Unsupervised Learning – Clustering

Styliani Kalaitzaki, aid24009

December 2023

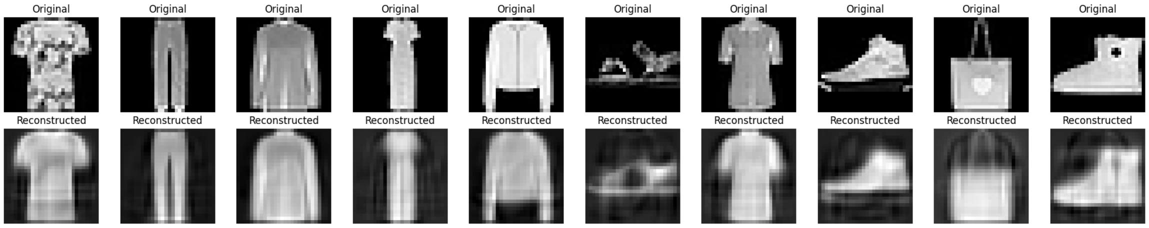
1 Introduction

This project focuses on evaluating dimensionality reduction and clustering techniques using the Fashion-MNIST dataset. It aims to compare the impact of raw data versus synthetically derived features on clustering performance. The study involves implementing five dimensionality reduction techniques, including 'PCA', 'Factor Analysis', 'FastICA', 'Stacked Autoencoder' and 'CSAE', along with five clustering methods. Evaluation criteria encompass four performance metrics.

2 Dimensionality Reduction Techniques

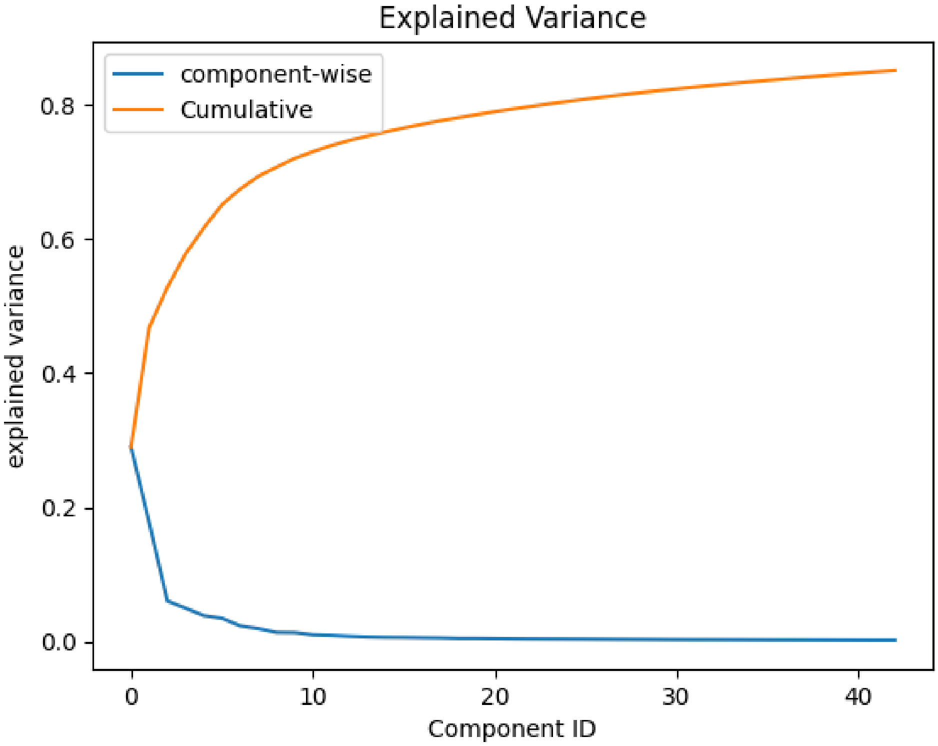
PCA

The reconstruction appears to capture the essential features of the original image quite well. There is a noticeable loss of detail naturally, as the images appear more blurry. Certain features with narrower dimensions, such as straps on shoes or bags, seem to be affected the most, resulting in some distortion. The shadows or areas of blackness have undergone a subtle transformation with a smoother and more abstract feel compared to the sharpness of the original. This may be positive if it captures the overall shape of a dress for example, but it might mistake a sandal for a shoe.



Explained variance is a concept often used in Principal Component Analysis (PCA). PCA aims to transform the original features of a dataset into a new set of uncorrelated features called principal components. These principal components are ordered by the amount of variance they capture in the data.

The explained variance represents the proportion of the total variance in the data that is retained, or ”explained,” by each principal component. In other words, it quantifies how much information each principal component contributes to the overall variability in the dataset.

