AE-DBSCAN: An Outlier Detection Method for High-Dimensional Datasets

Midé Mabadeje, Jose Hernandez Mejia, Stella Fillmore-Patrick

April 2023

1 Introduction and Literature Review

Outlier detection in high dimensional data is challenging because traditional methods typically rely on distance or density measures. As data dimensionality increases, the distance between data points becomes more uniform and sparse. This is called the 'curse of dimensionality': space increases exponentially as dimensions are added, resulting in this phenomenon. This paper proposes a new method for outlier detection in high dimensional datasets that combines data reduction with an autoencoder with DBSCAN clustering (AE-DBSCAN).

There are several current approaches to increasing the performance of outlier detection in high dimensional datasets. In general, they rely on three strategies: the use of a distance measure that is robust to increasing dimensions; the use of data reduction prior to outlier identification; the use of a method that does not rely on a distance measure.

An exemplary method from the first category is the Local Outlier Factor (LOF) method, which uses the Mahalanobis distance as its distance metric. Mahalanobis distance is robust to increasing dimensionality because it is a distance measure that depends in part upon the covariance structure of the dataset (Ghorbani, 2019). As dimensionality increases, however, the Mahalanobis distance decreases in performance due to computational complexity. In addition, the generalizability of the Mahalanobis distance metric depends upon accurately capturing the covariance of the data, making it sensitive to the presence of outliers. Finally, using LOF for outlier detection is sensitive to hyperparameter selection since a threshold above which a data point is an outlier needs to be selected (Breunig et al, 2000).

Belonging to the second category is outlier detection using Principal Component Analysis (PCA). PCA is a data reduction method that uses linear relationships between features in a dataset to construct principal components that explain the most variability in the dataset. This process reduces the number of features, making traditional methods of outlier detection possible, while effectively combating the effects of the 'curse of dimensionality' (Stefatos et al, 2007). PCA assumes a linear relationship between features in the dataset, making it inapplicable to both nonlinear and non-normally distributed data.

An example of the third category for outlier detection is Isolation Forest (IF). Rather than relying on a distance metric, IF uses random splitting via decision trees to isolate anomalies in the dataset. IF is sensitive to hyperparameter selection and doesn't perform well on datasets with varying data distributions or to detect local outlier populations (Cheng et al, 2019).

The above review of some existing methods for outlier detection in high dimensional datasets indicates that the performance of any given method depends on the properties of the dataset considered for outlier detection. Hence, any tool set for outlier detection should include a variety of methods to be utilized in different situations. To this end, we propose a new outlier detection method for high dimensional data AE-DBSCAN (Auto-Encoder Density-Based Spatial Clustering of Applications with Noise). Our method was developed for tabular data applications and tested with a petroleum engineering dataset. The proposed method is generalizable to other fields and applications such as data cleaning, financial fraud detection, early cancer diagnosis, bot activity detection on social media, etc.

2 Outlier Definition

Before introducing our proposed method for outlier detection, we briefly characterize and define what is meant by 'outlier'. An outlier is an observation in the dataset that constitutes an exception to the general trend observed in the data. Alternatively, an outlier is part of a distribution that differs from the majority of the observations in the dataset.

Although outliers are commonly characterized by extreme values or rare events, it must be noted that observations that are part of the primary distribution of the data can be 'extreme' in so far as they exist at the tails of the distribution. These observations would not necessarily constitute outliers.

Finally, it is poor practice to conduct an outlier analysis after fitting a model such that disconfirming observations within the dataset are excluded as 'outliers.' This ad-hoc definition of an outlier – an observation that fails to conform to the proposed statistical model – should also be rejected. Unless you are able to confirm that an outlier is a consequence of an erroneous measurement or other systemic errors, outliers are a part of the dataset. Instead, outliers should be investigated and understood by appealing to domain knowledge as they might give adequate inferential cues, and should not be removed for the dataset for being an outlier.

3 Data-set and Exploratory Data Analysis

The proposed outlier detection method was tested on a petroleum engineering dataset with 25 features. The outcome variable in this dataset is pump system failure type, while the predictor features include well properties such as well bore inclination, pump stroke length, and average oil volume production.

We selected the 17 numerical features in the dataset to conduct further analysis since we planned to implement our outlier detection method on numerical features. In Figure 1, we display the histograms we constructed of each of these predictor features. In Figure 2, we show the correlation matrix for the numerical features. Inspecting these visualizations, we ruled out the use of features that had little to no variation in value assignments and those that were highly correlated or redundant. Thus, we selected the top five numerical features that minimized missing values and correlation/anti-correlation between features: average tubing pressure (F), average casing pressure (G), average water volume (J), average oil volume (I), and average flowline pressure (E).

After we selected the five features to keep in the dataset, we constructed a spearman rank correlation matrix (Figure 3) and bivariate scatter plots for all the pairwise combinations (Figure 4). Next, we labeled potential outliers based on the bivariate scatter plot visualizations. These initial assessments are not reliable as they do not account for the full dimensionality of the data, however, it is a heuristic approach used in literature.

4 Hypothesis

As discussed above, data reduction is one way to combat poor performance of outlier detection methods in high dimensional datasets. Our hypothesis is that auto encoding prior to outlier detection will be a well-performing method for outlier detection in high dimensional data-sets

Autoencoders possess several properties that make them preferable to other methods of data reduction. Autoencoders do not assume linear relationships between features, make no distributional assumption, are flexible regarding data type, offer a method for decoding, and are unsupervised. Since we are working on the problem of outlier detection, it is particularly important that this method is unsupervised and does not require labeled data. It is uncommon for data to be labeled as outlier/inlier populations.

We also hypothesize that the use of DBSCAN in the latent space after encoding will be an effective method for outlier detection given it is an unsupervised clustering algorithm that is robust to noise. Current methods for outlier detection with autoencoders typically rely on MSE by virtue of a threshold to detect outliers, which we dubbed as 'AE-MSE'. Therefore, we propose the use of DBSCAN in the latent space for outlier identification instead.

5 Methodology

In this section, we outline our method of outlier detection using an autoencoder and DBSCAN on the latent space and our evaluation method.

5.1 Workflow for AE-DBSCAN

We utilize the following series of steps to perform AE-DBSCAN outlier detection:

1. Split dataset into train and test.

2. Train and tune the hyperparameters of the autoencoder using an appropriate loss function.

3. Using the trained AE, project all data into the latent space.

4. Tune the DBSCAN parameters.

5. Perform DBSCAN in the latent space using optimal paramters to identify outliers.

6. Set all clusters formed via density-reachability to cluster 1, denoted as inliers, and elsewhere outliers.

7. Use a decoder network to identify the detected outliers in the original space.

5.2 Evaluation

Our dataset lacks labels for the outlier/inlier populations. As such, we cannot use a straightforward accuracy analysis to evaluate the performance of our outlier detection method. To evaluate the performance of our method, we utilized two methods: we tested our method on a synthetic dataset that was corrupted with noise to simulate an outlier population, then we visually inspected the detected outlier points in a space of reduced dimensionality. We performed outlier detection with other commonly used methods such as IF, LOF, DBSCAN in high dimensional space to compare the performance on both the synthetic and real data. For the three comparative methods, PCA was used to assist visualization of the dataset since these methods are performed on the high dimensional data as-is

6 Results

6.1 Synthetic Data

We constructed a synthetic Gaussian dataset to test our method for outlier detection. Our synthetic dataset is composed of 3,000 observations with 4 features. We trained the autoencoder on the synthetic data. The visualization of the loss across 100 epochs indicated convergence (Figure 6).

