Available online at http://scik.org
Commun. Math. Biol. Neurosci. 2026, 2026:9
https://doi.org/10.28919/cmbn/9659
ISSN: 2052-2541

OUTLIER DETECTION IN MULTIVARIATE TIME SERIES: AN APPLICATION OF HYBRID DNN-DBSCAN TECHNIQUE

THEOPHILUS ASAMOAH[1,*], ANTHONY GICHUHI WAITITU[2], BISMARK KWAO NKANSAH[3], CYPRIAN OMARI[4]

[1]Department of Mathematics, Institute for Basic Sciences, Technology and Innovation, Pan African University, Juja, Kenya

[2]Department of Statistics and Actuarial Sciences, Jomo Kenyatta University of Agriculture and Technology, Juja, Kenya

[3]Department of Statistics, University of Cape Coast, Cape Coast, Ghana

[4]Department of Statistics and Actuarial Science, Dedan Kimathi University of Technology, Nyeri, Kenya

Abstract. In spite of on-going advances and utilization of Deep Neural Networks (DNN) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) techniques to enhance outlier detection in multivariate time series (MTS) data, research is yet to explore an approach that integrates the capabilities of the two techniques for complex data representation. The paper therefore combines the two techniques to obtain the hybrid DNN-DBSCAN technique. It is demonstrated in simulated data that the resulting technique achieves improved precision and recall of outlier detection based on a number of performance metrics. In particular, the new procedure adequately captures the complexity involved with the underlying high-dimensionality of MTS data which poses problems for outlier detection to traditional methods.

Keywords: DBSCAN; multivariate outlier detection; deep neural network.

2020 AMS Subject Classification: 62H30, 68T07, 68T10.

[*]Corresponding author

E-mail addresses: theasamoah36@gmail.com, theophilus.asamoah@students.jkuat.ac.ke

Received October 23, 2025

1. Introduction

Outlier detection in MTS data has garnered considerable interest owing to the critical need for precise outlier detection in many applications, including banking, healthcare, and industrial monitoring [1]. Outliers, characterized as data points that markedly diverge from expected trends, may result from measurement inaccuracies, emerging patterns, or infrequent occurrences. Their detection in MTS is notably difficult because of interdependency across variables and temporal points. Inaccurate detection of outliers may result in significant adverse effects [2]. This gap has prompted the development of techniques to improve outlier detection. Current methods for detecting outliers in time series are mostly categorized into statistical, machine learning, and deep learning. Statistical techniques including principal component analysis, autoregressive, and moving average models, although proficient in univariate scenarios, often exhibit diminished performance with high-dimensional data due to inadequacy in capturing non-linear patterns and temporal dependency characteristics of MTS data [3]. Machine learning techniques such as k-means and support vector machines have been used to mitigate some of these constraints. Nonetheless, they encounter issues related to scalability and accuracy [3].

Recently, deep learning models, those using Recurrent Neural Networks and Convolutional Neural Networks have been investigated to describe temporal relationships and high-dimensional data structures [4]. Although these models exhibit proficiency in capturing complex patterns, they often encounter challenges with model interpretability, scalability, and tendency to overfit, which may restrict their effectiveness in practical applications [5]. The DBSCAN technique, a prevalent clustering method, has shown efficacy in detecting outliers based on density rather than relying only on Euclidean distance, making it advantageous for data exhibiting variable shapes and heterogeneous densities [6]. However, the direct implementation of the DBSCAN technique to MTS data often proves inadequate since the high-dimensional characteristics of these data hinder precise distance-based clustering. Although this technique effectively detects noise (outliers), it is insufficient in dealing with complex temporal connections and feature interactions independently. So, existing outlier detection techniques may either overlook outliers or mistakenly classify normal patterns as outliers in MTS data.

The integration of DNN modeling and DBSCAN technique could offer a viable solution to the constraints inherent in each technique when used alone in detection of outliers. The DNN modeling component provides a feature extraction method that can detect complex, nonlinear correlations among variables, converting high-dimensional data into a more manageable lower-dimensional feature space for the DBSCAN technique processing [7, 8, 9]. After data transformation through complex features extraction, DBSCAN technique detects density-based outliers, facilitating robust detection of outliers that are less sensitive to high-dimensional noise [10]. This hybrid technique utilizes the advantages of DNN modeling for feature extraction and the effectiveness of the outlier detection abilities of the DBSCAN technique to provide a solution that is both scalable and interpretable. The application of the hybrid DNN-DBSCAN technique is important in fields characterized by high-dimensional, interrelated time series data, where precise detection of outliers is essential for informed decision-making. Therefore, this paper presents a novel strategy that integrates the ability of DNN modeling to describe complex data structures with robustness of traditional DBSCAN technique to enhance outlier detection in MTS. The paper is guided by the following objectives. To:

(a) Develop a hybrid DNN-DBSCAN technique to detect outliers in MTS data.

(b) Evaluate and compare the performance of the hybrid DNN-DBSCAN technique developed with existing outlier detection techniques.

The rest of the paper is organized as follows: Section 2 presents the procedures for incorporating the DNN and DBSCAN techniques to detect outliers and its performance measures. Also, Section 3 presents the main results while Section 4 presents the conclusion of the paper.

2. Methods

Detecting outliers by combining DNN modeling with DBSCAN technique entails extracting features of the MTS data using the DNN modeling technique and clustering the feature extracted data using DBSCAN technique as outliers or non-outliers. Starting with DNN technique, followed by DBSCAN technique, this section provides integration of the two to obtain the hybrid DNN-DBSCAN technique.

2.1. DNN Modeling Technique

Assuming Y_t represents an MTS data given by

$$Y_t = \begin{pmatrix} y_{11} & y_{21} & \cdots & y_{p1} \\ y_{12} & y_{22} & \cdots & y_{p2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1T} & y_{2T} & \cdots & y_{pT} \end{pmatrix} \tag{1}$$

where each y_{it} is an observation for the ith variable at time t, $t = 1,2,3,\dots,T$, and p is number of variables. Normalizing Equation (1) gives

$$X_t = (Y_t - \mathbf{1}_T \mu_Y^{'}) S_Y^{-1} \tag{2}$$

where X_t is normalized MTS data, Y_t is initial MTS data, μ_Y is mean vector, S_Y is diagonalized vector, σ_Y of standard deviations, and finally, $\mathbf{1}_T$ is a vector of ones. The DNN model is composed of multiple layers, where each layer performs a transformation on the input. The output of the previous layer is the input for the next, as exemplified in Figure 1. The DNN model has: input, hidden, and output layers. In the last layer, the output is compared to the true value and error criterion is applied to compute the loss. The Huber loss is applied. The trained DNN model is defined by its parameters: weight matrices, $W_{j,j-1}$ and bias vectors, b_j, with j from 1 to the number of desired hidden layers. The parameters are adjusted iteratively to minimize the loss, with the application of the Deterministic Finite Gradient Search. Given a MTS data, each hidden layer applies a non-linear transformation function g to the output of the previous layer.

