Considering Cluster Validity in Attribute Extension for Small Data Set Predictions

Luu-Ly Tran [1], Chih-Chieh Chang [2,*], and Hsiang-An Yu [3]

[1] Department of Information Management, National Taiwan University of Science and Technology, Taiwan
[2] School of Management, National Taiwan University of Science and Technology, Taiwan
[3] Business Administration Department, National Taiwan University of Science and Technology, Taiwan
Email: M11309813@mail.ntust.edu.tw (L.-L.T.); ccchang@mail.ntust.edu.tw (C.-C.C.);
m11121026@mail.ntust.edu.tw (H.-A.Y.)
*Corresponding author

***Abstract*—Cluster validity has been widely used in determining the optimal clusters with a huge data sample size in recent years. However, there is less discussion of the validity cluster in small data sizes. This study presents a new approach, which considers the cluster validity to improve predictive ability for small data set problems. The first step of the proposed method is the use of the K-means data clustering technique, with seven cluster validity indices to determine the optimal number of clusters; and the second step is to build up the attribute extending function for each attribute in clusters to generate new attributes by computing the membership possibility. Finally, cross-validation and t-tests are used on two real manufacturing cases of Thin-Film Transistor Liquid Crystal Display (TFT-LCD) quality and Photo-Spacer Height (PSH) to verify the effectiveness of the proposed method with backpropagation neural networks (BPNN) and support vector machine for regression (SVR) forecasting methods. The results show that the combinations of C-index and attribute extension yields significantly lower forecasting errors, reduced variance, and statistically validated superiority over other cluster-validity indices and over baseline BPNN and SVR and linear regression (LR) models without attribute extension.**

***Keywords*—cluster validity index, small data set, k-means, support vector regression, mega-trend diffusion**

I. Introduction

There are many small data set problems in our surroundings. For example, the earthquake in Japan on March 11, 2011, may provide very valuable data for predicting tsunamis. It is important to use small data sets to build prediction models, although a number of problems arise in such cases. Early studies addressing small data set learning have primarily relied on virtual sample generation techniques, such as Generative Adversarial Networks (GANs) [1]. While these methods can improve learning performance by increasing apparent data volume, they also introduce a risk of sampling bias, as GAN-generated data tend to propagate existing biases in real-world data and may further under-represent minority patterns [2, 3]. Sampling bias refers to the situation where virtual-sample generation changes the effective sampling distribution seen during training, either by altering class priors (frequency bias) or by unevenly covering rare modes (coverage bias). For example, consider a small manufacturing dataset with n = 20 where only 2 lots correspond to a rare defect mode (true prevalence ≈ 10%). If virtual-sample augmentation increases the minority class to 10–20 synthetic samples, the learner is effectively trained under an inflated defect prevalence (by 40%−50%), which can shift the decision threshold and yield excessive false alarms when deployed on the real process stream. Sampling bias also typically manifests as (i) missing low-frequency regimes (rare failures not observed at all), (ii) high sensitivity to a few high-leverage samples that shift cluster structure, and (iii) time-window or condition-specific sampling that induces covariate shift (e.g., pre- vs post-maintenance).

For example, our TFT-LCD quality dataset contains only 13 samples, meaning rare-defect conditions are represented by at most a few observations. If we attempt to address this imbalance by generating many virtual samples for the minority (rare-defect) regime, the synthetic points are necessarily extrapolated from an extremely limited and unrepresentative set of true defect samples. This can cause the predictor to overfit the synthetic defect pattern and degrade generalization to real production defects. Moreover, GAN-based augmentation trained on very few minority samples often suffers from mode collapse and limited minority coverage, generating many near-duplicates of the dominant minority subtype while failing to reproduce other rare but critical patterns [4]. Another approach involves the Mega-Trend Diffusion (MTD) technique, which produces artificial samples to enhance predictive performance in small data sets [5, 6]. However, this method depends on a single diffusion value defined over a global domain, limiting its ability to capture local structural variations in the data.

Clustering is a useful approach in machine learning, and cluster analysis is a classification technique for

Manuscript received October 17, 2025; revised November 24, 2025; accepted February 6, 2026; published May 13, 2026.

discovering whether the individuals in a population fall into different groups by making quantitative comparisons of multiple characteristics. Recent studies, such as projected cross-view learning for unbalanced incomplete multi-view clustering, combines missing samples recovery, projected cross-view learning, and graph learning to develop efficient and high-performance clustering [7]. Although such works primarily focus on large multi-view datasets, they reinforce the broader research direction in which clustering often requires additional mechanisms to restore information balance and enhance cluster reliability. Our study addresses a different but equally important scenario, small data sets, where traditional clustering becomes unstable. Another challenge in clustering approach is determining the optimal number of clusters [8]. Cluster validity index, which measures the compactness and separation of the data clustering results, is one way to overcome this problem. In recent years, many studies using cluster validity indices have been presented examining large data set clustering problems, such as using the mean of clustered data to determine the optimum number of clusters, a Hybrid FCM-WOA data clustering algorithm, but much less work has been implemented on small data sets [9, 10].

To address these limitations, we propose the method that integrates cluster validity analysis with attribute extension in a unified framework with two main phases. The first uses the K-means data clustering technique with seven different cluster validity indices to find the optimal number of clusters. Seven indices are the Silhouette index (S-index), Davies and Bouldin index (DB), Calinski-Harabasz index (CH), Dunn index (DUN), C-index, Krzanowski-Lai index (KL), and Hartigan index (H-index). Based on this number, the second step is to build up the attribute extending function for for each original attribute in the clusters. The membership grades produced by these functions are treated as additional attributes and concatenated with the original feature set to form an extended training dataset for forecasting models.

Unlike oversampling or GAN-based augmentation, the proposed framework does not generate synthetic or virtual data points does not reweight samples. Instead, it derives additional attributes deterministically from the original observations through cluster membership functions. Therefore, the method avoids the specific form of sampling bias introduced by virtual-sample generation which are distortions of sample frequencies (frequency bias) and distortions of rare-mode coverage caused by generating additional samples (coverage bias). Furthermore, in contrast to the MTD-based approach in [6], which assigns a single diffusion-based importance value to uniformly generated samples over a global domain, our work explicitly models cluster structure. By constructing multiple cluster-specific diffusion functions and integrating them into a cluster-aware feature extension scheme, it captures richer local and geometric relationships. These extended attributes encode soft cluster membership and similarity patterns that are not directly observable from the original variables, thereby enhancing the effective information content of the feature representation and improving robustness with respect to the choice of the number of clusters K.

To evaluate the effectiveness of the proposed approach, two small manufacturing datasets are examined. The first case involves predicting the quality of TFT-LCDs, and the second concerns the prediction of PSH. Cross-validation is employed to assess predictive performance, with comparisons against LR, BPNN, and SVR using metrics such as average Mean Squared Error (MSE), standard deviation (STD), and statistical t-tests.

In summary, our work fills the existing gaps in (i) scaling the utility of small datasets; (ii) an efficient approach for data generation which balances the productivity and computation complexity without generating synthetic samples; (iii) a comprehensive investigation into the integration of cluster validity analysis and attribute extension for small data set forecasting. The rest of this paper is organized as follows: Section II describes some studies related to small data set approaches, cluster validity indices, and forecasting models. The proposed method is shown in Section III, and in Section IV uses two real cases to demonstrate it. Finally, the conclusions and suggestions for future studies are presented in Section V.

II. Related Work and Background

This section presents a review of works related to this study. We first review some of the research into small data set problems that have been published in the past decade, and then some cluster validity-related studies. Finally, forecasting models are deployed as a benchmark to demonstrate the potential and effectiveness of our method.

A. Small Data Set with Clustering

Small data machine learning has gained significant attention in recent years due to its challenges and implications for various applications. In Computer Science, small data sets are defined by their limited size, with quantitative thresholds such as fewer than 200 modules (including classes, files, or similar units) being used to classify a data set as small in software engineering research [11]. In this paper, we define a small dataset as a learning setting where the number of labeled observations per prediction task is $n < 20$. This threshold reflects our manufacturing use cases, in which historical records for rare events (such as exceptional process conditions or low-frequency failures) are typically limited to only a few to tens of samples, making conventional data-hungry learners and resampling-based validation unstable.