We projected all the synthetic data into the latent space using the trained autoencoder (Figure 7).

To evaluate the performance of our autoencoder, we reconstructed our test set using a decoder and visualized the actual vs. predicted value for each feature (Figure 8). We found that the data reconstruction was satisfactory, although non-ideal.

Finally, we injected noise into our simulated dataset to replicate an outlier population and performed outlier detection on the synthetic data. We compared AE-DBSCAN outlier detection with the following other methods: AE-MSE, DBSCAN on high dimensional data, Isolation Forest on high dimensional data, and Local Outlier Factor method on high dimensional data.

We found that our proposed outlier detection method (AE-DBSCAN) performed better at identifying the noise population in the dataset than the use of an autoencoder with MSE threshold for outlier detection (AE-MSE). Additionally, the use of DBSCAN without data reduction was inconsistent in identifying the outlier populations, performing poorly on simulated data when the scatter of the outlier population resembled that of the inlier population. The other two methods we considered, Isolation Forest and Local Outlier Factor, both produced a high number of false positives.

6.2 Real Data

We applied the same workflow to the petroleum dataset, comparing the performance of AE-DBSCAN with the other methods, AE-MSE, IF, LOF, and DBSCAN in high dimensional space.

We trained the AE on the training set. In Figure 15 it can be observed that the loss function fails to converge for the test data. The real data projected to the latent space, seen in Figure 16, are linear. We again tested the performance of the decoder in reconstructing the high dimensional data (Figure 17) and found that the reconstructed data are noisy but roughly consistent with the actual data.

Similar to the synthetic dataset, we compared the performance of AE-DBSCAN and the other methods for outlier detection on the real dataset. By inspecting the visualization of the latent space, it can be seen that the AE-MSE method appears to pick a series of points clumped near the center of the dataset (Figure 19. Meanwhile, the AE-DBSCAN method marks a series of points on the fringe of the dataset as outliers, which is more in line with our expectations regarding data points that should be labeled outliers (Figure 20.

Finally, we used three other methods for outlier detection on the high dimensional data for comparison: DBSCAN without data reduction, Isolation Forest, and Local Outlier Factor method. We used Principal Component Analysis to visualize the results for inspection (Figure 21. Our results were consistent with those of the same process for the simulated data. The Isolation Forest and Local Outlier Factor method both produced an excess of false positives. DBSCAN without data reduction appeared to select outliers inconsistently and was more sensitive to location variations during outlier identification.

7 Conclusion, Limitations and Future Work

7.1 Conclusions

Based on our results, an autoencoder paired with DBSCAN in the latent space performed better than an autoencoder paired with an MSE threshold for outlier detection for synthetic and real data. In addition, the outlier detection methods we used for comparison (DBSCAN on the high dimensional space, Isolation Forest, and Local Outlier Factor method) performed poorly based on the visual inspection of the data in principal component space. It appears that AE-DBSCAN is the best-performing method on our synthetic and real data for outlier detection.

7.2 Limitations

Our analysis with synthetic data is limited by the contrived structure of the synthetic data. For example, we assumed an independent Guassian distribution over all our features. We replicated an outlier population using Guassian noise. The performance of our method on our synthetic data may not generalize to other distributions of inlier/outlier populations, more complex correlation structures in the data, or mixed distributions over the features in our dataset.

Our analysis with real data is limited by the amount of missing data in our dataset. We dealt with missing values by eliminating data points with missing values. We assumed that missing values are randomly distributed in our dataset.

Furthermore, our analysis with real data is limited by the lack of labels for the outlier/inlier populations. Without labels, we could not evaluate the accuracy of our detection method. Instead, we had to rely on more subjective methods of evaluation, like visualization and inspection.

7.3 Future Work

In future work, the performance of the AE-DBSCAN outlier detection method should be tested on non-guassian, mixed distributional simulated datasets with somewhat correlated features to mimic real datasets. Also, we would like to implement the use of a similarity matrix enabled decoder to find the detected outliers in the original space when the reconstruction error is poor. Lastly, we intend to verify using domain knowledge to determine if the detected points should be categorized as outliers to give some semblance of labeled data in order to evaluate the accuracy of the proposed method.

8 Acknowledgments

Midé: 100% Jose: 100% Stella: 100%

References

[1] Aggarwal, Charu and Philip S. Yu. 2001. Outlier detection for high dimensional data. SIGMOD Rec. 30, 2 (June 2001), 37–46.

[2] Breunig, Markus, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. SIGMOD Rec. 29, 2 (June 2000), 93–104.

[3] Cheng, Z, Wang, S, Zhang, P, Wang, S, Liu, X, Zhu, E. Improved autoencoder for unsupervised anomaly detection. Int J Intell Syst. 36, (2021), 7103-7125.

[4] Cheng, Zhangyu, Chengming Zou, and Jianwei Dong. 2019. Outlier detection using isolation forest and local outlier factor. In Proceedings of the Conference on Research in Adaptive and Convergent Systems (RACS '19). Association for Computing Machinery, New York, NY, USA, 161–168.

[5] Ghorbani, Hamid. "Mahalanobis Distance and Its Application for Detecting Multivariate Outliers." Ser. Math. Inform, 34, 3 (2019), 583-595.

[6] Jobe, J. Marcus, and Michael Pokojovy. "A Cluster-Based Outlier Detection Scheme for Multivariate Data." Journal of the American Statistical Association, 110, 512 (2015), 1543–51.

[7] Ro, Kwangil, et al. "Outlier Detection for High-Dimensional Data." Biometrika, 102, 3 (2015), 589–99.

[8] Sakurada, Mayu and Takehisa Yairi. 2014. Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. In Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis (MLSDA'14). Association for Computing Machinery, New York, NY, USA, 4–11.

[9] Stefatos, George and A. B. Hamza, "Cluster pca for outliers detection in high-dimensional data," 2007 IEEE International Conference on Systems, Man and Cybernetics, Montreal, QC, Canada, (2007), 3961-3966.

List of Figures

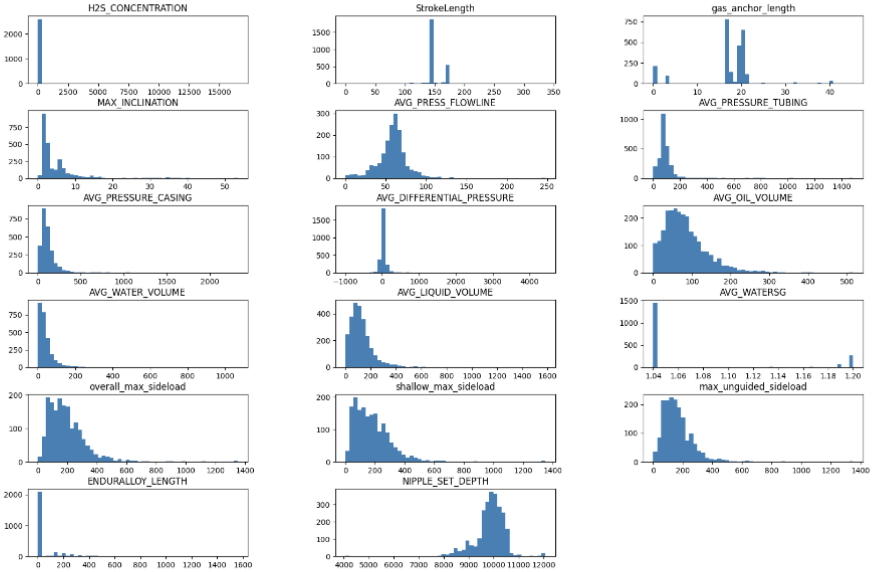

Figure 1: Histograms of numerical features.