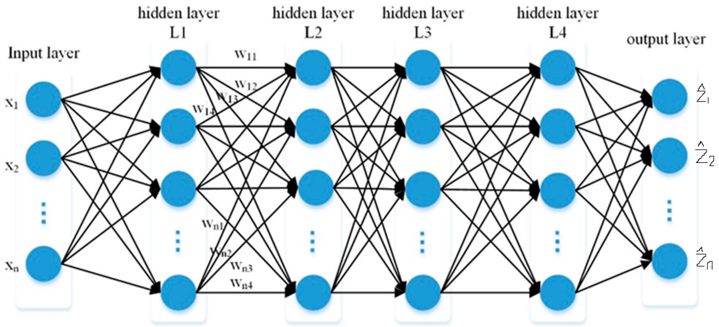


FIGURE 1. DNN Model Architecture

The transformation takes into accounts the parameters, linking a layer to its previous layer, and providing the activation values of neurons with

$$h_j(x^{(i)}) = g(W_{j,j-1}h_{j-1}(x^{(i)}) + b_j) \tag{3}$$

where *j = 2, . . . , T-1* and

$$h_1(x^{(i)}) = g(W_{1,0}x^{(i)} + b_1) \tag{4}$$

Hence, the DNN model architecture is of the form

$$Z = g\left(W_n\left(g\left(W_{n-1}\left(\cdots g\left(W_1X_t + b_1\right)\cdots\right) + b_{n-1}\right) + b_n\right)\right) \tag{5}$$

where Z is feature representation of the input. The ReLU activation function is applied because it remains the default choice in many applications. The ReLU is used due to its simplicity and computational efficiency, alleviation of vanishing gradient problem, sparsity in neural networks, better performance practically, biological inspiration and its variants and flexibility [11]. The ReLU is

$$\text{ReLU}(z) = \max(0, z) \tag{6}$$

This activation function introduces non-linearity, which allows the network to learn complex patterns. The Huber loss is applied for robust outlier detection, combining the advantages of the squared error for small errors and absolute error for large errors, defined as

$$L_\delta(r) = \begin{cases} \frac{1}{2}r^2, & \text{for } |r| \leq \delta \\ \delta(|r| - \frac{1}{2}\delta), & \text{otherwise} \end{cases} \tag{7}$$

where, $r = Z - \hat{Z}$ is residuals and δ is a threshold parameter that determines the switch between quadratic and linear behaviours. The threshold parameter helps in mitigating the impact of outliers, treating them with less sensitivity compared to the traditional squared losses. Particularly, the auto-encoder technique is applied as presented in Section 2.1.1.

2.1.1. AE Technique

The Auto-Encoder (AE) compresses and reconstructs non-linear feature representations of data, reducing noise while preserving informative structures relevant to outlier detection [12]. Given Equation (2), the encoder maps it to a latent vector $\mathbf{Z}$ defined as

$$\hat{\mathbf{Z}} = f_{\text{enc}}(\mathbf{H}) = g(\mathbf{W}_e\mathbf{H} + \mathbf{b}_e) \tag{8}$$

and the decoder reconstructs the input as

$$\hat{\mathbf{H}} = f_{\text{dec}}(\mathbf{Z}) = g(\mathbf{W}_d\mathbf{Z} + \mathbf{b}_d) \tag{9}$$

Unlike traditional AEs that use mean squared error, the *Huber loss* is adopted to enhance robustness against outliers or noisy observations, formulated as

$$L_{AE}^{\text{Huber}} = \frac{1}{N}\sum_{i=1}^{N} \begin{cases} \frac{1}{2}\|\mathbf{H}_i - \hat{\mathbf{H}}_i\|_2^2, & \text{if } \|\mathbf{H}_i - \hat{\mathbf{H}}_i\|_1 \leq \delta, \\ \delta\left(\|\mathbf{H}_i - \hat{\mathbf{H}}_i\|_1 - \frac{\delta}{2}\right), & \text{otherwise.} \end{cases} \tag{10}$$

where δ controls the transition between the quadratic and linear regimes of the loss. The per-window reconstruction error, serving as the outlier score, is defined as

$$\varepsilon_i = \begin{cases} \frac{1}{2}\|\mathbf{H}_i - \hat{\mathbf{H}}_i\|_2^2, & \text{if } \|\mathbf{H}_i - \hat{\mathbf{H}}_i\|_1 \leq \delta, \\ \delta\left(\|\mathbf{H}_i - \hat{\mathbf{H}}_i\|_1 - \frac{\delta}{2}\right), & \text{otherwise} \end{cases} \tag{11}$$

Finally, $\hat{\mathbf{Z}}$ are clustered using the DBSCAN algorithm for outlier detection as presented in Section 2.2

2.2. DBSCAN Technique

The results of Equation (8) are clustered to detect outliers using DBSCAN [6]. The DBSCAN detects its neighbourhood $N_\varepsilon(\hat{z}_t)$ within a radius ε and further determines if it is a core point, border point or an outlier. The DBSCAN application procedure is carried out by first determining two parameters, ε and *MinPts*. The ε is maximum distance between two points to consider them as neighbours, whereas *MinPts* is minimum number of points required to form a cluster, for a point to be a core, border point, or an outlier. The desired ε is determined using k-distance graph. This is an effective technique used in estimating the ideal ε for clustering. Thus, the k-graph enhances the efficacy of the DBSCAN, facilitating accurate parameter selection and superior clustering quality [13]. Finally, it helps in obtaining the best parameters to detect sufficient number of outliers. In the graph, mostly, the best ε for fine-tuning to obtain desired results is selecting a value around the elbow point. Furthermore, *MinPts* is determined based on dimensionality of the data, defined as

$$MinPts \geq 2(p+1) \tag{12}$$

After determining ε and *MinPts*, clusters are formed and points not belonging to any of the clusters are referred to as noise (outliers). The procedure is as follows

(1) Calculate the Euclidean distance between two points $\hat{z}_i$ and $\hat{z}_j$ in a p-dimensional space as:

$$dist(\hat{z}_i, \hat{z}_j) = \sqrt{\sum_{i,j=1}^{p} (\hat{z}_i - \hat{z}j)^2} \tag{13}$$

where $i,\ j$ = 1, 2, 3, ..., p but $i \neq j$, $dist(\hat{z}_i,\ \hat{z}_j)$ is the Euclidean distance between points at time i and time j, $\hat{z}_i$ is the value of the i^{th} feature at t, $\hat{z}_j$ is the value of the j^{th} feature at t, and p is the dimensional space.

