

Article

Clustering of Crimes Using Latent Representations Obtained via Autoencoders

Weronika Nadworska * , Magdalena Pilat-Rożek  and Ewa Łazuka 

Faculty of Mathematics and Information Technology, Lublin University of Technology, 20-618 Lublin, Poland; m.pilat-rozek@pollub.pl (M.P.-R.); e.lazuka@pollub.pl (E.Ł.)

* Correspondence: w.nadworska@pollub.pl

Abstract

This article presents the use of autoencoders as part of a dimensionality-reduction method in the task of crime clustering. The study was conducted on a real-world crime dataset from the city of Chicago, based on publicly available police records. The initial data processing involved selecting and extracting variables, aggregating the selected variables, and converting crime categories and incident locations into contextual embeddings. The data prepared in this way was used to train various autoencoder architectures, including Vanilla, convolutional, denoising and variational models. The representations obtained from the latent layer of the encoder were then used as input data for clustering methods, such as *k*-means, Gaussian mixture model, and spectral clustering. The experimental results showed that the use of autoencoders in the clustering process enabled the identification of distinct groups of offences, with the best results (measured using the ARI and NMI metrics) obtained for Vanilla autoencoders combined with *k*-means and GMM, particularly with intermediate latent-space dimensions. The results confirm the potential of autoencoders as effective tools for dimensionality reduction and feature extraction in crime data analysis, as well as their usefulness in the exploratory analysis of complex urban data.

Keywords: autoencoders; deep learning; latent representations; clustering

1. Introduction

In recent years, there has been rapid development in machine learning methods, which are used in a variety of fields. Deep learning models, whose architecture is based on neural networks, play a particularly significant role. These models are capable of recognising complex patterns contained within data. Thanks to the growing availability of high computing power and large datasets, these methods are successfully used in, amongst other things, image and audio analysis, natural language processing, and the exploration of high-dimensional data.

One important class of deep learning models is autoencoders, which enable the learning of a compressed representation of the input data by transforming it into a lower-dimensional space. These models are used in anomaly detection, feature extraction, and dimensionality reduction, serving as an effective tool for analysing complex datasets. In particular, the latent space of an autoencoder can serve as a representation of the input data for further analytical steps, such as clustering.

Crime data is characterised by high dimensionality, a complex structure and the presence of outliers. The effective identification of patterns and similarities between reported crimes is crucial for exploratory data analysis as well as a better understanding



Academic Editor: Luis Javier Garcia Villalba

Received: 5 June 2026

Revised: 18 July 2026

Accepted: 20 July 2026

Published: 22 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

of the dynamics of crime in urban environments. This paper proposes the use of various autoencoder architectures to train representations of crime data, which were subsequently used in a crime clustering task.

The aim of this study is to investigate the impact of the choice of autoencoder architecture and the dimension of the latent space on the quality of crime data clustering. The research was conducted using a real-world dataset on crime in the city of Chicago, and the results obtained allow us to assess the usefulness of autoencoders as a preliminary data processing step in crime analysis.

2. Related Works

Autoencoders (AEs) are based on artificial neural networks. Neural networks are models inspired by the functioning of the human brain and are used for information processing and data-based pattern recognition. These models form the foundation of modern deep learning methods, including autoencoders [1].

The groundwork of neural networks was set alongside work on the creation of the first modern neuron, known as the perceptron. The way it functions bears similarity to models of animal and human learning developed in psychology [2]. In the decades that followed, developments in the field of deep learning enabled the construction of models capable of extracting feature representations from high-dimensional data.

Autoencoders were first used as a method of unsupervised pre-training in Ballard's paper [3]. An autoencoder is a type of neural network designed to encode input data into a compressed representation, whilst preserving key information, followed by the decoding of said data in such a way that the reconstructed input data is as similar as possible to the original [4–6].

A significant extension of classical autoencoders (Vanilla autoencoders—Vanilla AEs) are variational autoencoders (VAEs), which aim to learn latent representations and probability distributions. This enables a new generation of data to be created and improves the effectiveness of input distribution modelling [7]. These models have found applications in many fields, including complex data analysis and generative artificial intelligence [8]. Convolutional autoencoders (CAEs), on the other hand, which are based on convolutional layers, effectively learn more complex data representations and reduce the input data to a lower-dimensional space [9,10].

Over the years, numerous clustering algorithms have been developed, ranging from classic partitioning methods to graph-based approaches capable of identifying clusters with irregular shapes and complex similarity structures [11,12]. The rapid development of deep learning has further expanded these methods by introducing representation learning, whereby latent feature spaces, which are learned by neural networks, improve clustering quality through nonlinear dimensionality reduction and feature extraction [13]. These developments are part of a broader trend in computational urban analytics, where machine-learning-based approaches are increasingly used to extract meaningful patterns from complex urban systems. Recent studies have demonstrated the potential of data-driven frameworks for analysing heterogeneous urban environments by integrating multiple indicators and deriving latent representations of urban characteristics [14]. Similar methodological principles have also been adopted in urban crime analytics, where machine learning and deep learning techniques are increasingly employed to analyse large-scale spatiotemporal crime datasets, identify crime hotspots and support predictive policing [15]. More recently, graph neural networks have attracted considerable attention due to their capacity to model spatial interactions between neighbouring urban regions while simultaneously being able to capture temporal dependencies, thus improving performance in crime-forecasting tasks [16].

The growing popularity of representation learning has sparked interest in the application of autoencoders to unsupervised data analysis. Studies conducted in domains such as bioinformatics and anomaly detection have demonstrated that clustering performed on latent representations learned by AEs can produce more compact, homogeneous and semantically meaningful clusters than clustering performed directly in the original feature space [13,17,18]. By learning compressed feature representations, autoencoders effectively reduce data dimensionality while preserving the most informative characteristics of the input data, making them particularly suitable for high-dimensional datasets.

AEs have also proven effective in anomaly detection, as they learn to accurately reconstruct typical observations while producing larger reconstruction errors for anomalous or previously unseen instances [19]. This property makes them especially useful in solving unsupervised learning problems involving complex and unstructured data.

Despite these developments, the application of autoencoder-based latent representations to the unsupervised exploration of urban crime data remains relatively limited, particularly with respect to comparative analyses involving multiple autoencoder architectures combined with different clustering algorithms. This study addresses this gap by systematically comparing classical, denoising, variational and convolutional autoencoders as feature extraction methods prior to clustering, thereby providing a comprehensive evaluation of their effectiveness in the analysis of real-world urban crime data.

3. Materials and Methods

3.1. Data and Preliminary Processing

The dataset used in the study was obtained from the City of Chicago Open Data Portal [20], where it is updated daily, and contains information on reported crimes in the city of Chicago between 2001 and 2024. The data includes, amongst other things, the type of crime, the location of the incident, details of any arrests, and the date the incident was reported.

For the purposes of this study, the dataset was limited to crimes reported between 2014 and 2024. Detailed geographical variables, including *Latitude*, *Longitude*, *X Coordinate*, *Y Coordinate*, *Location*, *Ward*, and *District*, were removed to reduce redundancy and avoid overrepresenting highly correlated spatial information. Instead, the *Community Area* variable was retained as a broader spatial descriptor. This variable assigns each incident to one of the 77 official community areas of Chicago and therefore preserves information about the urban and socio-spatial context while reducing feature redundancy. However, we acknowledge that removing precise coordinates limits the possibility of identifying micro-level crime hotspots, which is treated as a limitation of the present study. A new variable, *Season*, was created based on the *Date* variable to specify the season in which the crime was reported, after which the original variable was removed. Observations containing missing data in the *Location Description* column (crime location) were also removed from the dataset. Furthermore, the levels of selected categorical variables (*Primary Type*, *Description* and *Location Description*) were aggregated to reduce the number of unique categories.

Subsequently, the textual variables used for embedding consisted of 30 unique levels for *Primary Type*, 300 unique levels for *Description*, and 103 unique levels for *Location Description*. Each variable was transformed using a pre-trained *SentenceTransformer* model *all-MiniLM-L6-v2* [21], which produced a 384-dimensional embedding for each textual variable. Therefore, the three embedded variables would have increased the dimensionality of the dataset by 1152 additional features. To reduce computational complexity and avoid memory-related limitations during autoencoder training, Principal Component Analysis (PCA) [22] was applied to the embedding matrix and the resulting representation was reduced to 100 dimensions. This value was selected as a practical compromise between

preserving the semantic information contained in the embeddings and maintaining computational feasibility. The preliminary application of PCA enabled efficient training of the autoencoders on the high-dimensional *SentenceTransformer* representations. However, since PCA is a linear dimensionality reduction technique, it may discard nonlinear relationships that could otherwise be captured by deep neural networks. This limitation should be considered when interpreting the results, and future work may investigate architectures capable of processing the full embedding representations without preliminary PCA reduction.

The remaining categorical variables were encoded using the *One-hot encoding* method. Following all the aforementioned transformations, the resulting dataset contained 2,731,705 observations and 458 variables. The data was then split into a training set and a test set at a ratio of 80:20. The test set served as a validation set and was utilised during the autoencoder training phase. It was used to monitor the reconstruction error and limit model overfitting. After the training process was completed, an encoded representation was derived for the entire dataset and then used to perform clustering.

No additional feature standardization was applied before training the autoencoders. This decision was based on the numerical characteristics of the input data. The *SentenceTransformer* embeddings produced continuous values of comparable magnitude, whereas the remaining variables were encoded as binary features. Consequently, all input variables were already represented on similar numerical scales, and no feature exhibited a substantially larger numerical range that could dominate the Mean Squared Error (MSE) reconstruction loss.

3.2. Autoencoders

An autoencoder is a type of artificial neural network used to learn a compressed representation of raw data. This model consists of two parts: an encoder, which transforms the input data into a lower-dimensional space (known as the latent space), and a decoder, whose task is to reconstruct the data based on its compressed representation. Figure 1 illustrates how Vanilla AE works, where the input image is encoded into a compressed form using the encoder and then reconstructed from that representation using the decoder.

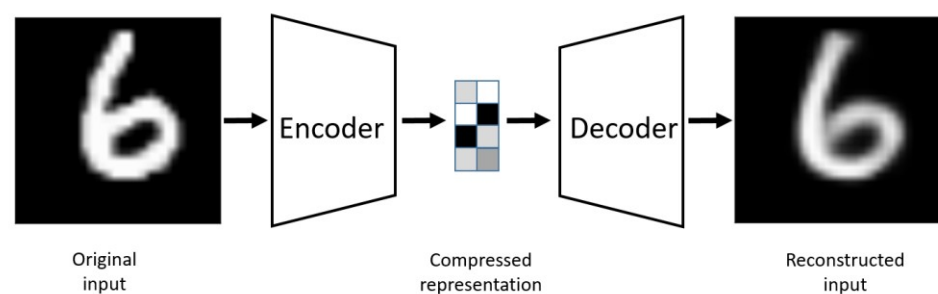


Figure 1. How an autoencoder works [5].

The training process for this model involves minimising the difference between the input data and its reconstruction, thereby enabling the model to learn to capture the most significant patterns present in the data [6].

To prevent a perfect mapping of the input data, various constraints are applied in the architecture of autoencoders, such as a bottleneck in the latent space or additional regularisation mechanisms. This forces the model to capture only the main features of the data. For this reason, AEs are used, among other things, in dimensionality reduction, feature extraction, anomaly detection and data clustering [5].

The literature identifies many variants of autoencoders. One such variant is CAE, which uses convolutional layers to extract local patterns in the data. These models are

particularly useful in the analysis of spatially structured data, such as images, where it is important to preserve the relationships between neighbouring data elements.

Another important class of models are VAEs, which are based on probabilistic modelling of latent representations. Unlike classical autoencoders, the encoder in a VAE model does not directly generate an encoded representation of the input sample, but instead learns the parameters of a probability distribution—the mean and variance for each latent variable. The latent representation is then sampled from this distribution, on the basis of which the decoder reconstructs the input data. However, random sampling introduces difficulties in applying the backpropagation algorithm. Therefore, to enable the correct learning of the distribution parameters, the so-called *reparameterisation trick* is used, which allows the random component to be separated from the parameters optimised during training [7].

This study also utilised denoising autoencoders (DAEs), which are an extension of Vanilla AE architecture. In these models, a so-called noise layer is introduced into the input data, whilst the decoder's task is to reconstruct the original, uncontaminated data. This approach allows for the learning of a more robust data representation, which reduces the risk of the model overfitting the training set [5].

3.3. Clustering Methods

The information contained in the latent layers of trained autoencoders was used as a representation of input features for the crime clustering task. The aim was to investigate whether compressed data representations allow for the identification of natural structures within the crime dataset. Three clustering methods were used in the analysis: *k*-means, a Gaussian mixture model (GMM), and spectral clustering (SC).

To justify the range of cluster numbers used in the experiments, an additional preliminary cluster-number analysis using the *k*-means method was performed for *k* values from 2 to 10 using the silhouette score on a sample of 100,000 observations with a fixed random seed. The analysis indicated that the most stable and interpretable solutions were obtained for *k* values within the range of 2 to 5. Therefore, the main experiments were restricted to this range to ensure comparability between clustering methods and autoencoder architectures. Calculations were performed for all analysed autoencoder models, i.e., Vanilla, DAE, CAE, VAE. The results are presented in Table A1 in Appendix A.

3.3.1. *k*-Means Method

The *k*-means method is one of the simplest and most commonly used clustering techniques. It involves dividing a dataset into *k* clusters by iteratively assigning points to the nearest centroid and updating the position of the centroids which serve as the centres of mass for the points assigned to them [11,23]. The aim of the algorithm is to minimise the sum of the squares of the points' distances from the group centroids. This method assumes that the clusters are spherical in shape and of similar size.

3.3.2. Gaussian Mixture Model

The GMM is a probabilistic approach to clustering, in which data are modelled as a combination of multiple Gaussian distributions. Each cluster is represented by a single normal distribution described by the parameters of mean and variance. These parameters are estimated using the Expectation-Maximisation algorithm, and data are assigned to clusters based on their probabilities [24]. Unlike the *k*-means method, GMM allows for the modelling of clusters of various shapes, sizes and orientations within the feature space.

3.3.3. Spectral Clustering

SC is a method based on the spectral analysis of a neighbourhood matrix or a similarity matrix between data points [12]. This method utilises the eigenvalues and eigenvectors

of the Laplacian matrix of the graph constructed from the data. Its eigenvalues and eigenvectors are then calculated and used to create a reduced representation of the data. The actual clustering is performed in this space, enabling the identification of clusters with complex and non-linear shapes.

3.4. Metrics

In this study, the ARI (*Adjusted Rand Index*) and NMI (*Normalized Mutual Information*) metrics were used to evaluate the performance of the analysed models [25].

ARI assesses the accuracy of assigning observations to clusters relative to the true labels, taking probability into account. It presents the result on a scale of $[-1, 1]$, where a value of 1 indicates perfect agreement with the true labels, a value of 0 indicates agreement equivalent to random assignment, and a value less than 0 indicates worse agreement than random assignment. ARI can be calculated using the following formula:

$$\text{ARI} = \frac{\sum_{ij} \binom{n_{ij}}{2} - \frac{\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}}{\binom{n}{2}}}{\frac{1}{2} \left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2} \right] - \frac{\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}}{\binom{n}{2}}}, \quad (1)$$

where n_{ij} represents the number of observations belonging to actual cluster i and a predicted cluster j , a_i stands for the number of observations in actual cluster i , b_j is the number of observations in predicted cluster j , and n is the total number of observations [26].

