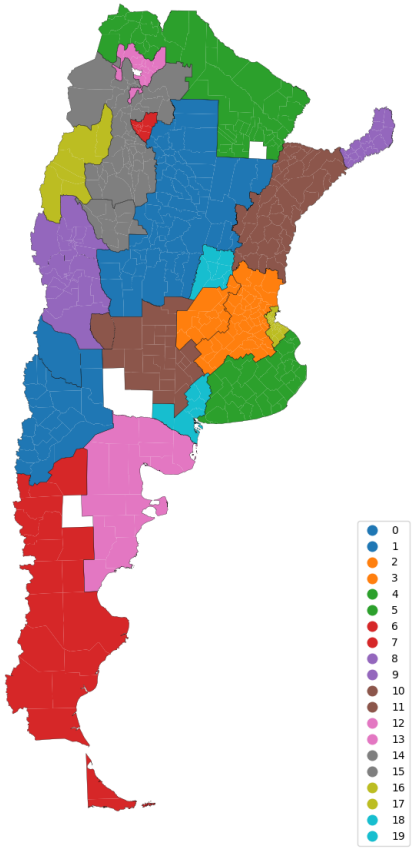
Table 4: Performance metrics of the model.

Metric	Category	Value	Description
RBTSS (Original)	Similarity	0.13	Baseline reconstruction quality
RBTSS (Encoders)	Similarity	0.90	Enhanced reconstruction post-encoding
Reconstruction loss	Error	0.18	Loss value from autoencoder reconstruction
Coefficient Silhouette	Clustering	0.37	Cohesion and separation of clusters
Calinski-Harabasz Index	Clustering	202.90	Cluster dispersion ratio
Davies-Bouldin Index	Clustering	2.52	Intra-cluster similarity vs separation

Note: Values reflect reconstruction accuracy and clustering effectiveness.

Figure 7 shows the regionalization results. As expected, the map shows that the regions maintain spatial contiguity.

Figure 7: SDEC results for the evolution of the wage and workers in Argentina



Note: Own elaboration based on data from the SDEC clustering.

Due to the wide variety of activity workers, the number of regions is twenty. A small number of regions in this partition fails to produce the desired level of within-region homogeneity and between-region heterogeneity for the evaluation. The number of regions was selected using the elbow method, a visual heuristic that helps identify the point at which further partitioning yields diminishing returns in model fit.

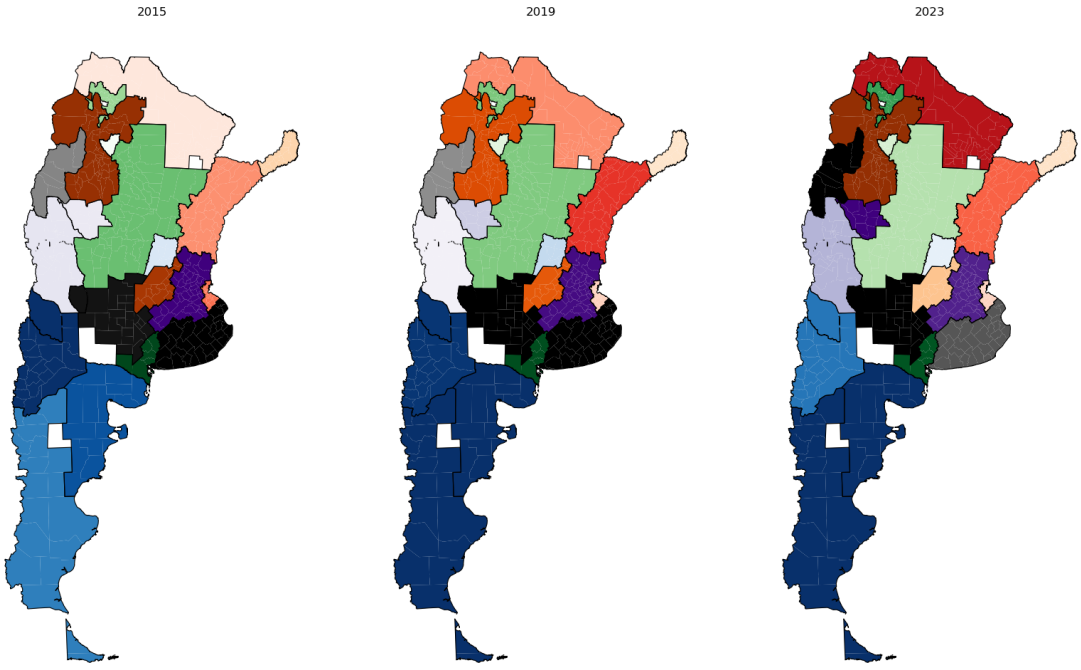
Evaluation and interpretation of the results

