Journal of Engineering Managment and Soft Computing (JEMSC)

Application of Deep Learning Algorithms in Clustering Production Units: Enhancing Regulatory Processes Using Soft Computing Approaches (Case Study: North Khorasan Province)

Atefe Pakzad[1] and Aylin Pakzad[2]

1. Assistant Professor, Department of Computer Engineering, Kosar University of Bojnord, Bojnord, Iran, Email: atefepakzad@kub.ac.ir
2. Corresponding author, Assistant Professor, Department of Industrial Engineering, Kosar University of Bojnord, Bojnord, Iran, Email: a.pakzad@kub.ac.ir

Article Info

Article type:
Research Article

Article history:
Received 3 July 2025
Received in revised form 28 November 2025
Accepted 16 December 2025
Published online 1 January 2025

Keywords:
spectral Clustering, HDBSCAN, deep embedded clustering, Gaussian mixture model, production units.

ABSTRACT

The oversight of production units ensures compliance with national and international standards. The National Standards Organization gathers the quality documentation of manufacturers within the Standard Implementation Monitoring System (SINA). Despite this extensive database, the intensity and frequency of sampling and inspection processes are conducted without considering quality records. This study develops an intelligent clustering and deep learning approach to categorize production units in North Khorasan. In this regard, algorithms such as Spectral Clustering, Density-based hierarchical clustering with noise detection, Deep embedded clustering (DEC), Gaussian mixture model, and Louvain were implemented. The Davies–Bouldin index, Average Silhouette score, and Calinski–Harabasz index were utilized to assess the quality of clusters. Furthermore, the efficacy of the chosen approach was contrasted with the findings of earlier studies (K-means). The results showed that Deep embedded clustering outperformed, revealing hidden data structures with more cohesive, separable clusters and superior metrics. Deep embedded clustering outperformed K-means, reducing Davies–Bouldin by 0.24, increasing silhouette by 0.052, and enhancing Calinski–Harabasz by 479. It identified five distinct production clusters by location, production volume, and quality metrics, enabling more efficient monitoring.

Cite this article: Pakzad, A. & et al, (2026)., Application of Deep Learning Algorithms in Clustering Production Units: Enhancing Regulatory Processes Using Soft Computing Approaches (Case Study: North Khorasan Province). *Journal of Engineering Management and Soft Computing*, 12 (1). 82-101.

DOI: https://doi.org/10.22091/jemsc.2026.13325.1286

DOI: https://doi.org/10.22091/jemsc.2026.13325.1286

Publisher: University of Qom

1) Introduction

As technology continues to advance and every facet of life becomes increasingly digitalized, a vast amount of data is being produced. The smart handling of this information can be crucial for societal progress and assist in tackling major issues, such as poverty, illness, and disparity (Jacobs, 2009). Data represents one of the most important assets for a company, and for cutting-edge firms, handling large datasets is a key concern for maintaining competitiveness (Harding et al., 2006). Effective data management can not only set companies apart from their competitors but also provide a competitive edge. Organizations that employ data-driven decision-making strategies tend to outshine their competitors, indicating an average increase of 5% in productivity, and 6% in profitability (McAfee et al., 2012). Nonetheless, regrettably, even with an awareness of their data's value, numerous companies frequently do not possess the essential knowledge to utilize it effectively, lacking a clear grasp of what metrics should be evaluated. As a result, the data's informational value and its useful, actionable insights are diminished (Harding et al., 2006). Consequently, enhancing the gathering, utilization, and distribution of data has become essential for numerous businesses (Kusiak, 2017).

Machine learning (ML) is a sector of artificial intelligence (AI) that emphasizes enhancing computer system performance by learning directly from input data, significantly contributing to fulfilling many of today's requirements. At present, ML is well-known for its capacity to uncover concealed patterns and attributes in data and to learn from these insights, being employed in various data-intensive fields, such as industry, banking, insurance, healthcare, and retail (Molaee Fard, 2023; Niavand et al., 2024; Younespour & Romoozi, 2023). The main objective of machine learning is to create algorithms and models that allow computer systems to learn from data and make predictions and decisions when dealing with extensive and complex datasets (Chen et al., 2024). The key strategies in machine learning can be divided into two main types: supervised learning and unsupervised learning (Dogan & Birant, 2020). Supervised learning entails employing labeled training datasets to create a model that learns the relationship between inputs and outputs. This model aims to acquire a generalized mapping from data samples, enabling it to generate accurate predictions for new data (Kampezidou et al., 2024). In comparison, unsupervised learning uses unlabeled training data, requiring the model to identify the underlying structure, patterns, and relationships present in the data without labels (Song et al., 2023).

A frequent issue in unsupervised learning is clustering, which seeks to gather alike data points so that samples in the same cluster show the highest similarity to each other while preserving the greatest dissimilarity from samples in different clusters. Major clustering methods can be organized into five categories. Partitioning clustering methods attempt to decompose the data into k clusters such that items in each cluster are closely related to each other. Hierarchical clustering methods construct a tree of clusters by either repeatedly merging smaller clusters into larger ones (agglomerative), or by splitting larger clusters into smaller ones (divisive). Density-based clustering methods aim to find high-density clusters separated by sparse areas that clusters can differ in terms of their size and shape. Additionally, there are alternative approaches known as grid-based and model-based methods (Dogan & Birant, 2020).

2. Literature Review

Currently, the extensive adoption of information and communication technology in various sectors has led to a considerable rise in the production of industrial data (Raptis et al., 2019). The information gathered from different sensors and control systems can enhance performance and facilitate improved decision-making in various industries. Through the examination of this data, one can recognize issues, anticipate equipment breakdowns, enhance product quality, and refine production processes (Dogan & Birant, 2020). For instance, in 2016, Zidek et al. (2016) proposed a machine learning-driven approach to improve quality control on production lines by utilizing machine vision technology integrated into robots for image capture and defect identification. This technique utilizes K-means clustering,

hierarchical clustering (FLANN1), and density-based spatial clustering (DBSCAN) algorithms to group product defects into various clusters, and subsequently employs six distinct classification methods to allocate each new defect to one of these clusters. This allows for the automatic identification of the defect type and enables the implementation of required corrective measures. Wang et al. (2017) introduced an innovative approach for examining and grouping extensive electricity usage data. Initially, they utilized symbolic aggregate approximation (SAX) to condense the extensive amount of electricity usage data, subsequently applying a Markov model to mimic the shifts in customer electricity usage patterns over time. Using a density-based clustering algorithm, customers were then divided into uniform groups to deliver personalized services. In aother study, a novel approach for examining electricity usage trends in industries was presented. This research combined K-Means clustering methods with association rules to categorize industrial units situated in an industrial park in Tehran province into low, medium, and high-consumption clusters based on their electricity consumption. Concealed trends in the data of each cluster were recognized, assisting industries in improving their energy use and cutting expenses (Rahimi et al., 2022). In a 2022 study conducted by Khadivar and Mojibian (2022), the significance of different factors in clustering was initially established using the analytic hierarchy process (AHP). Subsequently, applying K-Means and Kohonen neural network algorithms, industrial workshops were categorized into four primary clusters. This categorization relied on elements such as population distribution, income status, and value-added, helping managers create customized development initiatives for every category of workshops.

In the research conducted by Sheikh Shoaee (2021), a comprehensive review of two decades of machine learning research in the field of production examined the application of these approaches across four main areas: planning, monitoring, quality, and failure prediction. This study analyzed methods based on tasks (clustering, classification, regression), types of learning, and evaluation metrics, while also explaining the stages of knowledge discovery and benefits, and identifying challenges and future research directions. In the study conducted by Ghousi (2015), Ghousi employed data mining techniques to examine large datasets related to industrial accidents. The findings revealed that a significant portion of these incidents pertained to two main groups of workers: young workers aged 20 to 25, with a high school diploma and less than two years of work experience, and workers with 3 to 8 years of work experience and education levels below high school diploma. The research by Suman and Das (2020) proposed a data-driven approach for monitoring and fault diagnosis of multi-stage processes by combining fuzzy clustering and multi-block principal component analysis. This approach, without requiring prior process knowledge, categorizes variables into homogeneous blocks and identifies abnormal conditions. Its evaluation in a steel manufacturing facility demonstrated that this approach, in addition to accurately diagnosing faults and determining the contribution of each variable to their occurrence, reduces dependence on process knowledge and increases detection accuracy, thereby showing potential for broad applicability in complex industries to improve quality and reduce risk.