(2) A point M is core point if the number of points within the distance ε of M is at least *MinPts*. M is a single component categorized as noise. To identify core points, further define

$$\text{Core Point} = \begin{cases} \text{True}, & \text{if } |\{\hat{z} \in D : \text{dist}(\hat{z}_i, \hat{z}_j) \leq \varepsilon\}| \geq MinPts \\ \text{False}, & \text{otherwise} \end{cases} \tag{14}$$

(3) A point M is a border point if it is not a core point but it is in the neighbourhood of a core point.

(4) Finally, in detecting outliers, the DBSCAN technique forms clusters by

   (i) For each core point not assigned to a cluster, a new cluster is created, and the core points and all points within ε are added. The cluster is expanded by adding all reachable points within ε.

   (ii) Clusters are denoted as C_k, where, k is the number of clusters formed.

(iii) For each cluster C_k, the centroid, μ_k, is estimated as the mean of the feature vectors of the points in the cluster formed.

(iv) The distance, $dist(M, \mu_k)$ between each point M and the centroid, μ_k, of its cluster C_k is estimated.

(v) Finally, a threshold, τ, is defined to classify a point as outlier if;

$$dist(M, \mu_k) > \tau \tag{15}$$

2.3. Hybrid DNN-DBSCAN Technique

The combination of the DNN model and DBSCAN technique leverage the strengths of both methods. In the DNN-DBSCAN framework, the DNN model extracted robust features of the data, encoding the data in a low-dimensional space where clusters and outliers are more easily distinguishable [14]. The encoded features are passed to the DBSCAN technique, which clusters and isolates outliers based on density features [15]. By combining DNN modeling technique with DBSCAN technique to detect outliers, improvement in precision and recall of outlier detection is achieved. Some performance metrics of the hybrid technique are presented in Section 2.4.

2.4. Performance Metrics of Hybrid DNN-DBSCAN Technique

The following accuracy measures are used: Silhouette Coefficient [16], Davies-Bouldin Index [17], Calinski-Harabasz Index [18], Outlier Point Ratio [13], Precision, Recall and F1-Score [19, 20, 21], qualitative validation techniques with Principal Component Analysis and heat map, and finally, stability under parameter variation [13, 22, 23]. These are used to evaluate quality of clustering:

2.4.1. Silhouette Coefficient

The Silhouette Coefficient ($S(i)$) measures the similarity of a data point to its given cluster in relation to others [16]. The formula for $S(i)$ for a data point i is defined as

$$S(i) = \left[\frac{b(i) - a(i)}{\max(a(i), b(i))}\right] \tag{16}$$

where $a(i)$ is the average distance between i and all points in the same cluster (intra-cluster distance or cohesion) and $b(i)$ is the (inter-cluster separation). $S(i)$ ranges from -1 to 1, with 1 indicating perfect clustering, 0 suggesting overlapping clusters, and negative indicating wrong clustering.

2.4.2. Davies-Bouldin Index

The Davies-Bouldin Index (DBI) evaluates the ratio of within-cluster dispersion to the separation between clusters [17]. It is estimated as

$$DBI = \left[\frac{1}{k}\left(\sum_{i=1}^{k} \max_{i \neq j} \frac{s_i + s_j}{d_{ij}}\right)\right] \tag{17}$$

where s_i is the average distance of points in cluster i from its centroid, and d_{ij} is the distance between centroids of clusters i and j. DBI ranges from 0 to ∞ with lower values (preferably, at most 2) indicating better clustering. Zero is ideal, representing perfectly separated and cohesive cluster formation.

2.4.3. Calinski-Harabasz Index

The Calinski-Harabasz Index (CHI), also called the variance ratio criterion, measures ratio of between-cluster dispersion to within-cluster dispersion [18], defined as

$$CHI = \left[\left(\frac{\mathrm{Tr}(B_k)}{\mathrm{Tr}(W_k)}\right) \times \left(\frac{N-k}{k-1}\right)\right] \tag{18}$$

where $\mathrm{Tr}(B_k)$ is between-cluster scatter matrix trace, $\mathrm{Tr}(W_k)$ is within-cluster scatter matrix trace, N is number of data points and k is number of clusters. Higher values above 100 indicate better-defined clusters. A value greater than or equal to 100 indicates strong clustering performance, between 50 and 100 suggests an acceptable or moderate clustering performance, and a value less than or equal to 50 indicates poor clustering performance with significant cluster overlaps.

2.4.4. Outlier Point Ratio (DBSCAN-specific)

This refers to the percentage of data points the DNN-DBSCAN and traditional DBSCAN techniques detect as outliers in relation to the entire dataset. The conservativeness of the clustering is measured by this DBSCAN-specific metric [6]. This ratio aids in evaluating the algorithm's aggressiveness in detecting outliers [13]. Values too high may indicate oversensitivity, and too low may miss real outliers. The estimation formula is

$$\textit{Noise Point Ratio} = \left(\frac{\textit{Number of points labeled as outliers [-1]}}{\textit{Total number of points}}\right) \tag{19}$$

2.4.5. Precision, Recall and F1-Score

Precision is the ratio of correctly detected outliers to all points detected as outliers [21]. It is estimated as

$$\textit{Precision} = \left(\frac{TP}{TP+FP}\right) \tag{20}$$

where, *TP* is *True Positives*, data points correctly detected as outliers and *FN* is *False Positives* which are data points incorrectly flagged as outliers. Furthermore, *Recall* refers to the ratio of correctly detected outliers to all actual outliers. It measures the completeness of outlier detection [20]. It is applied in situations where missing outliers have severe consequences. It is estimated as

$$\textit{Recall} = \left(\frac{TP}{TP+FN}\right) \tag{21}$$

where, *FN* represents *False Negatives* which are true outliers missed by the model. The *F1-Score* is the harmonic mean of precision and recall which provides a value that balances precision and recall, and particularly useful with imbalanced data (common in outlier detection). It is preferred over accuracy when class distribution is skewed, such as containing outliers [19], estimated as

$$\textit{F1-Score} = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right) \tag{22}$$

2.4.6. Qualitative Validation Techiques

Performance measures such as the heat map, which shows the deviation of detected outliers from normal patterns, and plot of clusters with Principal Component Analysis (PCA) for dimensionality reduction are used to shed light on the effectiveness of the hybrid technique [13, 22, 23].

2.4.7. Stability Under Parameter Variation

This refers to the consistency of results when parameters (ε and MinPts) are varied within reasonable bounds. It estimates the variance in performance measures across parameter combinations, and critical for DBSCAN as it is sensitive to parameter choices [22]. The stability indicates the robustness of the technique [23]. This is also particularly important in unsupervised settings where parameters may not be easily tuned [13], as in the current study.