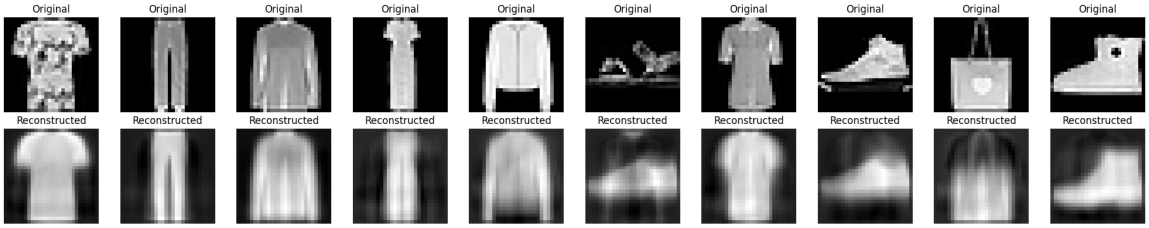


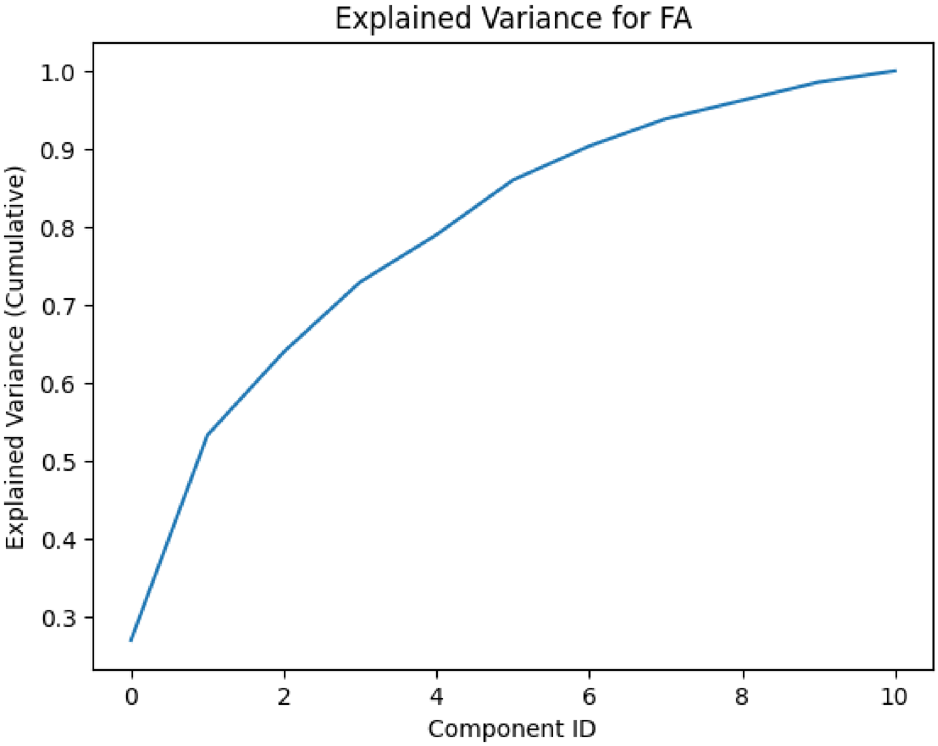
From the plot we can read off the percentage of the variance in the data explained as we add principal components. So the first principal component explains about 30% of the variance. The first ten principal components combined explain almost 72% of the variance in the data, after which it shows diminishing returns for retaining additional eigenvalues. After about 43 principal components almost 85% of all variance is accounted for.

These results indicate that a relatively small number of principal components, compared to the initial 784 features, can capture a significant portion of the variability in the data. This is a positive outcome because it means we can reduce the dimensions of the data while retaining most of the important information.

Factor Analysis (FA)

The reconstructed images appear similar with the PCA technique but more blurry. FA manages to effectively capture the main patterns while reducing the overall complexity of the image. Again, there might be a problem with the sandal resembling a sneaker.

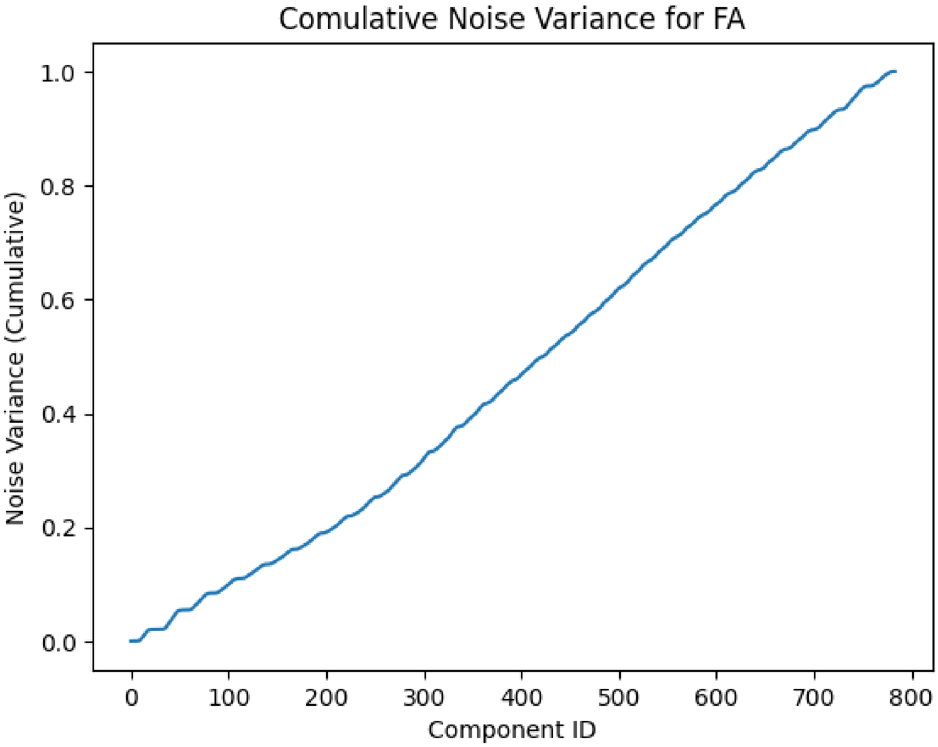




About 90% of the variance is explained by the initial 6 components. After 11 principal components almost 100% of all variance is accounted for. This suggests that a noteworthy reduction in data dimensionality can be achieved using even fewer components than the PCA technique.

Cumulative noise variance represents the proportion of total variance in the observed variables that is not explained by the underlying factors. A higher cumulative noise variance indicates that a larger proportion of the total variance is due to unexplained or random variability.

In factor analysis, the goal is typically to have a low cumulative noise variance. This implies that the factors extracted from the data account for a substantial portion of the observed variance, suggesting a more meaningful and interpretable model.