Grouping industrial facilities within a province acts as a tactical approach for economic and industrial progress. By recognizing clusters of production units with common traits, a better comprehension of the province's industrial framework can be attained. This enables more focused and efficient planning, resulting in better resource allocation, enhanced industrial collaboration, the creation of value chains, improved competitiveness, and sustainable progress. In other terms, the grouping of production units offers a guide for the province's industrial expansion and advancement.

The North Khorasan Provincial Department of Standards, as the representative of the Iranian National Standards Organization in the province, is responsible for supervising the quality of domestically produced and imported goods and services. This department provides a wide range of services to manufacturers, importers, and consumers, using technical knowledge, advanced laboratory equipment, and specialized personnel. Through the SINA system, this entity has access to valuable data, including sampling results, test outcomes, inspection reports, non-conformities, and negative scores of production units, and utilizes data mining techniques to identify hidden patterns and optimize supervision processes. In research conducted by Pakzad et al. (2024), production units were clustered

1. Fast library for approximate nearest neighbors

based on their qualitative attributes and performance characteristics utilizing clustering and multi-criteria decision-making methods. This has enabled the customization of inspection programs, ensuring that higher-risk units receive more targeted and regular oversight. This method not only enhances the quality of produced goods and increase consumer satisfaction but also results in lower monitoring expenses for the organization. Previous research in the field of personalizing sampling and inspection processes has largely been limited to partitional clustering algorithms such as K-means and K-medoids. Although these algorithms are practical in many cases, they may have limitations when dealing with the complexity of industrial data and identifying irregularly shaped clusters. Therefore, the present study aims to achieve more accurate clustering and an in-depth assessment of various techniques by exploring a wide range of clustering algorithms. Gaussian mixture models (GMM), spectral Clustering based on Laplacian matrix analysis, hierarchical density-based spatial clustering of applications with noise (HDBSCAN) for identifying clusters with varying density, DEC based on learning latent features, and the Louvain algorithm for detecting graph structures have been selected as representatives of statistical, graph-based, density-based, and deep learning approaches. These algorithms, due to their capability in identifying complex clusters and robustness to noise, facilitate the selection of an optimal method for the studied data. Moreover, by comparing the results from these algorithms with previous studies, more precise and comprehensive results in clustering production units can be achieved.

In the following sections of this study, the clustering techniques employed are described. Subsequently, Section 4 outlines the proposed research methodology. Section 5 presents the case study and the findings derived from applying the proposed model. Finally, the concluding section provides the results and outlines directions for future research.

3. Clustering Algorithms

Clustering is a key method in unsupervised learning, performed with the aim of grouping similar data into separate sets. This technique enables the discovery of hidden structures in data without the need for labeling. In the following subsections, five state-of-the-art clustering techniques, including Spectral Clustering, HDBSCAN, GMM, Louvain, and DEC, will be examined.

3-1 Spectral Clustering

Spectral Clustering is a sophisticated method employed in machine learning and graph theory for grouping data points into clusters based on the eigenvalues and eigenvectors derived from a similarity matrix. This technique utilizes the spectrum of the graph Laplacian to uncover cluster structures within the data. Unlike traditional clustering methods, such as K-means, which predominantly depend on distance-based criteria, spectral Clustering focuses on analyzing the global structure of the data. This characteristic renders it especially effective for datasets where clusters may not conform to spherical or convex shapes. The fundamental principle of this algorithm is to leverage the graph Laplacian to capture the inherent structure within the data, facilitating a more flexible and accurate clustering process (Ng et al., 2001). Despite its effectiveness across a diverse array of applications, spectral Clustering is not without its limitations. One significant challenge lies in the computational complexity associated with eigenvalue decomposition, particularly when dealing with large-scale datasets. Furthermore, the performance of spectral Clustering is highly contingent upon the selection of the similarity graph and the formulation of the graph Laplacian. The quality of the resultant clusters can vary considerably based on how well the graph encapsulates the underlying structure of the data. For example, if the graph is excessively sparse or if the similarity function is inappropriately selected, the algorithm may struggle to yield meaningful clustering outcomes (von Luxburg, 2007).

The theory of spectral Clustering is rooted in graph analysis and the characteristics of the Laplacian matrix. In this approach, the first step involves constructing a similarity graph among the data points, after which the spectral properties of the graph Laplacian are utilized to carry out the clustering process. The inputs for the spectral Clustering algorithm comprise a dataset with n data samples and the target number of clusters k. The output consists of cluster labels assigned to each individual data point. The procedure of the algorithm is outlined as follows:

Step 1: Construction of the Similarity Graph

In the first step, a graph G = (V, E) is constructed, where V represents the set of data points and E denotes the edges that encode the similarity relationships among these points. Typically, the similarity between two data points x_i and x_j is computed using a similarity function, such as the Gaussian (radial basis function) kernel, as illustrated in Equation (1) (Ng et al., 2001):

$$S(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (1)$$

Here, σ is a tuning parameter that regulates the scale of the distance. A smaller value of σ results in high similarity only among very close data points, while a larger value of σ allows for similarities between points that are farther apart. The function $exp()$ represents the exponential function, which maps the similarity values to the interval (0, 1]. This means that similarities approach one for points that are close together while decrease toward zero as the distance between the points increases.

Step 2: Computation of the Graph Laplacian Matrix

1-Computation of the degree matrix D 2-Computation of one of the variants of the graph Laplacian matrix. The degree matrix D is a diagonal matrix, with each diagonal element representing the sum of the similarity values found in the i-th row of the similarity matrix S. This relationship is expressed mathematically in Equation (2) (Liu & Han, 2008):

$$D_{ii} = \sum_j S_{ij} \quad (2)$$

Subsequently, the Laplacian matrix L is constructed in one of three forms: the unnormalized Laplacian, given by $L = S - D$; the symmetric normalized Laplacian, represented as $L_{sym} = I - DS^{-\frac{1}{2}}D^{-\frac{1}{2}}$; and the random-walk normalized Laplacian, expressed as $L_{rw} = I - SD^{-1}$.

Step 3: Computation of Eigenvectors

After constructing the Laplacian matrix, the next step is to perform eigenvalue decomposition. The eigenvalues and eigenvectors of the Laplacian matrix characterize the structural properties of the graph. Eigenvectors associated with eigenvalues smaller than a given threshold typically correspond to the natural groupings of the data. In practice, the first k eigenvectors are commonly used for dimensionality reduction, where k denotes the number of clusters. Accordingly, the third step consists of the following two stages (Liu & Han, 2008):

- Computation of the k eigenvectors corresponding to the smallest eigenvalues of the Laplacian matrix.
- Construction of the feature matrix U consisting of the k eigenvectors arranged as $U = [u_1, u_2, \dots, u_k]$.

Step 4: Application of K-Means Clustering

After performing eigenvalue decomposition in Step 3, the data points are mapped into a new feature space, where each point is represented as a feature vector based on the selected eigenvectors. A clustering algorithm, such as K-means, is then applied to these representations to identify the clusters. Consequently, Step 4 involves the following stages (Liu & Han, 2008):

1. Treat each row of U as a new feature vector representing the corresponding data point.
2. Apply the K-means algorithm to the rows of U in order to cluster the data.
3. Assign each data point to the nearest cluster center.

3-2 HDBSCAN Clustering

The HDBSCAN algorithm is an advanced extension of the DBSCAN algorithm that combines density-based clustering with hierarchical methods. It clusters data based on density, addressing the limitations of DBSCAN, which requires fixed values for both the minimum number of points in a neighborhood

and the radius parameter ε. HDBSCAN employs hierarchical density estimation to identify clusters with varying densities. By converting the dataset into a density-connected graph, it extracts stable clusters through hierarchical analysis, resulting in higher accuracy in detecting heterogeneous clusters. This algorithm effectively identifies clusters with differing densities without the need to predefine ε (Campello et al., 2013).

One of the key concepts in HDBSCAN is the elimination of the ε parameter, coupled with the use of a density-connected graph. This approach relies on a minimum spanning tree (MST) for clustering. Initially, a complete graph of the data points is constructed, with edge weights assigned based on the pairwise distances between points. By progressively removing edges according to density, a hierarchical cluster tree is formed, where points with higher density occupy higher levels. This process enables the algorithm to accurately identify clusters with irregular shapes and varying densities. Another important feature of HDBSCAN is its ability to automatically detect noise. Unlike DBSCAN, which classifies points as noise only when their local density falls below the minimum neighborhood threshold, HDBSCAN employs a cluster stability measure to inform its decisions. This measure evaluates the stability of each cluster across different hierarchical levels and designates points with low stability values as noise (McInnes et al., 2017).