3. Main Results

The summary results and discussion are presented in this section. Observations on five variables are simulated from normal distribution. The aim is to check the ability of the hybrid DNN-DBSCAN technique to detect outliers. There are 5 variables, $(X_1, X_2, X_3, X_4, X_5)$, with 61,440 observations generated. Each variable is assumed to be normally distributed. (multivariate normal distribution). A random seed of 42 is set to ensure reproducibility. Therefore, the generated matrix has a shape of (61440 by 5). The data is arranged into a MTS format such that

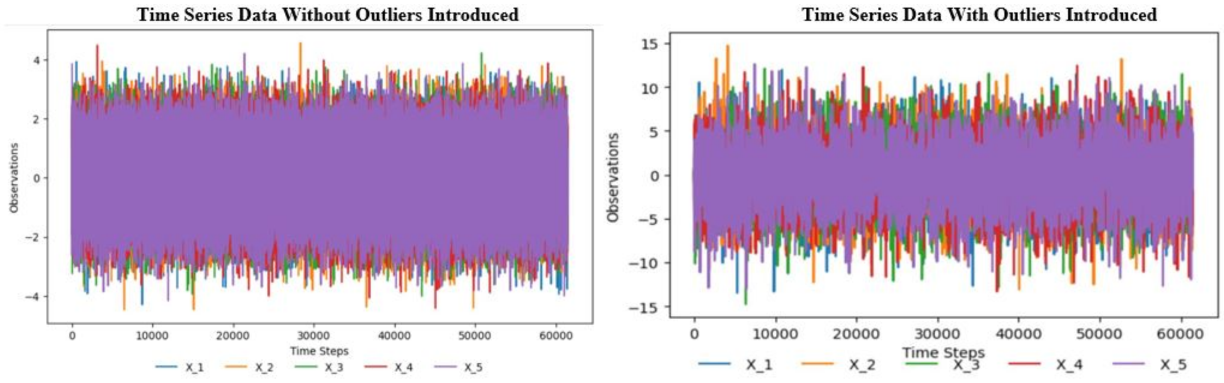


FIGURE 2. MTS Observations on 5 Variables (With and Without Outliers)

each row in the data matrix represents a time step, each column represents a distinct variable. Since the data are purely random from a multivariate normal distribution, white noise behaviour is observed. The data are visualized with plots of the time series (Figure 2). Specifically, the left panel refers to (data without outliers). In addition, 4,000 outliers are introduced into the data. This forms about 6.5% of the data being outliers. This is because by statistical convention, 5% outliers in the data are acceptable while above this threshold are unacceptable [24, 25]. The outliers introduced are visualized with a plot of the time series. Specifically, the right panel refers to (data with outliers). The aim of introducing such outliers is to check the ability of the new hybrid DNN-DBSCAN technique to detect them, together with others (if they are available in the data). Finally, the results are summarized and a discussion is presented. The extraction of features of the data is carried out with the DNN model based on Equations (1) to (15), and then the DBSCAN technique is used in detection of outliers. Thus, the hybrid technique is applied to the results of Equation (8). Next, the performance of the hybrid and traditional techniques are compared. However, in applying the DBSCAN technique, the first step is to determine the parameters of the technique, namely, ε and $MinPts$. These are presented in Sections 3.1 and 3.2.

3.1. Determination of Radius Parameter

Figure 3 illustrates the k-distance graphs, where the left panel is constructed on the basis of the feature extracted data, and the right panel is based on the non-feature extracted data (sorted by distance).

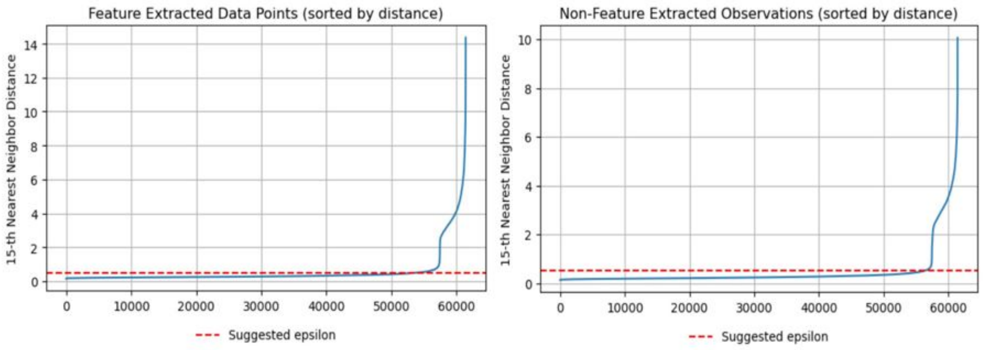


FIGURE 3. k-distance graphs for ε

These two graphs suggest the best ε values for fine-tuning to obtain the desired clustering results in the application of both techniques [13]. Specifically, values around the elbow point of the graphs are preferred. Following convention, a series of values ε are selected around the elbow points (Table 1) to arrive at the optimal value. In Table 1, it is observed that the best results for the detected outliers and performance measures are found at $\varepsilon = 0.75$. This is used as the ε value for optimal clustering and outlier detection. Next is the determination of Minimum Points, referred to as *MinPts*. This is presented in Section 3.2.

3.2. Determination of *MinPts Parameter*

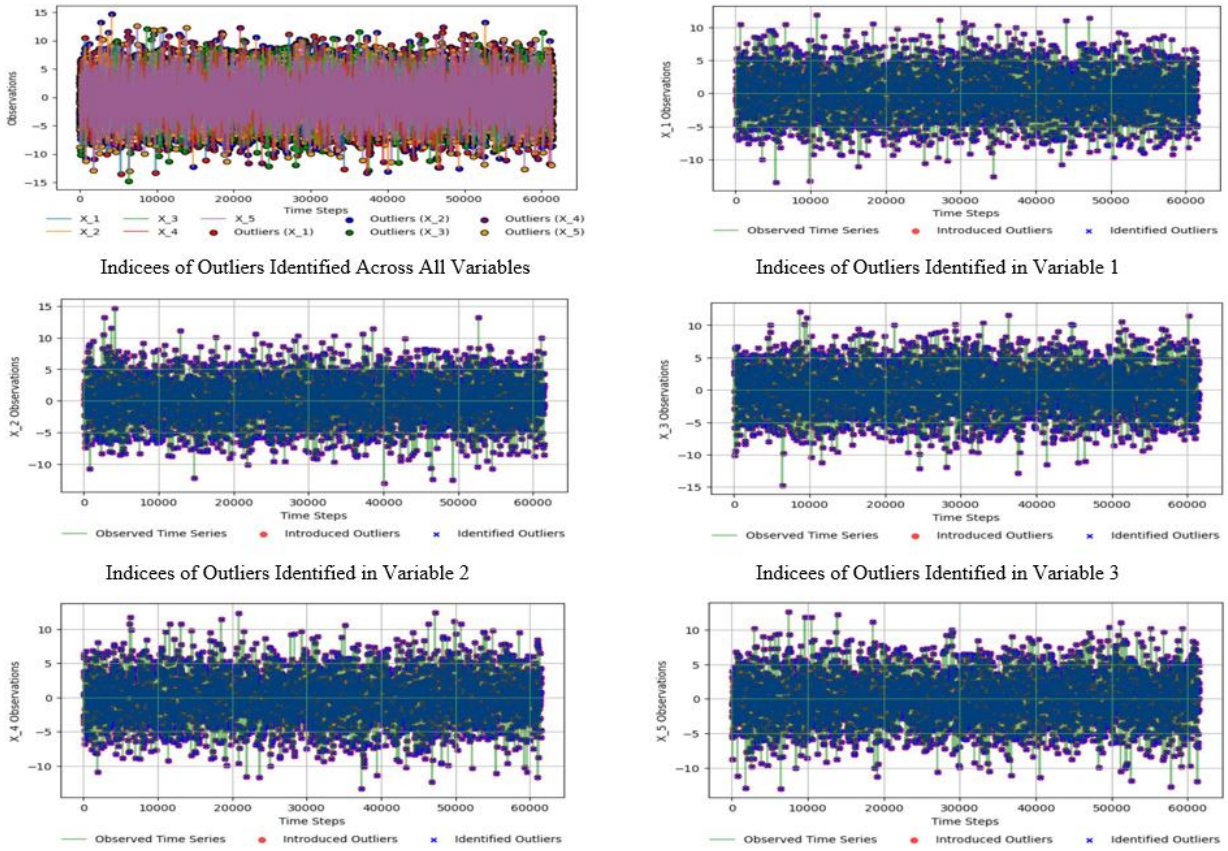
From the MTS data, the number of variables *(p)* is 5 and by Equation (12), the *MinPts* is