The NMI metric indicates the degree to which the values of the predicted clusters and the actual clusters resemble one another. Its value ranges from 0 to 1, where 0 indicates no agreement and 1 indicates a perfect match. The NMI can be calculated using the formula below:

$$\text{NMI} = \frac{2 \times I(T; P)}{H(P) + H(T)}, \quad (2)$$

where T and P are the actual labels and predicted clusters, respectively, $I(T; P)$ denotes the mutual information between the actual labels and the predicted clusters and $H(\cdot)$ is the entropy of a distribution [26].

For each experiment in this study, the cluster labels generated by the clustering algorithm were compared with the corresponding labels derived from the original dataset. For computational reasons and to ensure comparability across all experiments, the evaluation of ARI and NMI was performed on ten random samples of 10,000 observations drawn from the full dataset using fixed random seeds. The same samples were used for all compared clustering methods and autoencoder architectures. This sampling strategy was adopted to avoid potential bias that could arise from using the initial records of the dataset, while maintaining computational feasibility. The resulting ARI and NMI values quantified the degree of agreement between the identified clusters and the original class structure of the crime data. After completing all the calculations, the mean values of ARI and NMI, along with their standard deviations, were calculated for all ten samples.

4. Results

4.1. Autoencoder Models

Four types of autoencoders were used in the experiments:

- Vanilla AE—two architectures, each with two variants of the middle layer (four models in total);

- DAE—two architectures, two variants of the middle layer (four models in total);
- VAE—one architecture, three different dimensions of latent space (3 models in total);
- CAE—two architectures (2 models in total).

Each model was trained for 60 epochs using an *early stopping* mechanism (training was stopped if the loss function did not decrease for 5 consecutive epochs). The batch size was 256, and the *Adam* algorithm was used for optimisation. For Vanilla AEs, DAEs and CAEs the Mean Squared Error function was used as the reconstruction function.

$$L_{\text{MSE}} = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2, \quad (3)$$

where x_i is the expected outcome, $\hat{x}_i$ is the outcome obtained from the model, and n is the number of observations [27]. This function minimises the difference between the input data and the reconstruction generated by the model. The MSE is sensitive to outliers because it takes into account the squares of the errors, which means that larger errors have a much greater impact on the MSE [28].

In the case of DAE models, noise was added to the input data, whilst the models were trained to reconstruct the original data. Formally, $L_{\text{DAE}} = L_{\text{MSE}}$, where the model's input was the noised data $\tilde{x}$, and the reconstruction target remained the original data x .

For the CAE model, an identical MSE-based reconstruction function was used:

$$L_{\text{CAE}} = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2. \quad (4)$$

In this case, encoding and decoding were carried out using convolutional layers.

In this study, the convolutional autoencoder was implemented using one-dimensional convolutional layers (Conv1D). The input crime data were represented as 458-dimensional feature vectors, and the Conv1D layers were applied along this feature dimension. Although convolutional architectures are most commonly used for image or sequential data, they were included here as a case of experimental architecture to test whether local dependencies between neighbouring transformed features could be exploited for representation learning. At the same time, we acknowledge that tabular crime data do not have a natural spatial ordering comparable to images, which limits the interpretability of convolutional filters in this setting.

In a VAE, the loss function consists of two main components: the reconstruction loss function (here, MSE) and the latent loss function. The former measures how well the decoder reconstructs the input data from the sampled latent space. The latter, in turn, measures the discrepancy between the approximated distribution $q(z|x)$ (resulting from the encoder's operation) and the prior distribution $p(z)$. The Kullback–Leibler divergence (KL divergence) is typically used to express this difference. The total loss function L_{VAE} of the variational autoencoder can be represented as:

$$L_{\text{VAE}} = L_{\text{MSE}} + \text{KL}(q(z|x)||p(z)), \quad (5)$$

where the KL component measures the discrepancy between the learned and the expected latent distributions.

If the normal distribution is assumed to be the latent distribution, the KL component can be expressed as:

$$\text{KL} = -\frac{1}{2} \sum_{i=1}^N (1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2), \quad (6)$$

where N is the dimension of the latent space, μ_i and σ_i are the mean and standard deviation, respectively, of the i -th latent variable. The encoder outputs vectors μ and σ , containing the values μ_i and σ_i for all dimensions, respectively, [29]. Here, the weighting coefficient of the KL divergence was set to $\beta = 1$, corresponding to the standard VAE formulation. No KL annealing strategy was applied; therefore, the KL regularization term was active throughout the entire training process.

Learning Curves

Figures 2–5 show the MSE values at successive training epochs for the constructed models.

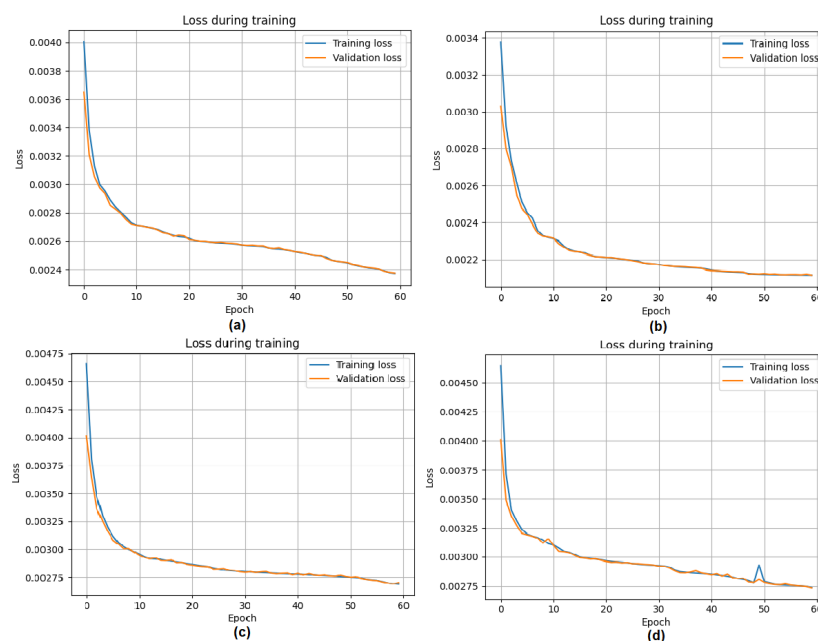


Figure 2. Plots of the training process for the constructed Vanilla AEs: (a) AE No. 1, (b) AE No. 2, (c) AE No. 3, (d) AE No. 4.

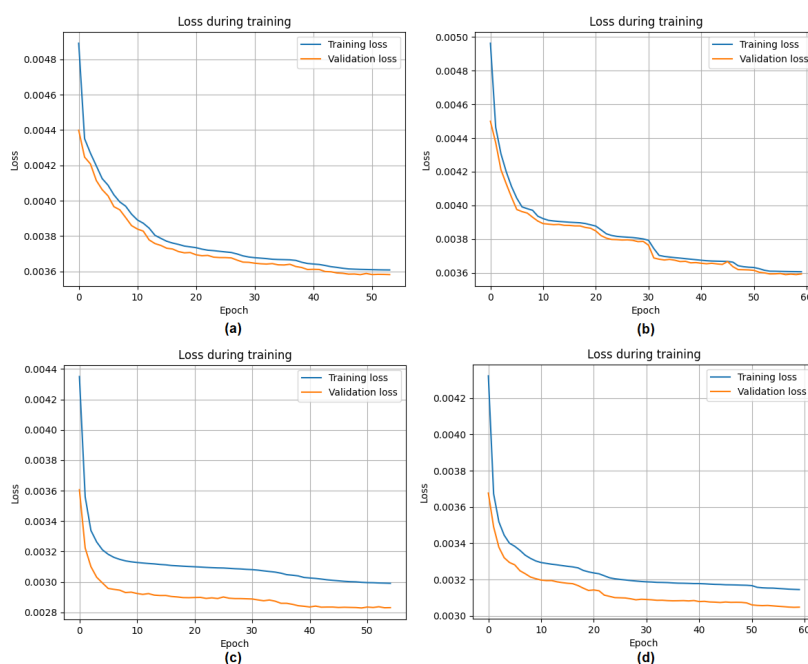


Figure 3. Plots of the training process for the DAEs: (a) DAE No. 1, (b) DAE No. 2, (c) DAE No. 3, (d) DAE No. 4.

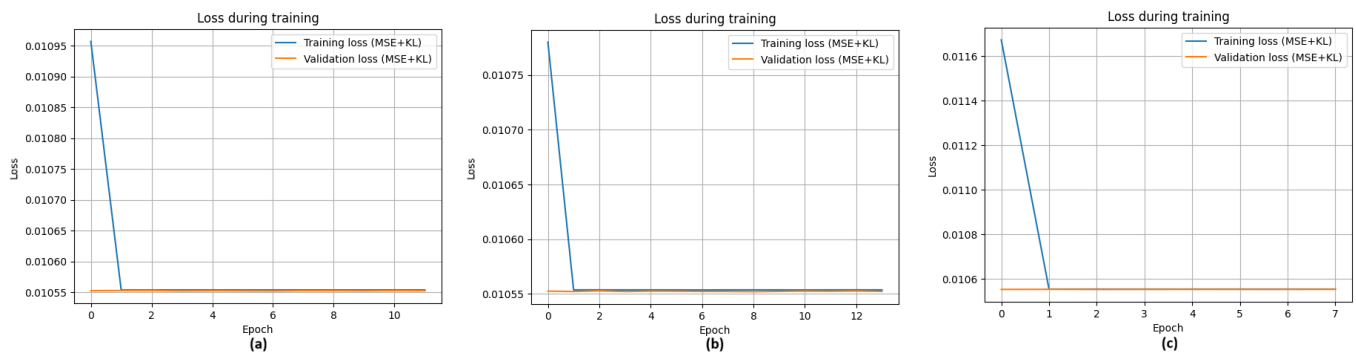


Figure 4. Plots of the training process for the constructed VAEs: (a) VAE No. 1, (b) VAE No. 2, (c) VAE No. 3.

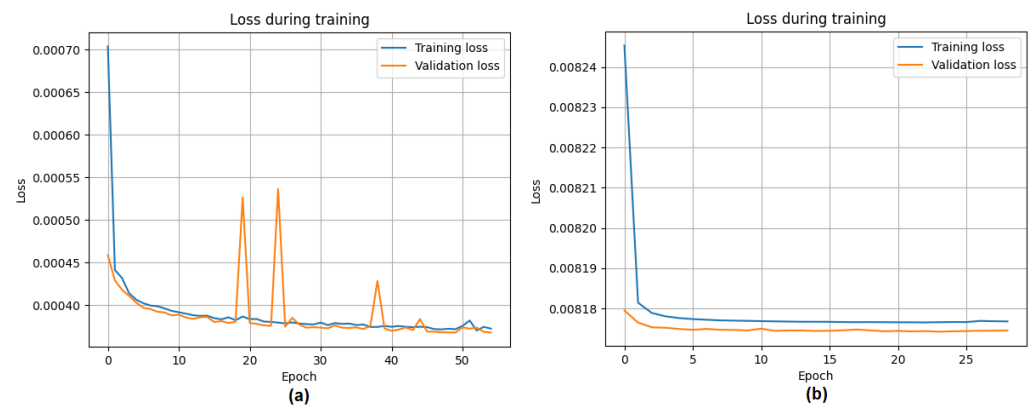


Figure 5. Training plots for the CAEs: (a) CAE No. 1, (b) CAE No. 2.

On the basis of the charts shown in the figures above, the following conclusions can be drawn:

- Vanilla AE: The initial epochs are characterised by the steepest decline in MSE; in subsequent epochs, the loss function decreases more slowly;
- DAE: A similar trend to that of Vanilla AEs; in models with a 90-dimensional hidden layer, the MSE on the validation set is lower than on the training set;
- VAE: The training was terminated by the early stopping mechanism due to limited improvement in the loss function values;
- CAE: The MSE value remains stable throughout the training process; the training process itself is terminated by the *early stopping* mechanism.

The detailed architectures of all models are presented in the Appendix B (Tables A2–A14).

4.2. Clustering Results

The results of crime clustering are presented using three approaches: the *k*-means method, GMM, and SC. For each algorithm, feature representations obtained from the latent layers of the autoencoders described earlier (Vanilla AE, DAE, VAE, CAE) were used. In the visualizations presented below, which illustrate the effectiveness of grouping the analysed data, each figure is shown using a sample of 2000 observations for clarity, but clustering was performed on the full dataset.

4.2.1. *k*-Means Clustering

Figure 6 show the results of *k*-means clustering in various configurations:

- DAE + 3-means (Figure 6a): the data were divided into three clusters based on the latent space of a denoising autoencoder with a hidden layer dimension of 50;

- DAE + 2-means (Figure 6b): a hidden layer of 90 was used, resulting in two clusters;
- CAE + 5-means (Figure 6c): a convolutional autoencoder with architecture No. 2 was used, resulting in five clusters.

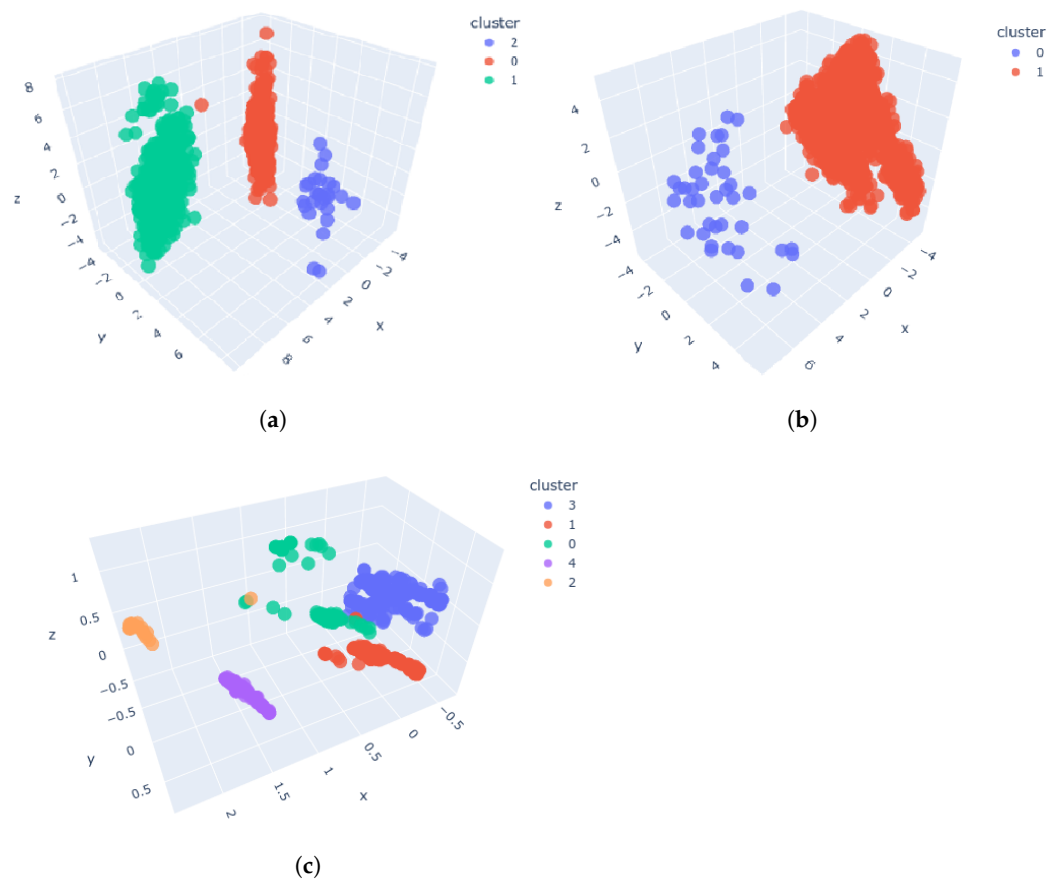


Figure 6. Clustering results using the *k*-means method (a) Clustering results for DAE + 3-means. (b) Clustering results for DAE + 2-means. (c) Clustering results for CAE + 5-means.