One of the major advantages of HDBSCAN over other methods is its ability to handle high-dimensional data and its applicability across a wide range of real-world scenarios. The algorithm has been utilized in financial market analysis, medical image processing, and large-scale data clustering in big data systems. Research by Tran et al. (2021) demonstrated that HDBSCAN can effectively analyze datasets with dense and complex features, allowing for the identification of clusters with nonlinear structures.

In conclusion, HDBSCAN is regarded as one of the most powerful density-based clustering algorithms due to its elimination of the sensitive ε parameter, its hierarchical clustering capabilities, automatic noise detection, and high accuracy in identifying complex structures. Compared to methods such as K-means and DBSCAN, it offers greater flexibility in detecting non-uniform clusters and minimizes the need for manual tuning of critical parameters (Tran et al. 2021). The following outlines the steps of the HDBSCAN clustering algorithm:

Step 1: Definition of Mutual Reachability Distance

A key component of HDBSCAN is the definition of the mutual reachability distance between data points. This distance is defined in Equation (3):

$$d_{mutual}(p,q) = max\{core_k(p), core_k(q), d(p,q)\} \quad (3)$$

where:

- d(p,q) is the Euclidean distance between two points p and q.
- $core_k(p)$ is the core distance of point p, defined as the distance to its k-th nearest neighbor.

The mutual reachability distance $d_{mutual}(p,q)$ ensures that the edges between points in the graph are weighted according to the estimated local density.

Step 2: Construction of the Minimum Spanning Tree (MST) and Density Hierarchy

In this step, a fully connected weighted graph is created using the mutual reachability distance. From this graph, the minimum spanning tree (MST) is extracted, which serves to connect the data points effectively. By systematically lowering the density threshold and eliminating low-density edges, a hierarchy of density-based clusters is developed.

Step 3: Extraction of Clusters Based on Stability

In this step, clusters showing the highest density stability across the hierarchical levels are identified. The stability measure of a cluster C is defined as follows:

$$Stability(C) = \sum_{p \in C} (\lambda_{min}(p) - \lambda_{max}(p)) \quad (4)$$

Here, $\lambda = \frac{1}{distance}$ represents the inverse of the distance, reflecting the local density. $\lambda_{max}(p)$ denotes the maximum density level at which the point p is present, while $\lambda_{min}(p)$ represents the minimum density level at which point p still remains in the cluster. Clusters with higher stability values are retained in the final results, whereas points with low stability are classified as noise or outliers.

Theoretical Advantages of HDBSCAN over DBSCAN

1. No Requirement to Specify ε: Unlike DBSCAN, which necessitates the explicit selection of the ε threshold, HDBSCAN autonomously discovers clusters with varying densities without the need for manual parameter tuning.

2. Improved Noise Handling: In HDBSCAN, noise points are identified based on cluster stability rather than solely relying on absolute density criteria.

3. Detection of Complex Clusters: HDBSCAN is proficient in identifying irregularly shaped clusters with different density levels, especially in large and noisy datasets (Campello et al., 2013).

3-3 Deep Embedded Clustering (DEC)

Deep Embedded Clustering (DEC) is a machine learning approach specifically designed to cluster data by utilizing deep neural networks. The primary objective of this algorithm is to learn a meaningful latent representation of the data, where similar samples are mapped closer together in the new feature space, while dissimilar samples are pushed further apart. DEC begins by employing a deep neural network to perform nonlinear dimensionality reduction. It then iteratively refines these latent features to enhance clustering performance using the K-means algorithm. This method is particularly well-suited for complex and high-dimensional data that traditional clustering techniques might struggle to handle effectively (Caron et al., 2018).

One of the distinguishing characteristics of the DEC clustering algorithm is its capacity to simultaneously perform data clustering and representation learning. The process begins with training a deep neural network to compress the data, followed by an initial clustering step. Afterward, the algorithm starts an optimization process that progressively refines the data representations, drawing similar data points closer together in the feature space while pushing dissimilar points further apart. This iterative refinement not only enhances clustering accuracy but also reduces model complexity (Guo et al., 2017). Another notable characteristic of DEC clustering is its effectiveness in handling complex, high-dimensional data, such as images and textual information. In contrast to traditional methods, which often struggle with high-dimensional and intricate datasets, DEC utilizes the representational power of deep neural networks to automatically learn discriminative features in conjunction with the clustering process. This capability allows DEC to achieve superior performance compared to conventional approaches, such as K-means (Caron et al., 2018). The DEC clustering algorithm is frequently utilized alongside autoencoder-based neural networks to learn nonlinear representations of data. These autoencoders facilitate DEC in mapping the input data into a lower-dimensional feature space while preserving essential information. This process results in a significant enhancement in clustering performance for complex datasets and is especially effective in applications such as image processing and other high-dimensional data domains (Xie et al., 2016).

In conclusion, the DEC clustering algorithm offers notable advantages over other clustering methods, such as K-means or Hidden Markov Model (HMM) based approaches. DEC is capable of automatically optimizing feature representations for each cluster and adapting its performance to the intrinsic characteristics of the data. This flexibility allows it to be effectively applied across a wide range of domains, including image processing, natural language processing, and medical data analysis, where it often delivers superior results compared to traditional methods (Guo et al., 2017). This algorithm is specifically developed for high-dimensional and complex datasets, simultaneously performing optimal clustering and representation learning. Rather than depending on predefined features, it autonomously learns discriminative features tailored for clustering tasks. In this framework, a deep neural network is initially trained to compress data into a latent feature space, after which clustering is applied.

Importantly, these two stages are executed concurrently through a unified optimization process (Caron et al., 2018).

In DEC clustering, a nonlinear representation of the data is first learned using a neural network known as an autoencoder. The dataset is denoted as $X = \{x_1, x_2, \dots, x_N\}$, where each x_i is a data point in a d-dimensional space. These data points are then mapped into a new latent feature space $Z = \{z_1, z_2, \dots, z_N\}$ through a transformation function $f_\theta(x)$, which effectively compresses complex features. This operation is modeled as shown in Equation (5):

$$f_\theta(x_i) = z_i \tag{5}$$

Here, f_θ denotes a neural network whose parameters are optimized using the training data. This network is responsible for transforming the data into a new latent space that is well-suited for clustering tasks (Guo et al., 2017).

After this stage, clustering is carried out using an optimization algorithm that integrates K-means clustering with a tailored cost function. The primary objective is to minimize the distance between similar data points while maximizing the separation between dissimilar ones. To accomplish this, a loss function J is utilized, which is optimized through K-means clustering in conjunction with various regularization terms to enhance the distribution of data points within the latent feature space. Specifically, the loss function is defined as follows:

$$J = \sum_{i=1}^{N} \|z_i - \mu_{c_i}\|^2 \tag{6}$$

where μ_{c_i} denotes the centroid of the cluster to which data point x_i is assigned, and c_i represents the corresponding cluster label (Caron et al., 2018). In the subsequent stage, to further enhance data representations and clustering performance, the DEC algorithm utilizes an optimization strategy known as soft assignment. In this process, each data point is not only assigned to its nearest cluster but the probability of its membership in each cluster is also computed. To determine this probability, a function p_{ij} is employed, quantifying the likelihood that data point z_i belongs to cluster j, as defined in Equation (7):

$$p_{ij} = \frac{\exp\left(-\frac{\|z_i - \mu_j\|^2}{\tau}\right)}{\sum_{k=1}^{K} \exp\left(-\frac{\|z_i - \mu_j\|^2}{\tau}\right)} \tag{7}$$

where μ_j denotes the centroid of cluster j, and τ is a temperature or tuning parameter that is typically adjusted during the optimization process (Caron et al., 2018). Finally, to further refine the neural network and enhance clustering performance, the DEC algorithm applies the backpropagation algorithm. During this phase, the neural network parameters are iteratively updated so that both clustering effectiveness and the quality of learned feature representations in the latent space are simultaneously optimized. This iterative process continues until convergence is achieved (Guo et al., 2017). Through these stages and optimization procedures, DEC is able to efficiently cluster complex, high-dimensional data and to consistently outperform traditional clustering methods.