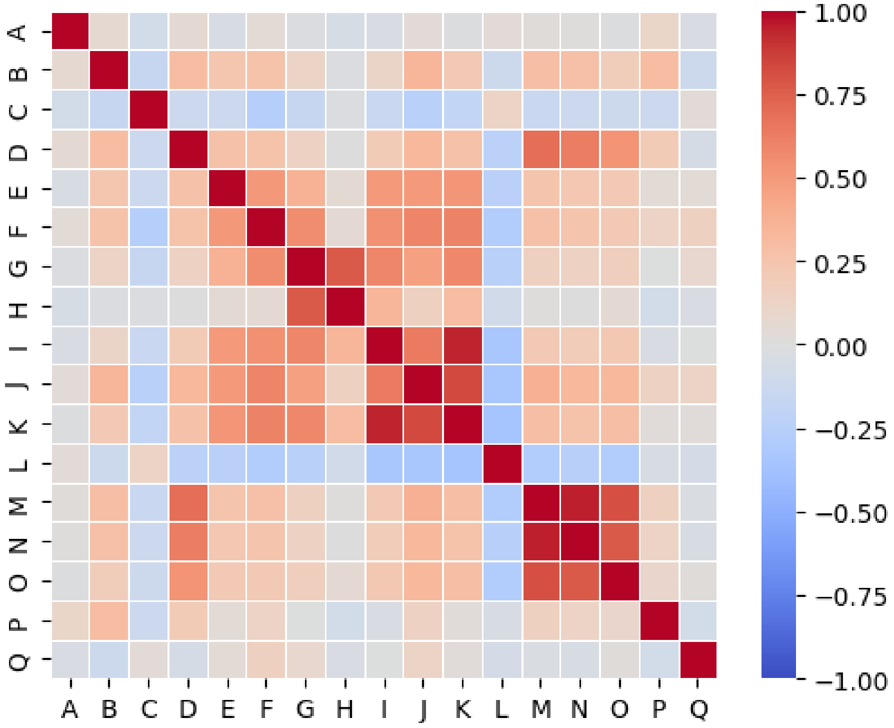

Figure 2: Correlation matrix for numerical features (feature names ommitted for legibility).

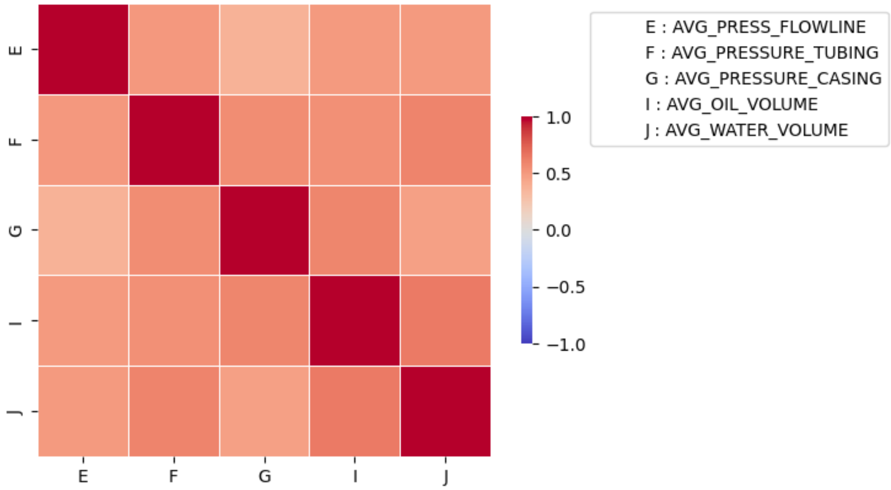

Figure 3: Correlation matrix for final features.

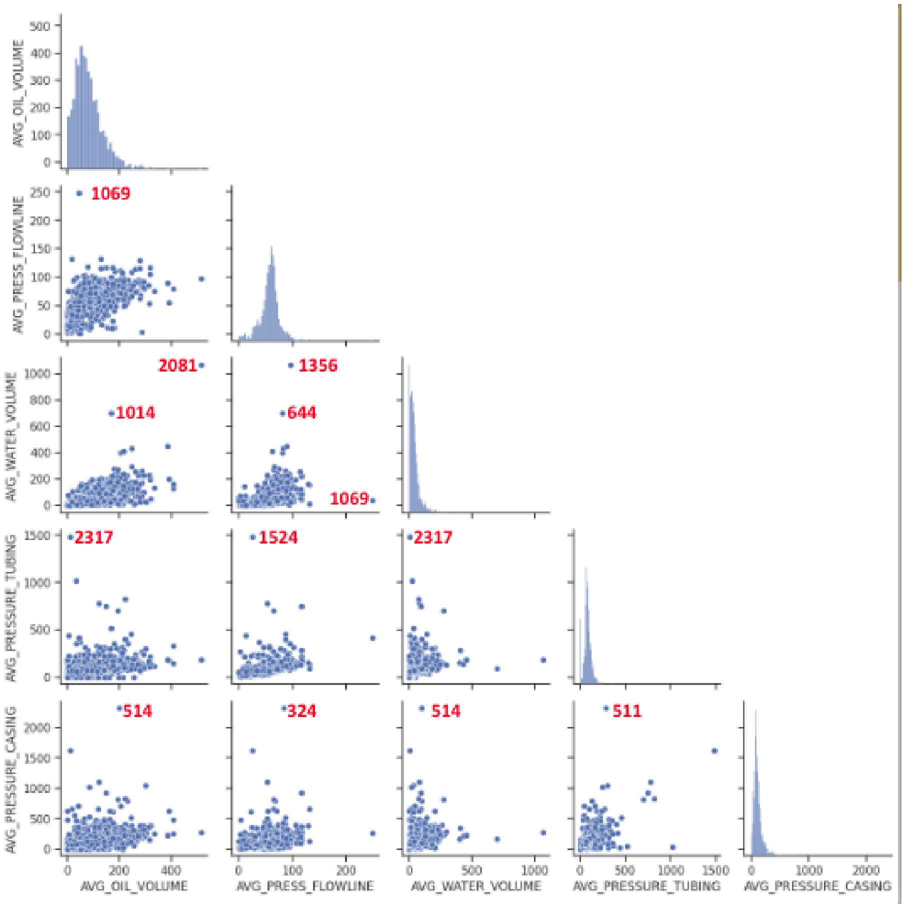

Figure 4: Histograms and bivariate scatter plots with potential outliers labeled.