4.2.2. Gaussian Mixture Model

The results of the clustering using GMM are shown in Figure 7:

- DAE + GMM (Figure 7a,b): DAE architecture No. 2 was used with a 50-dimensional hidden layer for three clusters and with a 90-dimensional hidden layer for two clusters;
- CAE + GMM (Figure 7c): convolutional autoencoder No. 2 was used to identify five groups.

4.2.3. Spectral Clustering

Figure 8 show the results of spectral clustering:

- Vanilla AE + SC (Figure 8a): division of the data into two clusters in the latent space of Vanilla AE;
- DAE + SC (Figure 8b,c): a denoising autoencoder with a hidden layer dimension of 50 was used, yielding two clusters; some observations were identified as outliers;
- CAE + SC (Figure 8d): the use of convolutional autoencoder No. 1 allowed the data to be divided into four groups.

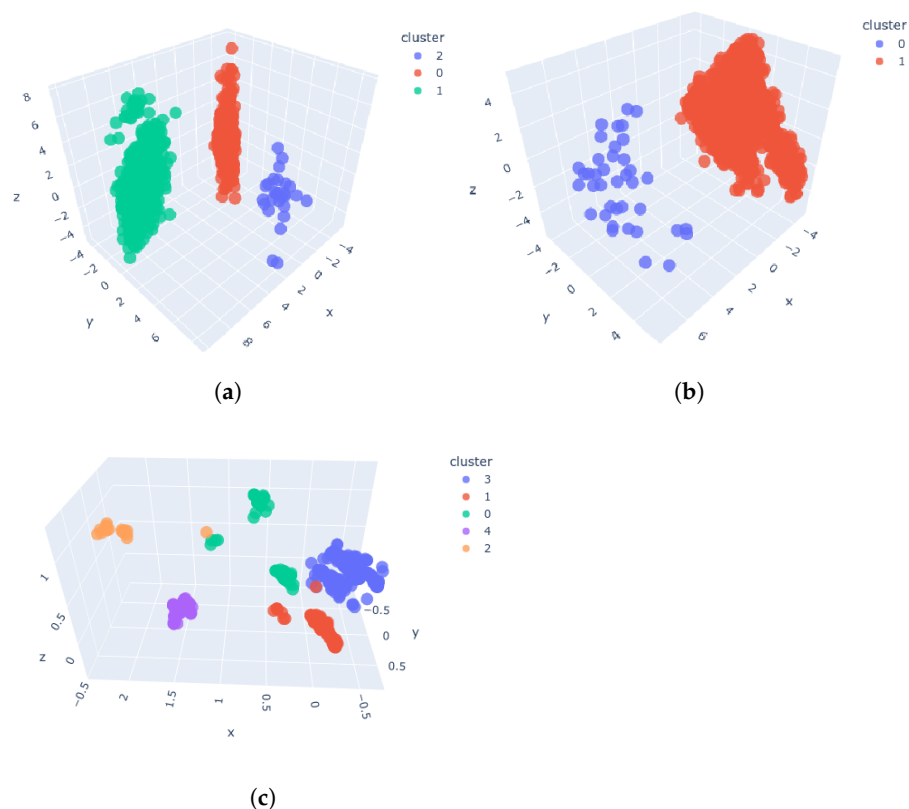


Figure 7. Clustering results using the GMM method (a) DAE + GMM clustering results. (b) DAE + GMM clustering results. (c) CAE + GMM clustering results.

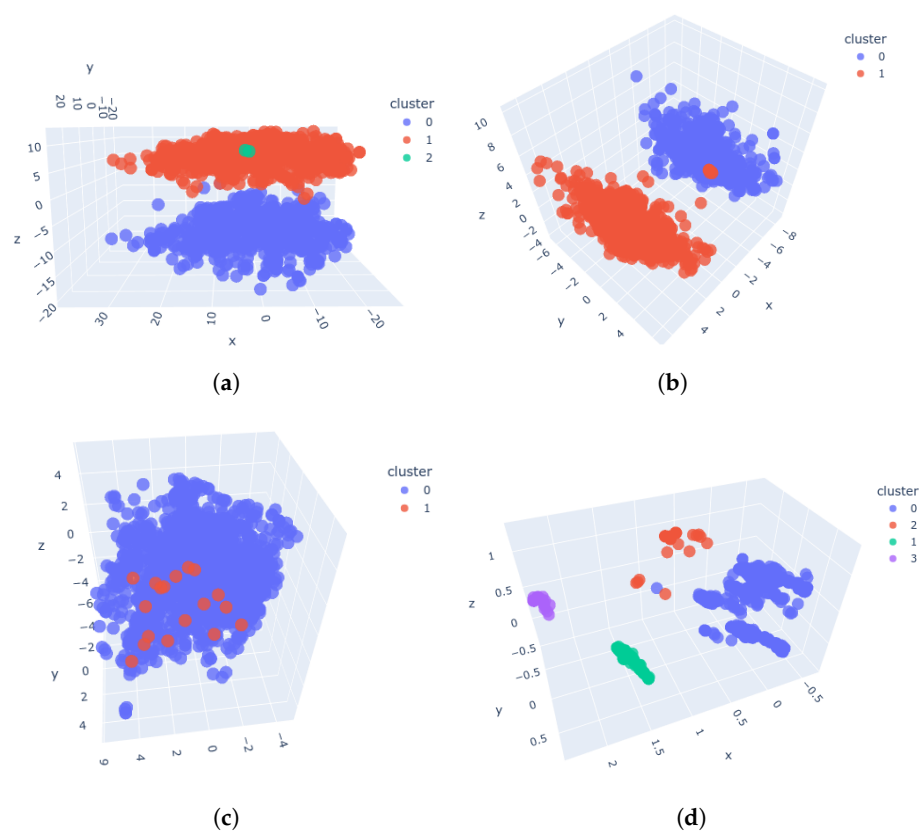


Figure 8. Clustering results using SC method (a) Clustering results for Vanilla + SC. (b) Clustering results for DAE + SC. (c) Clustering results for DAE + SC. (d) Clustering results for CAE + SC.

In all cases, the use of latent representations derived from autoencoders enabled the grouping of crime data in a lower-dimensional space, which facilitates the visualisation and further analysis of clusters.

5. Discussion

Table 1 summarises the mean values of the *Adjusted Rand Index* (ARI) and *Normalized Mutual Information* (NMI) with their standard deviations for various combinations of clustering methods (*k*-means, GMM, SC) and the number of clusters (2–5), trained on data representations obtained from selected autoencoder variants. The results were compared against three descriptive features: *Primary Type* (PT), *Description* (D) and *Location Description* (LD). The full results for all combinations of autoencoders and clustering methods are presented in Table A15 in Appendix C.

An analysis of the clustering results shows that the number of clusters had a significant impact on the quality of the data clustering. The best results were most often achieved with 4 or 5 groups. The Vanilla AE and *k*-means and GMM configurations were among the best in the overall results table. The mean ARI exceeded 0.5, and the mean NMI reached an average value of nearly 0.7 for the *Primary Type* variable. This may suggest that crime data can be divided into several main subgroups corresponding to different categories of criminal behaviour or different spatial contexts in which crime occurs.

From a socio-spatial perspective, these clusters can be interpreted as areas exhibiting an increased risk of specific types of crime, groups of incidents of a similar nature, or similar patterns of criminal behaviour. For example, some clusters corresponded to street crime, others to domestic violence, while the remaining instances centred on specific, rare crimes. The results therefore suggest that the latent representations obtained using autoencoders were capable of partially capturing hidden patterns in crime data.

At the same time, a decline in clustering quality was observed when the number of clusters was too small. For models with two clusters, the ARI values were lower. This means that dividing the criminal data into just two clusters led to an oversimplification of the data structure and the grouping of dissimilar categories of crime into a single cluster.

The DAE models in general achieved good results. For example, DAE No. 1, combined with the GMM method for 5 groups, achieved a mean ARI of 0.3858, whilst DAE No. 2, combined with GMM for 5 groups, achieved a mean ARI of 0.2949 for the *Primary Type* variable. Adding noise during training increased the model's resistance to noise and may have improved the generalisation of latent representations; however, it also led to a partial blurring of the differences between crime categories. The standard deviations obtained over repeated runs (typically around 0.07–0.10 for the best DAE configurations) indicate that these models produced reasonably stable clustering results, although they remained more variable than the best Vanilla AE models.

The CAE models achieved lower results, generally outperforming VAEs but underperforming Vanilla AEs and DAEs. This may indicate that convolutional architectures were not optimally suited to the nature of the data being analysed. Since the input consisted of heterogeneous tabular features rather than spatially organized data, neighbouring input dimensions did not necessarily correspond to semantically related variables. Consequently, the locality assumption underlying convolutional neural networks was only weakly satisfied, which may explain why CAEs did not outperform other architectures. Convolutional layers work best with image data or data exhibiting strong local spatial relationships. In the case of criminal data, the information analysed was mainly textual and categorical in nature, which is why the potential of convolutions may have been limited. Although the CAE models generally achieved lower clustering quality than the best Vanilla AE and DAE models, their stability across repeated runs was mixed. Some configurations

produced relatively consistent results, whereas others showed noticeably larger standard deviations, suggesting that differences in clustering quality were influenced more by the model architecture than by variability across repeated runs.

The VAE-based representations produced weaker clustering results than the best Vanilla AE and DAE configurations. This may be related to the probabilistic regularization imposed on the latent space, which can reduce the separability of categories in datasets dominated by categorical and text-derived features. Therefore, although VAEs are useful generative models, their latent representations were less suitable for the clustering task executed in this study. Although the VAE models often produced almost identical results across calculations, this apparent stability reflected consistently poor clustering performance rather than stable latent representations, as both ARI and NMI remained close to zero for most configurations.

When comparing clustering methods, it can be seen that the *k*-means method achieved the best results. The high effectiveness of this method suggests that the latent spaces generated by autoencoders contained relatively compact and well-separated clusters. GMM models appear among the top results, whilst additionally allowing for the modelling of more irregular data distributions. Spectral clustering, on the other hand, produced weaker and less stable results, which may indicate difficulties in mapping the latent space for criminal data.

The values obtained for the clustering quality metrics were highest for the *Primary Type* variable, whilst the results for the *Description* and *Location Description* variables were significantly lower. The *Primary Type* variable describes the main category of crime and constitutes the most general level of classification for criminal incidents. These categories are relatively well-separated in terms of similarity to one another and represent different types of criminal behaviour, such as theft, burglary, assault or drug-related offences. Autoencoder models are therefore able to generate more separable latent representations. The resulting clusters corresponded closely to actual crime categories, indicating the models' high ability to identify major patterns of criminality.

To assess the added value of autoencoder-based dimensionality reduction, an additional baseline experiment was performed by applying the same clustering algorithms directly to the 458-dimensional preprocessed dataset, without using latent representations. The baseline results are reported in Table A16 in Appendix D. In most configurations, direct clustering of the original feature space produced lower ARI and NMI values than the best autoencoder-based variants. This indicates that the learned latent representations improved the separability of the data and provided more useful inputs for clustering.

In summary, the results indicate that Vanilla AEs best preserved the structure of the crime data and enabled the creation of the most interpretable latent representations. The best results were achieved for 4–5 clusters and the *k*-means and GMM methods. CAE models achieved moderate results, whilst VAE models proved to be the least effective at clustering crime data.

Table 1. Comparison of the mean ARI and NMI results with their standard deviations for different clustering methods and numbers of clusters, with respect to the target variables: *Primary Type* (PT), *Description* (D), *Location Description* (LD) for 10 random samples (sorted by ARI values in decreasing order)—best results.

Method	No. Groups (k)	Target	Mean ARI $\pm$ sd	Mean NMI $\pm$ sd
GMM + Vanilla No. 4	5	PT	0.5299 $\pm$ 0.07	0.6950 $\pm$ 0.05
<i>k</i> -means + Vanilla No. 4	5	PT	0.5061 $\pm$ 0.08	0.6714 $\pm$ 0.06
<i>k</i> -means + Vanilla No. 3	5	PT	0.4664 $\pm$ 0.12	0.6391 $\pm$ 0.08
GMM + Vanilla No. 4	4	PT	0.4578 $\pm$ 0.05	0.6364 $\pm$ 0.04

Table 1. Cont.

Method	No. Groups (k)	Target	Mean ARI $\pm$ sd	Mean NMI $\pm$ sd
GMM + Vanilla No. 3	5	PT	0.4533 $\pm$ 0.11	0.6080 $\pm$ 0.09
k-means + Vanilla No. 4	4	PT	0.4361 $\pm$ 0.06	0.6092 $\pm$ 0.04
k-means + Vanilla No. 3	4	PT	0.3965 $\pm$ 0.08	0.5668 $\pm$ 0.08
GMM + DAE No. 1	5	PT	0.3858 $\pm$ 0.10	0.5647 $\pm$ 0.10
GMM + Vanilla No. 3	4	PT	0.3843 $\pm$ 0.08	0.5424 $\pm$ 0.09
k-means + DAE No. 1	5	PT	0.3767 $\pm$ 0.09	0.5493 $\pm$ 0.07
k-means + Vanilla No. 4	3	PT	0.3573 $\pm$ 0.03	0.5567 $\pm$ 0.02
GMM + DAE No. 1	4	PT	0.3563 $\pm$ 0.07	0.5408 $\pm$ 0.09
GMM + Vanilla No. 4	3	PT	0.3552 $\pm$ 0.03	0.5531 $\pm$ 0.02
k-means + DAE No. 1	4	PT	0.3468 $\pm$ 0.07	0.5236 $\pm$ 0.09
SC + Vanilla No. 3	5	PT	0.3303 $\pm$ 0.09	0.5699 $\pm$ 0.06
k-means + Vanilla No. 3	3	PT	0.3290 $\pm$ 0.05	0.5016 $\pm$ 0.07
GMM + Vanilla No. 3	3	PT	0.3001 $\pm$ 0.04	0.4611 $\pm$ 0.06
GMM + DAE No. 2	5	PT	0.2949 $\pm$ 0.07	0.4850 $\pm$ 0.08
SC + Vanilla No. 4	5	PT	0.2883 $\pm$ 0.10	0.5871 $\pm$ 0.09
k-means + DAE No. 2	5	PT	0.2835 $\pm$ 0.07	0.4562 $\pm$ 0.09
k-means + Vanilla No. 4	5	D	0.2693 $\pm$ 0.03	0.5572 $\pm$ 0.02
GMM + Vanilla No. 4	5	D	0.2632 $\pm$ 0.03	0.5564 $\pm$ 0.02
k-means + DAE No. 1	3	PT	0.2542 $\pm$ 0.07	0.4296 $\pm$ 0.09
SC + Vanilla No. 3	4	PT	0.2540 $\pm$ 0.03	0.4996 $\pm$ 0.02

6. Conclusions

This study demonstrated that autoencoder-based representations can improve the clustering of large-scale crime datasets by capturing latent relationships between categorical and textual attributes. Among the evaluated approaches, Vanilla AEs combined with the *k*-means and GMM algorithms consistently produced the most informative clustering structures. This approach also proved more effective than clustering the original data directly.