Dynamics of the wage regions are more complex than COVID-19 case. In this example we take the average for year for each variable used for a better representations. Some clusters had high wages in 2015 and then saw those wages decline over subsequent years; conversely, other clusters did not follow this pattern.

In the Figure 8, different time periods have been selected: three different years for the salary data (calculated as annual averages).

Each region is represented by a color scale that indicates the intensity of the analyzed variable. Light shades correspond to low values of the variable, whereas more intense or vivid colors represent the highest values, reaching maximum saturation at the peak of the scale.

Figure 8: Evolution of wage for region in the respective year.

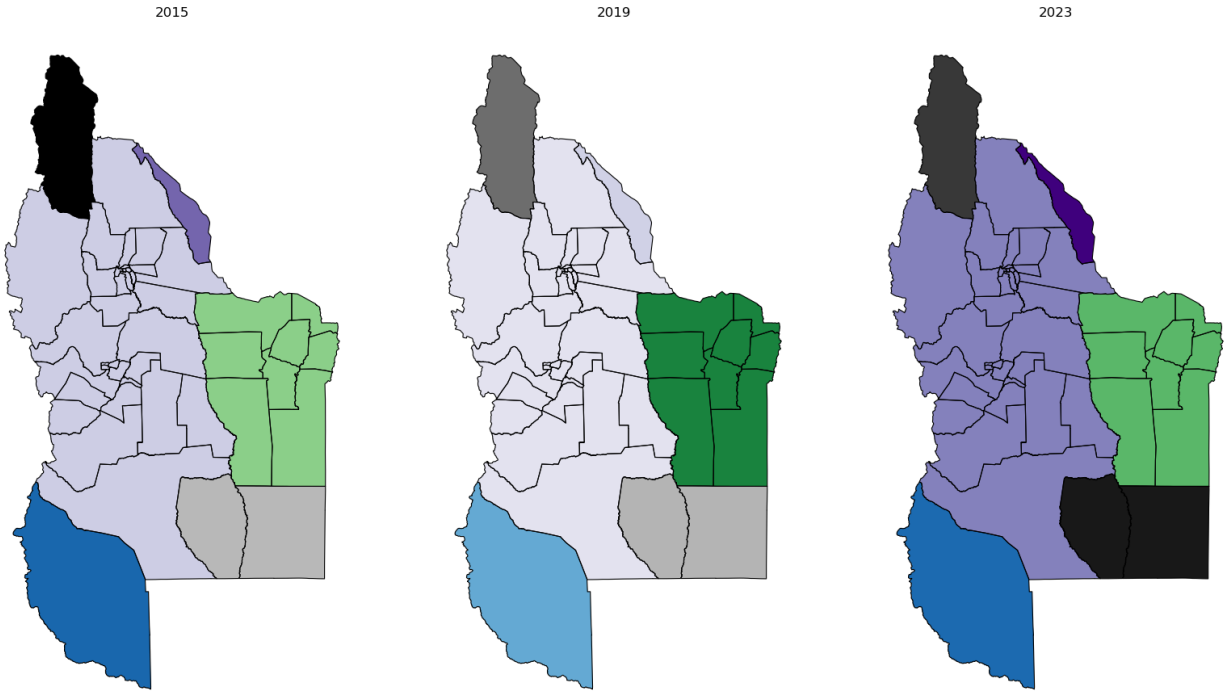


Note: Own elaboration based on data from the SDEC clustering.