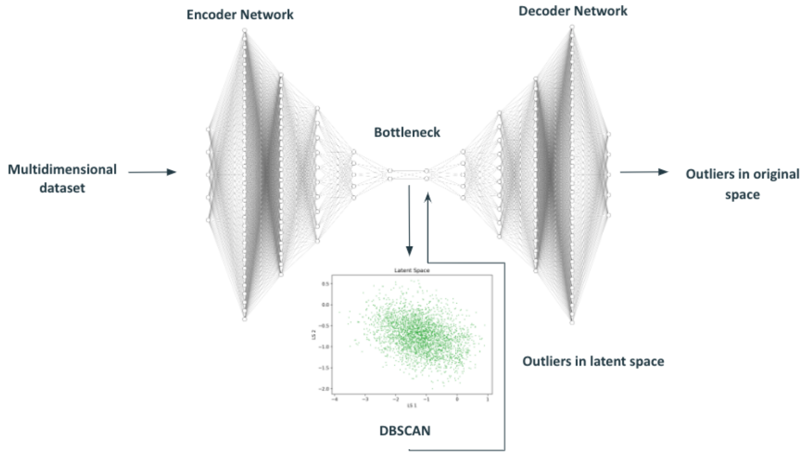

Figure 5: AE-DBSCAN for multidimensinal outlier detection.

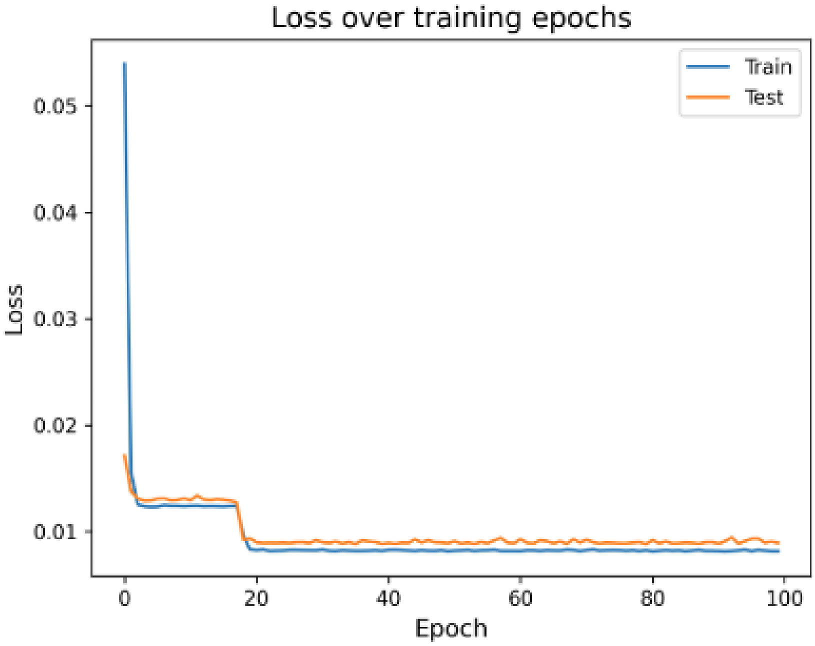

Figure 6: Training the AE on synthetic data.

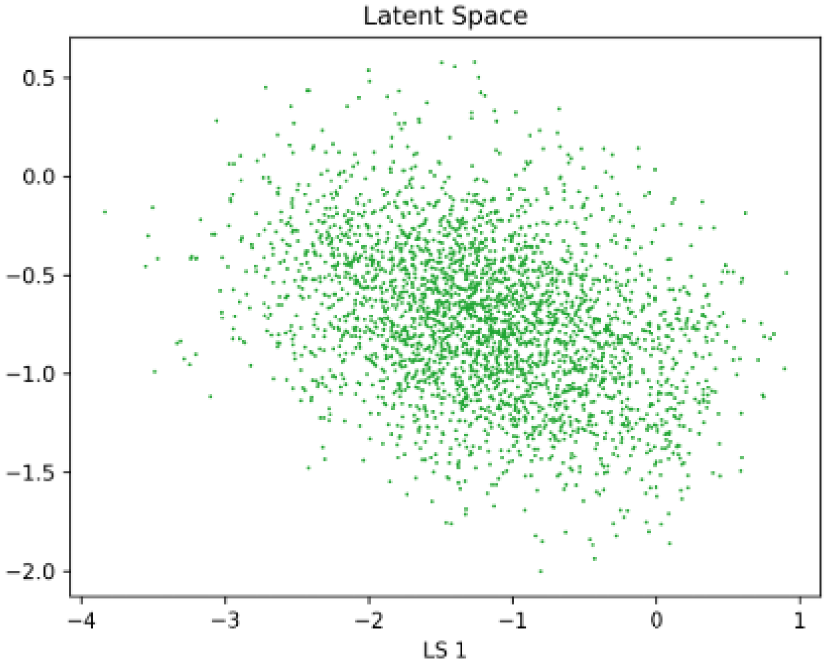

Figure 7: The synthetic data projected into the latent space.

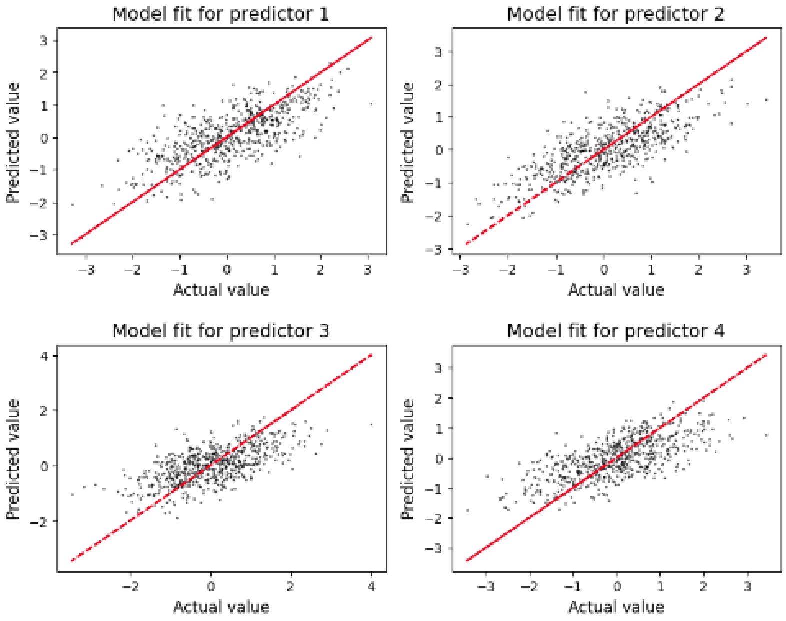

Figure 8: Evaluating the performance of the AE decoder in replicating the test set from the synthetic data.

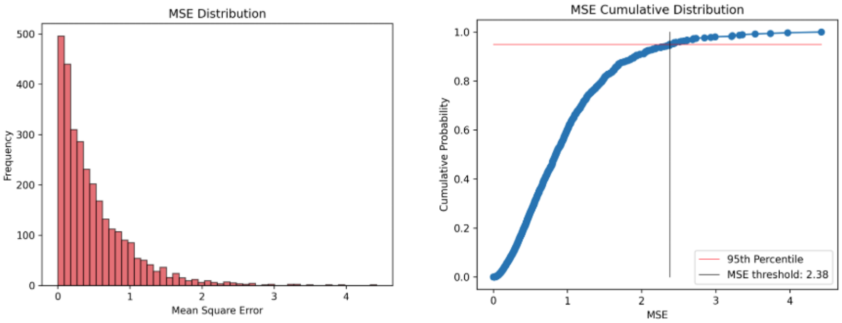

Figure 9: Finding the 95th percentile threshold using the distribution of the MSE.