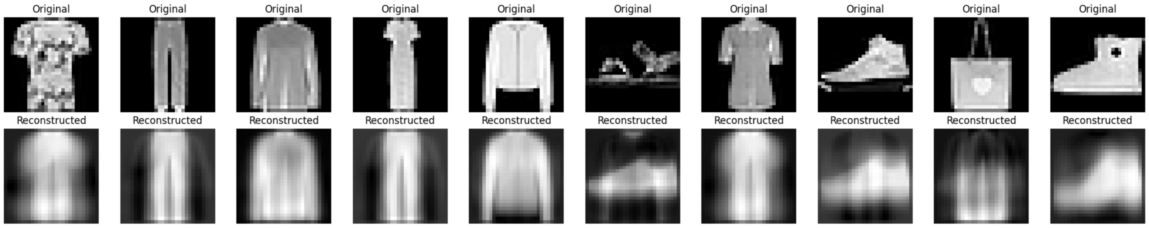


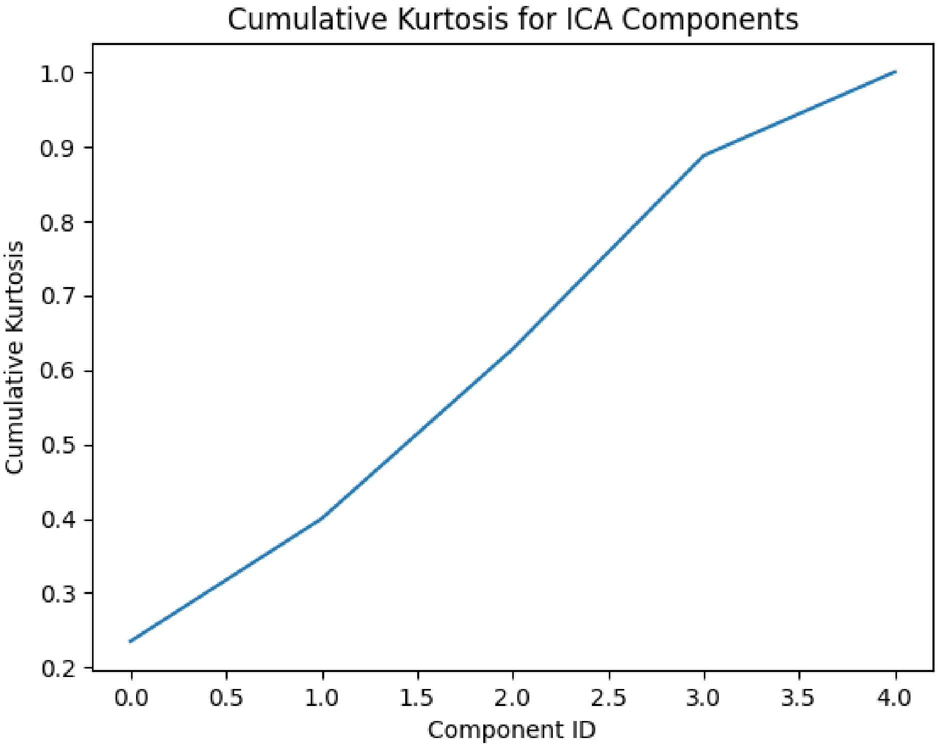
The plot demonstrates that the noise variance begins to increase significantly at an early stage. However, this escalation is contained due to the efficiency of the analysis—merely 11 principal components suffice to explain nearly 100% of the variance, confining the noise to less than 1%.

This efficient reduction in dimensionality allows for a concise representation of the data, with a minimal amount of unexplained variance or noise.

FastICA

The reconstruction process has significantly distorted the images, particularly notable in the scandal, which now resembles a sneaker, and the dress, which bears an uncanny resemblance to trousers. This outcome was anticipated, given that the fastICA algorithm was configured to reduce the dimensions from 784 to just 5. Despite the distortion, this configuration was chosen due to superior performance metrics in this reduced dimensional space.



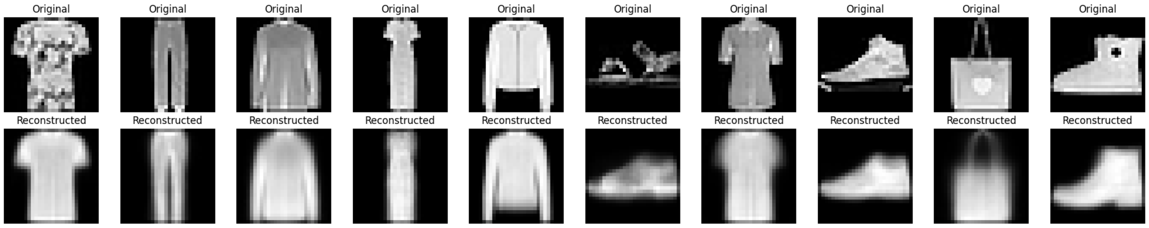


Up until the 3rd component there is a steep increases in the cumulative kurtosis plot. This indicates components that contribute significantly to the overall non-Gaussianity. Components associated with sharp increases may be crucial in capturing important features or sources in the data.

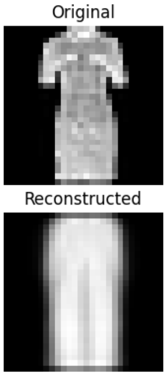
Stacked Autoencoder (SAE)

1. The following results make use of a dropout rate of 0.3:

The reconstruction does not distort the background of the images and manages to capture the main patterns of the original image. However, the sandal is perceptually transformed into what seems more like a sneaker. All reconstructed images exhibit a significant increase in brightness.



Here is another sample from a different run where the image resembles an entirely different piece of clothing:



Although other techniques yielded more significant distortions, the results below indicate that this particular approach was unsatisfactory.

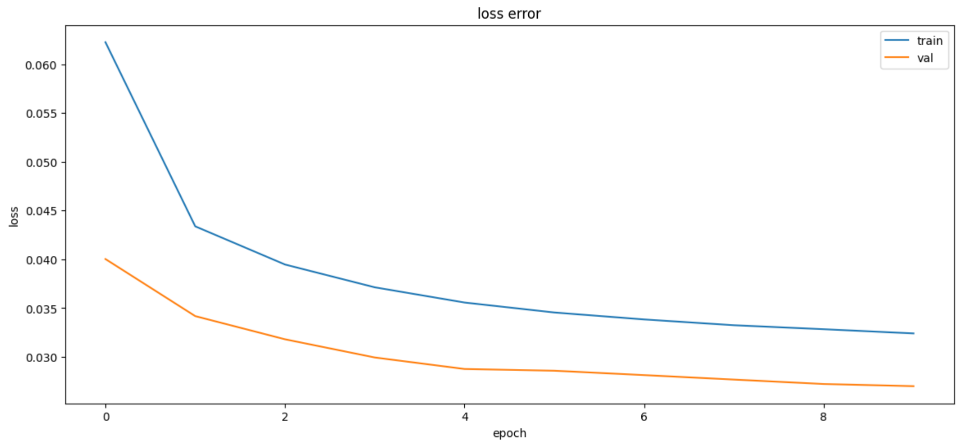


Figure 1: loss = mean squared error

The loss function measures the difference between the predicted output of the autoencoder and the actual input. The goal of training is to minimize this loss, which indicates how well the autoencoder is reconstructing its input data.