Clustering techniques have emerged as a valuable tool for addressing these challenges by extracting meaningful patterns from limited samples. Several studies have explored the application of clustering algorithms in the context of minor data problems. Xia *et al.* [12] proposed a multiple kernel k-means clustering method with matrix-induced regularization, enabling the discovery of meaningful clusters in small data sets. McNeish and Harring [13] conducted a rigorous simulation experiment to compare the performance of two clustering techniques:

Generalized Estimating Equations (GEE), a design-based method, and Generalized/Linear Mixed effects Models (GLMM), which are model-based methods. The objective was to evaluate the standard error estimates and relative statistical power of these methods when dealing with small data set problems.

Peng *et al.* [10] addressed the challenging problem of diversified cluering and proposed a novel approach. Their experiment specifically focused on small data sets, aiming to generate clustering solutions that exhibit high diversity. Their research employed robust methodologies and rigorous evaluation criteria to assess the effectiveness of their proposed technique. Weigand *et al.* [14] conducted a comprehensive analysis, comparing the performance of several clustering techniques, namely K-Means, Hierarchical Agglomerative Clustering (HAC), and Density-Based Spatial Clustering of Applications with Noise (DBSCAN), with a particular emphasis on their suitability for handling small data sets. The study employed sophisticated evaluation metrics and statistical analyses to determine the strengths and weaknesses of each method. The findings highlighted K-Means clustering as a promising approach, offering improved interpretability of machine learning-based clustering outcomes.

In a recent work by Yu *et al.* [15], a novel approach called the Gaussian Mixture Model with Wasserstein Generative Adversarial Network was proposed to address the challenges associated with clustering problems in the context of limited data set sizes. They demonstrated their method's effectiveness in generating meaningful clustering results when faced with insufficient data, leveraging the power of generative adversarial networks and Gaussian mixture models. The study employed rigorous experimentation and evaluation protocols to validate the superiority of the proposed approach. While these prior studies have proposed improved clustering algorithms for small data, they focus primarily on enhancing clustering quality or statistical estimation. In contrast, our present work leverages clustering as a feature-engineering mechanism rather than as an end task.

Besides, existing approaches still face critical limitations under small-sample conditions. GAN-based and highly flexible models (e.g., DBSCAN, GLMM) often suffer from overfitting or mode collapse [16], as their parameter complexity exceeds the available data signal. Diversified clustering techniques can expose the instability inherent in such settings, but do not fundamentally enhance the feature representation itself. Moreover, most studies emphasize clustering quality rather than establishing a direct linkage between clusters and predictive accuracy, leaving practical performance gains uncertain. Finally, while kernel regularization introduces a form of structural bias, it lacks an explicit and controllable prior that captures global or long-range trends across samples. This gap highlights our approach with the MTD mechanism to augment and smooth the feature space within and across clusters, enabling more stable, generalizable, and predictive representations under small-sample conditions. Overall, by explicitly integrating cluster validity indices to select optimal cluster numbers and constructing membership-based attribute extensions, our work directly enhances supervised prediction accuracy, a dimension not addressed by existing small-data clustering research.

B. Cluster Validity Indices

Cluster analysis is an exploratory method used to group objects based on their similarities or intrinsic characteristics [17]. Its main objective is to reveal the inherent cluster structure present in data [18], and it is therefore exploratory in nature. Data clustering has been widely employed across various domains, including gaining insights into underlying data structures, generating hypotheses, detecting anomalies, and identifying significant features. One approach involves exploring natural classifications or determining the degree of similarity among forms or organisms. Data compression is another technique that organizes and summarizes data using cluster prototypes. However, choosing the optimal number of clusters presents a challenge. One strategy is to utilize cluster validity indices to ascertain the most appropriate number of groups. This involves executing the clustering algorithm multiple times with different cluster sizes. Below is a brief overview of various cluster validity indices.

Calinski and Harabasz [19] used the variance ratio of the within-group sum of squares and the between-group sum of squares, and used the resulting maximum value of the CH as the optimal number of clusters. Dunn [20] proposed an index, called DUN, for crisp clustering, which is assigned to each cluster based on its distance from the within-cluster and between-cluster. The main objective is to maximize the DUN index to achieve the optimal number of clusters. Hartigan [21] indicated that the ratio of the square error to the maximum value in two clusters can be determined as the best number of clusters, and this is known as the H-index. Davies and Bouldin [22] presented the DB-index, which measures the average similarity between each cluster and its most similar one. A lower value of DB-index shows a better number of clusters. Milligan and Cooper [23] showed that minimizing the C-index produces the optiomal number of clusters. Rousseeuw [24] presented the S-index, which represents the average distance of all other points within the cluster, with the maximized value of the S-index indicating the optimal number of clusters. Krzanowski and Lai [25] proposed a new criterion to assess the ratio of different clusters, with the maximum value of the ratio being the best number, and this approach is called the KL. Iglesias *et al.* [26] conducted an evaluation of clustering results by performing geometric measurements on the solution space. Their study employed various indices to provide a coherent interpretation of the data structure, thereby improving the robustness of clustering applications for knowledge discovery. Recently, Naderipour *et al*. [27] introduced a structured and similarity-based validity index that calculates the compactness and separation between communities. This index offers valuable suggestions for datasets based on its computed metrics. Duan *et al.* [28] conducted research on

the augmented non-shared nearest neighbors index, which determines the difference between between-cluster separation and within-cluster compactness. Their work showed that this index outperformed others in terms of performance.

However, there are not many studies that discuss the combination of cluster validity index with a small data set attribute extension function. Consequently, our study considers seven cluster validity indices, namely CH, DUN, H-index, DB, C-index, S-index, and KL, to discover the performance within the attribute extension function in minor data set problems.

C. Forecasting Models

In recent years, there have been numerous studies comparing the predictive capabilities of various forecasting models. For instance, Li and Liu [29] proposed a neural network weight determination model, which employed concepts from cognitive theory to develop a weighted learning algorithm. Carrizosa *et al.* [30] proposed an automated Support Vector Machine (SVM)-based method capable of detecting important predictor variables and demonstrating excellent classification ability. Li *et al.* [5] developed a yield forecast model based on past manufacturing experience and compared it with methods using regression, BPNN, RBFNN, and SVR. Furthermore, their study also demonstrated that SVR had significant potential to improve the non-linear quality in manufacturing TFT-LCDs. More recently, studies in network science have shown that structural signals such as the Local Clustering Coefficient (LCC) can be exploited to forecast social connections, since increases in LCC often correspond to the formation of new links through triadic closure [31]. In this study, we will benchmark our approach's efficiency based on LR, SVR, and BPNN [32, 33].

1) Support Vector Machine for Regression (SVR)

SVM first proposed by Vapnik [34], are widely used classification tools grounded in the Structural Risk Minimization (SRM) principle of statistical learning theory. Unlike empirical-risk–based learners that often overfit in low-sample regimes, SVM explicitly controls model capacity by maximizing the geometric margin between data points and the separating hyperplane [35, 36]. This margin-maximization principle naturally produces classifiers with strong generalization ability, particularly when the number of samples is limited.

Drucker *et al.* [37] further extended the SVM framework to SVR, motivated by the observation that classical SVM formulations are not directly suited for prediction tasks. The main difference between the two approaches is that the original regression uses the whole data set to build the forecasting model, while SVR only uses the "valid data". The SVR constructs a model by applying acceptable values, which are in the interval with the acceptable error ε, as shown in Fig. 1.

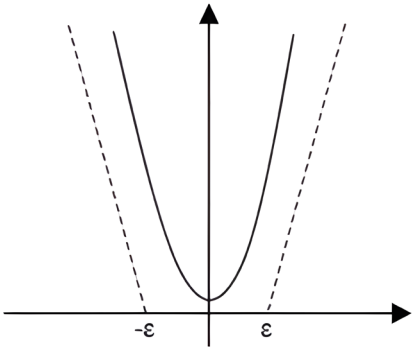

Fig. 1. Linear problem.