$$MinPts \geq 2(p+1) \geq 2(5+1) \geq 12 \tag{23}$$

Thus, the selected *MinPts* must be at least 12. To ensure that the best *MinPts* is selected, series of *MinPts* are conventionally used in conjunction with the series of ε values. Therefore, the best value selected is *MinPts* = 12. Thus, using *MinPts* = 12 and $(\varepsilon) = 0.75$, outliers detected are presented in Table 1. The detection of outliers using these two parameters are presented in Section 3.3.

3.3. Outliers Detection

As stated earlier, series of ε and *MinPts* values are tried to ensure the best parameter values are selected and used for optimal clustering and efficient detection of outliers. The same set of parameters are used in the application of the two techniques, namely, hybrid DNN-DBSCAN and traditional DBSCAN techniques. Figure 4 shows the indices at which outliers are detected across all the 5 variables. In particular, Figure 4 shows the superposition of the indices at which outliers are detected across the 5 variables, as well as for individual variables. Furthermore, Figure 4 indicates the indices at which outliers are detected for all variables superimposed on the same panel (top left corner). Furthermore, the other five images give the indices at which outliers are detected in all the other variables. These outliers are detected by the hybrid technique.



Indicees of Outliers Identified Across All Variables

Indicees of Outliers Identified in Variable 1

Indicees of Outliers Identified in Variable 2

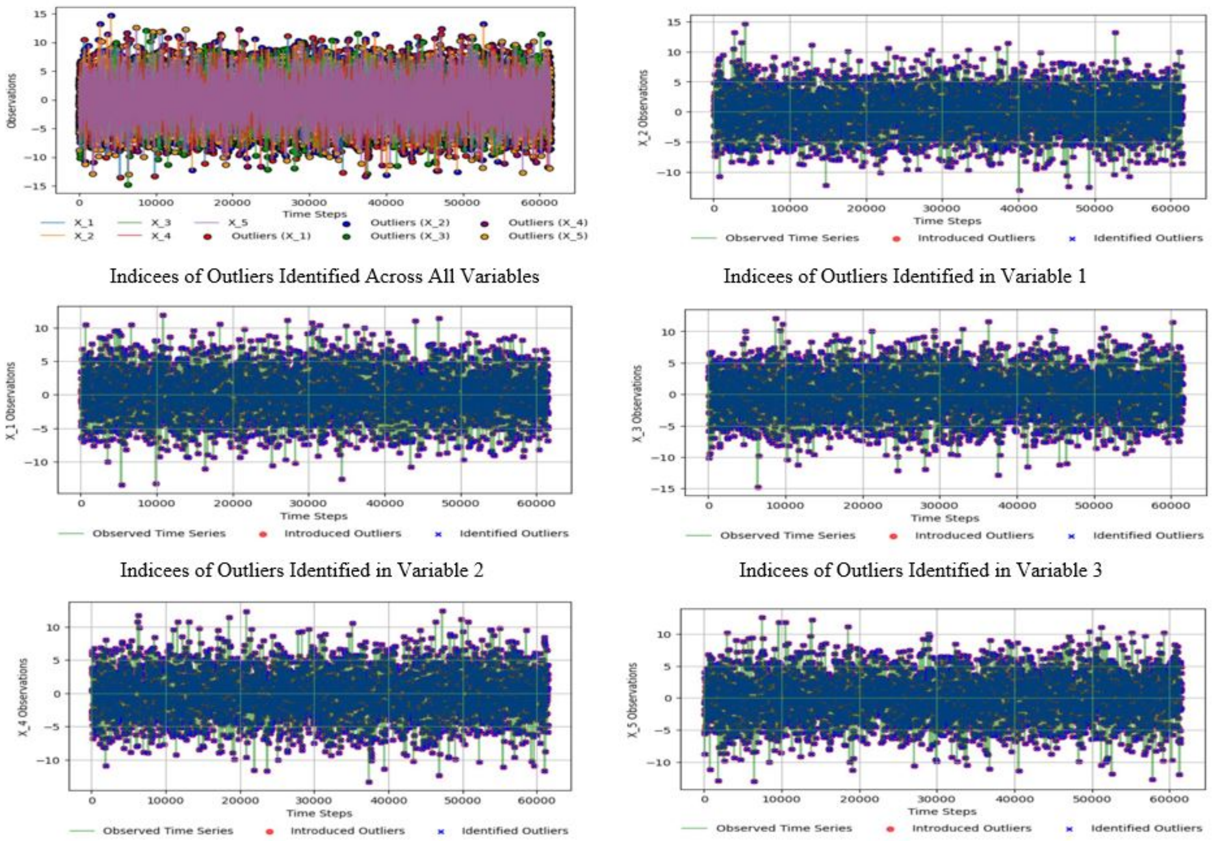
Indicees of Outliers Identified in Variable 3

Indicees of Outliers Identified in Variable 4

Indicees of Outliers Identified in Variable 5

FIGURE 4. Indices of Detected Outliers (Hybrid Technique)

Similarly, Figure 5 shows the superposition of the indices at which outliers are detected across the five variables, as well as for individual variables. Specifically, Figure 5 indicates the indices at which outliers are detected for all variables superimposed on the same panel (top left corner). Furthermore, the other 5 images give the indices at which outliers are detected in all other variables. These outliers are detected from the application of the traditional technique. The number of outliers detected when the hybrid technique is applied is seen to be slightly higher than those detected using the DBSCAN technique (4037 versus 3977). The hybrid technique behaves well in terms of the metrics that describe the goodness of fit of the clusters and detected outliers. This is attributed to the effectiveness of the combination of the DNN model for feature extraction and DBSCAN technique for clustering. Thus, the hybrid technique has contributed much more in detecting influential outliers relative to the DBSCAN technique. Therefore, the specific performance of the two techniques in terms of ability to perfectly cluster and detect



Indicees of Outliers Identified Across All Variables

Indicees of Outliers Identified in Variable 1

Indicees of Outliers Identified in Variable 2

Indicees of Outliers Identified in Variable 3

Indicees of Outliers Identified in Variable 4

Indicees of Outliers Identified in Variable 5

FIGURE 5. Indices of Detected Outliers (DBSCAN Technique)

outliers, and particular one performing much better relatively, is presented in Section 3.4, Tables 1 and 2, and Figures 6 and 7.