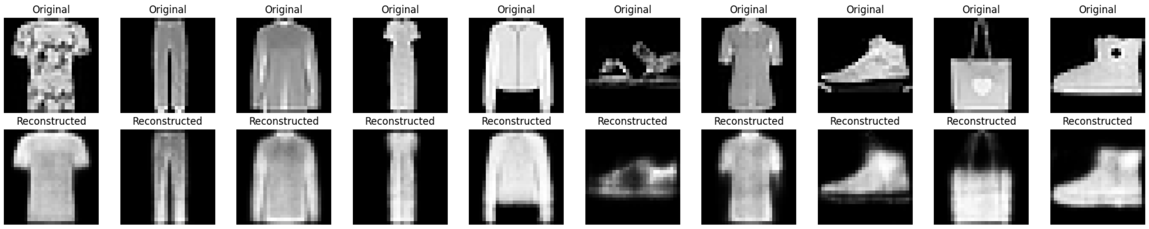
The loss decreases over time as the autoencoder learns to better reconstruct the input data. A decreasing trend suggests that the model is learning and improving its representation of the data. No sudden jumps or spikes in the loss which indicates no issues such as learning rate being too high, numerical instability, or problems with the data.

A decreasing validation loss suggests that the model is learning to generalize well to new, unseen data. This is crucial because the ultimate goal is to perform well on data it has not been trained on. The model is effectively learning meaningful representations from the input data. The lower validation loss indicates that the learned features are applicable not just to the training data but to a broader range of examples.

However, there exists a noticeable gap between the two lines. More precisely, the validation loss registers a lower value than the training loss. This discrepancy suggests that the validation dataset may be easier for the model to predict than the training dataset. While the curves exhibit a marginal, ongoing decrease towards the end of the plot, it's plausible that they are approaching a state of stability. The comparison between original and reconstructed images further underscores this observation, revealing poor results in the reconstructed images when contrasted with other dimensionality reduction techniques.

2. The following results make use of a dropout rate of 0.0:

In this scenario, the reconstruction has improved from before. That is expected since without a dropout the model has learned better this dataset.



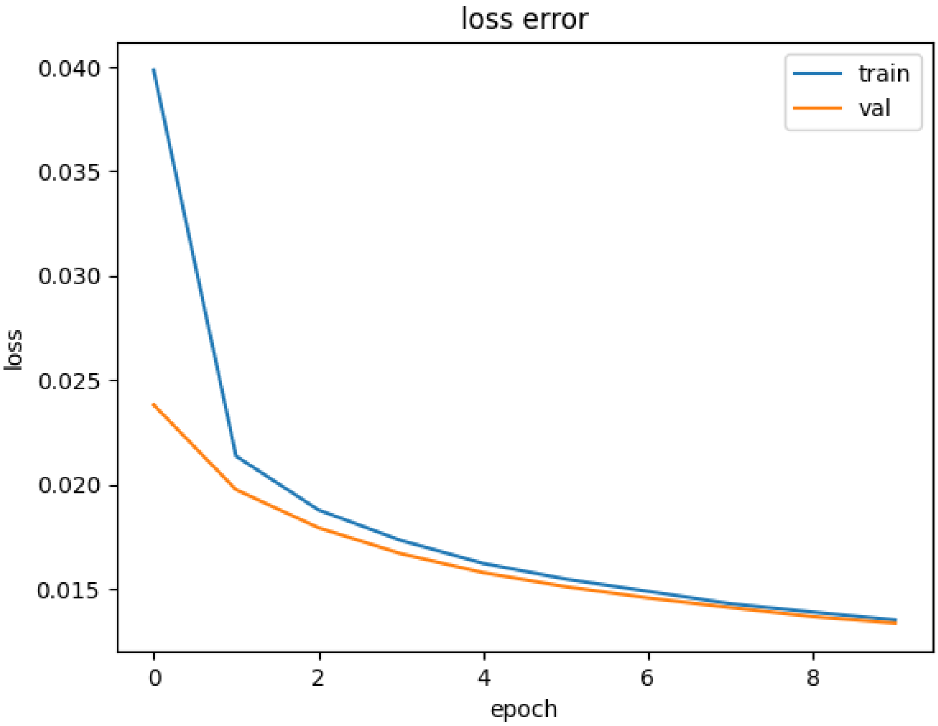
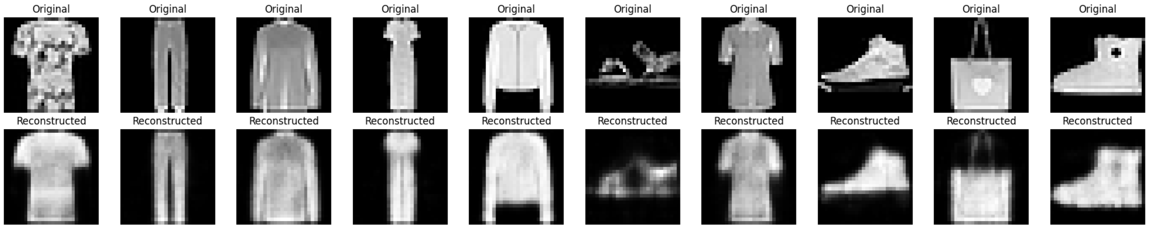


Figure 2: loss = mean squared error

A good fit is identified by a training and validation loss that decreases to a point of stability with a minimal gap between the two final loss values. In this case, the reconstructed images demonstrate an expected improvement, benefitting from the model's enhanced learning on the training data without dropout. Both lines depicting loss errors exhibit notable enhancements, with the training loss line showcasing a steep decline, indicative of rapid learning. Smaller error numbers further reinforce this progress, and the convergence of these lines signifies the model's improved ability to generalize and perform well on unseen data.

Convolutional Stacked Autoencoder (CSAE)

It's a good illustration of the trade-off between dimensionality reduction and maintaining image quality. The compressed version is visually close, but not identical to the original. The compressed image strikes a balance between reducing dimensions and maintaining the overall structure, although some finer details are inevitably lost. The technique exhibits better results on straps by blurring only the object and preserving the black background without distortion. This results in a pronounced contrast between the background and the brightly highlighted object, a notable improvement over previous techniques that tended to blur the entire image.