In contrast to traditional regression approaches that use the full dataset to minimize squared error, SVR relies only on "support vectors", the minority of points lying outside the ε-tube, to define the predictive function. This sparse representation enhances robustness, especially when data samples are insufficient or noisy. While SVMs efficiently handle linearly separable data, they can also address nonlinear problems by mapping the original inputs into higher-dimensional feature spaces through kernel functions, enabling flexible yet capacity-controlled decision boundaries. The variable ε represents the cost of error, and it is zero when the data point is within the acceptable ε, which is seen as "valid data". If ε is higher than 0, the data are considered invalid. Fig. 2 shows the typical results of using an SVM to process a nonlinear data set. Recent applications demonstrate their effectiveness in domains such as face recognition, handwriting recognition, and financial data mining.

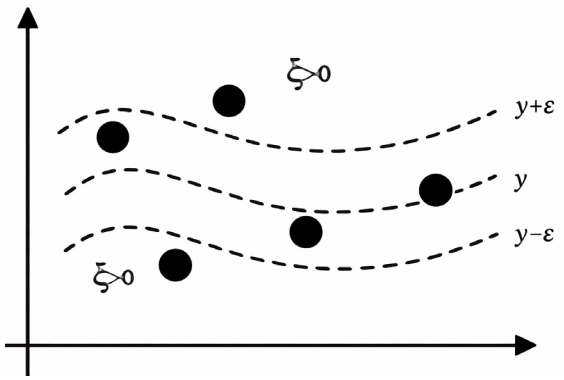

Fig. 2. Non-linear problem.

Crucially, SVM and SVR have been repeatedly shown to be high-performing learners under small-data conditions, which aligns directly with the goals of this research. Their reliance on margin maximization, capacity control (via penalty parameter C and kernel parameters (γ)), and sparsity of support vectors make them particularly resistant to overfitting when N is small. In fact, several studies [38] report that SVM-based models consistently outperform Neural Networks (NN), decision trees, and deep learning methods when sample sizes are limited, since they require far fewer parameters and impose strong theoretical constraints on solution complexity.

Our proposed extension is designed to increase intra-cluster cohesion and inter-cluster separation, effectively reshaping the feature space; SVMs respond to exactly these geometric changes by selecting decision functions that maximize the margin in the induced (possibly kernel) space, a principle that is strongly connected to generalization behavior via margin-based bounds. Moreover, margin/radius–margin analyses make explicit that generalization depends on geometric quantities of the embedded data, so improvements in cluster structure should translate into better performance without requiring higher model flexibility. This choice is also aligned with prior work showing that membership-derived feature mappings (cluster/fuzzy membership functions used to re-express inputs in a new feature space) pair naturally with SVM learning as a downstream evaluator, supporting SVM/SVR as an appropriate validator for membership-based attribute construction [39]. Finally, to ensure small-N fairness and avoid optimistic evaluation, we implement leakage-safe preprocessing by fitting the scaler within each fold and apply nested cross-validation to tune C, γ, and epsilon (ε). Thus, SVR not only provides strong predictive baselines but also offers diagnostic evidence that our method meaningfully improves the signal structure within small datasets.

2) Back-Propagation Neural Network (BPNN)

NN has been widely used in machine learning algorithms, with many applications in manufacturing analysis, medical diagnosis, and business forecasting. BPNN are the most representative of the supervised neural network algorithms. The main framework of BPNN was proposed by Rumelhart *et al*. [40], which used the gradient steepest descent method to adjust the weight and bias. The objective of the BPNN is to minimize the difference between the predicted and real values, and the BPNN can also be seen as a nonlinear expansion of the Least-Mean-Square (LMS) error approach. Fig. 3 shows the basic concept of the BPNN, which can be divided into three parts: input layer, hidden layer, and output layer. The initial data would be input into the hidden layer, using random weights and biases to construct models, and then computing the expected errors between the exact value and predicted value. Iterate this step until the expected errors are stable.

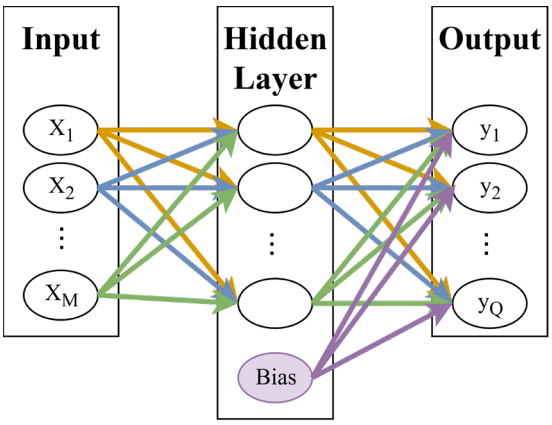

Fig. 3. The structure of a back-propagation neural network.

BPNN is leveraged as a benchmark due to the capacity-controllable universal approximator whose performance is tightly coupled to the quality of the representation that feeds it. If MTD denoises and structures the space uncovered by clustering, a properly regularized BPNN should (i) reach a given validation score in fewer epochs; (ii) achieve higher accuracy/AUC with the same or smaller architecture; and (iii) show a smaller generalization gap than when trained on raw features. Unlike SVM, BPNN can exploit residual nonlinear structure introduced by MTD while still being regularized to avoid overfitting in small-N settings, so gains are attributable to better features, not model size. To keep the comparison small-N fair and prevent the network from winning via excess capacity, we strictly cap model size, apply standard small-sample L2 weight decay, which is theoretically and empirically shown to improve generalization and early stopping, which curbs overfitting by halting training when validation performance ceases to improve. Evaluation follows the same leakage-safe protocol as SVR: preprocessing is fit within each fold, hyperparameters (hidden width, weight decay strength, learning rate) are selected via nested cross-validation.

III. Methodology

This section explains the proposed method in detail. The whole method starts with validity clustering and building attribute extension functions. We then develop the possibility attribute and combine the transformed data with the raw data set for the forecasting models.

A. K-Means and Cluster Validity Indices

This research applies the commonly used K-means clustering algorithm to classify the target value with natural data characteristics [41]. Considering a data set $X = x_i, i = 1, \ldots, n$, which is the set of n d-dimensional points to be clustered into a set of K clusters, $C = \{c_k \,|K = 1, \ldots, k\}$, the K-means algorithm will find a partition where the squared error between the empirical mean of a cluster and the points in the cluster is minimized [42]. Mathematically, let μ_k be the mean of cluster c_k, and then the squared error between k and the points in cluster c_k is defined as:

$$J(C) = \sum_{x_i \in C} \|x_i - \mu_k\|^2 \tag{1}$$

The goal of K-means clustering is to minimize the sum of the squared error over all K clusters, and it is expressed in a formula as:

$$J(C) = \sum_{k \in 1, \ldots k} \sum_{x_i \in C} \|x_i - \mu_k\|^2 \tag{2}$$

The main steps of the K-means algorithm are abbreviated as follows [43]:

1. Select an initial partition with K clusters.
2. Generate a new partition by assigning each pattern to its closest cluster center.
3. Compute new cluster centers.
4. Repeat steps 2 and 3 until cluster membership stabilizes.

The algorithm was run with multiple random initializations, and the best solution (minimum within-cluster sum of squares) was selected. In addition, K-means++ initialization was used to ensure well-separated starting centroids, reducing variance across runs. All features were standardized prior to clustering, and cluster validity indices were evaluated across repeated runs to avoid unstable partitions.

Seven Cluster Validity Indices (CVIs) including the S-index, DB, CH, DUN, C-index, KL, and H-index were selected. Since in our framework, clustering is a medium, not an end goal. The end goal is to determine the additional attributes directly from the derived membership functions. Therefore, these CVIs are suitable to be computable in small-n manufacturing scenarios and provide complementary diagnostics rather than redundant scoring.

It also relies only on the observed data for distances, within/between dispersion without assuming a specific generative model. Specifically, DB and CH quantify compactness–separation trade-offs using within- vs between-cluster dispersion [19]; Dunn emphasizes worst-case separation relative to cluster diameters, making it sensitive to narrow gaps and outliers that are common in small batches [20]; C-index assesses within-cluster pairwise distances relative to achievable bounds, providing a rank-like check against overly fragmented partitions [44]; KL and H are elbow-type criteria derived from within-cluster sum-of-squares trends across k, which helps detect diminishing returns when additional clusters only marginally reduce dispersion [25, 45]; finally, the S-index family captures symmetry/shape-driven structure and can detect non-spherical yet symmetric clusters that Euclidean-compactness indices may undervalue. Using them set mitigates the risk of over-trusting any single criterion in small-n settings by cross-checking dispersion-, separation-, pairwise-distance-, and shape-based signals.