3-4 Gaussian Mixture Model (GMM)

A Gaussian Mixture Model (GMM) is a parametric probability density function that is formulated as a weighted sum of multiple Gaussian component densities. GMMs are widely employed as parametric models to represent the probability distribution of continuous measurements or features. The parameters of a GMM are typically estimated from training data using the iterative Expectation Maximization (EM) algorithm or through Maximum A Posteriori (MAP) estimation based on a trained prior model. Specifically, a GMM comprises a weighted sum of M Gaussian component densities, as represented in Equation (8):

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \Sigma_i) \quad (8)$$

where x is a continuous-valued data vector with D dimensions, representing the number of features. The terms w_i for $i = 1, \dots, M$ denote the mixture weights, while $g(\mathrm{x}|\mu_i, \Sigma_i)$ for $i = 1, \dots, M$ represent the component Gaussian densities. Each component density corresponds to a D dimensional multivariate Gaussian distribution, which is defined as shown in Equation (9):

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \Sigma_i) \quad (9)$$

where μ_i is the mean vector, and Σ_i is the covariance matrix of the i-th Gaussian component. The mixture weights satisfy the constraint $\sum_{i=1}^{M} w_i = 1$. The complete Gaussian mixture model is parameterized by the set of mean vectors, covariance matrices, and mixture weights for all component densities. These parameters are collectively denoted as $\lambda = \{w_i, \mu_i, \Sigma_i\}$ for $i = 1, \dots, M$. The choice of model configuration, including the number of components, whether to use full or diagonal covariance matrices, and the extent of parameter sharing, is typically determined by the amount of available data for estimating the GMM parameters and the specific application context in which the GMM is employed.

Although GMM is fundamentally a statistical framework for modeling the probability density of continuous data, it can also be effectively utilized for clustering tasks. In this setting, each Gaussian component is interpreted as a distinct cluster, and data points are assigned to these components based on maximum likelihood estimation. Specifically, after fitting the GMM to the data, the probability that each data point belongs to each Gaussian component is calculated, enabling the assignment of cluster labels according to these probability distributions. Unlike algorithms such as K-means, which perform hard clustering by assigning each data point exclusively to a single cluster, GMM-based clustering employs a soft assignment approach. This means that the membership probability of each data point to all clusters is determined, allowing for more nuanced and flexible clustering results. As a result, GMM is particularly advantageous in scenarios where the data exhibits complex structures or substantial overlap between clusters, providing superior performance compared to traditional clustering methods (Reynolds, 2015).

3-5 Louvain Algorithm

The Louvain algorithm is a graph-based clustering technique designed to maximize the modularity Q of a given partition P. It utilizes a greedy optimization strategy and is recognized as one of the most widely adopted algorithms for graph clustering. Modularity serves as a metric to assess the quality of a network's division into clusters by comparing the density of edges within clusters to those between clusters. The primary objective of the Louvain algorithm is to identify a partition that achieves the highest possible modularity. The algorithm operates on an undirected graph G = (V, E), which may include self-loops and multiple edges between node pairs. Edge weights can be specified; if omitted, all edges are assigned a default weight of one. A significant advantage of the Louvain algorithm is that it does not require the number of clusters to be specified in advance, making it suitable for applications where the internal structure of the graph is unknown. The clusters detected by the Louvain algorithm are organized hierarchically, enabling users to investigate each cluster at multiple levels of granularity. This hierarchical structure facilitates the discovery of meaningful substructures within clusters that may not be evident at the top level of partitioning (Combe et al., 2015).

The overall procedure for implementing the Louvain algorithm is as follows:

1. Initialization with Singleton Partitioning: Each node is initially placed in its own individual cluster.
2. Iterative Optimization: Repeat the following steps until no substantial increase in modularity is observed or the improvement falls below a predefined threshold:

a) Modularity Optimization Sequence (MOS): For each node, identify the cluster transfer that produces the greatest increase in modularity and move the node accordingly. This step embodies the greedy optimization core of the algorithm.

b) Community Aggregation Sequence (CAS): Using the clusters obtained from the MOS step, construct a new graph in which each cluster is represented as a single node. This aggregation step forms the basis of the hierarchical clustering structure characteristic of the Louvain algorithm.

3. Return the Resulting Clustering of the Graph.

Through this iterative process, the Louvain algorithm is capable of detecting meaningful clusters at both the global network level and within local substructures (Dollmann, 2023).

3-6 Clustering Evaluation Metrics

3-6-1 Davies-Bouldin Index (DBI)

An important metric for simultaneously assessing clustering quality evaluates the compactness of clusters and the degree of separation between them. This index is computed based on the average ratio of within-cluster dispersion to between-cluster distances. Lower values of this metric indicate superior clustering performance, as they correspond to clusters that are both more compact and better separated from one another. The general formula for this metric is presented in Equation (10):

$$DBI = \frac{1}{N}\sum_{i=1}^{N} \max_{j \neq i}\left(\frac{\sigma_i + \sigma_j}{d_{ij}}\right) \tag{10}$$

where σ_i represents the average distance of points within cluster i to its centroid, d_{ij} denotes the distance between the centroids of clusters i and j, and N is the total number of clusters (Davies & Bouldin, 2009). A key advantage of the Davis-Bouldin index is that it does not require labeled data, making it suitable for evaluating unsupervised clustering results. However, this metric has certain limitations. For example, when clusters have irregular shapes or significantly different densities, the index may not accurately reflect clustering quality. Therefore, the Davis-Bouldin index is often used alongside other evaluation metrics, such as the Silhouette score and the Calinski-Harabasz index, to provide a more comprehensive assessment of clustering performance (Halkidi et al., 2001).

3-6-2 Average Silhouette Score

The Silhouette score is one of the most widely used metrics for evaluating clustering quality, as it simultaneously captures both cluster compactness and separation. This score is calculated for each data point in the dataset, and the average of these values provides the overall clustering performance. The Silhouette value ranges from +1 to -1. A value close to +1 indicates well-clustered points (i.e., points are close to their own cluster and distant from other clusters), a value near 0 suggests overlapping clusters, and values approaching -1 indicate that points may have been assigned to the wrong clusters. The formula for calculating the Silhouette score is presented in Equation (11):

$$S(i) = \frac{b(i) - a(i)}{\max\left(a(i), b(i)\right)} \tag{11}$$

where $a(i)$ is the average distance of sample i to all other points within the same cluster (intra-cluster distance), and $b(i)$ is the average distance of sample i to the points in the nearest neighboring cluster (inter-cluster distance) (Rousseeuw, 1987).

An important advantage of the Silhouette score is that it does not require labeled data, making it particularly suitable for evaluating unsupervised clustering algorithms. Additionally, the Silhouette score facilitates graphical analysis, enabling a more detailed assessment of the quality of individual clusters. However, the metric is sensitive to clusters with irregular shapes or varying densities, which can affect its reliability. Therefore, it is advisable to use the Silhouette score in conjunction with other

evaluation metrics, such as the Davis-Bouldin index and the Calinski-Harabasz index, to achieve a more comprehensive and accurate evaluation of clustering quality (Kaufman & Rousseeuw, 2009).

3-6-3 Calinski-Harabasz Index

The Calinski-Harabasz index, also referred to as the variance ratio criterion, is a widely adopted metric for assessing clustering quality. It is calculated as the ratio of between-cluster dispersion (the variance among cluster centroids) to within-cluster dispersion (the variance of points within each cluster). A higher Calinski-Harabasz index signifies superior clustering performance, indicating that clusters are well separated and data points are tightly grouped around their respective centroids. The formula for this index is provided in Equation (12):

$$CH = \frac{Tr(B_k)}{Tr(W_k)} \times \frac{N-k}{k-1} \tag{12}$$

where N is the total number of samples, and K is the number of clusters. $Tr(B_k)$ denotes the total between-cluster dispersion, and $Tr(W_k)$ represents the total within-cluster dispersion (Caliński & Harabasz, 1974).

A notable advantage of the Calinski-Harabasz index is its capability to automatically suggest the optimal number of clusters. This is achieved by calculating the index for various values of k, producing the highest Calinski-Harabasz score being selected as the optimal number of clusters. However, the accuracy of this metric may be reduced when clusters have irregular shapes or differing densities. Therefore, it is advisable to use the Calinski-Harabasz index alongside other evaluation metrics, such as the Silhouette score and the Davis-Bouldin index, to obtain a more robust and reliable assessment of clustering quality (Milligan & Cooper, 1985).