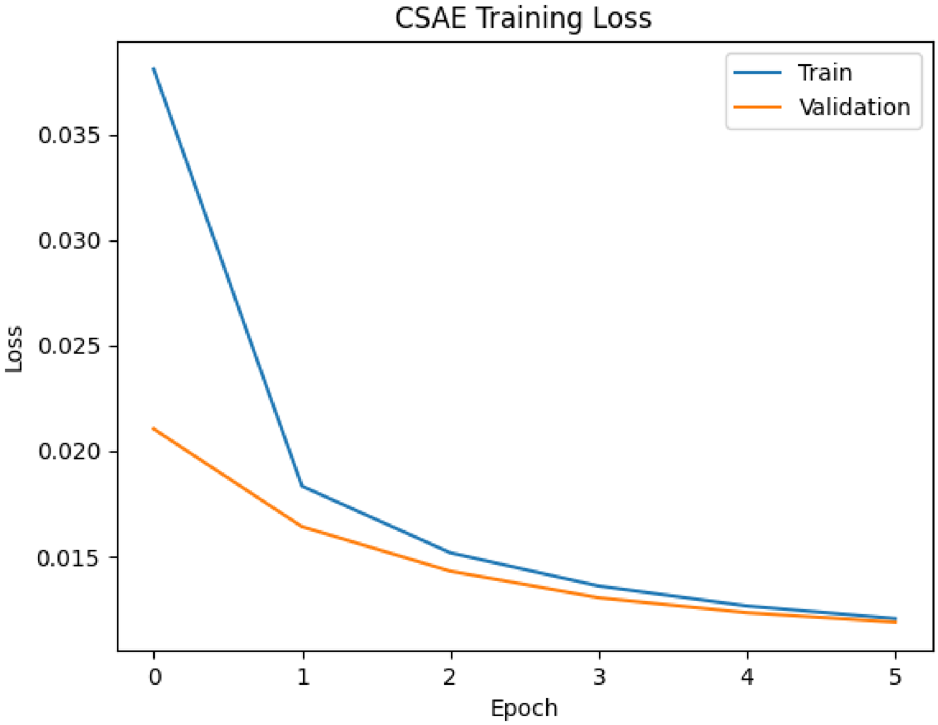
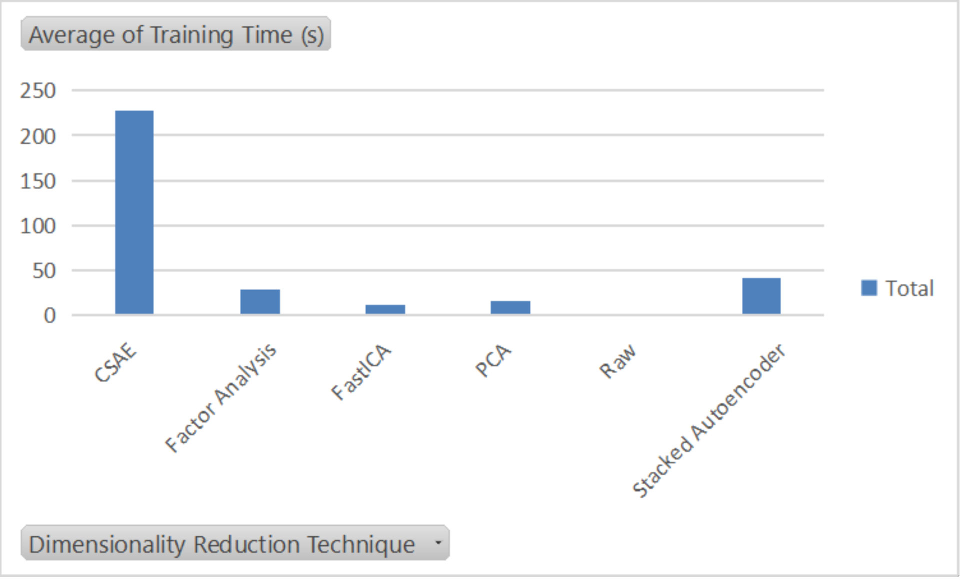


Figure 3: loss = mean squared error

The loss curve has a steep decrease, which suggests that the model is learning rapidly and making substantial improvements in reducing reconstruction error. This is a positive sign, especially in the early stages of training. Both loss curves decrease, and they are close to each other. This indicates that the model is learning well on the training data and is also able to generalize to unseen data (validation set). It suggests that the model is not overfitting or underfitting.

It strongly indicates that the dimensionality reduction process is likely functioning effectively within the autoencoder model. Overall, these positive trends collectively support the inference that the dimensionality reduction implemented by the autoencoder is robust and yields meaningful representations applicable to a broader range of examples.

2.1 Training time



FastICA stands out as the quickest method for dimensionality reduction, whereas CSAE requires the longest training time. This is attributed to the fact that neural networks, including autoencoders, involve complex architectures with multiple layers and numerous parameters that need to be fine-tuned through iterative training epochs. The training process involves adjusting the weights and biases in the network to minimize a specified loss function, and this optimization can be computationally demanding, particularly when dealing with large datasets or deep architectures. FastICA is configured to compute only 5 components, a considerably smaller number compared to other techniques, resulting in a faster training process.

3 Clustering Algorithms

Unlike the other four algorithms, where the option to set the number of clusters is available, DBSCAN faces a unique challenge. Given its nature, it does not have prior knowledge about the number of clusters to identify. Consequently, it may struggle to precisely discover ten clusters (Figure 4) unless its parameters are meticulously fine-tuned for the specific dataset.

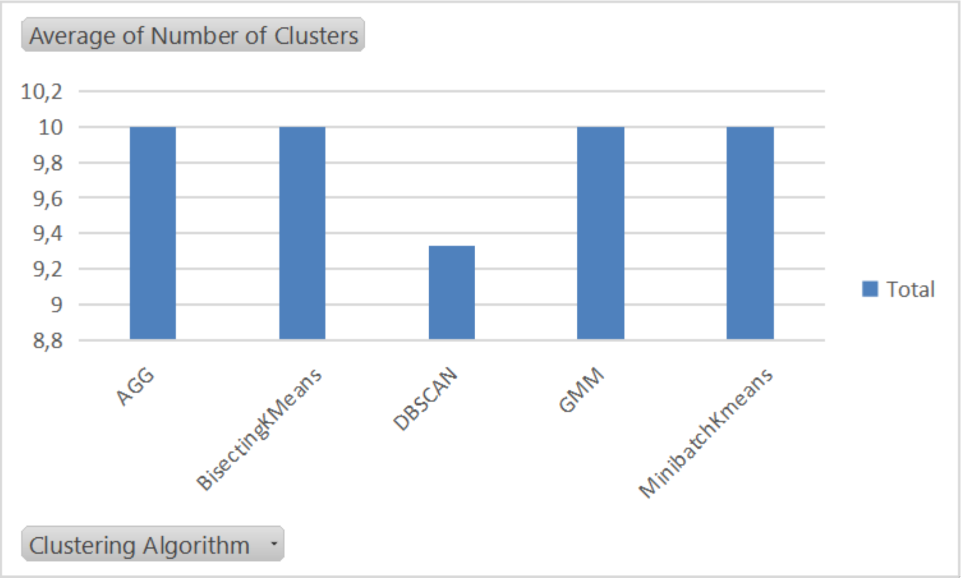


Figure 4: Number of suggested clusters per algorithm

DBSCAN

For DBSCAN to operate effectively, it is crucial to set key parameters, such as 'eps,' which represents the maximum distance between two samples for one to be considered in the neighborhood of the other, and 'min_samples'. These parameters hold significant importance and should be chosen appropriately based on the characteristics of the dataset.