We did not include several widely used criteria because they are either redundant with our selected diagnostics or misaligned with membership-based attribute extension under small n. For example, Silhouette and PBM/I-index primarily provide average compactness–separation summaries, which overlap with the dispersion and separation signals already covered by DB/CH (average trade-off) and Dunn (worst-case gap); adding them would increase redundancy rather than provide a new failure-mode check. We also avoided resampling- or simulation-driven criteria such as the Gap statistic and cluster stability/consensus measures because repeated subsampling (and reference distributions) can be high-variance when n is small, leading to unstable k and, consequently, unstable membership functions and extended attributes. Finally, model-based selection (e.g., AIC/BIC for Gaussian mixtures) imposes distributional assumptions that are not required by our distance/membership pipeline, and fuzzy-specific CVIs (e.g., Xie–Beni, partition coefficient/entropy) can become circular in our setting because they evaluate memberships produced by a particular fuzzy clustering objective, whereas our memberships are derived downstream for attribute extension rather than optimized by that fuzzy objective. The equations of each index are described as follows:

S-index:

$$S = \frac{D - C}{max(C, D)} \tag{3}$$

where S denotes the sum of the distances of all objects from the same cluster, C represents the average distance of a point from the other points of the cluster, D represents the minimum of the average distances of the point from the points of the other clusters.

Given R_{ij} is the similarity measure of clusters, d_{ij} is the cluster dissimilarity measure, and S_i is the dispersion measure of a cluster, the DB index is calculated as:

DB index:

$$DB = \frac{1}{n_c}\sum_{i=1}^{n_c} R_i \tag{4}$$

where:

$$R_i = \max_{i \neq j}(R_{ij})$$
$$R_{ij} = \frac{S_i + S_j}{d_{ij}}$$
$$d_{ij} = d(v_i, v_j)$$
$$S_i = \frac{1}{\|c_i\|}\sum_{x \in C_i} d(x, v_i)$$

CH index:

$$CH = \frac{\frac{B_k}{(k-1)}}{\frac{W_k}{(n-k)}} \tag{5}$$

K is the number of clusters and n is the total number of values, B_k represents the between-cluster sum of squares, and W_k represents the sum of squares of within-cluster [46].

DUN index:

$$DUN = \frac{\delta(C_i, C_j)}{\Delta(C_l)} \tag{6}$$

where:

$$\delta(C_i, C_j) = \{d(x_i, x_j) | x_i \in C_i, x_j \in C_j\}$$
$$\Delta(C_l) = max\{d(x_i, x_j) | x_i, x_j \in C_l\}$$

$\Delta(C_l)$ denotes the within-cluster compactness, $\delta(C_i, C_j)$ denotes the between-cluster separation, and C_l denotes the data belonging to the whole data set.

C-index:

$$C = \frac{S - S_{min}}{S_{max} - S_{min}} \tag{7}$$

where S denotes the sum of the distance of all objects from the same cluster, S_{min} is the sum of the n smallest distances if all objects are considered, S_{max} is the sum of the n largest distances.

KL-index:

$$KL = \frac{|diff_k|}{|diff_{k+1}|} \tag{8}$$

where:

$$\left|diff_k\right| = (k-1)^{\frac{2}{d}}W_{k-1} - k^{\frac{2}{d}}W_k$$

$\left|diff_k\right|$ denotes the within-cluster sums of squares of the partition, d is the number of features in the data set, and k is the number of clusters.
H-index:

$$H(K) = \left(\frac{W_k}{W_{k+1}}\right)(n-k-1) \tag{9}$$

where k is the number of clusters, n is the total number of data instance, and W_k is the within-cluster sum of squares.

This study uses the indices listed above, which are commonly used in K-means clustering to discover the optimal number of clusters in previous studies [27, 47]. In our framework, this seven-index set is used as a preliminary screening tool to reduce the search complexity of determining a suitable k, by narrowing attention to a small candidate range indicated by complementary compactness–separation diagnostics rather than conducting exhaustive end-to-end trials for every k. Although additional CVIs may further refine the selection under different data characteristics, investigating their impact is left as a natural extension; the present work therefore provides a baseline study that establishes how CVI-guided k selection propagates to cluster-aware MTD construction, attribute extension, and downstream prediction performance. The goals of each index are as follows: the S-index, CH, DUN, KL, and H-index aim to find the maximum value, while DB and C-index aim to select the minimum value, as indicated by the number of clusters in Table I.

TABLE I. THE PERFORMANCE OF EACH INDEX

Index	Criteria
S-index	Max value
DB	Min value
CH	Max value
DUN	Max value
C-index	Min value
KL	Max value

B. An Attribute Extension Function

An attribute extension function was employed to gather more information from each attribute. The initial idea was introduced in Li *et al.* [6], where the MTD function was proposed using the possibility calculation of the membership function in fuzzy set theory. Fig. 4 illustrates the application of the fuzzy theorem to the MTD function. The triangle shape represents the membership function, and values a and b are the boundaries of the MTD function. The height of samples m and n reflects the possibility values of the membership function, which fall within the range of 0 to 1. In summary, the attribute extension function employs fuzzy set theory to generate more information from each attribute through the MTD function.

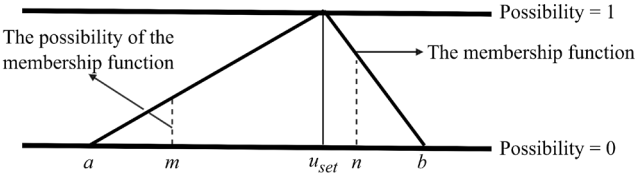

Fig. 4. The MTD function.

This paper presents the possibility of a sample belonging to each cluster by employing the MTD function. The detailed steps to build an MTD function are shown in below. Given the sample of $X = x_1, x_2, \ldots, x_n$, the boundaries a and b are defined as follows:

$$a = u_{set} - skew_L\sqrt{-2S_x^2\frac{1}{N_L}\ln(\alpha)} \tag{10}$$

$$b = u_{set} + skew_U\sqrt{-2S_x^2\frac{1}{N_U}\ln(\alpha)} \tag{11}$$

where $u_{set} = \frac{min(X)+max(X)}{2}$ and S_x^2 are the variance of X, N_L is the number of data points smaller than u_{set}, while N_u is the number of data points greater than u_{set}. $skew_L = \frac{N_L}{(N_L+N_U)}$ and $skew_U = \frac{N_U}{(N_L+N_U)}$ show the rates of skewness in the distribution of the data on each side of u_{set}. Here $\alpha \in (0,1)$ is a fixed tail threshold that specifies an "effectively-zero" membership level at the boundary; we set $\alpha = 10^{-20}$ (equivalently $\ln(\alpha) = \ln(10^{-20})$) so that $-\ln(\alpha)$ is large, yielding a stable and conservative boundary expansion while avoiding the numerical singularity at $\ln(0)$. Importantly, the factor $1/N_L$ (and $1/N_U$) makes the diffusion width increase when a cluster has very few samples, so the resulting domain $[a,b]$ expands more aggressively for small clusters. After obtaining $[a,b]$, we define the normalized triangular membership $M(x) \in [0,1]$ for any target value t, with $M(x) = 1$ at the central location and linearly decaying to 0 at a and b (and $M(x) = 0$ outside $[a,b]$). Consequently, $small - n$ clusters produce wider distribution domains and thus a softer (less confident) membership surface, which acts as an inherent regularization mechanism that mitigates sharp, unstable boundaries caused by unreliable mean/variance estimates in small-sample settings. In other words, wider MTD boundaries act as a built-in regularization mechanism, preventing overfitting and unreliable membership values when clusters contain very few samples.