4. Research Methodology

This study was conducted with the aim of proposing a novel approach to clustering production units, particularly those under the supervision of the North Khorasan Provincial Administration of the National Standards Organization, to enhance and customize the processes of sampling and inspection. The main stages of the proposed approach are as follows:

1. **Structural analysis and the Collection of Relevant Data on Production Units:** The proposed method was implemented on data from production units extracted from the SINA system of the North Khorasan Provincial Administration. In this stage, by utilizing the data available in the SINA system and consulting with domain experts, a database comprising active production units was constructed. The database included variables such as “Industry Type,” “Industry Classification (Small, Medium, Large),” “Number of Quality Control Managers,” “Highest Education Level of Quality Control Manager,” “Number of Manufactured Products,” “City of the Production Unit’s Location,” and “Average Total Negative Score of the Production Unit for the period 2015-2023.”
2. **Data preprocessing:** In this stage, the raw data was refined through operations including replacing missing values, transforming categorical variables, and normalizing numerical values.
3. **Clustering of production units clustering:** Five clustering algorithms (Spectral Clustering, HDBSCAN, GMM, Louvain, and DEC) were implemented and executed under various scenarios, using all features and by removing less important features.
4. **Evaluation of clustering performance and the selection of the superior clustering method:** The performance of the clustering algorithms was assessed and compared using standard clustering evaluation metrics, including the Davies-Bouldin Index, average Silhouette score, and Calinski-Harabasz Index. The best clustering method was then identified.

5. **Cluster labeling and interpretation:** The resulting clusters were labeled, and a content-based analysis of the clusters was conducted to interpret the findings.

5. Case Study

In this study, performance data from 217 active production units supervised by the Khorasan Shomali Provincial Standards Office between 2015 and 2023 were collected from the SINA system. These units represented a wide array of industrial sectors, including food and agriculture, textiles and leather, packaging and cellulose, chemical industries, automotive and powertrain, electrical and electronics, biomedical engineering, construction and mining, weights and measures, safety, health, energy and environment, mechanical metallurgy, and precious metals. Following preliminary data analysis and consultation with experts from the Khorasan Shomali Standards Office, seven variables were initially selected for clustering these units (as detailed in Table 1). However, findings from a previous study (Pakzad et al., 2024) suggested that excluding the variable "industry type" and incorporating the relative importance of different industries in calculating the seventh variable, "the average negative score of production units from 2015 to 2023" led to improved clustering results. Consequently, in the present study, the annual negative scores of each production unit were multiplied by industry-specific weights derived from the Best–Worst method described by Pakzad et al. (2024). The weighted average of these scores over the 2015–2023 period was then computed, and exponential smoothing with a smoothing coefficient of $\lambda = 0.2$ was applied to obtain the final negative performance index for each unit. As a result, the final dataset used for clustering consisted of 217 records (production units) and six clustering variables. To gain deeper insights into the performance structure of these units, five advanced clustering algorithms were employed: Spectral Clustering, HDBSCAN, GMM, Louvain, and DEC. For comparison with the previous study which utilized variable weighting based on information gain, the current study also performed clustering using variable weights determined by the same information gain criterion. This approach facilitated a direct comparison of cluster structures and outcomes with those reported in the prior study. All data analyses were conducted using Excel, RapidMiner, and Python.

Table 1) Clustering Variables for Production Units

No.	Variable	Description	Type
1	Industry type	Food and agriculture, …, services	Nominal
2	Industry classification	Small, Medium, Large	Nominal
3	Number of quality control managers	1, 2, …	Numeric
4	Highest education level of quality control managers	Diploma, Associate degree, …, PhD	Nominal
5	Number of products produced	1, 2, …	Numeric
6	City of the Production Unit's Location	Distance of the unit's city from Bojnord (km)	Numeric
7	Average total negative score of the production unit (2015–2023)	–	Numeric

5-1) Data Preprocessing

In the dataset utilized for this study, certain observations contained missing values. To address this, the "Replace Missing Values" operator in RapidMiner was employed to ensure data integrity and minimize potential bias in the analysis. By configuring the operator to use the "mean" method, missing values were automatically estimated and replaced with the mean of the respective variable. Following this, the "Nominal to Numerical" operator was used to convert categorical variables into numerical values suitable for further analysis. As the next step in the data preprocessing workflow, the "Normalization" operator was applied to scale all variables. Given the numerical nature of the dataset, range normalization was performed to rescale the data within the [0, 1] interval. All data preprocessing procedures were carried out using RapidMiner.

5-2 Clustering of Production Units

In this study, following data preparation, five clustering algorithms, Spectral Clustering, HDBSCAN, GMM, Louvain, and DEC, were employed to categorize the production units into five clusters. The selection of five clusters was guided by expert judgment from the North Khorasan Provincial Standards Office to ensure that the results would be interpretable and practically valuable for tailoring standardization processes. To facilitate comparison with a previous study (Pakzad et al., 2024), the same information gain-based weighting approach was adopted. In this framework, the variable "Production Unit Code" served as the target, and the weight of each variable was determined according to its influence on this target (see Table 2, first row for results). The clustering was conducted under six different scenarios, which are detailed in Table 2. In the initial scenario, all six variables, including industry classification, number of quality control managers, the highest education level of quality control managers, number of products produced, city of production unit location, and average negative score of the production unit from 2015 to 2023, were included with their respective calculated weights. In subsequent scenarios, variables with lower weights were systematically excluded. This strategy enabled the evaluation of dimensionality reduction effects on clustering quality and allowed for a structured comparison of the results with those of the previous study.

Table 2) Variables Used Along with Their Weights in Different Clustering Scenarios

Scenario	Industry Classification	Number of Quality Control Managers	Highest Education Level of Quality Control Managers	Number of Products Produced	City of Production unit's location	Average Negative Score of Production Unit (2015–2023)
1	0.289	0.428	0.626	0.957	0.971	1
2	✗	0.428	0.626	0.957	0.971	1
3	✗	✗	0.626	0.957	0.971	1
4	✗	✗	✗	0.957	0.971	1
5	✗	✗	✗	✗	0.971	1
6	✗	✗	✗	✗	✗	1

It is important to note that for the Spectral Clustering, HDBSCAN, GMM, and Louvain algorithms, all six scenarios outlined in Table 2 were evaluated. In contrast, due to the unique characteristics of the DEC algorithm which relies on learning optimal data representations through a neural network, only the first scenario (involving all variables) was considered. By nonlinearly transforming the data into a lower-dimensional space, DEC is capable of automatically capturing complex structures and nonlinear relationships among variables, eliminating the need to assess different weighting scenarios as required by other algorithms. All computations and algorithm implementations were performed in the Python environment. The results of these five clustering methods, along with comprehensive comparative analyses, are presented in the subsequent subsection of this study.

5-3) Evaluation of Clustering Performance

For the comprehensive evaluation of the clustering results, obtained from the spectral Clustering, HDBSCAN, DEC, GMM, and Louvain algorithms, three metrics including Davies-Bouldin, Calinski-Harabasz, and mean silhouette were used, each examining a different aspect of clustering. The Davies-Bouldin index (smaller value preferred) evaluates cluster density and distinctness by determining the ratio of intra-cluster distance to inter-cluster distance. The mean silhouette score (optimal value closer to 1) analyzes the quality of sample assignments at a micro level by examining the distance of each sample to its own cluster and the nearest other cluster. The Calinski-Harabasz index (higher value preferred) assesses the overall quality of clustering by examining the ratio of dispersion between clusters to dispersion within clusters. The combination of these three metrics enables a comprehensive evaluation of clustering in terms of compactness, separation, and structural cohesion.

- **Analysis of Spectral Clustering Results**

The spectral clustering algorithm maps data into a new space using the spectral features of the similarity matrix between data points and, then, employs the K-means algorithm to cluster the data in this new space. This method is particularly effective for data with complex structures and non-linear clusters. In this study, five clusters were identified across six distinct data scenarios utilizing the spectral clustering algorithm. Table 3 displays the results of the evaluation metrics.