Upon applying various dimensionality reduction techniques to the data, the distances between points undergo alterations. The transformation may introduce compression or distortion to the original data, consequently influencing the distances between data points. Following iterative experimentation, the 'eps' parameter was fine-tuned to values that yielded approximately 10 clusters for each clustering algorithm.

Nevertheless, it remains uncertain whether these fine-tuned values represent the optimal configuration. Further exploration and refinement may be necessary to ascertain the most suitable 'eps' values for robust and accurate clustering outcomes.

3.1 Execution Time

The computation time for the Gaussian Mixture Models algorithm significantly surpasses that of other clustering methods. Agglomerative clustering comes next in terms of time consumption, while both variants of the KMeans algorithm exhibit the fastest processing speeds. DBSCAN also demonstrates a noteworthy level of time efficiency.

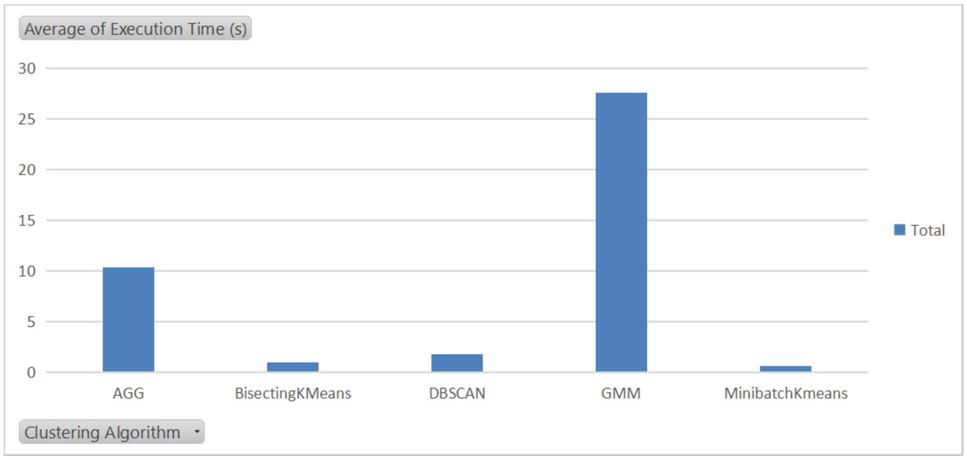


Figure 5: Execution Time of Clustering Algorithms

When examining the execution time of each clustering algorithm in conjunction with dimensionality reduction techniques, it becomes apparent that Gaussian Mixture Models (GMM) applied to raw data exhibit notably higher execution times. This discrepancy is not only evident when compared to other algorithms but also when contrasted with the same algorithm utilizing any dimensionality reduction technique.

Gaussian Mixture Models are known for their computational complexity, especially when dealing with high-dimensional data. The algorithm involves estimating parameters for each component in each dimension, which can be computationally expensive. This complexity is exacerbated when applied to raw image data without any dimensionality reduction.

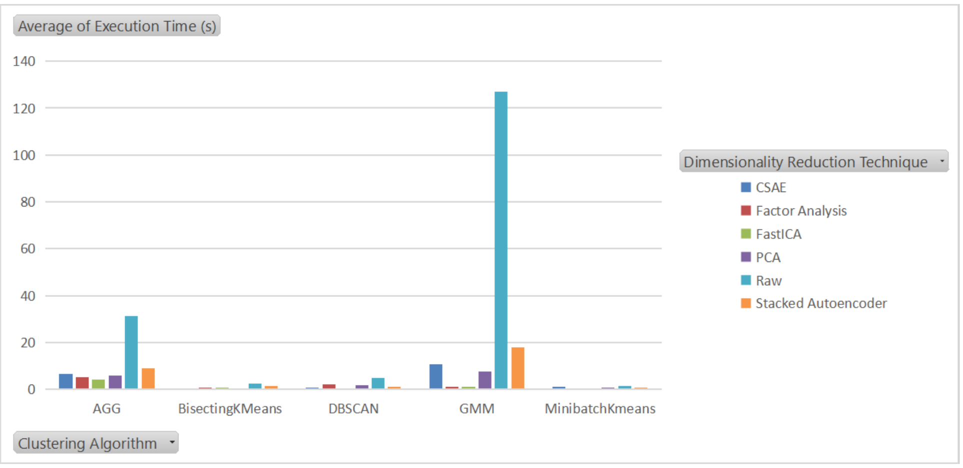


Figure 6: Execution Time of Clustering Algorithms by Dimensionality Reduction Technique

By removing the Raw data from the plot for better clarity, it is observed that the Gaussian Mixture models achieve the highest times in pair with the stacked autoencoders and PCA. Apparently, it works better when coupled with Factor Analysis and FastICA. This is probably attributed to the fact that these two techniques were set to compute fewer components than the rest.
Agglomerative Clustering (AGG) similarly benefits from low-dimensionality data, as it operates faster with FA and FastICA. BisectingKMeans demonstrates its best times with CSAE and PCA. Interestingly, MiniBatchKMeans exhibits its poorest performance when paired with these two methods. DBSCAN achieves its fastest performance with FastICA but takes longer with Factor Analysis. Both techniques reduce data to the lowest dimensions compared to others, suggesting that DBSCAN's time complexity is relatively independent of dimensionality.