For better visualization, Figure 9 presents a zoomed-in view of a specific area—Cuyo. This region, traditionally considered a homogeneous unit in Argentine economic analysis, reveals a

different reality when analyzed using the regionalization criteria proposed by SDEC. According to this methodology, the regional territory is fragmented into distinct sub-regions, defined over the same geographical space. Within this framework, it can be observed that the largest sub-region identified within Cuyo exhibited a slight increase in per capita wages during the initial year of the analyzed period, followed by a decline and, by the end of 2023, a significant recovery compared to previous years. In contrast, the remaining sub-regions did not follow similar trajectories. For instance, the sub-region represented in green began at an intermediate wage level, reached its peak in 2019, and then experienced a decline through to 2023. In summary, the dynamics identified through the regionalization proposed by SDEC reveal differentiated patterns that would remain obscured under conventional regional delimitation.

Figure 9: Evolution of wage for region in the respective years



Note: Own elaboration based on data from the SDEC clustering.

Table 5 presents the summary statistics for the SDEC-defined regions within the traditional region of Cuyo. Moreover, while the maps in earlier sections focus on selected reference years, the table reports the complete temporal dynamics expressed in mean of year from 2014 to 2023, allowing for a more comprehensive analysis.

The table highlights the evolution of the indicators for each SDEC-region over time. Note that for presentation clarity, only the means are emphasized for the years 2015, 2019, and 2023, corresponding to the years previously visualized in the maps.

Table 5: Temporal evolution of indicators, 2014–2023

Region	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
0	0.22	**-0.13**	-0.32	0.14	0.26	**-0.28**	-0.36	-0.19	0.12	**-0.14**
1	-0.32	**-0.33**	-0.05	-0.13	-0.01	**-0.13**	-0.40	0.00	-0.18	**-0.26**
9	-0.32	**-0.40**	-0.47	-0.34	-0.26	**-0.47**	-0.38	-0.22	-0.23	**-0.23**
11	-0.37	**-0.36**	0.15	-0.32	-0.27	**-0.35**	-0.39	0.06	0.23	**-0.05**
15	-0.30	**-0.18**	0.16	0.15	-0.22	**-0.41**	-0.44	-0.16	0.44	**-0.01**
17	-0.29	**0.15**	0.06	-0.01	0.36	**-0.21**	0.26	0.60	0.05	**-0.11**

Note: Mean values for selected regions from SDEC algorithm.

6 Conclusions and Future Agenda

This study presents Spatial Deep Embedded Clustering (SDEC) as a comprehensive solution to one of the long-standing challenges in regional science: the simultaneous preservation of spatial contiguity and temporal coherence in socioeconomic data analysis. By merging deep autoencoding techniques with spatially constrained clustering, SDEC captures latent temporal dynamics while adhering to spatial boundaries—an essential capability given the growing complexity of modern datasets.

Through Monte Carlo simulations and real-world applications, SDEC demonstrates superior performance over classical methods, offering improved regional delineation that is both internally homogeneous and spatially cohesive. By improving the precision and interpretability of regional delineation, this innovation holds promise for domains where spatial segmentation and temporal analysis are foundational—particularly public health, policymaking, urban development, labor economics, regional growth diagnostics, and spatial logistics modeling.

Empirical examples illustrate the benefits of the proposed SDEC methodology. As observed, the algorithm captures the spatial heterogeneity present within the study area, delineating regions that exhibit distinct dynamic behaviors. This is particularly evident when comparing the temporal trajectories of the different SDEC-regions. In contrast to traditional approaches that rely on provincial boundaries or standard regional divisions—such as the Cuyo region considered in this case—the SDEC methodology enables a data-driven construction of functional economic regions. This approach allows for a more accurate representation of the spatial structure underlying the phenomenon analyzed, overcoming the limitations of predefined administrative units.

Beyond its technical contribution, the SDEC framework provides a scalable and flexible platform for spatial data scientists, enabling more informed and data-driven decision-making.