3.4. Performance of Hybrid DNN-DBSCAN Technique

The performance of the hybrid DNN-DBSCAN technique is evaluated in relation to one of its variants, the traditional DBSCAN technique. The cluster evaluation and performance metrics in (16) to (22), together with other measures such as the heat map, plot of the clusters with PCA for dimensionality reduction, are used. These results are in Table 1 and Figures 6 and 7. Table 1 provides information on the number of outliers detected and performance measures for both techniques. On all 5 variables, the number of outliers detected and performance of the clustering techniques are presented. For example, using the same selected parameters (ε =0.75 and *MinPts*=12), as suggested by the hybrid technique detected 4025 outliers while the DBSCAN

TABLE 1. **Performance Comparison of DNN-DBSCAN and DBSCAN Techniques**

Hybrid DNN-DBSCAN Technique				**Parameters**		**DBSCAN Technique**			
Outliers	**S(i)**	**DBI**	**CHI**	ε	**MinPts**	**Outliers**	**S(i)**	**DBI**	**CHI**
27 872	−0.44	1.66	228.62	0.25	12	14 756	−0.10	43.05	2.94
29 189	−0.43	1.62	260.53		13	15 408	0.03	56.80	3.58
30 429	−0.43	1.63	312.17		14	16 032	0.02	57.60	4.41
31 677	−0.44	1.63	259.30		15	16 699	0.34	81.92	4.83
4840	0.81	0.88	69 952.18	0.50	12	4140	0.75	55.03	17.50
4892	0.81	0.89	68 687.59		13	4146	0.75	54.66	17.73
4931	0.81	0.89	67 753.06		14	4159	0.75	54.89	17.55
4987	0.81	0.90	66 469.06		15	4163	0.74	54.76	17.63
4025	0.85	0.75	95 000.90	0.75*	12*	3977	0.75	55.11	17.73
4027	0.85	0.75	94 958.56		13	3977	0.75	55.11	17.73
4031	0.85	0.75	94 821.67		14	3977	0.75	55.11	17.73
4037	0.85	0.75	94 650.89		15	3977	0.75	55.11	17.73
3970	0.85	0.74	98 345.57	1.00	12	3960	0.75	55.76	17.34
3970	0.85	0.74	98 345.57		12	3960	0.75	55.76	17.34
3970	0.85	0.74	98 345.57		12	3960	0.75	55.76	17.34
3970	0.85	0.74	98 345.57		12	3960	0.75	55.76	17.34
3970	0.85	0.74	98 345.57	1.25	12	3933	0.76	56.35	17.00
3970	0.85	0.74	98 345.57		13	3933	0.76	56.35	17.00
3970	0.85	0.74	98 345.57		14	3933	0.76	56.35	17.00
3970	0.85	0.74	98 345.57		15	3933	0.76	56.35	17.00
3958	0.85	0.74	98 787.46	1.50	12	3885	0.76	55.39	17.62
3958	0.85	0.74	98 787.46		13	3885	0.76	55.39	17.62
3958	0.85	0.74	98 787.46		14	3885	0.76	55.39	17.62
3958	0.85	0.74	98 787.46		15	3885	0.76	55.39	17.62
3915	0.85	0.74	98 505.22	1.75	12	0	0	0	0
3915	0.85	0.74	98 505.22		13	0	0	0	0
3915	0.85	0.74	98 505.22		14	0	0	0	0
3915	0.85	0.74	98 505.22		15	0	0	0	0
3928	0.85	0.74	100 036.29	2.00	12	0	0	0	0
3928	0.85	0.74	100 036.29		13	0	0	0	0
3928	0.85	0.74	100 036.29		14	0	0	0	0
3928	0.85	0.74	100 036.29		15	0	0	0	0

Note: Best performing configurations are in asterisks $(*)$

technique detected 3977 outliers. This is attributable to the efficiency of the hybrid technique in terms of the ability to detect outliers. The number of outliers detected and performance metrics improved significantly to the required range of acceptance (Sections 2.4.1 to 2.4.7), as recommended by [16, 17, 18]. This is attributed to the effectiveness and efficiency of the hybrid technique in detecting influential outliers. Likewise, the effectiveness of the hybrid technique lies in the fact that in Table 1, even after $MinPts = 12$ and $\varepsilon = 1.50$, it continues to detect outliers even though the numbers decrease to some extent. However, the traditional technique does not detect any further outliers at these same parameter values and beyond. Specifically, the hybrid technique is effective because of its better clustering metrics in relation to the traditional technique. That is, $S(i) = 0.85$ against 0.75, indicating that the hybrid technique clusters are more distinct, $DBI = 0.75$ against 55.11, implying the hybrid technique has much better separation, and $CHI = 96702.64$ against 17.73, indicating that the hybrid technique finds much stronger clusters relative to the traditional technique.

Furthermore, in terms of stability with respect to outlier detection [13, 23], the outlier counts of the hybrid technique stabilize at epsilon greater than or equal to 0.75. That of the traditional technique however, varies more with *MinPts*. Finally, the hybrid technique maintains high performance across *MinPts* while the traditional technique degrades quickly with small values of ε. The stability of the number of outliers detected and performance metrics indicate the robustness of the hybrid technique [23]. This is particularly important in unsupervised settings where parameters may not be easily tuned [13], as in the case of the current study, where the unsupervised learning technique is used in extracted feature of the data before clustering to detect outliers through the DBSCAN technique. Therefore, it is clear that the optimal outlier detection would be obtained at ε values not exceeding 1.50. It is observed that up to this value of $\varepsilon = 0.75$, the optimal detection is obtained at $MinPts = 12$, since they are associated with the best performance metrics. Thus, the optimal detection is obtained at the pair of parameter values of $\varepsilon = 0.75$ and $MinPts = 12$ based on the hybrid technique.