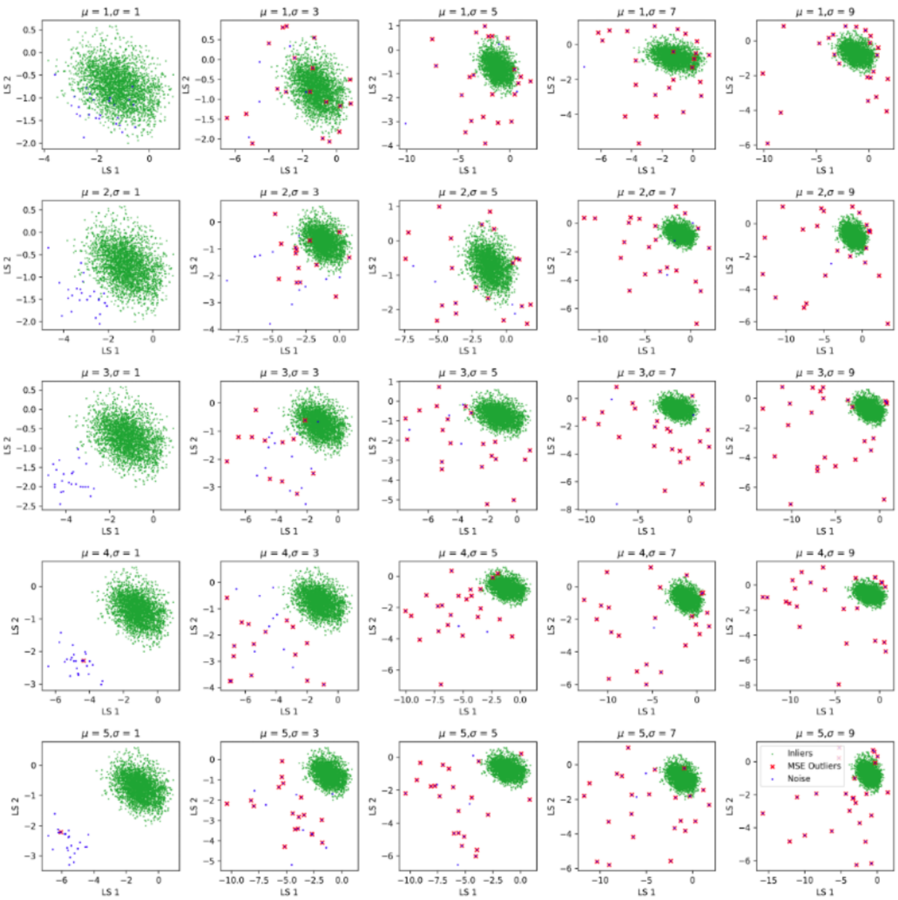

Figure 10: AE-MSE outlier detection on simulated data.

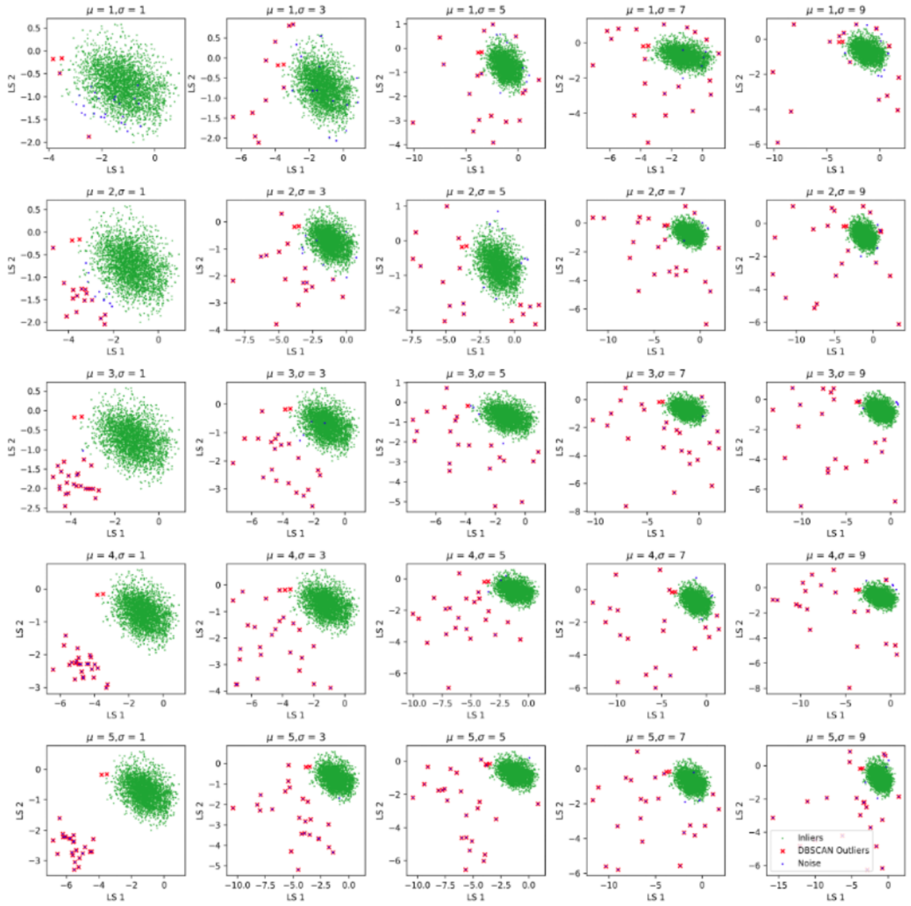

Figure 11: AE-DBSCAN outlier detection on simulated data.