Several limitations of this study should be acknowledged. First, due to computational constraints, PCA was applied before training the autoencoders, which may have removed part of the nonlinear information contained in the original embedding vectors.

Second, the preprocessing stage involved the removal of several location-related variables that were considered redundant, while this reduced the dimensionality of the dataset, some potentially useful spatial information may have been lost, which could have influenced the quality of the learned latent representations and the subsequent clustering results.

Thirdly, the variables *Primary Type*, *Description*, and *Location Description* were used both to construct the input representation (through sentence embeddings) and as external reference labels for evaluating clustering quality using ARI and NMI. Consequently, the agreement between the obtained clusters and these variables may be partially influenced by the fact that the corresponding semantic information was already present in the input data. It should therefore be emphasized that the reported ARI and NMI values measured how well the learned latent representations preserved the original structure of the dataset rather than the ability of the models to discover completely independent crime categories.

Finally, the proposed AE architectures use a single MSE reconstruction loss for all input features. The input representation consisted of heterogeneous data types, including binary variables obtained through *One-hot encoding* and continuous features derived from PCA-reduced sentence embeddings. Although MSE provided a simple and unified optimisation objective, it does not fully reflect the statistical properties of different feature types. Future research may therefore investigate composite reconstruction losses combining

binary cross-entropy for binary variables with MSE or other appropriate loss functions for continuous features.

Despite these limitations, the proposed framework demonstrates the potential of representation learning for supporting exploratory analysis of urban crime and may assist decision-makers in identifying patterns of criminal activity.

Further prospective inquiry into this topic may focus on the application of deep clustering methods, adversarial learning, and models that account for spatiotemporal dependencies. Such approaches may improve cluster separation, increase the interpretability of the results, and provide a more detailed understanding of crime patterns in urban environments. In addition, future studies should investigate methods capable of processing high-dimensional embedding representations without preliminary PCA compression. Although the proposed framework achieved encouraging results, additional research is required to investigate automatic cluster selection strategies and representation learning techniques that preserve the full semantic information contained in textual embeddings while remaining computationally feasible. Future studies may also include ablation experiments in which variables used for external clustering evaluation are excluded from the autoencoder input. This would enable a more rigorous assessment of whether the learned latent representations capture general structural patterns in crime data rather than primarily preserving information contained in the evaluation variables.

Author Contributions: Conceptualization, E.Ł. and M.P.-R.; methodology, M.P.-R.; software, W.N.; validation, W.N., M.P.-R. and E.Ł.; formal analysis, W.N.; investigation, M.P.-R.; resources, E.Ł.; data curation, W.N.; writing—original draft preparation, W.N.; writing—review and editing, M.P.-R. and E.Ł.; visualization, W.N.; supervision, E.Ł.; project administration, E.Ł. and M.P.-R.; funding acquisition, M.P.-R. and E.Ł. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by grants from the Polish Ministry of Science and Higher Education: FD-20/IŚ-6/047, FD-20/DN-10/037, FD-20/DN-10/999 and FJ-KMS.

Institutional Review Board Statement: Ethical review and approval were waived for this study because it used publicly available, de-identified administrative crime records and did not involve interaction with human participants.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original crime dataset analysed in this study is openly available from the City of Chicago Data Portal, Crimes 2001 to Present, at <https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2> (accessed on 4 November 2024). The source code used for data preprocessing, model training, clustering, and evaluation is available in the GitHub repository at https://github.com/Weronika59/article_files2026 (accessed on 15 July 2026). The processed dataset generated during preprocessing is not deposited separately due to its size, but it can be reproduced from the public source data using the provided scripts and configuration files. Further inquiries can be directed to the corresponding author.

Acknowledgments: The computations were performed using computational resources provided by the Department of Applied Mathematics at Lublin University of Technology.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Silhouette score values for various numbers of clusters k in the k -means method and for different architectures of Vanilla AEs, DAEs, CAEs and VAEs (sorted by silhouette score in decreasing order).

Model	k	Silhouette Score
DAE No. 3	4	0.3870
DAE No. 3	3	0.3737
DAE No. 3	2	0.3641
Vanilla No. 4	10	0.2765
Vanilla No. 4	2	0.2729
Vanilla No. 4	8	0.2631
Vanilla No. 4	7	0.2623
Vanilla No. 1	2	0.2570
Vanilla No. 4	9	0.2514
Vanilla No. 4	6	0.2482
Vanilla No. 4	4	0.2356
CAE No. 2	2	0.2268
Vanilla No. 3	2	0.2242
Vanilla No. 4	3	0.2235
DAE No. 4	2	0.2201
Vanilla No. 3	5	0.2162
Vanilla No. 3	4	0.2090
Vanilla No. 4	5	0.2022
Vanilla No. 3	9	0.1915
Vanilla No. 3	8	0.1825
Vanilla No. 3	6	0.1799
Vanilla No. 3	3	0.1772
CAE No. 2	4	0.1390
VAE No. 2	7	0.1266
VAE No. 2	2	0.1262
Vanilla No. 1	10	0.0967
DAE No. 2	2	0.0963
Vanilla No. 1	6	0.0918
Vanilla No. 2	9	0.0897
Vanilla No. 1	5	0.0888
Vanilla No. 1	8	0.0815
VAE No. 1	10	0.0595
VAE No. 1	9	0.0584
VAE No. 1	8	0.0578
VAE No. 3	10	0.0132
VAE No. 3	9	0.0132
VAE No. 3	8	0.0131
VAE No. 3	6	0.0131
VAE No. 3	7	0.0131

Appendix B

Appendix B.1. Vanilla Autoencoders

Table A2. Architecture of Vanilla AE No. 1.

No.	Module	Layer Type	Output Dimension	Activation Function
1	encoder	input	458	–
2		dense	256	ReLU
3		dense	128	ReLU
4	decoder	hidden	50	ReLU
5		dense	64	ReLU
6		dense	128	ReLU
7		output	458	–

Table A3. Architecture of Vanilla AE No. 2.

No.	Module	Layer Type	Output Dimension	Activation Function
1	encoder	input	458	–
2		dense	256	ReLU
3		dense	128	ReLU
4	hidden		90	ReLU
5	decoder	dense	64	ReLU
6		dense	128	ReLU
7		output	458	–

Table A4. Architecture of Vanilla AE No. 3.

No.	Module	Layer Type	Output Dimension	Activation Function
1	encoder	input	458	–
2		dense	128	ReLU
3		dense	64	ReLU
4		dense	32	ReLU
5	hidden		50	ReLU
6	decoder	dense	32	ReLU
7		dense	64	ReLU
8		dense	128	ReLU
9		output	458	–

Table A5. Architecture of Vanilla AE No. 4.

No.	Module	Layer Type	Output Dimension	Activation Function
1	encoder	input	458	–
2		dense	128	ReLU
3		dense	64	ReLU
4		dense	32	ReLU
5	hidden		90	ReLU
6	decoder	dense	32	ReLU
7		dense	64	ReLU
8		dense	128	ReLU
9		output	458	–

Appendix B.2. Denoising Autoencoders

Table A6. Architecture of DAE No. 1.

No.	Module	Layer Type	Output Dimension	Activation Function
1	encoder	input	458	–
2		Gaussian noise	458	–
3		dense	64	ReLU
4		dense	32	ReLU
5	hidden		50	ReLU
6	decoder	dense	32	ReLU
7		dense	64	ReLU
8		output	458	–

Table A7. Architecture of DAE No. 2.

No.	Module	Layer Type	Output Dimension	Activation Function
1	encoder	input	458	–
2		Gaussian noise	458	–
3		dense	64	ReLU
4		dense	32	ReLU
5	decoder	hidden	90	ReLU
6		dense	32	ReLU
7		dense	64	ReLU
8		output	458	–

Table A8. Architecture of DAE No. 3.

No.	Module	Layer Type	Output Dimension	Activation Function
1	encoder	input	458	–
2		Gaussian noise	458	–
3		dense	128	ReLU
4		dropout	–	–
5	decoder	hidden	50	ReLU
6		dense	64	ReLU
7		dense	128	ReLU
8		output	458	–

Table A9. Architecture of DAE No. 4.

No.	Module	Layer Type	Output Dimension	Activation Function
1	encoder	input	458	–
2		Gaussian noise	458	–
3		dense	128	ReLU
4		dropout	–	–
5	decoder	hidden	90	ReLU
6		dense	64	ReLU
7		dense	128	ReLU
8		output	458	–

Appendix B.3. Variational Autoencoders

Table A10. Architecture of VAE No. 1.

No.	Module	Layer Type	Output Dimension	Activation Function
1	encoder	input	458	–
2		dense	64	ReLU
3	decoder	latent	10	ReLU
4		dense	64	ReLU
5		output	458	–

Table A11. Architecture of VAE No. 2.

No.	Module	Layer Type	Output Dimension	Activation Function
1	encoder	input	458	–
2		dense	64	ReLU
3		latent	5	ReLU
4	decoder	dense	64	ReLU
5		output	458	–

Table A12. Architecture of VAE No. 3.

No.	Module	Layer Type	Output Dimension	Activation Function
1	encoder	input	458	–
2		dense	64	ReLU
3		latent	40	ReLU
4	decoder	dense	64	ReLU
5		output	458	–

*Appendix B.4. Convolutional Autoencoders***Table A13.** Architecture of CAE No. 1.

No.	Module	Layer Type	Output Dimension	No. of Filters	Kernel Size	Activation Function	Padding
1	encoder	input	458	–	–	–	–
2		convolutional	458	32	3	ReLU	same
3		pooling	229	–	2	–	same
4		convolutional	229	16	3	ReLU	same
5		pooling	115	–	2	–	same
6		convolutional	115	16	3	ReLU	same
7	decoder	upsampling	230	–	2	–	–
8		convolutional	230	32	3	ReLU	same
9		upsampling	460	–	2	–	–
10		convolutional	460	1	2	–	same
11		output	458	–	–	–	–

Table A14. Architecture of CAE No. 2.

No.	Module	Layer Type	Output Dimension	No. of Filters	Kernel Size	Activation Function	Padding
1	encoder	input	458	–	–	–	–
2		convolutional	458	64	5	ReLU	same
3		normalization	458	–	–	–	–
4		pooling	229	–	2	–	same
5		convolutional	458	32	5	ReLU	same
6		normalization	458	–	–	–	–
7		pooling	229	–	2	–	same
8		convolutional	458	16	3	ReLU	same
9		dropout	458	–	–	–	–
10		pooling	229	–	2	–	same
11		convolutional	229	16	3	ReLU	same
12	decoder	upsampling	458	–	2	–	–
13		convolutional	458	32	5	ReLU	same
14		upsampling	916	–	2	–	–
15		convolutional	916	64	5	ReLU	same
16		upsampling	1832	–	2	–	–
17		convolutional	1832	1	3	–	same
18		cropping	458	–	–	–	–

Appendix C

Table A15. Comparison of the mean ARI and NMI results with their standard deviations for different clustering methods and numbers of clusters, with respect to the target variables: *Primary Type* (PT), *Description* (D), *Location Description* (LD) for 10 random samples (sorted by ARI values in decreasing order)—all results.

Method	No. Groups (k)	Target	Mean ARI $\pm$ sd	Mean NMI $\pm$ sd
GMM + Vanilla No. 4	5	PT	0.5299 $\pm$ 0.07	0.6950 $\pm$ 0.05
k-means + Vanilla No. 4	5	PT	0.5061 $\pm$ 0.08	0.6714 $\pm$ 0.06
k-means + Vanilla No. 3	5	PT	0.4664 $\pm$ 0.12	0.6391 $\pm$ 0.08
GMM + Vanilla No. 4	4	PT	0.4578 $\pm$ 0.05	0.6364 $\pm$ 0.04
GMM + Vanilla No. 3	5	PT	0.4533 $\pm$ 0.11	0.6080 $\pm$ 0.09
k-means + Vanilla No. 4	4	PT	0.4361 $\pm$ 0.06	0.6092 $\pm$ 0.04
k-means + Vanilla No. 3	4	PT	0.3965 $\pm$ 0.08	0.5668 $\pm$ 0.08
GMM + DAE No. 1	5	PT	0.3858 $\pm$ 0.10	0.5647 $\pm$ 0.10
GMM + Vanilla No. 3	4	PT	0.3843 $\pm$ 0.08	0.5424 $\pm$ 0.09
k-means + DAE No. 1	5	PT	0.3767 $\pm$ 0.09	0.5493 $\pm$ 0.07
k-means + Vanilla No. 4	3	PT	0.3573 $\pm$ 0.03	0.5567 $\pm$ 0.02
GMM + DAE No. 1	4	PT	0.3563 $\pm$ 0.07	0.5408 $\pm$ 0.09
GMM + Vanilla No. 4	3	PT	0.3552 $\pm$ 0.03	0.5531 $\pm$ 0.02
k-means + DAE No. 1	4	PT	0.3468 $\pm$ 0.07	0.5236 $\pm$ 0.09
SC + Vanilla No. 3	5	PT	0.3303 $\pm$ 0.09	0.5699 $\pm$ 0.06
k-means + Vanilla No. 3	3	PT	0.3290 $\pm$ 0.05	0.5016 $\pm$ 0.07
GMM + Vanilla No. 3	3	PT	0.3001 $\pm$ 0.04	0.4611 $\pm$ 0.06
GMM + DAE No. 2	5	PT	0.2949 $\pm$ 0.07	0.4850 $\pm$ 0.08
SC + Vanilla No. 4	5	PT	0.2883 $\pm$ 0.10	0.5871 $\pm$ 0.09
k-means + DAE No. 2	5	PT	0.2835 $\pm$ 0.07	0.4562 $\pm$ 0.09
k-means + Vanilla No. 4	5	D	0.2693 $\pm$ 0.03	0.5572 $\pm$ 0.02
GMM + Vanilla No. 4	5	D	0.2632 $\pm$ 0.03	0.5564 $\pm$ 0.02
k-means + DAE No. 1	3	PT	0.2542 $\pm$ 0.07	0.4296 $\pm$ 0.09
SC + Vanilla No. 3	4	PT	0.2540 $\pm$ 0.03	0.4996 $\pm$ 0.02
GMM + DAE No. 1	3	PT	0.2536 $\pm$ 0.08	0.4359 $\pm$ 0.11
k-means + Vanilla No. 1	5	PT	0.2494 $\pm$ 0.06	0.3993 $\pm$ 0.04
GMM + DAE No. 2	4	PT	0.2482 $\pm$ 0.08	0.4596 $\pm$ 0.13
k-means + DAE No. 2	4	PT	0.2397 $\pm$ 0.09	0.4389 $\pm$ 0.14
GMM + Vanilla No. 4	2	PT	0.2286 $\pm$ 0.01	0.4167 $\pm$ 0.01
k-means + Vanilla No. 4	2	PT	0.2261 $\pm$ 0.01	0.4109 $\pm$ 0.01
GMM + Vanilla No. 3	5	D	0.2259 $\pm$ 0.04	0.4981 $\pm$ 0.06
k-means + Vanilla No. 3	5	D	0.2259 $\pm$ 0.04	0.5050 $\pm$ 0.05
k-means + Vanilla No. 4	4	D	0.2126 $\pm$ 0.02	0.5000 $\pm$ 0.01
GMM + Vanilla No. 4	4	D	0.2106 $\pm$ 0.02	0.5004 $\pm$ 0.01
k-means + Vanilla No. 3	2	PT	0.2062 $\pm$ 0.07	0.3601 $\pm$ 0.13
GMM + Vanilla No. 1	5	PT	0.2023 $\pm$ 0.06	0.3476 $\pm$ 0.06
GMM + DAE No. 2	5	D	0.2008 $\pm$ 0.04	0.4662 $\pm$ 0.05
k-means + DAE No. 2	5	D	0.2004 $\pm$ 0.04	0.4465 $\pm$ 0.05
GMM + Vanilla No. 3	2	PT	0.2003 $\pm$ 0.07	0.3584 $\pm$ 0.13
GMM + DAE No. 1	5	D	0.2002 $\pm$ 0.04	0.4686 $\pm$ 0.06
GMM + DAE No. 4	5	PT	0.1968 $\pm$ 0.11	0.3209 $\pm$ 0.13
k-means + Vanilla No. 1	4	PT	0.1913 $\pm$ 0.04	0.3334 $\pm$ 0.04
k-means + DAE No. 1	5	D	0.1869 $\pm$ 0.04	0.4547 $\pm$ 0.05
GMM + Vanilla No. 3	4	D	0.1841 $\pm$ 0.03	0.4486 $\pm$ 0.06
k-means + Vanilla No. 3	4	D	0.1839 $\pm$ 0.03	0.4489 $\pm$ 0.06
k-means + Vanilla No. 2	5	PT	0.1802 $\pm$ 0.06	0.3374 $\pm$ 0.07
GMM + DAE No. 2	3	PT	0.1769 $\pm$ 0.08	0.3474 $\pm$ 0.12
k-means + DAE No. 4	5	PT	0.1752 $\pm$ 0.11	0.2790 $\pm$ 0.13
k-means + DAE No. 1	2	PT	0.1748 $\pm$ 0.06	0.3366 $\pm$ 0.07
GMM + DAE No. 1	2	PT	0.1738 $\pm$ 0.06	0.3370 $\pm$ 0.07
SC + Vanilla No. 4	4	PT	0.1716 $\pm$ 0.09	0.4670 $\pm$ 0.10