Moreover, the proportion of points detected as outliers by the hybrid, and DBSCAN techniques relative to the total data are estimated using Equation (19). For the hybrid technique, 6.57% of the entire data points are detected as outliers while that for the traditional technique is 6.47%. These proportions fall within the acceptable range for outliers to have significant effects on parameter estimations in any modeling [24, 25]. This shows an improvement in performance of the metrics. Although the proportions appear the same, the performance metrics stated otherwise, with the hybrid technique showing much more improvement.

Further, Table 2 presents the performance metrics comparing the DNN-DBSCAN hybrid technique with the traditional DBSCAN technique for outlier detection. The precision (real detected outliers), recall (ability to find actual outliers) and F1-Score (balancing precision and recall) of the techniques are presented using Equations (20) to (22). The hybrid technique achieves high *Precision* and shows much better performance on all metrics. The higher *Recall* indicates that it misses none of the outliers introduced. The *F1-Score* reflects a better balance between precision and recall. The hybrid technique is deemed effective because it has caught all possible outliers (100% Recall). This is because the primary aim is to ensure no outlier is missed, making the hybrid technique the better choice.

TABLE 2. Precisions, Recalls and F1-Scores

Performance Metrics	**Hybrid DNN-DBSCAN Technique**	**DBSCAN Technique**
True Positives	4025.00	3977.00
False Negatives	0.00	23.00
False Positives	25.00	0.00
Precision (%)	99.38	100.00
Recall (%)	100.00	99.43
F1-Score (%)	99.39	99.71

Additionally, the degree of deviation of the detected outliers from normal patterns is presented in Figure 6. The heat Map shows the variables and observations that significantly deviate from expected patterns. The outlier indices axis corresponds to the specific data points classified as outliers, whereas the variable axis denotes the 5 variables. According to the scale, the blue colour indicates negative deviations from normal patterns while the red represents positive deviations. The degree of divergence from normal pattern is indicated by colour intensity. Variables, X_1, X_3, and X_4 show significant deviations, suggesting potential systemic outliers in these

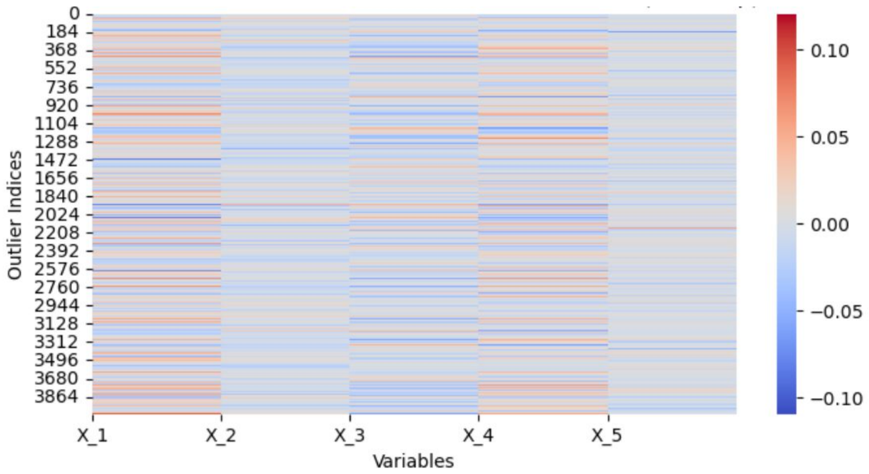


FIGURE 6. Heat Map (Hybrid Technique)

variables. Also, X_2 and X_5 demonstrate reduced variances compared to the others. Observations concentrate on specific index ranges, indicating that some periods or events are prone to unusual patterns across all or particular variables. The results indicate that the combination of DNN and DBSCAN techniques is effective and efficient in detecting outliers. Figure 6 further highlights the contributions of specific variables to the overall variation, showing the concentration on variables or time intervals prone to anomalies. This visualization enables further analysis or corrective actions.

Finally, plots of the PCA-formed clusters for dimensionality reduction is shown in Figure 7. The hybrid technique distinctly defines outliers, primarily situated outside the main triangular data cluster. The data distribution has a clear triangular configuration, with dense area accurately designated as clusters. The traditional technique also detects some outliers, but appears less defined in terms of the clusters detected by the technique [23]. However, the more circular and homogeneous clusters (right hand side and bottom panels) suggest that the traditional DBSCAN technique might have encountered difficulties in capturing the high-dimensionality or complex nature of the non-feature extracted data. It is observed that differentiation between clusters and outliers appear to be challenging for the traditional DBSCAN technique, leading to the neglect of influential outliers. The hybrid technique, thus, markedly improves data representation and the accuracy of influential outlier detection (clear triangular configuration). Hence, the hybrid technique is much more effective in detecting outliers compared to the traditional

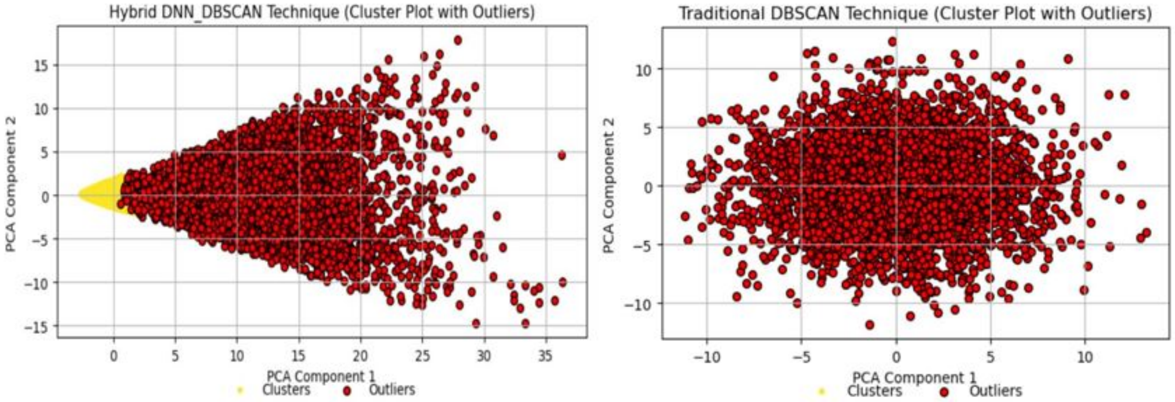


FIGURE 7. Cluster Plot With Outliers (Both Techniques Compared)

DBSCAN technique. This highlights the need for integrating advanced feature extraction with clustering techniques to enhance accuracy and interpretability. These results are congruent with the findings in literature [3, 26, 8, 27, 10, 13, 23].