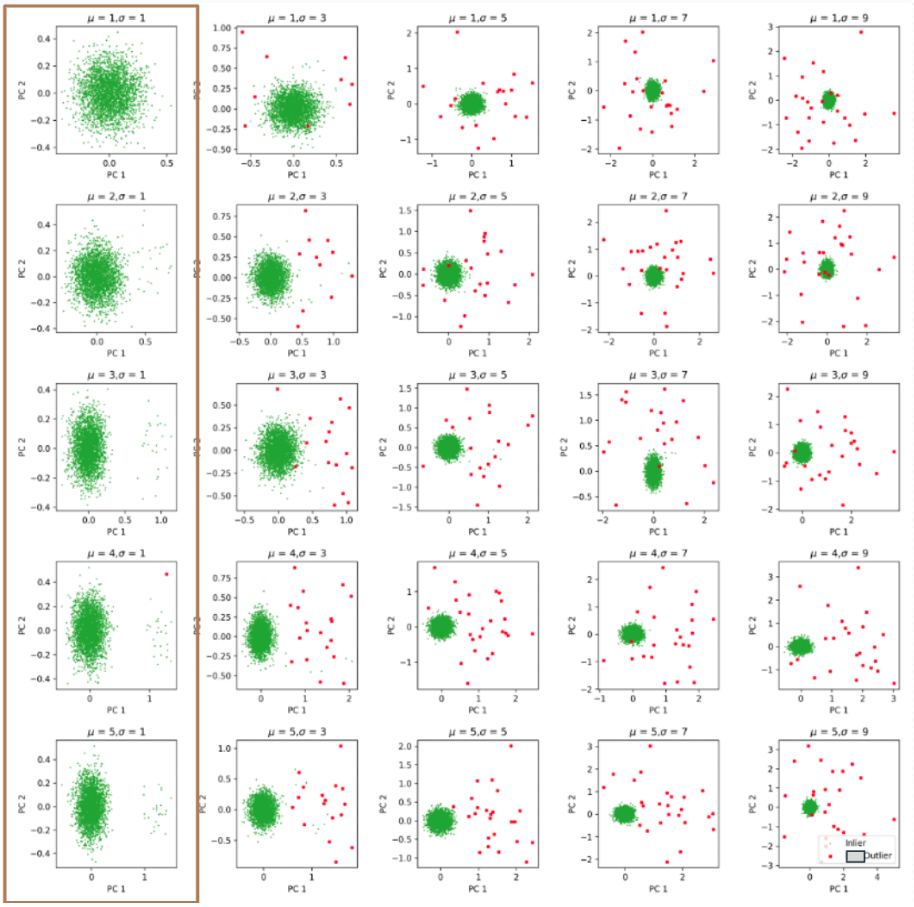

Figure 12: DBSCAN outlier detection on simulated data, no data reduction.

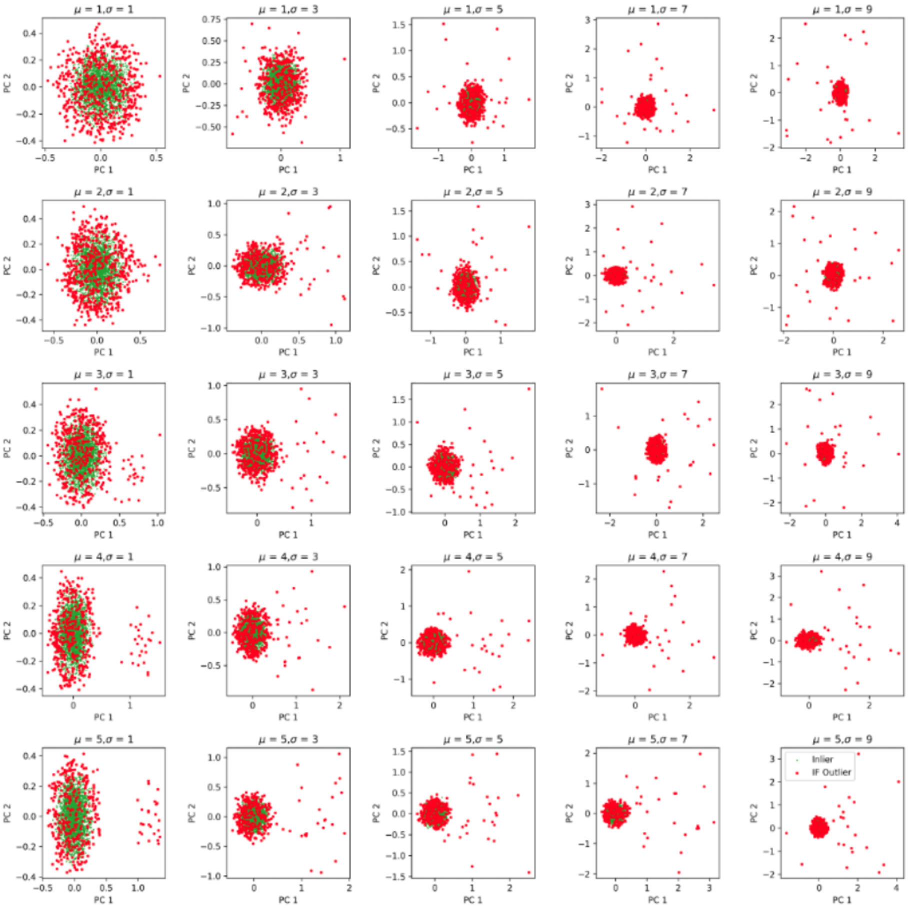

Figure 13: Isolation forest outlier detection on simulated data, no data reduction.

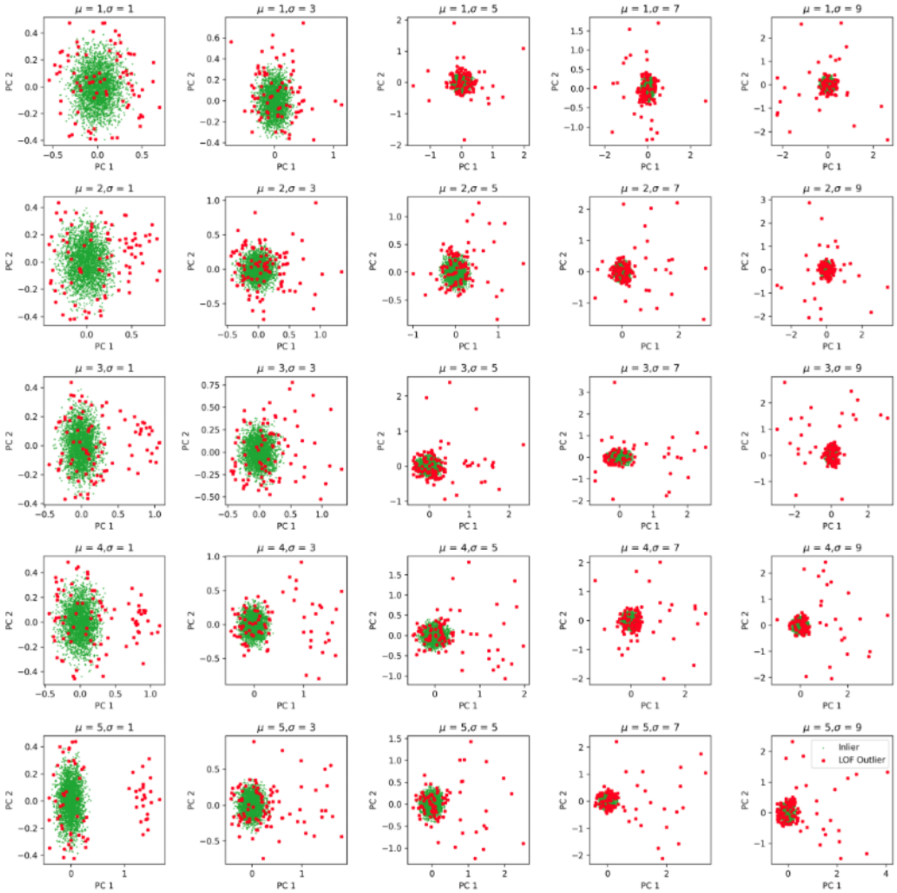

Figure 14: Local outlier factor method outlier detection on simulated data, no data reduction.