Table A15. Cont.

Method	No. Groups (k)	Target	Mean ARI $\pm$ sd	Mean NMI $\pm$ sd
GMM + DAE No. 1	4	D	0.1700 $\pm$ 0.03	0.4363 $\pm$ 0.06
k-means + DAE No. 1	4	D	0.1679 $\pm$ 0.03	0.4274 $\pm$ 0.05
GMM + Vanilla No. 1	4	PT	0.1644 $\pm$ 0.05	0.2984 $\pm$ 0.07
GMM + CAE No. 2	5	PT	0.1629 $\pm$ 0.05	0.3824 $\pm$ 0.04
GMM + CAE No. 1	5	PT	0.1616 $\pm$ 0.09	0.2386 $\pm$ 0.11
k-means + DAE No. 2	3	PT	0.1616 $\pm$ 0.08	0.3274 $\pm$ 0.13
k-means + Vanilla No. 1	5	D	0.1608 $\pm$ 0.05	0.3842 $\pm$ 0.04
k-means + CAE No. 2	5	PT	0.1588 $\pm$ 0.06	0.3801 $\pm$ 0.03
SC + Vanilla No. 4	5	D	0.1588 $\pm$ 0.06	0.4702 $\pm$ 0.09
GMM + DAE No. 2	4	D	0.1538 $\pm$ 0.05	0.4198 $\pm$ 0.09
k-means + DAE No. 2	4	D	0.1523 $\pm$ 0.05	0.4042 $\pm$ 0.10
GMM + DAE No. 4	4	PT	0.1510 $\pm$ 0.12	0.2700 $\pm$ 0.15
k-means + Vanilla No. 3	3	D	0.1500 $\pm$ 0.02	0.4043 $\pm$ 0.05
GMM + Vanilla No. 2	5	PT	0.1490 $\pm$ 0.05	0.3077 $\pm$ 0.06
GMM + Vanilla No. 4	3	D	0.1472 $\pm$ 0.01	0.4186 $\pm$ 0.01
GMM + Vanilla No. 1	5	D	0.1470 $\pm$ 0.05	0.3565 $\pm$ 0.05
k-means + Vanilla No. 2	4	PT	0.1461 $\pm$ 0.06	0.2890 $\pm$ 0.06
k-means + Vanilla No. 4	3	D	0.1461 $\pm$ 0.01	0.4155 $\pm$ 0.01
GMM + Vanilla No. 3	3	D	0.1412 $\pm$ 0.01	0.3906 $\pm$ 0.05
SC + Vanilla No. 3	3	PT	0.1404 $\pm$ 0.05	0.3452 $\pm$ 0.07
k-means + Vanilla No. 2	5	D	0.1375 $\pm$ 0.03	0.3827 $\pm$ 0.04
k-means + DAE No. 4	5	LD	0.1367 $\pm$ 0.07	0.2439 $\pm$ 0.08
k-means + DAE No. 4	4	LD	0.1361 $\pm$ 0.07	0.2449 $\pm$ 0.10
SC + Vanilla No. 3	5	D	0.1334 $\pm$ 0.04	0.4108 $\pm$ 0.04
GMM + DAE No. 4	5	LD	0.1319 $\pm$ 0.06	0.2265 $\pm$ 0.07
k-means + DAE No. 4	4	PT	0.1262 $\pm$ 0.11	0.2221 $\pm$ 0.13
GMM + DAE No. 4	4	LD	0.1261 $\pm$ 0.06	0.2230 $\pm$ 0.10
GMM + DAE No. 1	3	D	0.1224 $\pm$ 0.04	0.3625 $\pm$ 0.08
GMM + Vanilla No. 2	5	D	0.1216 $\pm$ 0.03	0.3504 $\pm$ 0.04
GMM + CAE No. 1	4	PT	0.1204 $\pm$ 0.09	0.2000 $\pm$ 0.12
k-means + CAE No. 1	5	PT	0.1200 $\pm$ 0.08	0.2129 $\pm$ 0.11
k-means + DAE No. 1	3	D	0.1193 $\pm$ 0.04	0.3537 $\pm$ 0.07
GMM + Vanilla No. 2	4	PT	0.1186 $\pm$ 0.07	0.2739 $\pm$ 0.08
k-means + Vanilla No. 1	4	D	0.1183 $\pm$ 0.03	0.3230 $\pm$ 0.03
GMM + CAE No. 2	4	PT	0.1146 $\pm$ 0.02	0.3085 $\pm$ 0.02
k-means + DAE No. 4	3	LD	0.1133 $\pm$ 0.07	0.2020 $\pm$ 0.13
GMM + DAE No. 4	5	D	0.1132 $\pm$ 0.05	0.3079 $\pm$ 0.10
k-means + CAE No. 2	4	PT	0.1125 $\pm$ 0.01	0.3139 $\pm$ 0.02
GMM + CAE No. 2	5	D	0.1092 $\pm$ 0.03	0.3696 $\pm$ 0.03
GMM + DAE No. 2	3	D	0.1080 $\pm$ 0.03	0.3403 $\pm$ 0.08
GMM + DAE No. 4	3	LD	0.1078 $\pm$ 0.07	0.1952 $\pm$ 0.12
k-means + Vanilla No. 2	3	PT	0.1077 $\pm$ 0.01	0.2154 $\pm$ 0.03
k-means + CAE No. 2	5	D	0.1059 $\pm$ 0.03	0.3659 $\pm$ 0.03
GMM + Vanilla No. 1	4	D	0.1049 $\pm$ 0.03	0.2984 $\pm$ 0.05
GMM + DAE No. 2	2	PT	0.1033 $\pm$ 0.06	0.2362 $\pm$ 0.11
k-means + Vanilla No. 2	4	D	0.1017 $\pm$ 0.03	0.3278 $\pm$ 0.04
k-means + Vanilla No. 1	3	PT	0.1015 $\pm$ 0.01	0.2297 $\pm$ 0.01
GMM + CAE No. 2	3	PT	0.1012 $\pm$ 0.03	0.2620 $\pm$ 0.03
SC + Vanilla No. 3	4	D	0.1001 $\pm$ 0.01	0.3526 $\pm$ 0.02
k-means + DAE No. 2	3	D	0.0995 $\pm$ 0.03	0.3147 $\pm$ 0.09
k-means + DAE No. 4	5	D	0.0959 $\pm$ 0.05	0.2684 $\pm$ 0.09
k-means + DAE No. 3	5	LD	0.0951 $\pm$ 0.04	0.1858 $\pm$ 0.05
k-means + CAE No. 2	3	PT	0.0944 $\pm$ 0.04	0.2611 $\pm$ 0.04
GMM + CAE No. 1	5	D	0.0934 $\pm$ 0.04	0.2236 $\pm$ 0.09
k-means + DAE No. 3	5	PT	0.0932 $\pm$ 0.04	0.1400 $\pm$ 0.06
SC + Vanilla No. 4	4	D	0.0921 $\pm$ 0.05	0.3671 $\pm$ 0.09
k-means + CAE No. 1	4	PT	0.0917 $\pm$ 0.08	0.1758 $\pm$ 0.12
k-means + DAE No. 2	5	LD	0.0908 $\pm$ 0.07	0.1864 $\pm$ 0.09

Table A15. Cont.

Method	No. Groups (k)	Target	Mean ARI $\pm$ sd	Mean NMI $\pm$ sd
<i>k</i> -means + DAE No. 4	2	LD	0.0902 $\pm$ 0.09	0.1701 $\pm$ 0.16
<i>k</i> -means + DAE No. 2	2	PT	0.0887 $\pm$ 0.06	0.2136 $\pm$ 0.12
GMM + DAE No. 4	2	LD	0.0884 $\pm$ 0.08	0.1727 $\pm$ 0.16
GMM + Vanilla No. 2	4	D	0.0873 $\pm$ 0.03	0.3153 $\pm$ 0.06
GMM + Vanilla No. 4	2	D	0.0865 $\pm$ 0.00	0.2987 $\pm$ 0.00
GMM + DAE No. 4	3	PT	0.0860 $\pm$ 0.09	0.1761 $\pm$ 0.15
<i>k</i> -means + Vanilla No. 4	2	D	0.0860 $\pm$ 0.00	0.2976 $\pm$ 0.00
GMM + DAE No. 3	5	PT	0.0844 $\pm$ 0.07	0.1416 $\pm$ 0.09
GMM + CAE No. 2	2	PT	0.0842 $\pm$ 0.07	0.2230 $\pm$ 0.10
<i>k</i> -means + Vanilla No. 2	2	PT	0.0832 $\pm$ 0.00	0.1318 $\pm$ 0.01
<i>k</i> -means + CAE No. 2	2	PT	0.0827 $\pm$ 0.07	0.2199 $\pm$ 0.10
<i>k</i> -means + DAE No. 3	4	LD	0.0805 $\pm$ 0.06	0.1519 $\pm$ 0.09
<i>k</i> -means + Vanilla No. 3	2	D	0.0796 $\pm$ 0.03	0.2685 $\pm$ 0.09
GMM + DAE No. 4	4	D	0.0782 $\pm$ 0.05	0.2545 $\pm$ 0.10
GMM + Vanilla No. 3	2	D	0.0770 $\pm$ 0.03	0.2610 $\pm$ 0.09
GMM + DAE No. 2	5	LD	0.0765 $\pm$ 0.06	0.1616 $\pm$ 0.08
<i>k</i> -means + DAE No. 4	3	PT	0.0763 $\pm$ 0.08	0.1590 $\pm$ 0.13
GMM + DAE No. 1	2	D	0.0762 $\pm$ 0.03	0.2756 $\pm$ 0.06
<i>k</i> -means + DAE No. 1	2	D	0.0755 $\pm$ 0.03	0.2642 $\pm$ 0.06
SC + Vanilla No. 4	3	PT	0.0755 $\pm$ 0.04	0.3030 $\pm$ 0.12
GMM + CAE No. 2	4	D	0.0754 $\pm$ 0.01	0.3046 $\pm$ 0.02
<i>k</i> -means + DAE No. 3	3	LD	0.0754 $\pm$ 0.06	0.1353 $\pm$ 0.10
<i>k</i> -means + CAE No. 2	4	D	0.0750 $\pm$ 0.01	0.3070 $\pm$ 0.01
GMM + CAE No. 1	3	PT	0.0749 $\pm$ 0.08	0.1453 $\pm$ 0.15
<i>k</i> -means + DAE No. 2	3	LD	0.0730 $\pm$ 0.10	0.1333 $\pm$ 0.14
<i>k</i> -means + DAE No. 2	4	LD	0.0729 $\pm$ 0.10	0.1404 $\pm$ 0.13
<i>k</i> -means + Vanilla No. 2	3	D	0.0729 $\pm$ 0.01	0.2475 $\pm$ 0.04
GMM + CAE No. 2	5	LD	0.0728 $\pm$ 0.05	0.1471 $\pm$ 0.04
<i>k</i> -means + Vanilla No. 1	2	PT	0.0722 $\pm$ 0.00	0.1609 $\pm$ 0.01
<i>k</i> -means + CAE No. 1	5	D	0.0716 $\pm$ 0.03	0.2031 $\pm$ 0.08
<i>k</i> -means + CAE No. 2	5	LD	0.0715 $\pm$ 0.05	0.1448 $\pm$ 0.04
<i>k</i> -means + DAE No. 4	4	D	0.0671 $\pm$ 0.05	0.2148 $\pm$ 0.09
GMM + Vanilla No. 2	2	PT	0.0644 $\pm$ 0.01	0.1547 $\pm$ 0.01
GMM + Vanilla No. 2	3	PT	0.0632 $\pm$ 0.02	0.1778 $\pm$ 0.05
GMM + CAE No. 1	4	D	0.0627 $\pm$ 0.04	0.1790 $\pm$ 0.09
<i>k</i> -means + DAE No. 3	4	PT	0.0623 $\pm$ 0.06	0.0986 $\pm$ 0.09
<i>k</i> -means + CAE No. 1	3	PT	0.0608 $\pm$ 0.08	0.1297 $\pm$ 0.15
GMM + CAE No. 2	3	D	0.0605 $\pm$ 0.01	0.2531 $\pm$ 0.03
GMM + DAE No. 3	4	PT	0.0597 $\pm$ 0.07	0.1090 $\pm$ 0.11
SC + Vanilla No. 1	5	PT	0.0589 $\pm$ 0.00	0.1585 $\pm$ 0.01
<i>k</i> -means + CAE No. 2	3	D	0.0585 $\pm$ 0.02	0.2497 $\pm$ 0.04
SC + Vanilla No. 1	4	PT	0.0581 $\pm$ 0.00	0.1556 $\pm$ 0.01
GMM + Vanilla No. 1	3	PT	0.0577 $\pm$ 0.03	0.1623 $\pm$ 0.08
SC + Vanilla No. 1	3	PT	0.0564 $\pm$ 0.00	0.1451 $\pm$ 0.01
<i>k</i> -means + DAE No. 1	4	LD	0.0559 $\pm$ 0.03	0.0705 $\pm$ 0.03
<i>k</i> -means + DAE No. 3	5	D	0.0557 $\pm$ 0.02	0.1695 $\pm$ 0.06
SC + Vanilla No. 3	3	D	0.0541 $\pm$ 0.02	0.2367 $\pm$ 0.05
GMM + DAE No. 2	4	LD	0.0537 $\pm$ 0.06	0.1182 $\pm$ 0.10
<i>k</i> -means + CAE No. 2	4	LD	0.0530 $\pm$ 0.02	0.1238 $\pm$ 0.03
<i>k</i> -means + Vanilla No. 1	3	D	0.0529 $\pm$ 0.00	0.2076 $\pm$ 0.00
GMM + Vanilla No. 2	3	D	0.0528 $\pm$ 0.02	0.2260 $\pm$ 0.06
GMM + CAE No. 2	4	LD	0.0523 $\pm$ 0.02	0.1256 $\pm$ 0.03
GMM + CAE No. 2	3	LD	0.0514 $\pm$ 0.02	0.1091 $\pm$ 0.03
GMM + DAE No. 3	5	D	0.0513 $\pm$ 0.04	0.1688 $\pm$ 0.07
<i>k</i> -means + Vanilla No. 2	2	D	0.0500 $\pm$ 0.00	0.1525 $\pm$ 0.00
<i>k</i> -means + CAE No. 2	3	LD	0.0494 $\pm$ 0.03	0.1030 $\pm$ 0.03
GMM + DAE No. 2	2	D	0.0494 $\pm$ 0.02	0.2108 $\pm$ 0.08
SC + Vanilla No. 1	2	PT	0.0494 $\pm$ 0.02	0.1257 $\pm$ 0.04
GMM + DAE No. 2	3	LD	0.0493 $\pm$ 0.06	0.1071 $\pm$ 0.10

Table A15. Cont.