The value of the membership function $M(x)$ presents the possibility value of x and is defined as follows:

$$M(x) = \begin{cases} \frac{x-a}{u_{set}-a}, a \le x \le u_{set}, \\ \frac{b-x}{b-u_{set}}, u_{set} \le x \le b \\ 0, otherwise \end{cases} \tag{12}$$

Based on the MTD distribution, $M(x)$, considering the forecast problem with K-clustering, the transformed x produced by the attribute extension is: $\psi(x) = x, M^1(x), M^2(x), \ldots, M^k(x)$. Each $M^i(x)$ for $i = 1, \ldots, k$ represents a new attribute derived from the MTD distribution. This mapping $\psi(x)$: $R \rightarrow R^{K+1}$ extends a one-

dimensional input x into an $(K + 1)$-dimensional feature vector, making it straightforward to compute the transformed representation once the functions M^i are available.

C. General Flow

The steps of the complete procedure of the proposed method are as follows: Assuming that we have a sample set $X = \{(x_1, t_1), (x_2, t_2), \ldots, (x_1, t_1)\}$ where each sample $x_i, i = 1, \ldots, N$, in X has M attributes, and t_i, are the target value of X_i. The K-means algorithm is used to compute the k clusters from based on the target value.

- Step 1: Compute the value of seven cluster validity indices for the raw data set with M attributes and N samples through K-means clustering on the target variable.
- Step 2: Assume that the optimal number of clusters in each index is K. Separate the sample set of each attribute into K clusters.
- Step 3: Start with Attribute 1 in Cluster 1 to construct the MTD function, and iterate this step for every attribute in all clusters to build up the transformation functions.
- Step 4: Start with Attribute 1 in all clusters, $X_{11}, X_{21}, \ldots, X_{N1}$, to compute the possibility of its own MTD function until all attributes have completed the calculation.
- Step 5: Combine the extended data sets and the raw data sets as the new data sets simply by concatenating. For every sample $xi \in X$ with M attributes, use the transformation function to extend the attribute from M attributes.
- Step 6: Use forecasting models (LR, SVR, and BPNN) to compare the predictive ability by using the raw data and the transformed data sets.

IV. Manufacturing Case Studies

In this section, this paper examines two manufacturing scenarios to validate the efficacy of our proposed method. The first case pertains to the TFT-LCD process, where we employ predictive techniques to control the relevant variables and ensure superior product quality by predicting cell verniers. The second case revolves around the PSH value, which represents the final step in TFT-LCD manufacturing and can impact overall production costs. These cases address common challenges encountered in the TFT-LCD industry.

A. Learning Parameter Settings and Experiment Design

The experiment is designed so that each cluster should have at least two data points, and thus the cluster processing stops when a cluster has only one such point. The parameter settings are specified in Table II.

TABLE II. Parameter Settings of the Models

SVR	BPNN	Regression	MTD
C = 1.0 RBF kernel Degree = 3 Lambda = 1e−7	Learning rate = 0.3 Momentum = 0.2 Training time = 500	Ridge = 1.0e−8	$\alpha = ln\,(10^{-20})$

The experiment is designed for small data sets, so in both cases, the data sizes are below 20. We thus use Leave-One-Out Cross-Validation (LOOCV) to ensure the experiment's validity. As the name suggests, LOOCV involves using a single observation from the original sample as the validation data, and the remaining observations as the training data. We use the average MSE, STDs, and t-tests ($p < 0.05$) to compare the extended data with the raw data using LR, SVR, and BPNN. The null hypothesis is that there is no significant difference between the prediction errors produced by the cluster-validity–extended model and the prediction errors produced by the corresponding raw-data baseline model. LR faces certain problems because the representation now becomes piecewise smooth (locally linear inside a cluster) but curved across clusters and along the manifold. We thus evaluate only the performance of BPNN and SVR for the extended data.

B. Sampling-Bias Analysis

In this paper, we conduct frequency- and coverage-bias diagnostics to examine whether the proposed attribute-extension scheme introduces sampling distortions, using a SMOTE-like synthetic sampling baseline for comparison [48]. The frequency-bias test reports the sample count (n), as well as summary statistics of the target variable y (mean, STD, and empirical quantiles) for the original data and the corresponding augmented/extended representations. The coverage-bias test evaluates the geometry of the data in the standardized original input space via nearest-neighbor distance statistics (minimum, mean, median, and the p10/p90 percentiles of the nearest-neighbor distances). Results for Case 1 (TFT-LCD quality dataset) are summarized in Tables III and IV, and results for Case 2 (PSH prediction dataset) are summarized in Tables V and VI. Overall, the proposed method yields identical frequency and coverage statistics to the original dataset in both cases, which is expected because it does not generate or reweight samples, but instead appends membership-derived attributes deterministically to the same observations. In contrast, the SMOTE-like baseline increases the effective training set size and produces measurable distortions in both the empirical target distribution and the input-space coverage statistics, as reflected by consistently smaller nearest-neighbor distances (indicating overconcentration/ near-duplicate behavior). These tests assess sampling distortions due to synthetic generation; although biases inherent in the original dataset may persist, the proposed method prevents additional frequency/coverage distortions arising from virtual-sample creation.

TABLE III. FREQUENCY BIAS TEST FOR CASE 1

No.	Method	n	mean(y)	Std(y)	Q05	Q25	Q50	Q75	Q95
0	Original	13	3.526962	0.063308	3.408300	3.517800	3.5350	3.56000	3.607140
1	Our method	13	3.526962	0.063308	3.408300	3.517800	3.5350	3.56000	3.607140
2	SMOTE-like virtual samples	43	3.526962	0.054019	3.415205	3.498715	3.5354	3.55803	3.598892

TABLE IV. COVERAGE BIAS TEST FOR CASE 1

No.	Method	n	min (d)	p10 (d)	median (d)	mean (d)	p90 (d)
0	Original	13	0.372668	0.409977	0.808787	1.010462	1.767984
1	Our method	13	0.372668	0.409977	0.808787	1.010462	1.767984
2	SMOTE-like virtual samples	43	0.029378	0.121035	0.331460	0.398982	0.832370

TABLE V. FREQUENCY BIAS TEST FOR CASE 2

No.	Method	n	mean (y)	std (y)	Q05	Q25	Q50	Q75	Q95
0	Original	19	1.319021	0.290777	0.976530	1.100000	1.269200	1.4601	1.886690
1	Our method	19	1.319021	0.290777	0.976530	1.100000	1.269200	1.4601	1.886690
2	SMOTE-like virtual samples	57	1.235550	0.209033	1.060634	1.103799	1.167969	1.2865	1.691406

TABLE VI. COVERAGE BIAS TEST FOR CASE 2

No.	Method	n	min (d)	p10 (d)	median (d)	mean (d)	p90 (d)
0	Original	19	0.583039	0.621553	1.063691	1.062482	1.589576
1	Our method	19	0.583039	0.621553	1.063691	1.062482	1.589576
2	SMOTE-like virtual samples	57	0.110446	0.149131	0.459596	0.536817	0.935779

C. *First Case (Case 1): Quality of Thin-Film Transistor Liquid-Crystal Display (TFT-LCD)*

In addition to the information mentioned earlier, it is worth noting that the accurate prediction and control of cell verniers play a vital role in ensuring the overall quality and performance of display panels. High values of cell vernier can lead to undesirable effects such as reduced contrast, degraded image quality, and decreased panel durability. Therefore, it is of utmost importance for manufacturers to effectively manage and minimize cell verniers during the production process.

The traditional approach of adjusting the Transfer Printing Entities (TPEs) in a Color Filter (CF) to match those of a TFT in the alignment process has been widely used but known to be complex and costly. By simplifying the process and assuming fixed TPE values in the TFT, the focus shifts to adjusting the CF values instead. Our approach raises an opportunity to streamline production operations and potentially reduce manufacturing costs, while still maintaining product quality.

To develop an accurate forecasting model for predicting cell verniers, the study leverages a data set comprising 19 valid yield data points. Although the data set is relatively small, it was obtained from a Statistical Process Control (SPC) database, ensuring the reliability and integrity of the collected information. Despite the limited number of observations, we aim to provide valuable insights into effectively controlling cell verniers and maintaining high-quality panel production. By analyzing the relationship between the input data X1, X4, Y1, Y6, D1, and D2, and the corresponding cell vernier values, the forecasting model seeks to identify patterns and correlations that can be used for accurate predictions. The successful development and implementation of such a model would not only enhance production efficiency but also contribute to the overall advancement of display panel manufacturing processes.