Table 3) Results of Evaluation Metrics for Spectral Clustering under Different Scenarios

Evaluation Metric \ Scenario	1	2	3	4	5	6
Davies-Bouldin Index	1.74	1.31	1.26	0.99	0.72	0.54
Mean Silhouette Score	0.1439	0.2467	0.3034	0.2395	0.4402	0.5586
Calinski-Harabasz Index	42	62	79	135	297	338

According to Table 3, the Spectral Clustering algorithm performed poorly in Scenario 1 (utilizing all weighted variables), as shown by a high Davies-Bouldin Index (1.74), and a low mean Silhouette Score (0.1439), indicating weak intra-cluster cohesion and inadequate separation. The stepwise elimination of less significant variables (Scenarios 2 to 6) resulted in a noticeable enhancement in the evaluation metrics. In Scenario 4, the decline of the Davies-Bouldin Index to 0.99, and the increase of the Calinski-Harabasz Index to 135 indicate an improved variable combination. Although Scenarios 5 and 6 (featuring the least variables) produce superior index values (for instance, a Calinski-Harabasz Index of 338 in Scenario 6), Scenario 4 is suggested as the ideal option for achieving a balance between variable count and clustering quality. By keeping the three key variables, "number of products manufactured," "city of production unit's location," and "Average total negative score of the production unit for the period 2015-2023," Scenario 4 achieves the optimal balance between within-cluster cohesion and between-cluster separation, a conclusion further supported by the significant improvement in metrics from Scenario 3 to 4.

- **Analysis of HDBSCAN Clustering Results**

The HDBSCAN algorithm is a density-based clustering method that extracts stable clusters by analyzing their hierarchical structures. This algorithm is capable of identifying clusters with varying densities and performs well in effectively separating noise points from the data. In this research, the HDBSCAN algorithm effectively detected four clusters based on density within the data. Table 4 shows the outcomes of the clustering assessment conducted with this algorithm.

Table 4) Results of Evaluation Metrics for HDBSCAN Clustering Under Different Scenarios

Evaluation Metric \ Scenario	1	2	3	4	5	6
Davies-Bouldin Index	1.74	1.63	1.53	1.75	1.73	4.79
Mean Silhouette Score	0.1439	0.1641	0.1820	0.1242	0.1247	-0.13
Calinski-Harabasz Index	42	53	61	72	101	79.4

The analysis of HDBSCAN clustering results in Table 4 indicates that Scenario 3 (after removing two low-priority variables) provided the best performance, demonstrated by enhanced cluster cohesion and separation, as shown by a reduced Davies-Bouldin Index. Additional variable reduction in Scenarios 4 and 5 compromised clustering quality, probably because of the elimination of important density-influencing variables; however, Scenario 6, utilizing only one variable, failed entirely (mean Silhouette: -0.13). This validates the necessity of a balanced set of density-related variables for HDBSCAN. Consequently, Scenario 3 is suggested as the ideal configuration, achieving the most favorable equilibrium between variable quantity and cluster validity, yielding a Davies-Bouldin Index of 1.53, with improvements in the mean Silhouette score to 0.1820, and the Calinski-Harabasz Index to 61. These results highlight HDBSCAN's pronounced sensitivity to variable choice and its unique performance relative to distance-based algorithms.

- **Analysis of GMM Clustering Results**

The analysis of the GMM clustering outcomes in Table 5 indicates that this statistically-based algorithm achieved a significant improvement in clustering quality by incrementally excluding low-importance variables. Scenario 4 was recognized as the best point, with the Davies-Bouldin Index dropping to 0.86, the mean Silhouette Score rising to 0.4503, and the Calinski-Harabasz Index reaching 181. These enhancements suggest the creation of more cohesive clusters more distinct boundaries. While Scenario 6, which utilizes a single variable, demonstrates the highest values (Davies-Bouldin Index: 0.51, mean Silhouette: 0.5364, Calinski-Harabasz: 639), Scenario 4 is recommended as the final option for its superior balance between the quantity of variables and the quality of clustering. The enhancement in metrics, especially the 466% growth in the Calinski-Harabasz Index from Scenario 1 to 4, validates the effectiveness of the GMM method in uncovering the inherent structure of the data.

Table 5) Results of Evaluation Metrics for GMM Clustering Under Different Scenarios

Evaluation Metric \ Scenario	1	2	3	4	5	6
Davies-Bouldin Index	1.57	1.21	1.36	0.86	1.09	0.51
Mean Silhouette Score	0.2329	0.3726	0.3951	0.4503	0.4173	0.5364
Calinski-Harabasz Index	32	77	88	181	220	639

- **Analysis of Louvain Clustering Results**

The analysis of the Louvain clustering results in Table 6 indicates that this modularity optimization-based algorithm achieved continuous improvement in clustering quality through the gradual removal of low-importance variables. Scenario 5 was identified as the optimal point, where the Davies-Bouldin Index decreased to 0.81, the mean Silhouette Score increased to 0.3986, and the Calinski-Harabasz Index reached 288. These improvements signify the formation of more cohesive clusters with stronger internal connections. Although Scenario 6, with a single variable, shows the best values (Davies-Bouldin Index: 0.51, mean Silhouette: 0.5868, Calinski-Harabasz: 330), Scenario 5 is proposed as the optimal choice, as it establishes a better balance between the number of variables and clustering quality, while also demonstrating a 476% increase in the Calinski-Harabasz Index from Scenario 1. These findings validate the effectiveness of the Louvain algorithm in detecting network patterns and inherent communities within the data.

Table 6) Results of Evaluation Metrics for Louvain Clustering under Different Scenarios

Evaluation Metric \ Scenario	1	2	3	4	5	6
Davies-Bouldin Index	1.43	1.39	1.20	1.11	0.81	0.51
Mean Silhouette Score	0.2241	0.2499	0.3024	0.2911	0.3986	0.5868
Calinski-Harabasz Index	50	61	76	108	288	330

- **Analysis of DEC Clustering Results**

This research utilized the DEC algorithm alongside a particular neural network design. The network design consisted of a shallow autoencoder with an input layer of six neurons (matching the data features), a hidden encoding layer with three neurons and ReLU activation function to generate a three-dimensional embedded space, and a reconstruction layer with six neurons and a sigmoid function for reconstructing the normalized data. The model was trained using the MSE loss function and the Adam optimizer (learning rate set to 0.001) for 100 epochs (batch size of 32). Following the initial training, the obtained latent vectors served as input for the K-Means algorithm to execute the final clustering. By combining deep learning and clustering, this method was able to automatically extract optimal variables and, without the need for manual variable removal, to generate five high-quality clusters. Table 7 displays the findings of the DEC clustering evaluation metrics.

Table 7) Results of Evaluation Metrics for DEC Clustering in Scenario 1

valuation Metric / Scenario	**Davies-Bouldin Index**	**Mean Silhouette Score**	**Calinski-Harabasz Index**
1	0.49	0.5415	683

The analysis of the DEC outcomes in Table 7 indicates that this sophisticated algorithm, which merges deep learning with clustering, has achieved remarkable performance. Through the use of an autoencoder neural network, the DEC algorithm effectively mapped the data into a refined space where clusters are clearly distinguished. The findings from this approach demonstrate strong clustering quality, as the Davies-Bouldin Index achieves a favorable value of 0.49, indicating remarkable separability among clusters. A mean Silhouette Score of 0.415 indicates strong internal cohesion and a compact cluster structure, while the Calinski-Harabasz Index of 683 additionally verifies that the clusters are statistically distinct. These findings indicate that DEC, as an innovative approach for clustering complex data, has a strong ability to discover latent structures within the data.

- **Comprehensive Analysis of Clustering Algorithms' Performance Compared with Previous Study**

This section systematically compares the performance of five advanced clustering algorithms, Spectral Clustering, HDBSCAN, DEC, GMM, and the Louvain algorithm, against the best reported scenario from a prior study (Pakzad et al., 2024), which utilized the traditional K-means algorithm. The main goal of this comparison is to assess the level of enhancement obtained by utilizing contemporary clustering techniques compared to the K-means algorithm, which was identified as the optimal choice in the earlier study. Table 8 displays the best performance of the clustering algorithms according to three assessment metrics: Davies-Bouldin Index, average Silhouette Score, and Calinski-Harabasz Index.