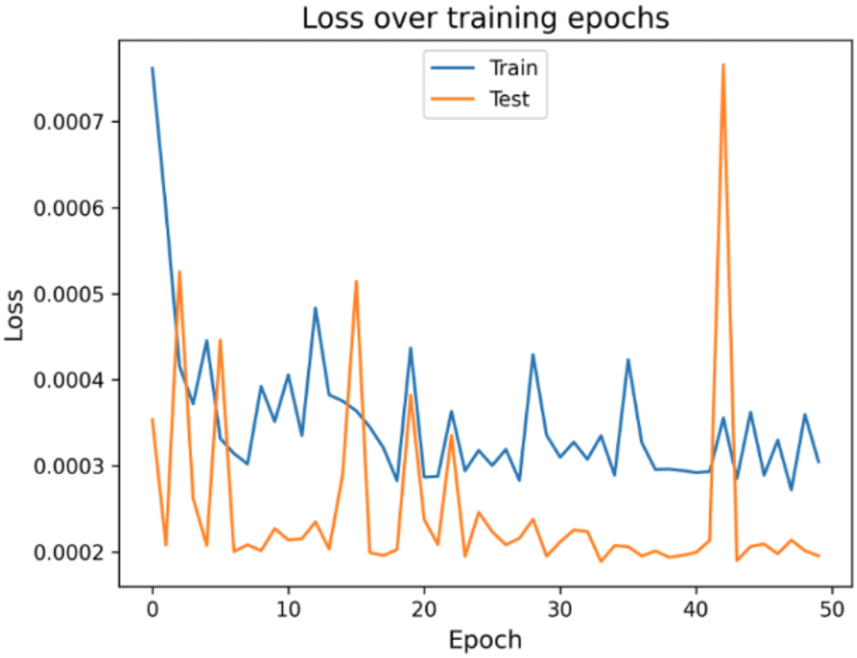

Figure 15: Training the AE on real data.

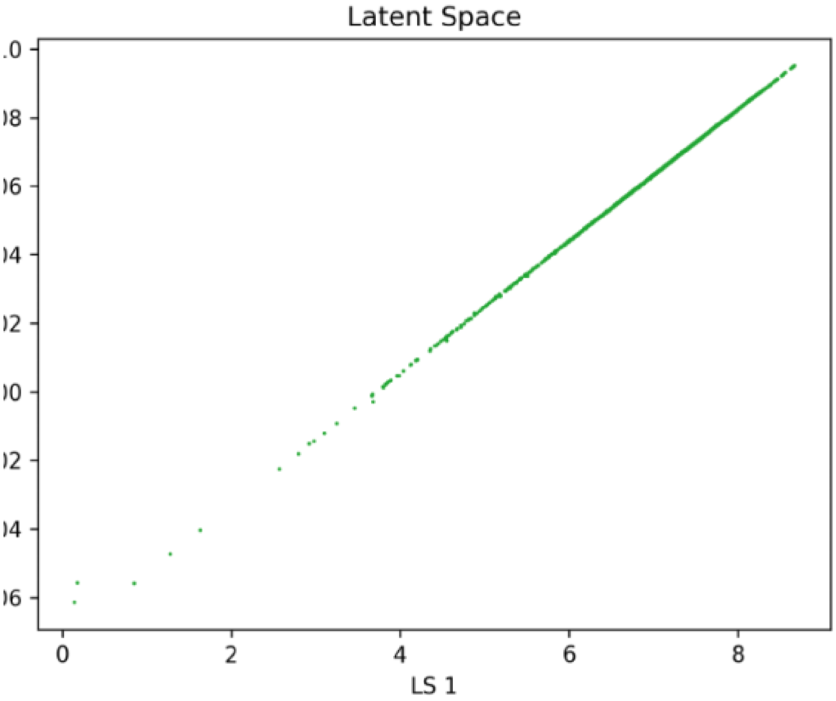

Figure 16: The real data projected into the latent space.

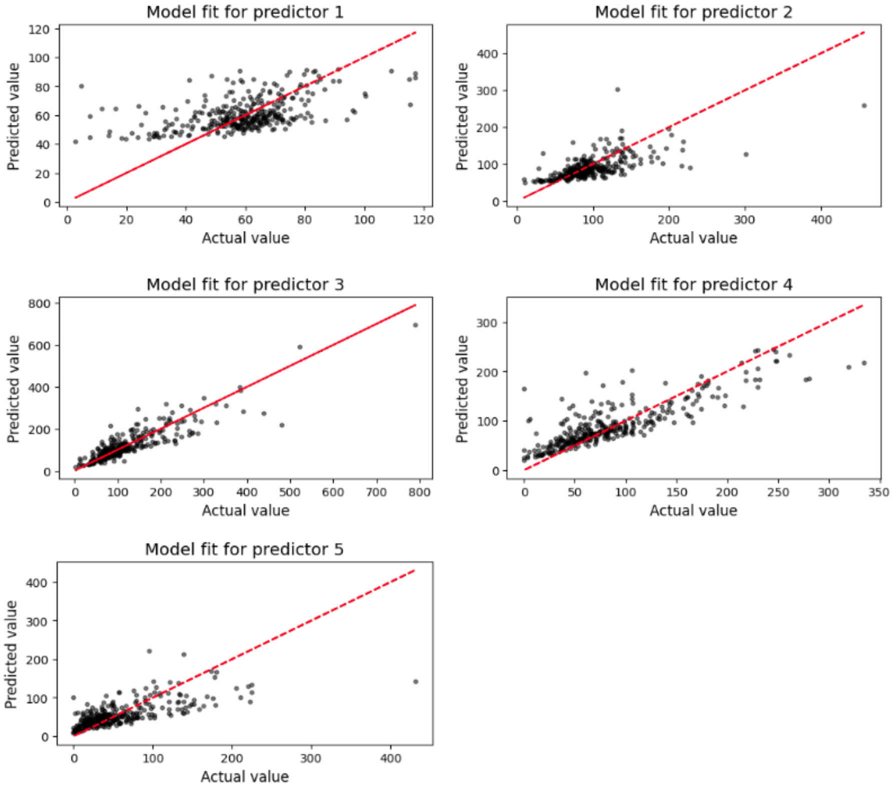

Figure 17: Evaluating the performance of the AE decoder in replicating the test set from the synthetic data.

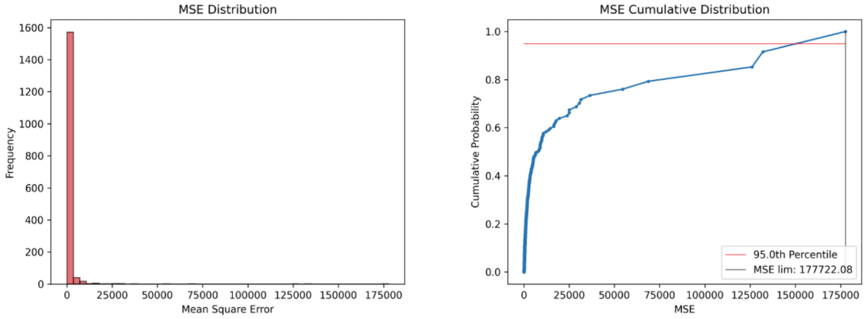

Figure 18: Finding the 95th percentile threshold using the distribution of the MSE.