4. CONCLUSIONS

The paper has integrated the DNN and traditional DBSCAN techniques to achieve a high performance in detecting outliers. This is accomplished by making optimal use of the qualities of each of the techniques. Precisely, the DNN modeling technique (Autoencoders) extracted robust features from the data and encoded it into a space with few dimensions. The fact that the DNN model technique makes it possible to differentiate between clusters and outliers to a large extent is a key advantage. Also, the hybrid technique clusters the encoded features and consequently detect outliers that may be present in the data. Both the accuracy and the recall of outlier detection process has been been improved as a consequence of the combination of the two techniques. Consequently, domains such as predictive maintenance, fraud detection, and environmental monitoring may significantly benefit from the hybrid DNN-DBSCAN technique. The hybrid technique could be verified on a variety of MTS data across different domains (financial markets, healthcare, and climate) to ascertain its generalisability and robustness. Additionally, exploration could be made on various topologies of the deep neural networks (such as depth, neuron count, and activation functions) to determine the optimal configurations for feature selection based on the hybrid technique.

ACKNOWLEDGEMENTS

The authors are grateful to the Pan African University Institute for Basic Sciences, Technology and Innovation (PAUSTI) for the support.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] Z. Zhao, Z. Lai, H. Zhi, Y. Zou, Y. Jin, et al., Automated Workflow of EIS Data Validation and Quality Improvement Based on the Definition, Detection, and Removal of Outliers, Electrochimica Acta 461 (2023), 142661. https://doi.org/10.1016/j.electacta.2023.142661.

[2] V. Chandola, A. Banerjee, V. Kumar, Anomaly Detection, ACM Comput. Surv. 41 (2009), 1–58. https://doi.org/10.1145/1541880.1541882.

[3] Y. Yang, C. Fan, L. Chen, H. Xiong, IPMOD: An Efficient Outlier Detection Model for High-Dimensional Medical Data Streams, Expert Syst. Appl. 191 (2022), 116212. https://doi.org/10.1016/j.eswa.2021.116212.

[4] L. Bontemps, V.L. Cao, J. McDermott, N.-A. Le-Khac, Collective Anomaly Detection Based on Long Short-Term Memory Recurrent Neural Networks, in: T.K. Dang, R. Wagner, J. Küng, N. Thoai, M. Takizawa, E. Neuhold (Eds.), Future Data and Security Engineering, Springer, Cham, 2016: pp. 141–152. https://doi.org/10.1007/978-3-319-48057-2_9.

[5] D.T. Kieu, A. Kepic, Integration of Borehole Data in Geophysical Inversion Using Fuzzy Clustering, ASEG Ext. Abstr. 2018 (2018), 1–6. https://doi.org/10.1071/aseg2018abp083.

[6] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases With Noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press, pp. 226–231, 1996.

[7] W. Xu, Y. Zhang, X. Tang, Parallelizing DNN Training on GPUs: Challenges and Opportunities, in: Companion Proceedings of the Web Conference 2021, ACM, New York, NY, USA, 2021, pp. 174-178. https://doi.org/10.1145/3442442.3452055.

[8] Z. Jiao, P. Hu, H. Xu, Q. Wang, Machine Learning and Deep Learning in Chemical Health and Safety: A Systematic Review of Techniques and Applications, ACS Chem. Health Saf. 27 (2020), 316–334. https://doi.org/10.1021/acs.chas.0c00075.

[9] J. Brownlee, Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python, Machine Learning Mastery, 2018.

[10] F. Cheng, J. Zhang, Z. Li, M. Tang, Double Distribution Support Vector Machine, Pattern Recognit. Lett. 88 (2017), 20–25. https://doi.org/10.1016/j.patrec.2017.01.010.

[11] V. Nair, G.E. Hinton, Rectified Linear Units Improve Restricted Boltzmann Machines, in: Proceedings of the 27th International Conference on Machine Learning, Omnipress, pp. 807–814, 2010.

[12] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, Cambridge, MA, 2016.

[13] E. Schubert, J. Sander, M. Ester, H.P. Kriegel, X. Xu, DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN, ACM Trans. Database Syst. 42 (2017), 1–21. https://doi.org/10.1145/3068335.

[14] S. Siddiqui, M. Arifeen, A. Hopgood, A. Good, A. Gegov, et al., Deep Learning Models for the Diagnosis and Screening of COVID-19: A Systematic Review, SN Comput. Sci. 3 (2022), 397. https://doi.org/10.1007/s42979-022-01326-3.

[15] S. Chaudhary, A. Jatain, Performance Evaluation of Clustering Techniques in Test Case Prioritization, in: 2020 International Conference on Computational Performance Evaluation (ComPE), IEEE, 2020, pp. 699-703. https://doi.org/10.1109/compe49325.2020.9200083.

[16] P.J. Rousseeuw, Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis, J. Comput. Appl. Math. 20 (1987), 53–65. https://doi.org/10.1016/0377-0427(87)90125-7.

[17] D.L. Davies, D.W. Bouldin, A Cluster Separation Measure, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-1 (1979), 224–227. https://doi.org/10.1109/tpami.1979.4766909.

[18] T. Calinski, J. Harabasz, A Dendrite Method for Cluster Analysis, Commun. Stat. - Simul. Comput. 3 (1974), 1–27. https://doi.org/10.1080/03610917408548446.

[19] D.M.W. Powers, Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation, arXiv:2010.16061 (2020). https://doi.org/10.48550/ARXIV.2010.16061.

[20] C.C. Aggarwal, Outlier Analysis, Springer, (2017).

[21] J. Davis, M. Goadrich, The Relationship Between Precision-Recall and ROC Curves, in: Proceedings of the 23rd international conference on Machine learning - ICML ’06, ACM Press, New York, New York, USA, 2006, pp. 233-240. https://doi.org/10.1145/1143844.1143874.

[22] R.J.G.B. Campello, D. Moulavi, A. Zimek, J. Sander, Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection, ACM Trans. Knowl. Discov. Data 10 (2015), 1–51. https://doi.org/10.1145/2733381.

[23] C. Hennig, Cluster-Wise Assessment of Cluster Stability, Comput. Stat. Data Anal. 52 (2007), 258–271. https://doi.org/10.1016/j.csda.2006.11.025.

[24] F.R. Hampel, The Influence Curve and Its Role in Robust Estimation, J. Am. Stat. Assoc. 69 (1974), 383–393. https://doi.org/10.1080/01621459.1974.10482962.

[25] J.W. Tukey, A Survey of Sampling From Contaminated Distributions, in: I. Olkin, et al. (Eds.), Contributions to Probability and Statistics, Stanford University Press, pp. 448–485, 1960.

[26] X. Lu, J. Wang, Y. Yan, L. Zhou, W. Ma, Estimating Hourly PM2.5 Concentrations Using Himawari-8 AOD and a Dbscan-Modified Deep Learning Model Over the YRDUA, China, Atmospheric Pollut. Res. 12 (2021), 183–192. https://doi.org/10.1016/j.apr.2020.10.020.

[27] H. Liu, Y. Liu, Z. Qin, R. Zhang, Z. Zhang, et al., A Novel DBSCAN Clustering Algorithm via Edge Computing-Based Deep Neural Network Model for Targeted Poverty Alleviation Big Data, Wirel. Commun. Mob. Comput. 2021 (2021), 5536579. https://doi.org/10.1155/2021/5536579.