Table 8) Performance Comparison of Clustering Algorithms Based on Evaluation Metrics

Evaluation Metric / Clustering Algorithm	**Optimal Scenario**	**Davies-Bouldin Index**	**Mean Silhouette Score**	**Calinski-Harabasz Index**
Spectral	4	0.99	0.2395	135
HDBSCAN	3	1.53	0.1820	61
DEC	1	0.49	0.5415	683
GMM	4	0.86	0.4503	181
Louvain	5	0.81	0.3986	288
K-means	4	0.73	0.4891	204

Table 8 provides a systematic comparison of six clustering algorithms, highlighting the DEC method as the most efficient, with superior performance across all metrics: a Davies-Bouldin Index of 0.49, a mean Silhouette Score of 0.54, and a Calinski-Harabasz Index of 683. While the GMM and Louvain algorithms demonstrate a commendable balance between the number of variables and cluster validity in their respective intermediate scenarios (4 and 5), the apparent metric improvements observed in the final, most reduced-variable scenarios are deemed artificial, resulting from an oversimplification of the clustering problem rather than a genuine enhancement in model capability.

The evaluation results indicate that the DEC method has achieved a significant improvement in clustering quality compared to K-means. In the Davies-Bouldin Index, DEC demonstrated better cluster separation with a reduction of 0.24 units (from 0.73 to 0.49). The mean Silhouette Score also improved by 0.052 units (from 0.4891 to 0.5415), indicating increased intra-cluster cohesion. The most notable improvement is observed in the Calinski-Harabasz Index, which increased by 479 units (from 204 to 683), highlighting the substantial superiority of DEC in identifying complex data structures. These figures clearly demonstrate that DEC outperforms the traditional K-means method in all aspects of clustering, including internal cohesion, separation between clusters, and overall quality.

Figure 1 provides a comprehensive visualization of the performance across all tested scenarios for the clustering algorithms. It clearly shows that the DEC method, executed in a single configuration, outperforms all others, achieving the best scores across every evaluation metric. The chart also reveals a consistent trend across methods such as K-means, Spectral Clustering, GMM, and Louvain; clustering quality initially improves with the removal of low-importance variables (from Scenario 1 onwards),

peaks at an optimal intermediate scenario (e.g., Scenario 4 for K-means, GMM, and Spectral; Scenario 5 for Louvain), and then declines, with further metric gains in the most reduced scenarios being misleading due to the over-simplification of the problem. In stark contrast, DEC achieved its optimal performance without any manual variable reduction, demonstrating its inherent ability to capture and preserve complex data structures. These results highlight the essential need for choosing the appropriate algorithm to ensure the precise analysis of multi-dimensional data.

5-4 Cluster Labeling and Interpretation

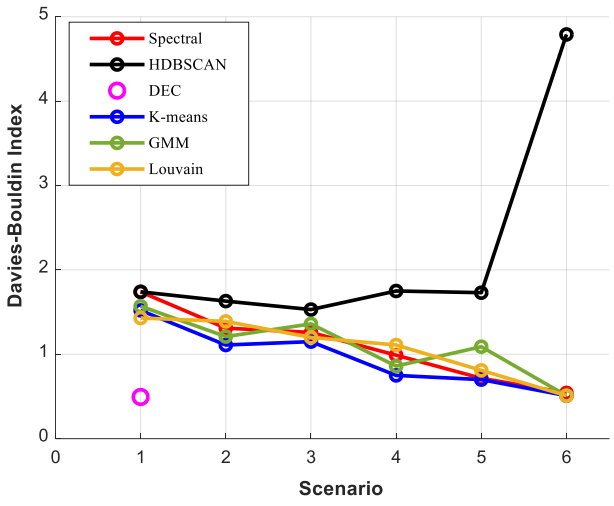
Based on comprehensive evaluations and in accordance with clustering standards, the DEC method was selected as the optimal algorithm for analyzing production units. Building on the results of this method, the identified clusters were labeled according to the average cluster centers presented in Table 9. The labeling was performed using the DEC approach and determined by key variables such as “city of the production unit’s location,” “number of manufactured products," and “average total negative score of the production unit for the period 2015–2023.”

Table 9) Cluster Centers Extracted Using the DEC Method

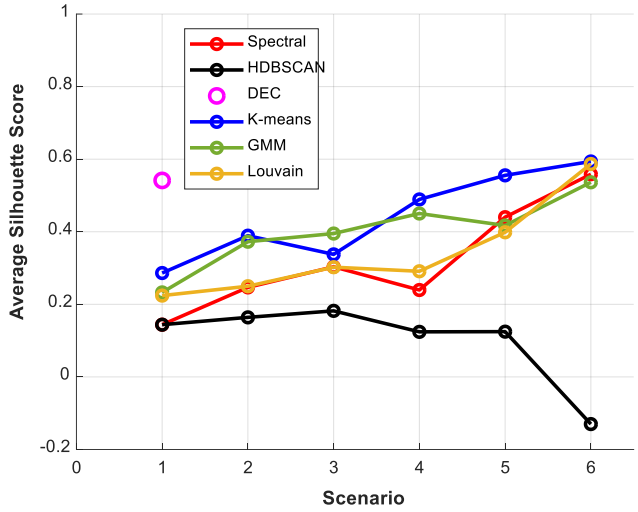
Variable	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
City of the Production Unit’s Location	8.24	109.14	63.46	94.57	34.57
Number of Manufactured Products	1.6	3.41	2.87	16.86	2.54
Average Total Negative Score of the Production Unit for the period 2015-2023	3.72	6.80	6.40	6.96	4.13

Figure 1) Comparison of Different Clustering Methods Based on (a) Davies–Bouldin Index, (b) Average Silhouette Score, and (c) Calinski–Harabasz Index

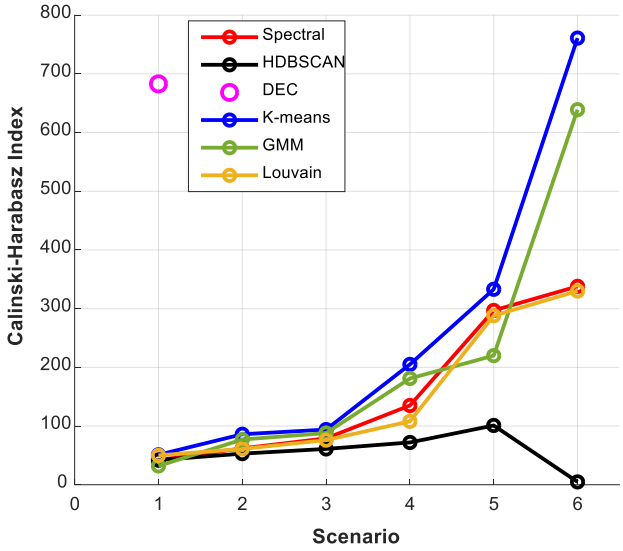
a



b



c



- **Cluster 1:** Production units located near Bojnurd, with a relatively good average negative score and a low number of products. This cluster is considered low-risk due to its favorable geographical location and relatively satisfactory performance.
- **Cluster 2:** Production units located far from Bojnurd, with a poor average negative score and a medium number of products. The combination of geographical distance and poor performance makes this cluster high-risk.
- **Cluster 3:** Production units located relatively far from Bojnurd, with an average negative score and a low number of products. Moderate supervision is required due to their intermediate distance and performance.
- **Cluster 4:** Production units located relatively far from Bojnurd, with a poor average negative score and a high number of products. High production volume coupled with poor performance makes this cluster critical.
- **Cluster 5:** Production units located relatively close to Bojnurd, with an average negative score and a low number of products. Despite their favorable location, their average performance necessitates monitoring.

Based on the comprehensive analyses conducted, and considering the key variables of distance from the center, production volume, and average negative score, the studied production units were classified into five distinct clusters. The clustering reveals that 20.3% of the units fall into high-risk clusters

(Clusters 2 and 4), requiring strict supervision, while 24.9% of the units (Cluster 1) are categorized as low-risk due to their favorable location and appropriate performance. Clusters 3 and 5, comprising 54.8% of the units, require moderate supervision. This classification, validated by experts from the North Khorasan Provincial Administration of the National Standards Organization, provides a scientific foundation for establishing a tiered monitoring system. Such a system ensures quality control through intelligent resource allocation while reducing unnecessary supervisory burden. The implementation of this system fully complies with the guidelines of the Iranian National Standards Organization and is designed with the capability for periodic revision.