Tables AI and AII show the computations of each sample in the seven cluster validity indices. Sample1 can be separated into two, three, and four clusters. S-index, DB, CH, DUN, and H-index suggest two clusters, while KL suggests three clusters, and C-index suggests four. In this case, the largest number of clusters is six, and the smallest number is two. Table AIII summarizes the predictive performance under multiple modeling configurations, showing different performance of cluster-validity–guided attribute extension in SVR and BPNN. Note that the S-index, CH, DUN, and H-index give the same suggestion for determining clusters, and thus, we use Group A to represent these four indices. With SVR, all of the indices improve the outcome both in average MSE and STD, the C-index reaches the lowest average MSE and STD (0.20 and 0.14), and the p-value is 0.03 ($P < 0.05$). C-index with average MSE = 0.25, STD = 0.18 is also the only index that shows that superior with raw data's performance (average MSE = 0.37 and STD = 0.30) in BPNN algorithm. Other indices' outcomes show that increasing attributes contribute to the complexity of this algorithm, leading to overfit.

D. *Second Case (Case 2): the PSH of TFT-LCD*

In the domain of TFT-LCD manufacturing, CF plays a pivotal role as a vital component. The production process for CF involves the application of multiple layers, including black matrix, green, red, blue, indium tin oxide, and photo-spacer layers. Among these layers, the coating of the photo-spacer layer holds paramount importance as it determines the target PSH value of the CF. This value is contingent upon the height of the photo-spacer layer itself. Ensuring that the PSH value meets the specified requirements is crucial, as any deviations can lead to

substantial reprocessing costs, which are significantly higher than those of other layers. To mitigate these expenses, a quality control strategy employing random sampling is employed. However, the availability for this kind of dataset is extremely scarce. In this study, an extremely limited production batch of only 13 samples from SPC database was available for training and validating. Our proposed approach addresses this challenge by extracting valuable insights and relationship between available measured data and PSH value, and then perform data enrichment based on those valuable conclusions. Therefore, the CF production process can be optimized and mitigate the reprocessing costs.

Table AIV shows the complete computations in the seven indices for each sample of Case 2. The smallest number of clusters is two, while the largest number is five. Table AV shows the prediction results of the seven indices in each sample, demonstrates that cluster-validity–driven attribute extension can outperform both SVR and BPNN. DUN and H-index have the same result, which is represented as DUN H-index. In this case, all the indices indicate better results with both SVR and BPNN models. In the result of t-tests, the C-index shows a significant difference and the superiority in performance with SVR (p = 0.0253). Meanwhile, DUN H-index demonstrates the significance when working with the BPNN model (p = 0.0251), with the lowest average MSE and STD of 0.0119 and 0.0089 respectively. Notably, while C-index remains the best performer with SVR (average MSE = 0.0114, STD = 0.0081), the DUN & H-index combination yields the largest improvement for BPNN, achieving an average MSE of 0.0119 (below the baseline BPNN error of 0.0290) and being the only index to reach statistical significance (p = 0.0251). This mismatch between SVR and BPNN is expected in Case 2 (N = 13), because the benefit of attribute extension depends on both the data geometry and the learner's inductive bias. C-index emphasizes within-cluster compactness, which tends to produce smoother, less noisy feature neighborhoods that SVR can exploit effectively under strong regularization. In contrast, DUN/H are more sensitive to worst-case separation relative to cluster diameter; when the process contains clearer regime separation, these indices can yield cluster-derived attributes that a nonlinear model such as BPNN leverages more strongly.

Furthermore, under the same PSH TFT–LCD forecasting setting, our proposed method consistently outperforms the MTD–based approach of Li *et al.* [5]. Specifically, their work reported an MSE of 0.012 with an STD of 0.014, whereas our proposed method, after attribute extension using C-index–based clustering, achieves a lower MSE of 0.0114 and a substantially reduced STD of 0.0081. While the improvement in MSE indicates enhanced prediction accuracy, the pronounced reduction in STD (approximately 42%) demonstrates significantly improved stability and robustness. This result confirms that explicitly modeling cluster structure and deriving cluster-aware attributes yields more reliable forecasting performance than relying on randomly generated samples weighted by a single diffusion measure.

V. Discussion and Conclusion

This study introduces a novel approach to enhance predictive ability by extending attributes using the most suitable cluster validity indices for overcoming small data set drawbacks. A direct, dataset-level benchmark against several small-data-specific clustering methods would be desirable; however, a number of related small-data clustering approaches are not accompanied by publicly available implementations, and several reports provide insufficient low-level details (e.g., preprocessing choices, distance definitions, initialization strategies, and model-selection protocols) to reproduce results under the same small-n setting. Under these constraints, any comparison would require substantial re-implementation and validation effort and could introduce confounding differences unrelated to the methods themselves, thereby weakening the fairness of the evaluation. To mitigate this, we strengthen the empirical evidence in two ways: (i) we report consistent comparisons against widely used supervised baselines (LR, BPNN, and SVR) under the same data splits and evaluation protocol, and (ii) we emphasize that the proposed framework targets attribute extension under extremely limited labels (small-n), where robustness and stability are critical. We believe these baseline comparisons, together with the ablation and sensitivity analyses provided, offer an indirect yet practical validation of the proposed method's value in real manufacturing small-data settings.

The findings reveal that all cluster validity indices improve the average MSE and STD relative to the raw data. In Case 1, the C-index demonstrates the lowest average MSE (0.20) and STD (0.14), and it is the only index that achieves statistical significance in the t-test (p = 0.03), indicating that the improvement over raw SVR is unlikely due to chance. In Case 2, the C-index again achieves the lowest average MSE (0.0114) and STD (0.0081) within the SVR-based models, with a statistically significant improvement (p = 0.0253). However, it is important to note that within the BPNN models for Case 2, the DUN & H-index achieves the best performance rather than the C-index. This discrepancy exhibits that while the C-index consistently provides the most compact and informative cluster structure for SVR, it is not universally optimal across all learning algorithms. While Case 1 was not compared with Li *et al.* [5] due to data privacy and the unavailability of the original framework; the superiority of Case 2 has verified the method's effectiveness.

In conclusion, our proposed method, which incorporates cluster validity assessment and attribute extension, outperforms existing methods and proves effective in addressing small data set problems. Future work will expand the experimental scope by benchmarking against other additional prior methods (e.g., Li *et al.* [5]) on both datasets and by validating on further real manufacturing datasets across product types, thereby reinforcing the comparative evidence and generalizability of the proposed framework. Additional validation using real data from different product types is recommended for future research to consolidate the approach's efficacy.

APPENDIX A: RESULTS OF SEVEN CLUSTER VALIDITY INDICES COMPUTATION AND PREDICTION ERROR

TABLE AI. THE COMPUTATION OF SEVEN CLUSTER VALIDITY INDICES IN EACH SAMPLE OF CASE 1

Cluster	Sample 1			Sample 2				Sample 3				Sample 4				
	2	3	4	2	3	4	5	2	3	5	6	2	3	4	5	6
S-index	**0.49**	0.35	0.32	**0.47**	0.32	0.34	0.31	**0.47**	0.32	0.28	0.26	**0.49**	0.35	0.34	0.30	0.32
DB	0.71	**0.95**	0.87	0.75	1.00	0.75	**0.72**	**0.74**	0.97	0.81	0.88	0.71	**0.95**	0.78	0.74	0.83
CH	**29.58**	22.34	19.21	**25.93**	18.79	15.88	16.15	**26.94**	20.03	15.86	14.83	**29.94**	22.51	18.81	18.38	17.65
DUN	**2.70**	1.94	1.52	**2.52**	1.78	1.77	1.56	**2.52**	1.79	1.56	1.27	**2.68**	1.93	1.75	1.55	1.67
C-index	0.15	0.11	**0.09**	0.14	0.10	0.09	**0.09**	0.13	0.09	**0.09**	0.09	**0.14**	0.11	0.10	0.08	0.07
KL	6.82	**1.56**	0.88	**7.16**	1.41	0.74	1.84	**6.71**	1.55	1.38	1.54	**6.98**	1.77	0.81	1.50	1.52
H-index	**29.58**	5.95	4.01	**25.93**	5.07	3.59	4.62	**26.94**	5.51	3.48	2.65	**29.94**	5.90	3.60	4.20	3.06