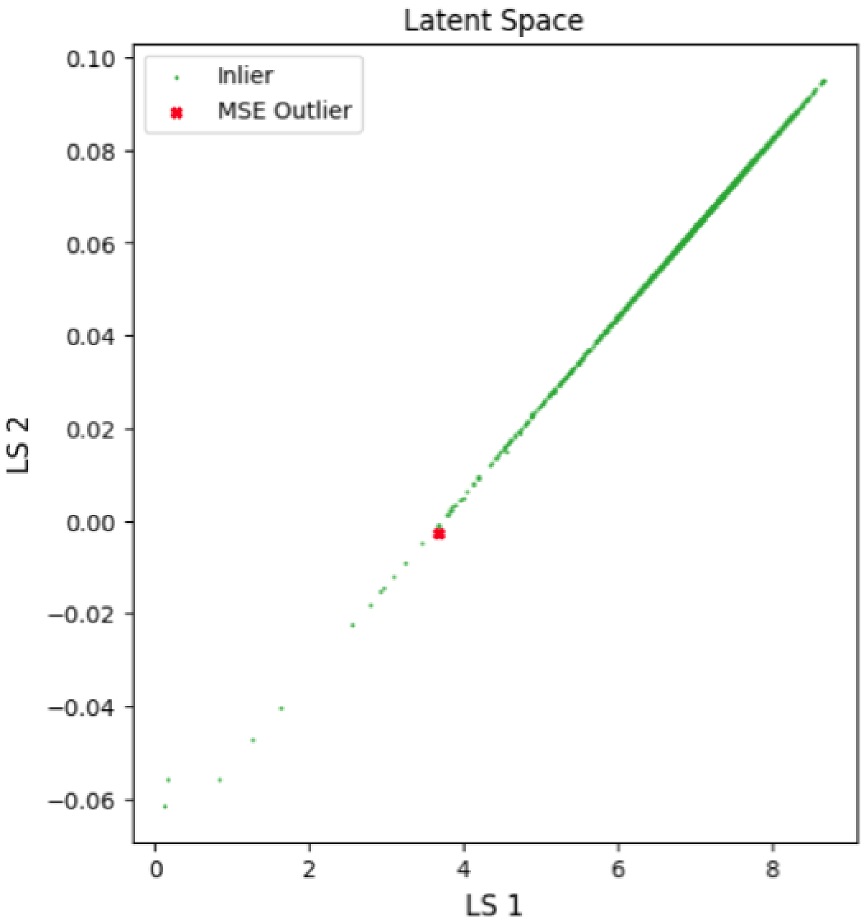

Figure 19: AE-MSE outlier detection on real data visualized in the latent space.

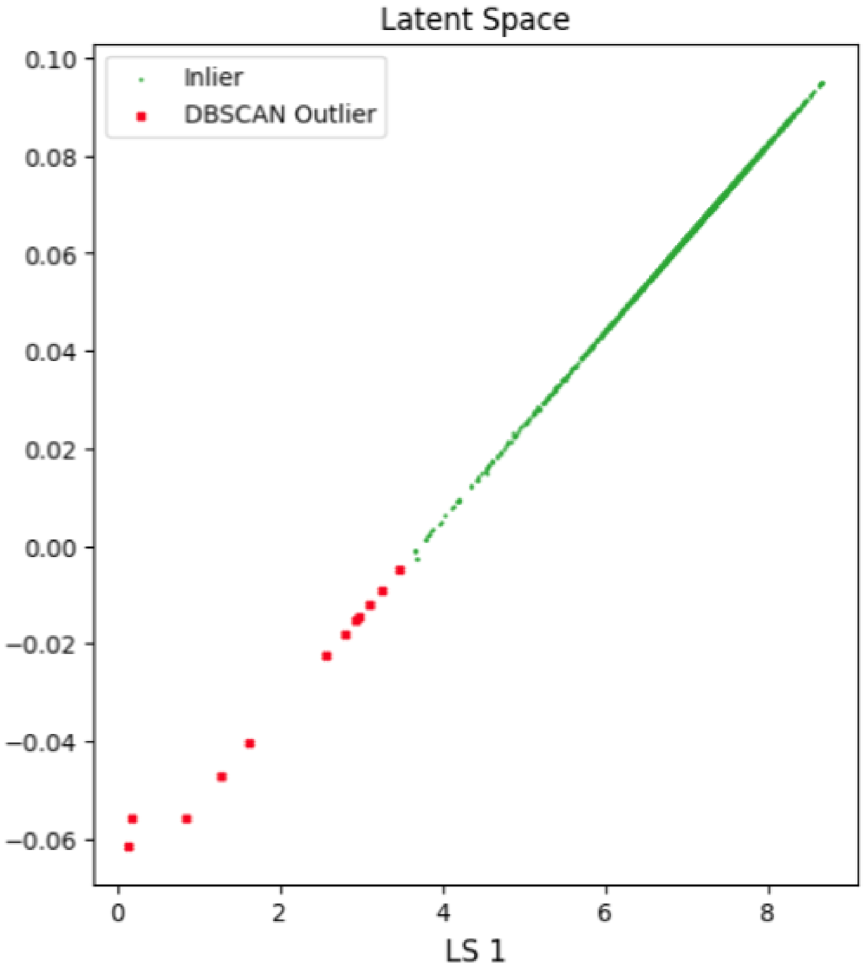

Figure 20: AE-DBSCAN outlier detection on real data visualized in the latent space.

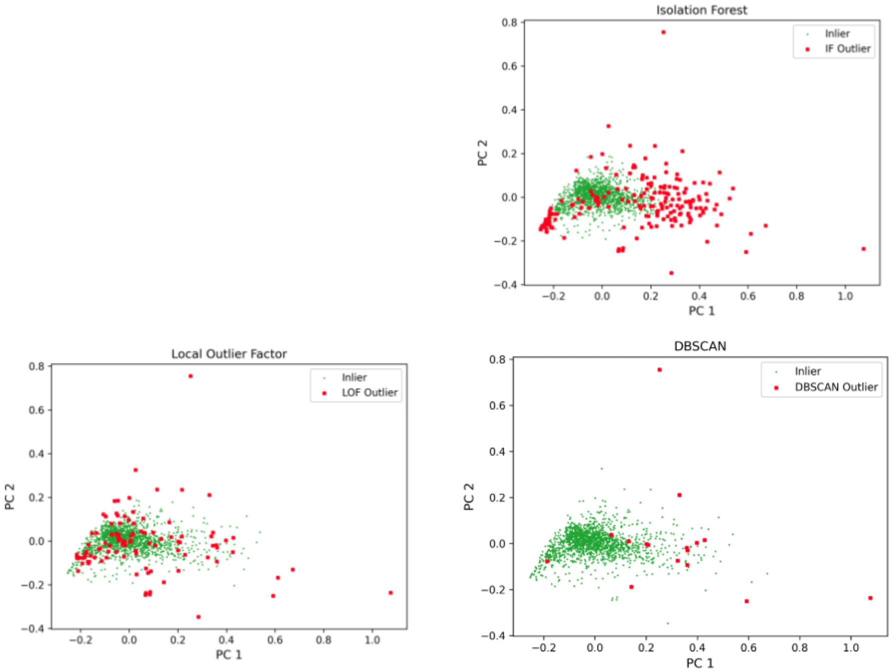

Figure 21: Outlier detection on real data with other methods for comparison, visualized in principal component space.