Future work can extend SDEC's capabilities in several directions. One area is the detection of non-contiguous regions that behave similarly over time. Enhancing the model to identify spatially separate but functionally similar areas would provide deeper insight into regional patterns—such as synchronized economic activity or mobility flows—that cross administrative boundaries.

Another promising direction is integrating real-time data. This would allow SDEC to update regional clusters dynamically, which is especially important for urgent situations like disease outbreaks or natural disasters where quick identification of shifting hotspots enables timely interventions.

Finally, causal analysis is also a useful extension. By embedding causal models into SDEC, researchers can evaluate how policies affect spatial clusters over time. This would improve the measurement of localized impacts from interventions such as health campaigns or financial support programs, while accounting for spatial and temporal dependencies.

These extensions would strengthen SDEC's role as a powerful tool for spatial econometric analysis, reinforcing the relevance of deep learning in exploring dynamic, high-dimensional spatial data.

References

Alqahtani, A., Ali, M., Xie, X., and Jones, M. W. (2021). Deep time-series clustering: A review. *Electronics*, 10(23):3001.

Anselin, L., Syabri, I., and Kho, Y. (2010). GeoDa: an introduction to spatial data analysis. In *Handbook of Applied Spatial Analysis*, pages 73–89. Springer.

Asadi, R. and Regan, A. (2019). Spatio-temporal clustering of traffic data with deep embedded clustering. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Prediction of Human Mobility*, pages 45–52.

Asadi, R. and Regan, A. (2021). Clustering of time series data with prior geographical information. *arXiv preprint arXiv:2107.01310*.

Assunção, R., Neves, M., Câmara, G., and da Costa Freitas, C. (2006). Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees. *International Journal of Geographical Information Science*, 20(7):797–811.

Berry, B. J. L. (1964). Approaches to regional analysis: A synthesis. *Annals of the Association of American Geographers*, 54(1):2–11.

Chavent, M., Kuentz-Simonet, V., Labenne, A., and Saracco, J. (2018). ClustGeo: an R package for hierarchical clustering with spatial constraints. *Computational Statistics*, 33(4):1799–1822.

Cheruvelil, K. S., Yuan, S., Webster, K., Tan, P.-N., Lapierre, J.-F., Collins, S., Fergus, E., Scott, C., Norton Henry, E., Soranno, P. A., Filstrup, C., and Wagner, T. (2017). Creating multithemed ecological regions for macroscale ecology: Testing a flexible, repeatable, and accessible clustering method. *Ecology and Evolution*, 7(9):3046–3058.

Combes, P.-P. and Duranton, G. (2006). Labour pooling, labour poaching, and spatial clustering. *Regional Science and Urban Economics*, 36(1):1–28.

Duque, J., Anselin, L., and Rey, S. (2012). The Max-P regions problem. *Journal of Regional Science*, 52(3):397–419.

Duque, J., Church, R., and Middleton, R. (2011). The p-Regions Problem. *Geographical Analysis*, 43(1):104–126.

Duque, J. C., Ramos, R., and Suriñach, J. (2007). Supervised regionalization methods: A survey. *International Regional Science Review*, 30(3):195–220.

Faloutsos, C., Ranganathan, M., and Manolopoulos, Y. (1994). Fast subsequence matching in time-series databases. *ACM Sigmod Record*, 23(2):419–429.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Gordon, A. (1996). A survey of constrained classification. *Computational Statistics & Data Analysis*, 21(1):17–29.

Guo, D. (2008). Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP), sapatially. *International Journal of Geographical Information Science*, 22(7):801–823.

Haining, R., Wise, S., and Ma, J. (2000). Designing and implementing software for spatial statistical analysis in a GIS environment. *Journal of Geographical Systems*, 2(3):257–286.

Huang, P., Huang, Y., Wang, W., and Wang, L. (2014). Deep embedding network for clustering. In *2014 22nd International conference on pattern recognition*, pages 1532–1537.