Method	No. Groups (k)	Target	Mean ARI $\pm$ sd	Mean NMI $\pm$ sd
<i>k</i> -means + CAE No. 1	4	D	0.0470 $\pm$ 0.03	0.1594 $\pm$ 0.09
GMM + Vanilla No. 2	4	LD	0.0470 $\pm$ 0.01	0.1096 $\pm$ 0.02
GMM + Vanilla No. 3	4	LD	0.0466 $\pm$ 0.01	0.0653 $\pm$ 0.01
<i>k</i> -means + DAE No. 1	5	LD	0.0464 $\pm$ 0.02	0.0700 $\pm$ 0.01
<i>k</i> -means + Vanilla No. 3	4	LD	0.0458 $\pm$ 0.01	0.0663 $\pm$ 0.01
<i>k</i> -means + DAE No. 2	2	D	0.0455 $\pm$ 0.02	0.1937 $\pm$ 0.08
GMM + Vanilla No. 3	5	LD	0.0450 $\pm$ 0.02	0.0725 $\pm$ 0.01
SC + Vanilla No. 4	3	D	0.0446 $\pm$ 0.02	0.2433 $\pm$ 0.11
<i>k</i> -means + Vanilla No. 2	5	LD	0.0444 $\pm$ 0.01	0.0737 $\pm$ 0.01
<i>k</i> -means + Vanilla No. 3	5	LD	0.0444 $\pm$ 0.02	0.0729 $\pm$ 0.01
<i>k</i> -means + DAE No. 1	2	LD	0.0437 $\pm$ 0.02	0.0492 $\pm$ 0.02
GMM + DAE No. 1	4	LD	0.0436 $\pm$ 0.01	0.0618 $\pm$ 0.01
GMM + DAE No. 1	5	LD	0.0435 $\pm$ 0.01	0.0683 $\pm$ 0.01
GMM + Vanilla No. 1	2	PT	0.0431 $\pm$ 0.01	0.1021 $\pm$ 0.01
GMM + Vanilla No. 2	5	LD	0.0423 $\pm$ 0.01	0.1022 $\pm$ 0.01
GMM + DAE No. 4	3	D	0.0422 $\pm$ 0.04	0.1682 $\pm$ 0.10
GMM + CAE No. 1	2	PT	0.0418 $\pm$ 0.05	0.0977 $\pm$ 0.12
<i>k</i> -means + Vanilla No. 2	4	LD	0.0414 $\pm$ 0.01	0.0685 $\pm$ 0.01
<i>k</i> -means + DAE No. 1	3	LD	0.0408 $\pm$ 0.02	0.0503 $\pm$ 0.01
GMM + DAE No. 1	2	LD	0.0405 $\pm$ 0.02	0.0471 $\pm$ 0.02
GMM + Vanilla No. 2	3	LD	0.0405 $\pm$ 0.02	0.0968 $\pm$ 0.02
GMM + DAE No. 4	2	PT	0.0398 $\pm$ 0.05	0.0953 $\pm$ 0.09
<i>k</i> -means + DAE No. 4	3	D	0.0396 $\pm$ 0.03	0.1548 $\pm$ 0.09
GMM + Vanilla No. 3	3	LD	0.0391 $\pm$ 0.01	0.0533 $\pm$ 0.01
<i>k</i> -means + DAE No. 3	4	D	0.0389 $\pm$ 0.03	0.1256 $\pm$ 0.09
<i>k</i> -means + Vanilla No. 3	3	LD	0.0389 $\pm$ 0.01	0.0568 $\pm$ 0.01
GMM + Vanilla No. 2	2	D	0.0387 $\pm$ 0.00	0.1771 $\pm$ 0.01
<i>k</i> -means + DAE No. 2	2	LD	0.0379 $\pm$ 0.06	0.0881 $\pm$ 0.11
GMM + CAE No. 2	2	LD	0.0378 $\pm$ 0.03	0.0672 $\pm$ 0.05
GMM + DAE No. 1	3	LD	0.0378 $\pm$ 0.01	0.0495 $\pm$ 0.01
<i>k</i> -means + DAE No. 3	2	LD	0.0371 $\pm$ 0.06	0.0698 $\pm$ 0.11
<i>k</i> -means + CAE No. 1	2	PT	0.0365 $\pm$ 0.04	0.0968 $\pm$ 0.11
GMM + Vanilla No. 1	3	LD	0.0353 $\pm$ 0.01	0.0859 $\pm$ 0.03
GMM + DAE No. 2	2	LD	0.0352 $\pm$ 0.06	0.0800 $\pm$ 0.11
<i>k</i> -means + CAE No. 2	2	LD	0.0339 $\pm$ 0.02	0.0632 $\pm$ 0.04
<i>k</i> -means + Vanilla No. 2	3	LD	0.0338 $\pm$ 0.01	0.0484 $\pm$ 0.01
GMM + CAE No. 1	3	D	0.0328 $\pm$ 0.03	0.1287 $\pm$ 0.11
GMM + DAE No. 3	3	LD	0.0328 $\pm$ 0.04	0.0815 $\pm$ 0.08
<i>k</i> -means + CAE No. 2	2	D	0.0327 $\pm$ 0.02	0.1699 $\pm$ 0.05
GMM + DAE No. 3	4	D	0.0321 $\pm$ 0.03	0.1278 $\pm$ 0.10
GMM + CAE No. 2	2	D	0.0319 $\pm$ 0.02	0.1699 $\pm$ 0.06
GMM + DAE No. 3	5	LD	0.0309 $\pm$ 0.03	0.0655 $\pm$ 0.04
<i>k</i> -means + Vanilla No. 1	2	D	0.0307 $\pm$ 0.00	0.1336 $\pm$ 0.00
SC + Vanilla No. 3	5	LD	0.0299 $\pm$ 0.01	0.0595 $\pm$ 0.01
GMM + Vanilla No. 1	3	D	0.0295 $\pm$ 0.01	0.1541 $\pm$ 0.05
<i>k</i> -means + Vanilla No. 4	5	LD	0.0291 $\pm$ 0.01	0.0671 $\pm$ 0.01
GMM + DAE No. 3	4	LD	0.0289 $\pm$ 0.03	0.0734 $\pm$ 0.06
GMM + Vanilla No. 4	5	LD	0.0277 $\pm$ 0.01	0.0663 $\pm$ 0.01
<i>k</i> -means + CAE No. 1	3	D	0.0270 $\pm$ 0.03	0.1188 $\pm$ 0.11
<i>k</i> -means + Vanilla No. 4	4	LD	0.0268 $\pm$ 0.01	0.0583 $\pm$ 0.01
SC + Vanilla No. 3	4	LD	0.0267 $\pm$ 0.01	0.0508 $\pm$ 0.00
GMM + DAE No. 4	2	D	0.0266 $\pm$ 0.02	0.1152 $\pm$ 0.08
GMM + DAE No. 3	2	LD	0.0257 $\pm$ 0.06	0.0538 $\pm$ 0.10
<i>k</i> -means + DAE No. 3	3	PT	0.0257 $\pm$ 0.04	0.0447 $\pm$ 0.05
<i>k</i> -means + Vanilla No. 1	3	LD	0.0254 $\pm$ 0.00	0.0348 $\pm$ 0.00
GMM + Vanilla No. 4	4	LD	0.0254 $\pm$ 0.01	0.0569 $\pm$ 0.01
SC + DAE No. 1	5	PT	0.0245 $\pm$ 0.01	0.1188 $\pm$ 0.06
<i>k</i> -means + DAE No. 4	2	PT	0.0240 $\pm$ 0.02	0.0655 $\pm$ 0.04
SC + CAE No. 2	5	D	0.0239 $\pm$ 0.02	0.1825 $\pm$ 0.09
SC + Vanilla No. 4	2	PT	0.0236 $\pm$ 0.02	0.1445 $\pm$ 0.10

Table A15. Cont.

Method	No. Groups (k)	Target	Mean ARI $\pm$ sd	Mean NMI $\pm$ sd
GMM + CAE No. 1	2	D	0.0232 $\pm$ 0.03	0.1008 $\pm$ 0.11
SC + Vanilla No. 4	2	D	0.0228 $\pm$ 0.02	0.1358 $\pm$ 0.10
GMM + CAE No. 1	5	LD	0.0226 $\pm$ 0.02	0.0586 $\pm$ 0.03
k-means + CAE No. 1	2	D	0.0224 $\pm$ 0.03	0.1020 $\pm$ 0.11
k-means + Vanilla No. 1	5	LD	0.0222 $\pm$ 0.00	0.0520 $\pm$ 0.00
k-means + Vanilla No. 2	2	LD	0.0221 $\pm$ 0.00	0.0238 $\pm$ 0.00
k-means + Vanilla No. 1	4	LD	0.0215 $\pm$ 0.01	0.0449 $\pm$ 0.00
SC + Vanilla No. 1	5	D	0.0208 $\pm$ 0.00	0.1277 $\pm$ 0.01
SC + Vanilla No. 1	4	D	0.0205 $\pm$ 0.00	0.1232 $\pm$ 0.01
SC + Vanilla No. 1	3	D	0.0200 $\pm$ 0.00	0.1128 $\pm$ 0.01
SC + DAE No. 2	5	PT	0.0194 $\pm$ 0.02	0.1348 $\pm$ 0.10
k-means + Vanilla No. 3	2	LD	0.0182 $\pm$ 0.01	0.0295 $\pm$ 0.01
GMM + DAE No. 3	3	PT	0.0178 $\pm$ 0.03	0.0358 $\pm$ 0.05
SC + Vanilla No. 2	5	PT	0.0178 $\pm$ 0.02	0.0610 $\pm$ 0.06
k-means + CAE No. 1	5	LD	0.0176 $\pm$ 0.02	0.0585 $\pm$ 0.03
k-means + DAE No. 3	3	D	0.0176 $\pm$ 0.02	0.0661 $\pm$ 0.07
SC + Vanilla No. 1	2	D	0.0176 $\pm$ 0.01	0.0962 $\pm$ 0.03
SC + Vanilla No. 2	5	D	0.0173 $\pm$ 0.02	0.0894 $\pm$ 0.08
GMM + Vanilla No. 4	3	LD	0.0173 $\pm$ 0.00	0.0412 $\pm$ 0.00
GMM + Vanilla No. 1	2	D	0.0171 $\pm$ 0.00	0.0868 $\pm$ 0.01
k-means + Vanilla No. 4	3	LD	0.0168 $\pm$ 0.00	0.0407 $\pm$ 0.00
SC + CAE No. 1	3	LD	0.0166 $\pm$ 0.00	0.0241 $\pm$ 0.00
SC + DAE No. 2	5	D	0.0166 $\pm$ 0.02	0.1223 $\pm$ 0.09
SC + DAE No. 1	4	PT	0.0162 $\pm$ 0.02	0.0664 $\pm$ 0.06
GMM + Vanilla No. 2	2	LD	0.0162 $\pm$ 0.00	0.0461 $\pm$ 0.00
SC + CAE No. 2	5	PT	0.0161 $\pm$ 0.02	0.1785 $\pm$ 0.09
SC + Vanilla No. 3	2	PT	0.0153 $\pm$ 0.04	0.0461 $\pm$ 0.11
k-means + DAE No. 4	2	D	0.0152 $\pm$ 0.00	0.0789 $\pm$ 0.02
SC + Vanilla No. 4	5	LD	0.0151 $\pm$ 0.00	0.0606 $\pm$ 0.01
SC + Vanilla No. 3	3	LD	0.0148 $\pm$ 0.01	0.0367 $\pm$ 0.01
GMM + Vanilla No. 1	4	LD	0.0146 $\pm$ 0.01	0.0620 $\pm$ 0.02
GMM + Vanilla No. 3	2	LD	0.0145 $\pm$ 0.01	0.0271 $\pm$ 0.01
SC + DAE No. 2	3	D	0.0129 $\pm$ 0.01	0.0981 $\pm$ 0.08
GMM + DAE No. 3	3	D	0.0127 $\pm$ 0.02	0.0622 $\pm$ 0.07
GMM + Vanilla No. 1	2	LD	0.0112 $\pm$ 0.01	0.0365 $\pm$ 0.02
SC + DAE No. 1	5	D	0.0107 $\pm$ 0.01	0.0935 $\pm$ 0.05
GMM + Vanilla No. 4	2	LD	0.0106 $\pm$ 0.00	0.0279 $\pm$ 0.00
k-means + Vanilla No. 4	2	LD	0.0105 $\pm$ 0.00	0.0277 $\pm$ 0.00
SC + DAE No. 1	3	PT	0.0099 $\pm$ 0.02	0.0402 $\pm$ 0.06
SC + DAE No. 2	4	D	0.0095 $\pm$ 0.01	0.0728 $\pm$ 0.08
GMM + CAE No. 1	4	LD	0.0093 $\pm$ 0.01	0.0347 $\pm$ 0.01
SC + CAE No. 2	4	D	0.0093 $\pm$ 0.01	0.0962 $\pm$ 0.08
SC + DAE No. 4	5	PT	0.0091 $\pm$ 0.01	0.0649 $\pm$ 0.04
GMM + Vanilla No. 1	5	LD	0.0091 $\pm$ 0.01	0.0514 $\pm$ 0.02
SC + DAE No. 2	3	PT	0.0088 $\pm$ 0.01	0.1062 $\pm$ 0.08
k-means + CAE No. 1	4	LD	0.0083 $\pm$ 0.01	0.0360 $\pm$ 0.01
SC + Vanilla No. 4	4	LD	0.0083 $\pm$ 0.01	0.0426 $\pm$ 0.01
GMM + DAE No. 3	2	PT	0.0080 $\pm$ 0.01	0.0150 $\pm$ 0.02
k-means + Vanilla No. 1	2	LD	0.0073 $\pm$ 0.00	0.0119 $\pm$ 0.00
SC + CAE No. 1	4	LD	0.0072 $\pm$ 0.00	0.0269 $\pm$ 0.00
SC + DAE No. 2	5	LD	0.0069 $\pm$ 0.01	0.0491 $\pm$ 0.03
SC + CAE No. 1	5	LD	0.0066 $\pm$ 0.00	0.0315 $\pm$ 0.00
SC + CAE No. 1	2	PT	0.0064 $\pm$ 0.00	0.0079 $\pm$ 0.00
SC + CAE No. 2	4	PT	0.0061 $\pm$ 0.02	0.0920 $\pm$ 0.08
SC + CAE No. 2	5	LD	0.0061 $\pm$ 0.01	0.0605 $\pm$ 0.03
SC + DAE No. 1	5	LD	0.0061 $\pm$ 0.00	0.0332 $\pm$ 0.01
SC + Vanilla No. 2	5	LD	0.0058 $\pm$ 0.00	0.0182 $\pm$ 0.00
SC + Vanilla No. 3	2	D	0.0057 $\pm$ 0.02	0.0309 $\pm$ 0.07
SC + DAE No. 1	4	D	0.0056 $\pm$ 0.01	0.0488 $\pm$ 0.04
k-means + DAE No. 3	2	PT	0.0056 $\pm$ 0.01	0.0132 $\pm$ 0.02

Table A15. Cont.