Cluster	Sample 5				Sample 6				Sample 7				
	2	3	4	5	2	3	4	5	2	3	4	5	6
S-index	**0.51**	0.38	0.29	0.27	**0.53**	0.32	0.30	0.30	**0.48**	0.33	0.31	0.32	0.32
DB	**0.68**	0.83	**0.91**	**0.91**	**0.66**	0.85	0.89	0.78	**0.74**	**0.99**	0.79	0.80	0.78
CH	**32.17**	24.27	19.18	18.63	**32.46**	21.21	18.26	17.77	**27.68**	20.29	17.60	15.10	16.85
DUN	**2.77**	2.12	1.18	1.27	**2.97**	1.62	1.46	1.56	**2.61**	1.81	1.50	1.32	1.70
C-index	0.14	0.09	0.10	0.08	0.12	0.10	0.09	**0.08**	0.15	0.12	0.10	0.10	**0.07**
KL	**7.28**	2.53	0.58	1.42	**13.31**	0.82	0.96	1.61	**7.17**	1.33	1.88	0.43	2.82
H-index	**32.17**	6.10	2.89	4.13	**32.46**	3.96	3.97	4.12	**27.68**	5.36	4.03	2.38	5.05

Cluster	Sample 8						Sample 9			Sample 10			
	2	3	4	5	6	7	2	3	4	2	3	4	5
S-index	**0.48**	0.34	0.32	0.32	0.29	0.29	**0.49**	0.36	0.35	**0.48**	0.34	0.31	0.29
DB	**0.73**	0.96	0.76	0.75	0.85	0.74	0.71	0.93	**0.68**	**0.74**	0.97	0.90	0.90
CH	**27.99**	21.08	17.54	17.12	16.16	15.08	**29.85**	23.22	19.40	**26.42**	19.55	17.24	14.95
DUN	**2.67**	1.89	1.68	1.79	1.27	1.51	**2.71**	1.99	1.76	**2.71**	1.88	1.56	1.26
C-index	0.15	0.11	0.09	**0.07**	**0.07**	0.08	0.08	**0.14**	0.09	0.14	**0.10**	0.11	0.11
KL	**6.61**	1.79	0.79	1.61	1.38	1.38	**6.30**	1.98	1.32	**6.73**	1.27	1.86	0.86
H-index	**27.99**	5.79	3.48	4.12	2.81	2.12	**29.85**	6.44	3.62	**26.42**	5.40	4.22	2.51

TABLE AII. THE COMPUTATION OF SEVEN CLUSTER VALIDITY INDICES IN EACH SAMPLE OF CASE 1

Cluster	Sample 11				Sample 12				Sample 13			
	2	3	4	5	2	3	4	5	2	3	4	5
S-index	**0.52**	0.34	0.32	0.31	**0.47**	0.32	0.30	0.30	**0.48**	0.34	0.33	0.31
DB	**0.65**	0.80	0.77	0.73	**0.74**	**0.98**	0.82	0.75	**0.73**	0.96	0.79	0.83
CH	**33.36**	24.50	20.65	18.88	**27.44**	20.23	17.27	16.98	**28.11**	21.12	17.40	17.51
DUN	**3.04**	1.69	1.76	1.56	**2.62**	1.80	1.46	1.53	**2.67**	1.89	1.64	1.56
C-index	0.12	0.10	0.09	**0.08**	0.15	0.11	0.09	**0.08**	0.15	0.11	0.10	**0.08**
KL	**8.14**	1.58	1.14	1.11	**7.00**	1.45	0.88	1.85	**6.58**	1.88	0.67	2.00
H-index	**33.36**	5.74	3.80	3.31	**27.44**	5.43	3.80	4.21	**28.11**	5.76	3.35	4.56

Cluster	Sample 14				Sample 15					Sample 16			
	2	3	4	5	2	3	4	5	6	2	3	4	5
S-index	**0.47**	0.32	0.29	0.30	**0.49**	0.36	0.31	0.30	0.31	**0.48**	0.33	0.30	0.31
DB	**0.74**	0.79	0.92	0.75	**0.71**	0.92	0.88	0.96	0.78	**0.73**	0.97	0.90	0.74
CH	**27.36**	20.70	17.12	16.68	**28.86**	22.31	19.17	16.39	17.48	**27.82**	20.75	17.60	17.28
DUN	**2.64**	1.79	1.47	1.49	**2.70**	1.88	1.56	1.22	1.27	**2.64**	1.84	1.49	1.59
C-index	0.14	0.11	0.09	**0.07**	0.13	**0.09**	0.10	0.11	**0.09**	0.15	0.11	0.10	**0.08**
KL	**6.95**	1.51	0.87	1.84	**6.25**	1.69	1.85	0.48	2.21	**6.79**	1.57	0.85	1.63
H-index	**27.36**	5.44	3.70	4.07	**28.86**	6.26	3.99	2.38	4.44	**27.82**	5.63	3.73	4.21

Cluster	Sample 17					Sample 18		Sample 19			
	2	3	4	5	6	2	3	2	3	4	5
S-index	**0.52**	0.33	0.28	0.30	0.26	**0.47**	0.34	**0.48**	0.33	0.29	0.28
DB	**0.68**	1.01	0.91	0.83	0.85	**0.74**	0.95	**0.73**	0.95	0.80	0.80
CH	**32.42**	21.55	18.40	17.95	16.46	**26.29**	20.09	**27.73**	21.00	17.95	16.53
DUN	**2.78**	1.95	1.46	1.56	1.20	**2.63**	1.88	**2.59**	1.88	1.44	1.56
C-index	0.13	0.10	0.09	**0.08**	**0.08**	0.14	**0.10**	0.13	0.09	0.10	**0.09**
KL	**12.08**	0.94	0.91	1.90	0.97	**6.04**	2.71	**6.45**	1.60	1.15	1.26
H-index	**32.42**	4.20	3.86	4.16	2.45	**26.29**	5.88	**27.73**	5.86	3.85	3.33

TABLE AIII. THE PREDICTION ERRORS OF THE SEVEN INDICES IN EACH SAMPLE OF CASE 1

Sample No.	Cluster validity with SVR				Cluster validity with BPNN				Raw data		
	Group A	DB	C-index	KL	Group A	DB	C-index	KL	SVR	LR	BPNN
Sample1	0.61	0.61	0.59	0.62	0.45	0.45	0.40	0.21	0.64	0.43	0.33
Sample2	0.11	0.14	0.15	0.11	0.19	0.11	0.36	0.19	0.11	0.54	0.12
Sample3	0.24	0.24	0.22	0.24	0.36	0.36	0.18	0.36	0.26	0.62	1.03
Sample4	0.56	0.56	0.38	0.56	0.66	0.66	0.35	0.66	0.58	0.42	0.34
Sample5	0.24	0.24	0.18	0.24	0.17	0.17	0.50	0.17	0.35	0.04	0.98
Sample6	0.23	0.23	0.21	0.23	0.67	0.67	0.10	0.67	0.17	0.45	0.40
Sample7	0.05	0.05	0.04	0.05	0.21	0.21	0.18	0.21	0.01	0.22	0.12
Sample8	0.20	0.20	0.27	0.20	0.22	0.22	0.23	0.22	0.29	0.06	0.29
Sample9	0.24	0.33	0.33	0.24	0.05	0.31	0.31	0.05	0.38	0.17	0.01
Sample10	0.05	0.05	0.11	0.05	0.48	0.48	0.39	0.48	0.03	0.20	0.79
Sample11	0.02	0.02	0.05	0.02	0.61	0.61	0.19	0.61	0.02	0.35	0.49
Sample12	0.36	0.36	0.33	0.36	0.36	0.36	0.34	0.36	0.36	0.07	0.37
Sample13	0.35	0.35	0.23	0.35	0.55	0.55	0.12	0.55	0.26	0.36	0.08
Sample14	0.10	0.10	0.04	0.10	0.20	0.20	0.03	0.20	0.01	0.20	0.02
Sample15	0.40	0.40	0.32	0.40	0.58	0.58	0.01	0.58	0.40	0.17	0.20
Sample16	0.08	0.08	0.11	0.08	0.29	0.29	0.09	0.29	0.16	0.36	0.33
Sample17	0.02	0.02	0.02	0.02	0.64	0.64	0.10	0.64	0.09	0.36	0.13
Sample18	0.18	0.18	0.17	0.18	0.69	0.69	0.69	0.69	0.19	0.23	0.27
Sample19	0.11	0.11	0.06	0.11	0.02	0.02	0.12	0.02	0.17	0.79	0.69
Average-MSE	0.22	0.23	**0.20**	0.22	0.39	0.40	**0.25**	0.38	0.24	0.32	0.37
STD	0.17	0.17	**0.14**	0.17	0.22	0.21	**0.18**	0.22	0.18	0.19	0.30
T-test	0.25	0.42	0.03*	0.26	0.81	0.73	0.11	0.92			