6. Conclusion and Suggestions

In this study, data mining and deep learning techniques were applied to cluster production units in North Khorasan Province. The clustering was conducted using six key variables including: "Industry Type," "Industry Classification," "Number of Quality Control Managers," "Highest Education Level of Quality Control Manager," "Number of Manufactured Products," "City of the Production Unit's Location," and "Average Total Negative Score of the Production Unit for the period 2015-2023." This process successfully identified five distinct clusters of units with shared geographical and qualitative characteristics, each assigned a specific supervisory level, ranging from low to strict monitoring. Performance evaluation showed that the DEC method outperformed the previously used K-means algorithm, achieving a reduction of 0.24 in the Davies–Bouldin Index, an increase of 0.052 in the average Silhouette score, and a remarkable improvement of 479 in the Calinski–Harabasz Index. These results highlight DEC's ability to uncover hidden structures with greater accuracy, producing clusters with stronger internal cohesion and clearer separability. Consequently, this approach facilitates a risk-based, targeted supervisory strategy that can enhance the efficiency and effectiveness of regulatory processes, particularly for high-risk units with poor quality records.

For future research, it is recommended to develop hybrid fuzzy–deep clustering approaches in a staged manner, beginning with basic algorithms, such as Fuzzy C-Means, and advancing to more sophisticated models, such as Fuzzy DEC. This would preserve operational efficiency while enabling more precise analysis of borderline and ambiguous data. Additionally, enriching the dataset with complementary indicators, such as laboratory test results, quality parameters of raw materials, and inspection reports, would enhance the robustness of the analyses. Finally, examining the generalizability of these findings to other provinces could contribute to the development of a unified national monitoring framework, thereby improving the effectiveness of supervisory processes across the country.

Acknowledgements

We sincerely thank the North Khorasan Standard Office for providing reliable data that was crucial for achieving practical results and enhancing supervision quality.

References

Caliński, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-Theory and Methods, 3*(1), 1–27. https://doi.org/10.1080/03610927408827101

Campello, R. J. G. B., Moulavi, D., & Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 160–172). Springer. https://doi.org/10.1007/978-3-642-37456-2_14

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 132–149).

Chen, K., Zhang, P., Yan, H., Chen, G., Sun, T., Lu, Q., Chen, Y., & Shi, H. (2024). A review of machine learning in additive manufacturing: Design and process. *The International Journal of Advanced Manufacturing Technology, 135*(3), 1051–1087. https://doi.org/10.1007/s00170-024-13094-w

Combe, D., Largeron, C., Géry, M. & Egyed-Zsigmond, E., (2015, October) . I-louvain: An attributed graph clustering method. In *International Symposium On Intelligent Data Analysis* (pp. 181-192). Springer International Publishing.

Davies, D. L., & Bouldin, D. W. (2009). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 2*(2), 224–227. https://doi.org/10.1109/TPAMI.1979.4766909

Dogan, A., & Birant, D. (2021). Machine learning and data mining in manufacturing. *Expert Systems with Applications, 166*, 114060. https://doi.org/10.1016/j.eswa.2020.114060

Dollmann, M.M., (2023). Graph Clustering: A comparison of Louvain and Leiden. In *Conf. Ser* (Vol. 2129, p. 012028).

Ghousi, R. (2015). Applying a decision support system for accident analysis by using data mining approach: A case study on one of the Iranian manufactures. *Journal of Industrial and Systems Engineering, 8*(3), 59-76.

Guo, X., Liu, X., Zhu, E., & Yin, J. (2017). Deep clustering with convolutional autoencoders. In *Neural Information Processing* (pp. 373–382). Springer. https://doi.org/10.1007/978-3-319-70096-0_39

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems, 17*(2–3), 107–145. https://doi.org/10.1023/A:1012801612483

Harding, J. A., Shahbaz, M., Srinivas, & Kusiak, A. (2006). Data mining in manufacturing: A review. *Journal of Manufacturing Science and Engineering, 128*(4), 969–976. https://doi.org/10.1115/1.2194554

Jacobs, A. (2009). The pathologies of big data. *Communications of the ACM, 52*(8), 36–44. https://doi.org/10.1145/1536616.1536632

Kampezidou, S. I., Ray, A. T., Bhat, A. P., Fischer, O. J. P., & Mavris, D. N. (2024). Fundamental components and principles of supervised machine learning workflows with numerical and categorical data. *Eng, 5*(1), 384–416. https://doi.org/10.3390/eng5010021

Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: An introduction to cluster analysis*. John Wiley & Sons.

Khadivar, A., & Mojibian, F. (2022). Workshops clustering using a combination approach of data mining and MCDM. *Modern Researches in Decision Making, 7*(2), 1–20.

Kusiak, A. (2017). Smart manufacturing must embrace big data. *Nature, 544*(7648), 23–25. https://doi.org/10.1038/544023a

Liu, J., & Han, J. (2018). Spectral clustering. In *Data clustering* (pp. 177–200). Chapman and Hall/CRC.

McAfee, A., Brynjolfsson, E., Davenport, T. H., Patil, D. J., & Barton, D. (2012). Big data: The management revolution. *Harvard Business Review, 90*(10), 60–68.

McInnes, L., Healy, J., & Astels, S. (2017). HDBSCAN: Hierarchical density-based clustering. *Journal of Open Source Software, 2*(11), 205. https://doi.org/10.21105/joss.00205

Milligan, G. W., & Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika, 50*(2), 159–179. https://doi.org/10.1007/BF02294245

Molaee Fard, R. (2023). Provide a method to diagnose and optimize diabetes using data mining methods and firefly algorithm. *Engineering Management and Soft Computing, 9*(1), 36-48. https://doi.org/10.22091/JEMSC.2022.6575.1147

Ng, A., Jordan, M., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems, 14*, 849–856.

Niavand, M., Adibi, M. A., & Pourghader Chobar, A. (2024). Selection of green supplier by multi-moora combination method and two-stage clustering. *Engineering Management and Soft Computing, 10*(1), 14-49. https://doi.org/10.22091/jemsc.2024.10977.1181

Pakzad, A., Vahdani, M., & Khoran, M. (2024). Personalization of sampling and standard inspection based on data mining. In *The 10th International Conference on Industrial Engineering and Systems*. Mashhad, Iran. https://civilica.com/doc/2119451

Rahimi, F., Kamranrad, R., & Zarei, A. (2022). Design of integrated clustering-association data mining model to study the electricity consumption behavior of industrial units. *Iranian Journal of Energy, 25*(3), 65–78.

Raptis, T. P., Passarella, A., & Conti, M. (2019). Data management in Industry 4.0: State of the art and open challenges. *IEEE Access, 7*, 97052–97093. https://doi.org/10.1109/ACCESS.2019.2929291

Reynolds, D., (2015). Gaussian mixture models. In *Encyclopedia of biometrics* (pp. 827-832). Springer. https://doi.org/10.1007/978-0-387-73003-5_196

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics, 20*, 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

Sheikh Shoaee, H. (2021). A review of machine learning and data mining in the manufacturing industry. In *The 2nd National Conference on Management and Tourism Industry*. Tehran, Iran.

Song, H., Li, C., Fu, Y., Li, R., Zhang, H., & Wang, G. (2023). A two-stage unsupervised approach for surface anomaly detection in wire and arc additive manufacturing. *Computers in Industry, 151*, 103994. https://doi.org/10.1016/j.compind.2023.103994

Suman, S., & Das, A. (2020). Fuzzy clustering-based process-monitoring strategy for a multistage manufacturing facility. In *Soft Computing: Theories and Applications: Proceedings of SoCTA 2018* (pp. 459-469). Springer Singapore.

Tran, T.-H., Cao, T.-D., & Tran, T.-T.-H. (2021). HDBSCAN: Evaluating the performance of hierarchical clustering for big data. In *Soft computing: Biomedical and related applications* (pp. 273–283). Springer. https://doi.org/10.1007/978-3-030-73103-8_20

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing, 17*(4), 395–416. https://doi.org/10.1007/s11222-007-9033-z

Wang, Y., Chen, Q., Kang, C., & Xia, Q. (2016). Clustering of electricity consumption behavior dynamics toward big data applications. *IEEE Transactions on Smart Grid, 7*(5), 2437–2447. https://doi.org/10.1109/TSG.2016.2548565

Xie, J., Girshick, R., & Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning* (pp. 478–487). PMLR.

Younespour, M. S., & Romoozi, M. (2023). wireless sensor network clustering based on label propagation algorithm. *Engineering Management and Soft Computing, 8*(2), 16-29.

Zidek, K., Maxim, V., Pitel, J., & Hosovsky, A. (2016). Embedded vision equipment of industrial robot for inline detection of product errors by clustering–classification algorithms. *International Journal of Advanced Robotic Systems, 13*(5), 1–14. https://doi.org/10.1177/1729881416664901