Method	No. Groups (k)	Target	Mean ARI $\pm$ sd	Mean NMI $\pm$ sd
GMM + CAE No. 1	2	LD	0.0055 $\pm$ 0.01	0.0186 $\pm$ 0.01
SC + DAE No. 3	5	PT	0.0054 $\pm$ 0.00	0.0137 $\pm$ 0.01
GMM + CAE No. 1	3	LD	0.0053 $\pm$ 0.01	0.0236 $\pm$ 0.01
SC + CAE No. 1	5	PT	0.0051 $\pm$ 0.00	0.0159 $\pm$ 0.00
GMM + DAE No. 3	2	D	0.0050 $\pm$ 0.00	0.0281 $\pm$ 0.02
SC + DAE No. 3	3	PT	0.0048 $\pm$ 0.00	0.0098 $\pm$ 0.01
SC + DAE No. 4	4	PT	0.0048 $\pm$ 0.01	0.0178 $\pm$ 0.02
k-means + DAE No. 3	2	D	0.0043 $\pm$ 0.01	0.0218 $\pm$ 0.03
k-means + CAE No. 1	3	LD	0.0042 $\pm$ 0.01	0.0231 $\pm$ 0.01
SC + CAE No. 1	4	PT	0.0041 $\pm$ 0.00	0.0153 $\pm$ 0.00
SC + DAE No. 3	4	PT	0.0041 $\pm$ 0.00	0.0105 $\pm$ 0.01
SC + DAE No. 2	4	PT	0.0040 $\pm$ 0.01	0.0771 $\pm$ 0.09
SC + Vanilla No. 2	4	D	0.0040 $\pm$ 0.01	0.0299 $\pm$ 0.04
SC + CAE No. 1	2	LD	0.0038 $\pm$ 0.00	0.0144 $\pm$ 0.00
SC + DAE No. 1	3	D	0.0034 $\pm$ 0.01	0.0300 $\pm$ 0.04
SC + DAE No. 4	5	D	0.0032 $\pm$ 0.00	0.0572 $\pm$ 0.02
SC + DAE No. 1	4	LD	0.0030 $\pm$ 0.00	0.0161 $\pm$ 0.01
SC + DAE No. 3	5	D	0.0026 $\pm$ 0.00	0.0302 $\pm$ 0.01
SC + Vanilla No. 2	4	LD	0.0026 $\pm$ 0.00	0.0123 $\pm$ 0.00
SC + CAE No. 1	3	D	0.0023 $\pm$ 0.00	0.0274 $\pm$ 0.00
SC + Vanilla No. 2	3	LD	0.0023 $\pm$ 0.00	0.0093 $\pm$ 0.00
SC + DAE No. 3	3	D	0.0022 $\pm$ 0.00	0.0215 $\pm$ 0.01
SC + Vanilla No. 4	3	LD	0.0022 $\pm$ 0.00	0.0259 $\pm$ 0.01
SC + DAE No. 2	2	D	0.0021 $\pm$ 0.01	0.0198 $\pm$ 0.05
SC + DAE No. 3	4	D	0.0020 $\pm$ 0.00	0.0225 $\pm$ 0.01
SC + CAE No. 1	2	D	0.0018 $\pm$ 0.00	0.0151 $\pm$ 0.00
SC + Vanilla No. 2	4	PT	0.0018 $\pm$ 0.01	0.0180 $\pm$ 0.03
SC + DAE No. 1	3	LD	0.0015 $\pm$ 0.00	0.0104 $\pm$ 0.01
k-means + CAE No. 1	2	LD	0.0014 $\pm$ 0.00	0.0148 $\pm$ 0.01
SC + DAE No. 4	4	D	0.0014 $\pm$ 0.00	0.0200 $\pm$ 0.02
SC + DAE No. 4	3	PT	0.0013 $\pm$ 0.00	0.0067 $\pm$ 0.01
SC + CAE No. 1	4	D	0.0012 $\pm$ 0.00	0.0313 $\pm$ 0.00
SC + CAE No. 1	5	D	0.0011 $\pm$ 0.00	0.0354 $\pm$ 0.00
SC + DAE No. 3	3	LD	0.0011 $\pm$ 0.00	0.0071 $\pm$ 0.00
SC + DAE No. 3	4	LD	0.0010 $\pm$ 0.00	0.0091 $\pm$ 0.00
SC + Vanilla No. 2	2	LD	0.0010 $\pm$ 0.00	0.0041 $\pm$ 0.00
SC + Vanilla No. 3	2	LD	0.0010 $\pm$ 0.00	0.0058 $\pm$ 0.01
SC + DAE No. 3	5	LD	0.0009 $\pm$ 0.00	0.0124 $\pm$ 0.00
SC + DAE No. 2	2	LD	0.0007 $\pm$ 0.00	0.0060 $\pm$ 0.01
SC + DAE No. 3	2	PT	0.0007 $\pm$ 0.00	0.0025 $\pm$ 0.00
SC + CAE No. 1	3	PT	0.0004 $\pm$ 0.00	0.0155 $\pm$ 0.00
SC + DAE No. 1	2	LD	0.0004 $\pm$ 0.00	0.0026 $\pm$ 0.00
SC + CAE No. 2	2	LD	0.0002 $\pm$ 0.00	0.0025 $\pm$ 0.00
SC + DAE No. 1	2	PT	0.0002 $\pm$ 0.00	0.0031 $\pm$ 0.00
SC + DAE No. 1	4	LD	0.0030 $\pm$ 0.00	0.0161 $\pm$ 0.01
SC + DAE No. 4	3	D	0.0002 $\pm$ 0.00	0.0091 $\pm$ 0.01
SC + VAE No. 1	3	PT	0.0002 $\pm$ 0.00	0.0019 $\pm$ 0.00
SC + VAE No. 2	3	LD	0.0002 $\pm$ 0.00	0.0072 $\pm$ 0.00
SC + CAE No. 2	2	D	0.0001 $\pm$ 0.00	0.0031 $\pm$ 0.00
SC + DAE No. 1	2	D	0.0001 $\pm$ 0.00	0.0037 $\pm$ 0.00
SC + DAE No. 2	3	LD	0.0001 $\pm$ 0.00	0.0179 $\pm$ 0.01
SC + DAE No. 2	4	LD	0.0001 $\pm$ 0.00	0.0150 $\pm$ 0.01
GMM + VAE No. 1	3	PT	0.0001 $\pm$ 0.00	0.0019 $\pm$ 0.00
GMM + VAE No. 1	4	PT	0.0001 $\pm$ 0.00	0.0025 $\pm$ 0.00
SC + VAE No. 1	2	PT	0.0001 $\pm$ 0.00	0.0011 $\pm$ 0.00
SC + VAE No. 1	3	D	0.0001 $\pm$ 0.00	0.0128 $\pm$ 0.00
SC + VAE No. 1	3	LD	0.0001 $\pm$ 0.00	0.0070 $\pm$ 0.00
SC + VAE No. 1	4	D	0.0001 $\pm$ 0.00	0.0174 $\pm$ 0.00
SC + VAE No. 1	4	PT	0.0001 $\pm$ 0.00	0.0026 $\pm$ 0.00
SC + VAE No. 2	4	LD	0.0001 $\pm$ 0.00	0.0100 $\pm$ 0.00

Table A15. Cont.

Method	No. Groups (k)	Target	Mean ARI $\pm$ sd	Mean NMI $\pm$ sd
<i>k</i> -means + VAE No. 3	5	PT	0.0001 $\pm$ 0.00	0.0031 $\pm$ 0.00
GMM + VAE No. 1	2	D	0.0000 $\pm$ 0.00	0.0073 $\pm$ 0.00
GMM + VAE No. 1	2	LD	0.0000 $\pm$ 0.00	0.0040 $\pm$ 0.00
GMM + VAE No. 1	2	PT	0.0000 $\pm$ 0.00	0.0010 $\pm$ 0.00
GMM + VAE No. 1	3	D	0.0000 $\pm$ 0.00	0.0127 $\pm$ 0.00
GMM + VAE No. 1	3	LD	0.0000 $\pm$ 0.00	0.0068 $\pm$ 0.00
GMM + VAE No. 1	4	D	0.0000 $\pm$ 0.00	0.0170 $\pm$ 0.00
GMM + VAE No. 1	4	LD	0.0000 $\pm$ 0.00	0.0090 $\pm$ 0.00
GMM + VAE No. 1	5	D	0.0000 $\pm$ 0.00	0.0209 $\pm$ 0.00
GMM + VAE No. 1	5	LD	0.0000 $\pm$ 0.00	0.0115 $\pm$ 0.00
GMM + VAE No. 1	5	PT	0.0000 $\pm$ 0.00	0.0029 $\pm$ 0.00
<i>k</i> -means + VAE No. 1	2	D	0.0000 $\pm$ 0.00	0.0073 $\pm$ 0.00
<i>k</i> -means + VAE No. 1	2	LD	0.0000 $\pm$ 0.00	0.0039 $\pm$ 0.00
<i>k</i> -means + VAE No. 1	2	PT	0.0000 $\pm$ 0.00	0.0010 $\pm$ 0.00
<i>k</i> -means + VAE No. 1	3	D	0.0000 $\pm$ 0.00	0.0126 $\pm$ 0.00
<i>k</i> -means + VAE No. 1	3	LD	0.0000 $\pm$ 0.00	0.0069 $\pm$ 0.00
<i>k</i> -means + VAE No. 1	3	PT	0.0000 $\pm$ 0.00	0.0018 $\pm$ 0.00
<i>k</i> -means + VAE No. 1	4	D	0.0000 $\pm$ 0.00	0.0170 $\pm$ 0.00
<i>k</i> -means + VAE No. 1	4	LD	0.0000 $\pm$ 0.00	0.0091 $\pm$ 0.00
<i>k</i> -means + VAE No. 1	4	PT	0.0000 $\pm$ 0.00	0.0024 $\pm$ 0.00
<i>k</i> -means + VAE No. 1	5	D	0.0000 $\pm$ 0.00	0.0209 $\pm$ 0.00
<i>k</i> -means + VAE No. 1	5	PT	0.0000 $\pm$ 0.00	0.0030 $\pm$ 0.00
SC + VAE No. 1	2	D	0.0000 $\pm$ 0.00	0.0073 $\pm$ 0.00
SC + VAE No. 1	2	LD	0.0000 $\pm$ 0.00	0.0039 $\pm$ 0.00
SC + VAE No. 1	4	LD	0.0000 $\pm$ 0.00	0.0093 $\pm$ 0.00
SC + VAE No. 1	5	D	0.0000 $\pm$ 0.00	0.0210 $\pm$ 0.00
SC + VAE No. 1	5	PT	0.0000 $\pm$ 0.00	0.0031 $\pm$ 0.00
GMM + VAE No. 2	2	D	0.0000 $\pm$ 0.00	0.0076 $\pm$ 0.00
GMM + VAE No. 2	2	LD	0.0000 $\pm$ 0.00	0.0041 $\pm$ 0.00
GMM + VAE No. 2	2	PT	0.0000 $\pm$ 0.00	0.0010 $\pm$ 0.00
GMM + VAE No. 2	3	D	0.0000 $\pm$ 0.00	0.0126 $\pm$ 0.00
GMM + VAE No. 2	3	PT	0.0000 $\pm$ 0.00	0.0017 $\pm$ 0.00
GMM + VAE No. 2	4	D	0.0000 $\pm$ 0.00	0.0172 $\pm$ 0.00
GMM + VAE No. 2	4	LD	0.0000 $\pm$ 0.00	0.0099 $\pm$ 0.00
GMM + VAE No. 2	4	PT	0.0000 $\pm$ 0.00	0.0024 $\pm$ 0.00
GMM + VAE No. 2	5	D	0.0000 $\pm$ 0.00	0.0212 $\pm$ 0.00
GMM + VAE No. 2	5	LD	0.0000 $\pm$ 0.00	0.0118 $\pm$ 0.00
<i>k</i> -means + VAE No. 2	2	D	0.0000 $\pm$ 0.00	0.0076 $\pm$ 0.00
<i>k</i> -means + VAE No. 2	2	LD	0.0000 $\pm$ 0.00	0.0041 $\pm$ 0.00
<i>k</i> -means + VAE No. 2	2	PT	0.0000 $\pm$ 0.00	0.0010 $\pm$ 0.00
<i>k</i> -means + VAE No. 2	3	D	0.0000 $\pm$ 0.00	0.0127 $\pm$ 0.00
<i>k</i> -means + VAE No. 2	3	LD	0.0000 $\pm$ 0.00	0.0069 $\pm$ 0.00
<i>k</i> -means + VAE No. 2	3	PT	0.0000 $\pm$ 0.00	0.0017 $\pm$ 0.00
<i>k</i> -means + VAE No. 2	4	D	0.0000 $\pm$ 0.00	0.0171 $\pm$ 0.00
<i>k</i> -means + VAE No. 2	4	LD	0.0000 $\pm$ 0.00	0.0098 $\pm$ 0.00
<i>k</i> -means + VAE No. 2	5	D	0.0000 $\pm$ 0.00	0.0211 $\pm$ 0.00
<i>k</i> -means + VAE No. 2	5	LD	0.0000 $\pm$ 0.00	0.0117 $\pm$ 0.00
SC + VAE No. 2	2	D	0.0000 $\pm$ 0.00	0.0075 $\pm$ 0.00
SC + VAE No. 2	2	PT	0.0000 $\pm$ 0.00	0.0010 $\pm$ 0.00
SC + VAE No. 2	3	D	0.0000 $\pm$ 0.00	0.0128 $\pm$ 0.00
SC + VAE No. 2	3	PT	0.0000 $\pm$ 0.00	0.0019 $\pm$ 0.00
SC + VAE No. 2	4	D	0.0000 $\pm$ 0.00	0.0173 $\pm$ 0.00
SC + VAE No. 2	4	PT	0.0000 $\pm$ 0.00	0.0026 $\pm$ 0.00
SC + VAE No. 2	5	D	0.0000 $\pm$ 0.00	0.0213 $\pm$ 0.00
SC + VAE No. 2	5	LD	0.0000 $\pm$ 0.00	0.0121 $\pm$ 0.00
SC + VAE No. 2	5	PT	0.0000 $\pm$ 0.00	0.0031 $\pm$ 0.00
GMM + VAE No. 3	2	D	0.0000 $\pm$ 0.00	0.0071 $\pm$ 0.00
GMM + VAE No. 3	2	LD	0.0000 $\pm$ 0.00	0.0038 $\pm$ 0.00
GMM + VAE No. 3	2	PT	0.0000 $\pm$ 0.00	0.0010 $\pm$ 0.00
GMM + VAE No. 3	3	D	0.0000 $\pm$ 0.00	0.0125 $\pm$ 0.00

Table A15. Cont.