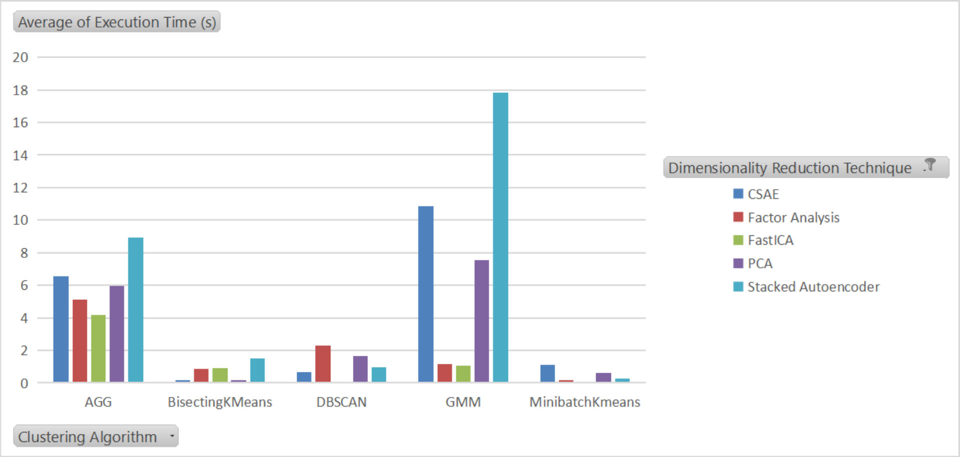


Figure 7: Execution Time (Without Raw)

4 Evaluation

Four metrics were computed for the evaluation of the clustering techniques, Calinski–Harabasz index, Davies–Bouldin index, Silhouette score and Adjusted Rand Index (ARI).

The Calinski-Harabasz index is a measure used to evaluate the goodness of a clustering algorithm. It assesses the ratio of the between-cluster variance to the within-cluster variance. The index aims to find clusters that are well-separated from each other and compact internally. A higher Calinski-Harabasz index value indicates better-defined clusters. The optimal number of clusters is usually the one that maximizes this index.

In the plot, FastICA demonstrates the highest values among all clustering algorithms, except for DBSCAN, which exhibits significantly lower values with all dimensionality techniques compared to the other algorithms. The pair that stands out is FastICA - MiniBatch.

Most techniques yield better values than raw data. Exception to this is FA which has the same or in one instance worse results.

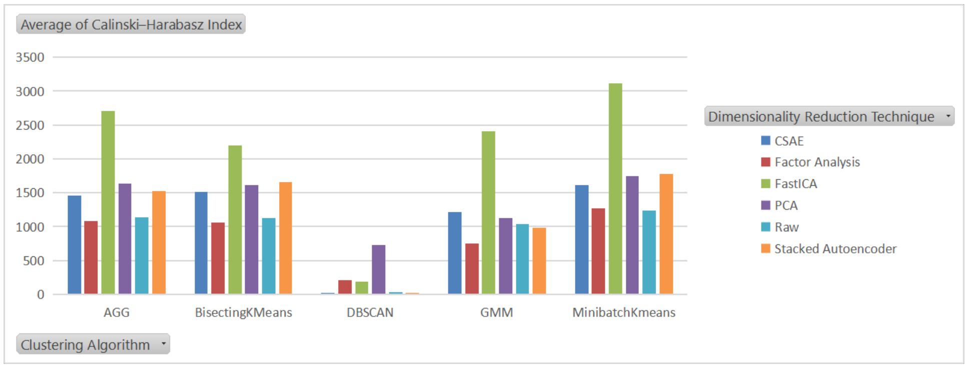


Figure 8: Calinski-Harabasz index

The Davies-Bouldin Index (DBI) is a metric used to evaluate the performance of clustering algorithms. It measures the compactness and separation between clusters in a clustering result. The lower the Davies-Bouldin Index, the better the clustering result. A lower DBI indicates that clusters are more compact and well-separated. Zero is the lowest possible score. Values closer to zero indicate a better partition.

Once again, FastICA stands out with superior values, achieving the lowest scores across all clustering algorithms. The combination that appears to yield the best results is, yet again, FastICA with MiniBatch, which marginally outperforms the combination of FastICA with DBSCAN.

Something to note here is that the Davies-Boulding index is generally higher for convex clusters than other concepts of clusters, such as density based clusters like those obtained from DBSCAN. This characteristic arises from the nature of how the index is computed and its sensitivity to the compactness and separation of clusters.

The raw data consistently exhibit elevated values, with the exception of the Gaussian Mixture Model (GMM), which demonstrates performance (with Raw data) that is not the least favorable.

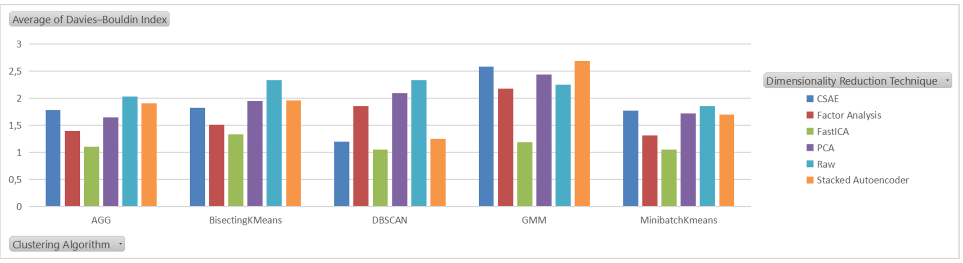


Figure 9: Davies-Boulding index

The silhouette score is a metric used to measure the goodness of a clustering technique. It quantifies how well-separated the clusters are in a given dataset. The silhouette score ranges from -1 to 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

- Close to +1: The data point is well matched to its own cluster and poorly matched to neighboring clusters.
- Around 0: The data point is on or very close to the decision boundary between two neighboring clusters.
- Close to -1: The data point is poorly matched to its own cluster and well matched to a neighboring cluster.

While no clustering algorithm approaches close to 1 or even exceeds 0.5, DBSCAN exhibits the worst values. They are all bellow zero indicating that the data points have been assigned to the wrong clusters. In other words, the clustering algorithm is not performing well, and the clusters are not well-separated. Noteworthy is that the Silhouette Coefficient is generally higher for convex clusters than other concepts of clusters, such as density based clusters like those obtained through DBSCAN.
FastICA - MiniBatch KMeans once again distinguishes itself as a combination, with a score of 0.3, suggesting a certain degree of separation between clusters, albeit not very pronounced.

The Raw data again exhibit poor performance across all clustering algorithms.