Huang, X., Ye, Y., Xiong, L., Lau, R. Y., Jiang, N., and Wang, S. (2016). Time series k-means: A new k-means type smooth subspace clustering for time series data. *Information Sciences*, 367:1–13.

Isard, W. (1956). Regional science, the concept of region, and regional structure. *Papers in Regional Science*, 2(1):13–26.

Jain, A. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666.

Li, H. (2019). Multivariate time series clustering based on common principal component analysis. *Neurocomputing*, 349:239–247.

Li, Q., Guan, X., Wu, P., Wang, X., Zhou, L., Tong, Y., Ren, R., Leung, K., Lau, E., and Wong, J. (2020). Early transmission dynamics in Wuhan, China, of novel coronavirus–infected pneumonia. *New England Journal of Medicine*.

Li, W., Church, R., and Goodchild, M. (2014). An extendable heuristic framework to solve the p-compact-regions problem for urban economic modeling. *Computers, Environment and Urban Systems*, 43:1–13.

López, F., Mora, A., and Magán-Carrión, R. (2023). Influencia de la selección de hiper-parámetros en el rendimiento de autoencoders para la detección de ataques en red. In *Actas de las VIII Jornadas Nacionales de Investigación en Ciberseguridad. Universidad de Vigo*.

Murtagh, F. (1985). A survey of algorithms for contiguity-constrained clustering and related problems. *The Computer Journal*, 28(1):82–88.

Nathan, M. and Overman, H. (2013). Agglomeration, clusters, and industrial policy. *Oxford review of economic policy*, 29(2):383–404.

Openshaw, S. (1977). A geographical solution to scale and aggregation problems in region-building, partitioning and spatial modelling. *Transactions of the Institute of British Geographers*, pages 459–472.

Openshaw, S. and Rao, L. (1995). Algorithms for Reengineering 1991 Census Geography. *Environment and Planning A: Economy and Space*, 27(3):425–446.

Paparrizos, J. and Gravano, L. (2015). k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1855–1870.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Portnov, B. (2006). Urban clustering, development similarity, and local growth: A case study of Canada. *European Planning Studies*, 14(9):1287–1314.

Rey, S., Arribas-Bel, D., and Wolf, L. J. (2023). *Geographic data science with python*. Chapman and Hall/CRC.

Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.

She, B., Duque, J., and Ye, X. (2017). The network-max-p-regions model. *International Journal of Geographical Information Science*, 31(5):962–981.

Singhal, A. and Seborg, D. (2005). Clustering multivariate time-series data. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 19(8):427–438.

Song, C., Liu, F., Huang, Y., Wang, L., and Tan, T. (2013). Auto-encoder based data clustering. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 18th Iberoamerican Congress, CIARP, Havana, Cuba, November 20-23, Proceedings, Part I 18*, pages 117–124. Springer.

Tian, F., Gao, B., Cui, Q., Chen, E., and Liu, T.-Y. (2014). Learning deep representations for graph clustering. In *Proceedings of the AAAI conference on artificial intelligence*.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11).

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., and Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(12).

Webster, R. and Burrough, P. (1972). Computer-Based soil mapping of small areas from sample data: I. Multivariate Classification and Ordination. *Journal of Soil Science*, 23(2):210–221.

Wei, L. and Keogh, E. (2006). Semi-supervised time series classification. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 748–753.

Wise, S., Haining, R., and Ma, J. (1997). Regionalisation tools for the exploratory spatial analysis of health data. In *Recent developments in spatial analysis*, pages 83–100. Springer.

Xie, J., Girshick, R., and Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487.

Ye, X., She, B., and Benya, S. (2018). Exploring regionalization in the network urban space. *Journal of Geovisualization and Spatial Analysis*, 2(1):4.

Yuan, S., Tan, P.-N., Cheruvelil, K. S., Collins, S., and Soranno, P. (2015). Constrained spectral clustering for regionalization: Exploring the trade-off between spatial contiguity and landscape homogeneity. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE.

Appendix A

Figure A: Argentina's subnational divisions

Provinces and Autonomous City (CABA)

Macrorregions of Argentina