Method	No. Groups (k)	Target	Mean ARI $\pm$ sd	Mean NMI $\pm$ sd
GMM + VAE No. 3	3	LD	0.0000 $\pm$ 0.00	0.0071 $\pm$ 0.00
GMM + VAE No. 3	4	D	0.0000 $\pm$ 0.00	0.0171 $\pm$ 0.00
GMM + VAE No. 3	4	LD	0.0000 $\pm$ 0.00	0.0095 $\pm$ 0.00
k-means + VAE No. 3	2	D	0.0000 $\pm$ 0.00	0.0071 $\pm$ 0.00
k-means + VAE No. 3	2	LD	0.0000 $\pm$ 0.00	0.0037 $\pm$ 0.00
k-means + VAE No. 3	2	PT	0.0000 $\pm$ 0.00	0.0009 $\pm$ 0.00
k-means + VAE No. 3	3	D	0.0000 $\pm$ 0.00	0.0127 $\pm$ 0.00
k-means + VAE No. 3	3	LD	0.0000 $\pm$ 0.00	0.0070 $\pm$ 0.00
k-means + VAE No. 3	3	PT	0.0000 $\pm$ 0.00	0.0018 $\pm$ 0.00
k-means + VAE No. 3	4	D	0.0000 $\pm$ 0.00	0.0172 $\pm$ 0.00
k-means + VAE No. 3	4	LD	0.0000 $\pm$ 0.00	0.0096 $\pm$ 0.00
k-means + VAE No. 3	4	PT	0.0000 $\pm$ 0.00	0.0025 $\pm$ 0.00
k-means + VAE No. 3	5	D	0.0000 $\pm$ 0.00	0.0213 $\pm$ 0.00
k-means + VAE No. 3	5	LD	0.0000 $\pm$ 0.00	0.0117 $\pm$ 0.00
SC + VAE No. 3	2	D	0.0000 $\pm$ 0.00	0.0072 $\pm$ 0.00
SC + VAE No. 3	2	LD	0.0000 $\pm$ 0.00	0.0040 $\pm$ 0.00
SC + VAE No. 3	3	D	0.0000 $\pm$ 0.00	0.0127 $\pm$ 0.00
SC + VAE No. 3	3	LD	0.0000 $\pm$ 0.00	0.0069 $\pm$ 0.00
SC + VAE No. 3	3	PT	0.0000 $\pm$ 0.00	0.0018 $\pm$ 0.00
SC + VAE No. 3	4	D	0.0000 $\pm$ 0.00	0.0174 $\pm$ 0.00
SC + VAE No. 3	4	LD	0.0000 $\pm$ 0.00	0.0096 $\pm$ 0.00
SC + VAE No. 3	4	PT	0.0000 $\pm$ 0.00	0.0024 $\pm$ 0.00
SC + VAE No. 3	5	D	0.0000 $\pm$ 0.00	0.0211 $\pm$ 0.00
SC + VAE No. 3	5	PT	0.0000 $\pm$ 0.00	0.0031 $\pm$ 0.00
SC + Vanilla No. 2	3	D	0.0000 $\pm$ 0.00	0.0073 $\pm$ 0.00
SC + CAE No. 2	2	PT	−0.0001 $\pm$ 0.00	0.0031 $\pm$ 0.00
SC + DAE No. 2	2	PT	−0.0001 $\pm$ 0.00	0.0210 $\pm$ 0.05
SC + DAE No. 4	2	PT	−0.0001 $\pm$ 0.00	0.0025 $\pm$ 0.00
k-means + VAE No. 1	5	LD	−0.0001 $\pm$ 0.00	0.0115 $\pm$ 0.00
SC + VAE No. 1	5	LD	−0.0001 $\pm$ 0.00	0.0116 $\pm$ 0.00
GMM + VAE No. 2	3	LD	−0.0001 $\pm$ 0.00	0.0069 $\pm$ 0.00
GMM + VAE No. 2	5	PT	−0.0001 $\pm$ 0.00	0.0028 $\pm$ 0.00
k-means + VAE No. 2	4	PT	−0.0001 $\pm$ 0.00	0.0024 $\pm$ 0.00
k-means + VAE No. 2	5	PT	−0.0001 $\pm$ 0.00	0.0028 $\pm$ 0.00
SC + VAE No. 2	2	LD	−0.0001 $\pm$ 0.00	0.0040 $\pm$ 0.00
GMM + VAE No. 3	3	PT	−0.0001 $\pm$ 0.00	0.0016 $\pm$ 0.00
GMM + VAE No. 3	4	PT	−0.0001 $\pm$ 0.00	0.0024 $\pm$ 0.00
GMM + VAE No. 3	5	D	−0.0001 $\pm$ 0.00	0.0211 $\pm$ 0.00
GMM + VAE No. 3	5	PT	−0.0001 $\pm$ 0.00	0.0030 $\pm$ 0.00
SC + VAE No. 3	2	PT	−0.0001 $\pm$ 0.00	0.0009 $\pm$ 0.00
SC + VAE No. 3	5	LD	−0.0001 $\pm$ 0.00	0.0118 $\pm$ 0.00
SC + Vanilla No. 2	2	D	−0.0001 $\pm$ 0.00	0.0033 $\pm$ 0.00
SC + CAE No. 2	4	LD	−0.0002 $\pm$ 0.01	0.0334 $\pm$ 0.02
GMM + VAE No. 3	5	LD	−0.0002 $\pm$ 0.00	0.0116 $\pm$ 0.00
SC + DAE No. 3	2	LD	−0.0003 $\pm$ 0.00	0.0028 $\pm$ 0.00
SC + DAE No. 4	2	D	−0.0003 $\pm$ 0.00	0.0032 $\pm$ 0.00
SC + CAE No. 2	3	D	−0.0004 $\pm$ 0.00	0.0293 $\pm$ 0.02
SC + Vanilla No. 4	2	LD	−0.0004 $\pm$ 0.00	0.0131 $\pm$ 0.01
SC + CAE No. 2	3	PT	−0.0005 $\pm$ 0.00	0.0278 $\pm$ 0.02
SC + Vanilla No. 2	2	PT	−0.0008 $\pm$ 0.00	0.0025 $\pm$ 0.00
SC + DAE No. 4	2	LD	−0.0012 $\pm$ 0.00	0.0025 $\pm$ 0.00
SC + Vanilla No. 2	3	PT	−0.0015 $\pm$ 0.00	0.0048 $\pm$ 0.00
SC + DAE No. 4	3	LD	−0.0022 $\pm$ 0.00	0.0066 $\pm$ 0.00
SC + CAE No. 2	3	LD	−0.0023 $\pm$ 0.00	0.0173 $\pm$ 0.01
SC + DAE No. 4	4	LD	−0.0024 $\pm$ 0.00	0.0119 $\pm$ 0.00
SC + DAE No. 4	5	LD	−0.0040 $\pm$ 0.01	0.0269 $\pm$ 0.01
SC + Vanilla No. 1	2	LD	−0.0044 $\pm$ 0.00	0.0063 $\pm$ 0.00
SC + Vanilla No. 1	3	LD	−0.0051 $\pm$ 0.00	0.0093 $\pm$ 0.00
SC + Vanilla No. 1	4	LD	−0.0055 $\pm$ 0.00	0.0137 $\pm$ 0.00
SC + Vanilla No. 1	5	LD	−0.0057 $\pm$ 0.00	0.0160 $\pm$ 0.00

Appendix D

Table A16. Comparison of NMI and ARI results on the original 458 dimensional data for different clustering methods and numbers of clusters, with respect to the variables: *Primary Type* (PT), *Description* (D), *Location Description* (LD).

Method	No. of Groups	NMI (PT)	NMI (D)	NMI (LD)	ARI (PT)	ARI (D)	ARI (LD)
k-means	2	0.0521	0.0531	0.0109	0.0227	0.0104	0.0048
k-means	3	0.2636	0.2499	0.1214	0.2043	0.0997	0.1062
k-means	4	0.0037	0.0196	0.0130	0.0005	0.0004	0.0003
k-means	5	0.1119	0.0982	0.0297	0.0342	0.0113	0.0019
GMM	2	0.0426	0.0439	0.0093	0.0107	0.0028	0.0014
GMM	3	0.1611	0.1651	0.0826	0.0834	0.0457	0.0461
GMM	4	0.0037	0.0196	0.0130	0.0005	0.0004	0.0003
GMM	5	0.1199	0.1046	0.0295	0.0366	0.0122	0.0022
SC	2	0.0232	0.0265	0.0394	−0.0085	0.0016	0.0219
SC	3	0.0260	0.0296	0.0507	−0.0092	0.0014	0.0236
SC	4	0.0209	0.0284	0.0509	−0.0092	0.0011	0.0221
SC	5	0.1274	0.1053	0.0697	0.0212	0.0127	0.0230

References

1. Qamar, R.; Zardari, B.A. Artificial Neural Networks: An Overview. *Mesopotamian J. Comput. Sci.* **2023**, *2023*, 130–139. [\[CrossRef\]](#)
2. Fradkov, A.L. Early History of Machine Learning. *IFAC-PapersOnLine* **2020**, *53*, 1385–1390. [\[CrossRef\]](#)
3. Ballard, D.H. Modular Learning in Neural Networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Seattle, WA, USA, 13–17 July 1987; pp. 279–284.
4. Schmidhuber, J. Deep Learning in Neural Networks: An Overview. *arXiv* **2014**, arXiv:1404.7828.
5. Bank, D.; Koenigstein, N.; Giryas, R. Autoencoders. *arXiv* **2020**, arXiv:2003.05991.
6. Michelucci, U. An Introduction to Autoencoders. *arXiv* **2022**, arXiv:2201.03898.
7. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. In Proceedings of the International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2014.
8. Liu, Y.; Yang, Z.; Yu, Z.; Liu, Z.; Liu, D.; Lin, H.; Li, M.; Ma, S.; Avdeev, M.; Shi, S. Generative Artificial Intelligence and Its Applications in Materials Science: Current Situation and Future Perspectives. *J. Mater.* **2023**, *9*, 798–816. [\[CrossRef\]](#)
9. Purwono, P.; Ma'arif, A.; Rahmiani, W.; Fathurrahman, H.I.K.; Frisky, A.Z.K.; ul Haq, Q.M. Understanding of Convolutional Neural Network (CNN): A Review. *Int. J. Robot. Control Syst.* **2023**, *2*, 739–748. [\[CrossRef\]](#)
10. Zakeri, A.; Xia, Y.; Ravikumar, N.; Frangi, A.F. *Deep Learning for Vision and Representation Learning*; Elsevier: Amsterdam, The Netherlands, 2024; pp. 451–474.
11. Jain, A.K. Data Clustering: 50 Years beyond K-Means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666. [\[CrossRef\]](#)
12. Theodoridis, S.; Koutroumbas, K. Clustering Algorithms IV. In *Pattern Recognition*, 4th ed.; Academic Press: Burlington, MA, USA, 2009; pp. 765–862. [\[CrossRef\]](#)
13. Xie, J.; Girshick, R.; Farhadi, A. Unsupervised Deep Embedding for Clustering Analysis. In Proceedings of the 33rd International Conference on Machine Learning (ICML 2016), New York NY, USA, 19–24 June 2016; Volume 48, pp. 478–487.
14. ArfaeiYazdiPour, M.; Javidi, N.; Mohammadi, A.; Choobchian, P.; Hair, J.F., Jr.; Mohammadian, A.K. ActivAbility: A Structural Framework for Quantifying Built Environment Impacts on Active Travel Using Data-Driven Sub-Scores. *SSRN Electron. J.* **2025**. [\[CrossRef\]](#)
15. İlgin, E.G.; Dener, M. Exploratory Data Analysis, Time Series Analysis, Crime Type Prediction, and Trend Forecasting in Crime Data Using Machine Learning, Deep Learning, and Statistical Methods. *Neural. Comput. Applic.* **2025**, *37*, 11773–11798. [\[CrossRef\]](#)
16. Cui, Z.; Ke, R.; Pu, Z.; Wang, Y. Crime Prediction with Graph Neural Networks and Multivariate Normal Distributions. *arXiv* **2021**, arXiv:2111.14733.
17. Karim, M.R.; Beyan, O.; Zappa, A.; Costa, I.G.; Rebholz-Schuhmann, D.; Cochez, M.; Decker, S. Deep Learning-Based Clustering Approaches for Bioinformatics. *Brief. Bioinform.* **2020**, *22*, 393–415. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Zong, B.; Song, Q.; Min, M.R.; Cheng, W.; Lumezanu, C.; Cho, D.; Chen, H. Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.

19. Agrawal, R. Complete Guide to Anomaly Detection with Autoencoders Using TensorFlow. Analytics Vidhya. 2022. Available online: <https://www.analyticsvidhya.com/blog/2022/01/complete-guide-to-anomaly-detection-with-autoencoders-using-tensorflow/> (accessed on 10 April 2026).
20. City of Chicago. Crimes–2001 to Present. Available online: https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2/about_data (accessed on 4 November 2024).
21. Hugging Face. Sentence Transformers. Available online: <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2> (accessed on 10 April 2026).
22. Hotelling, H. Analysis of a Complex of Statistical Variables into Principal Components. *J. Educ. Psychol.* **1933**, *24*, 417–441. [CrossRef]
23. McQueen, J. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*; University of California Press: Berkeley, CA, USA, 1967; Volumn 1, pp. 281–297.
24. Pearson, K. Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous Material. *Philos. Trans. R. Soc. Lond. A* **1894**, *185*, 71–110. [CrossRef]
25. Hubert, L.; Arabie, P. Comparing Partitions. *J. Classif.* **1985**, *2*, 193–218. [CrossRef]
26. Vinh, N.X.; Epps, J.; Bailey, J. Information Theoretic Measures for Clusterings Comparison: Is a Correction for Chance Necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, Montreal, Canada, 14–18 June 2009; pp. 1073–1080. [CrossRef]
27. Elharrouss, O.; Mahmood, Y.; Bechqito, Y.; Serhani, M.A.; Badidi, E.; Riffi, J.; Tairi, H. Loss Functions in Deep Learning: A Comprehensive Review. *arXiv* **2025**, arXiv:2504.04242.
28. Raschka, S. Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. *arXiv* **2018**, arXiv:1811.12808.
29. Terven, J.; Cordova-Esparza, D.M.; Romero-González, J.A.; Ramírez-Pedraza, A.; Chavez-Urbiola, E.A. A Comprehensive Survey of Loss Functions and Metrics in Deep Learning. *Artif. Intell. Rev.* **2025**, *58*, 195. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.