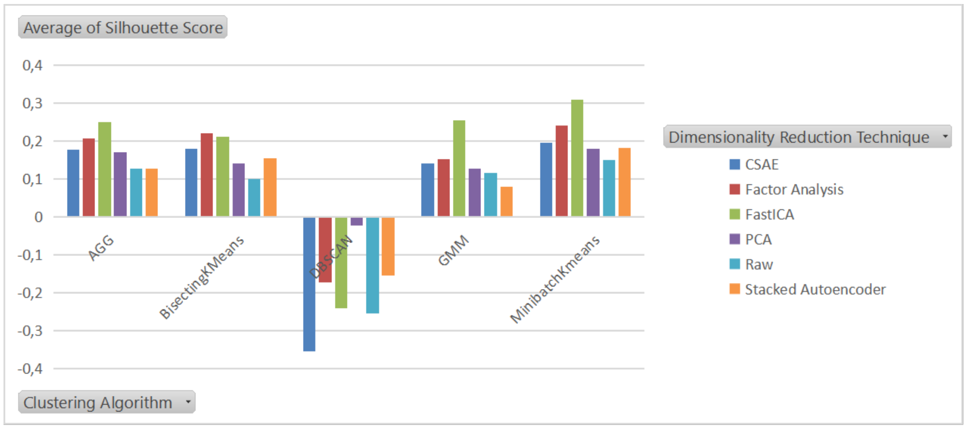
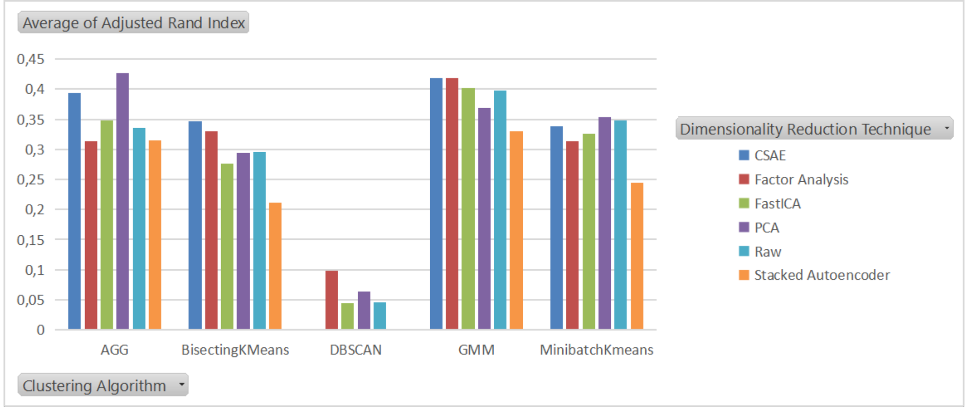


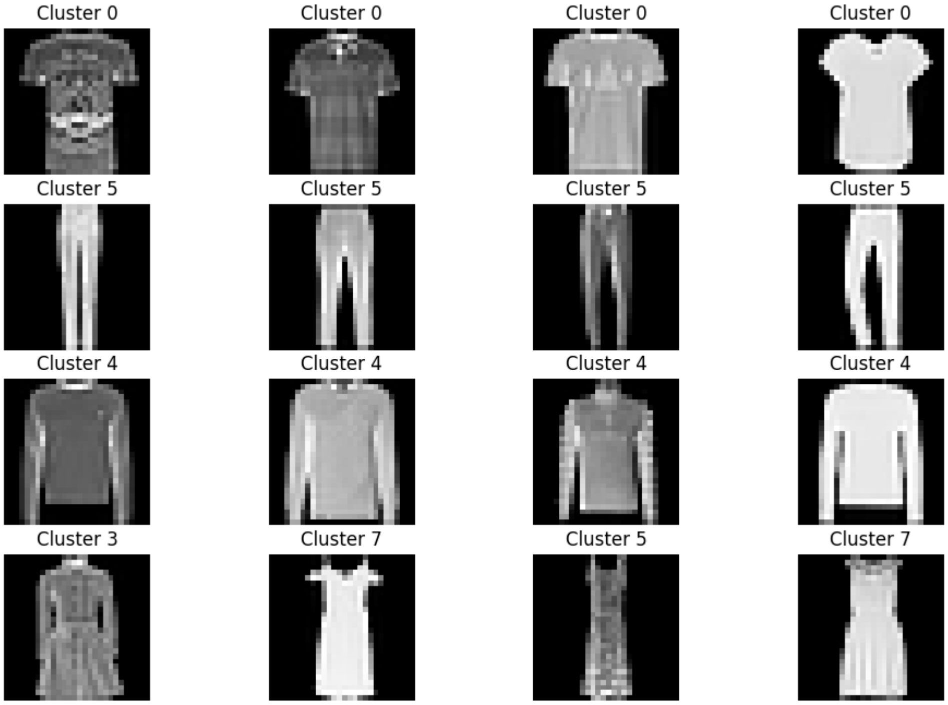
Figure 10: Silhouette score

The Adjusted Rand Index (ARI) uses the true labels (ground truth) to compare how well a clustering algorithm's results match the actual categories or classes of the data. The score is a number between -0.5 and 1. A higher ARI indicates better agreement between the clustering and the true labels, while a lower ARI suggests that the clustering might be no better than random. It shows how well a clustering algorithm performs in terms of grouping similar items together according to the true underlying structure of the data.
The combinations that seem to work better together are PCA - AGG, FA - GMM, CSAE - GMM. DBSCAN exhibits poor results.

This metric demonstrates that the raw data exhibits comparable values to those obtained through other dimensionality reduction techniques.



Presented below is a sample clustering result for random images, obtained from a combination of Factor Analysis (FA) and Gaussian Mixture Model (GMM):



This combination effectively distinguishes between t-shirts, trousers, and pullovers in the provided sample, as each category forms distinct clusters. However, there is some difficulty in accurately clustering dresses; in one instance, a dress has been erroneously grouped in the same cluster as trousers. Notably, the limitation of computing only 5 components in Factor Analysis (FA) introduces distortions in the images, potentially causing dresses to appear more akin to trousers.

5 Conclusions

The metrics indicate that clustering, when applied to data with reduced dimensions, performs comparably or even surpasses its performance when applied to raw data. This suggests that the reduction in dimensions not only does not compromise the efficacy of clustering but, in some cases, enhances its performance. The findings underscore the potential benefits of utilizing reduced-dimensional data for clustering purposes.

There was no single conclusive combination that universally delivered optimal results across all metrics. Nevertheless, among the options considered, FastICA - MiniBatchKMeans consistently outperformed the others in three out of four metrics.