Table AIII summarizes the predictive performance under multiple modeling configurations, including the seven cluster validity indices combined with SVR or BPNN, as well as baseline forecasts using raw data without attribute extension. Cluster-validity–guided attribute extension leads to consistent improvements in SVR performance across nearly all samples. Among these, the C-index exhibits the strongest improvement, achieving both the lowest average MSE (0.20) and the lowest STD (0.14). It is also the only index that reaches statistical significance ($p = 0.03$), indicating that its performance gain is unlikely to be attributable to random variation.

Collectively, the results indicate that compactness-oriented indices such as the C-index are particularly effective in small-data scenarios, as they tend to generate cluster structures that are tighter, less noisy, and therefore more informative for constructing extended features. In contrast, when paired with BPNN, the advantages of attribute extension largely diminish. Indeed, BPNN applied to raw data often outperforms all cluster-validity–extended configurations, yielding a lower average MSE (0.37) compared with the extended variants (0.38–0.40), except for the case of the C-index. This outcome reflects a limitation of neural networks in low-sample regimes: increasing feature dimensionality through extension introduces complexity that the model cannot reliably estimate, thereby amplifying overfitting.

TABLE AIV. THE COMPUTATION OF THE SEVEN CLUSTER VALIDITY INDICES IN EACH SAMPLE OF CASE 2

Cluster	Sample 1			Sample 2				Sample 3			Sample 4	
	2	3	4	2	3	4	5	2	3	4	2	3
S-index	0.4481	0.4388	0.4529	0.4704	0.4245	0.3929	0.4334	0.4644	0.3931	0.3793	0.5988	3973
DB	0.6716	0.6134	0.564	0.6584	0.6318	0.6214	0.6048	0.644	0.695	0.6597	0.4125	6861
CH	13.7786	17.5722	22.2275	15.3979	17.2184	19.5238	19.7244	15.3911	17.7085	18.8224	17.2824	.9276
DUN	2.8493	2.263	2.2395	2.9899	2.1801	2.0368	1.7674	2.9895	2.1959	2.2043	3.5216	0483
C-index	0.2321	0.1452	0.0872	0.1994	0.1346	0.0947	0.0976	0.204	0.1062	0.0956	0.1837	1108
KL	1.6773	1.7397	2.881	2.2389	1.67	2.0394	2.0394	2.1505	2.0047	2.4288	2.5941	2693
H-index	13.7786	9.5648	7.226	15.3979	8.1025	5.7934	3.3225	15.3911	8.4931	5.0627	17.2824	1742

Cluster	Sample 5			Sample 6			Sample 7			Sample 8		Sample 9	
	2	3	4	2	3	4	2	3	4	2	2	3	4
S-index	0.4357	0.4084	0.3976	0.4523	0.3483	0.3733	0.4416	0.3991	0.4188	0.4786	0.4408	0.3904	0.405
DB	0.6836	0.679	0.6555	0.6782	0.6356	0.6305	0.6753	0.5949	0.5458	0.7271	0.6834	0.5854	0.6009
CH	13.5678	17.3539	18.0604	13.8986	14.3433	17.1536	13.8362	17.29	19.1848	16.2207	13.3767	15.41	18.2494
DUN	2.727	2.1454	1.8031	2.8905	1.9375	1.7674	2.7755	2.1442	1.899	2.4303	2.795	2.1367	1.7594
C-index	0.2158	0.1027	0.1004	0.2174	0.159	0.0991	0.2178	0.1358	0.1069	0.1753	0.2246	0.1445	0.0946
KL	1.6431	2.4503	1.8947	2.2919	1.2323	2.243	1.7261	2.0736	2.6873	2.4847	1.8503	1.5642	2.5901
H-index	13.5678	9.5456	4.8039	13.8986	6.7694	6.2	13.8362	9.2832	5.5381	16.2207	13.3767	8.034	6.1821

Cluster	Sample 10			Sample 11	Sample 12			Sample 13	
	2	3	4	2	2	3	4	2	3
S-index	0.4589	0.3888	0.3876	0.3722	0.5166	0.3839	0.3771	0.3888	0.376
DB	0.6482	0.7114	0.6509	0.9054	0.6065	0.6753	0.6777	0.8435	0.4669
CH	15.0327	17.0866	18.8316	9.4605	17.8227	18.3463	18.6717	10.5515	10.8296
DUN	2.9561	2.1829	2.2766	2.0103	3.2937	2.1145	1.8312	2.2748	2.0221
C-index	0.2108	0.1082	0.0933	0.2146	0.1848	0.1157	0.0789	0.2095	0.1505
KL	2.1339	1.8044	2.6974	1.1606	2.9419	1.8322	2.0055	1.7356	1.0177
H-index	15.0327	8.2467	5.4448	9.4605	17.8227	7.4228	4.609	10.5515	5.9182

TABLE AV. THE PREDICTION ERRORS OF THE SEVEN INDICES IN EACH SAMPLE OF CASE 2

Sample No.	Cluster validity with SVR						Raw data
	S-index	DB	CH	DUN & H	C-index	KL	SVR
sample1	0.0254	0.0254	0.0254	0.0160	0.0254	0.0254	0.0320
sample2	0.0260	0.0467	0.0467	0.0260	0.0307	0.0260	0.0287
sample3	0.0051	0.0051	0.0091	0.0051	0.0091	0.0091	0.0275
sample4	0.0178	0.0178	0.0164	0.0178	0.0164	0.0178	0.0085
sample5	0.0033	0.0023	0.0023	0.0033	0.0023	0.0091	0.0021
sample6	0.0216	0.0070	0.0070	0.0216	0.0070	0.0216	0.0078
sample7	0.0217	0.0097	0.0097	0.0217	0.0097	0.0097	0.0310
sample8	0.0065	0.0065	0.0065	0.0065	0.0065	0.0065	0.0256
sample9	0.0173	0.0011	0.0071	0.0173	0.0071	0.0071	0.0060
sample10	0.0092	0.0092	0.0096	0.0092	0.0096	0.0096	0.0221
sample11	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0101
sample12	0.0256	0.0256	0.0096	0.0256	0.0096	0.0256	0.0382
sample13	0.0042	0.0131	0.0131	0.0042	0.0131	0.0042	0.0117
Average-MSE	0.0143	0.0132	0.0126	0.0136	**0.0114**	0.0134	0.0193
STD.	0.0091	0.0125	0.0114	0.0085	**0.0081**	0.0084	0.0115
T-test	0.1332	0.0879	0.0902	0.1006	0.0253*	0.0794	
Sample No.	**Cluster validity with BPNN**						**Raw data**
	S-index	**DB**	**CH**	**DUN & H**	**C-index**	**KL**	**LR**	**BPNN**
sample1	0.0384	0.0384	0.0384	0.0047	0.0384	0.0384	0.0080	0.0010
sample2	0.0121	0.0418	0.0418	0.0121	0.0169	0.0121	0.0270	0.0470
sample3	0.0143	0.0143	0.0098	0.0143	0.0098	0.0098	0.0040	0.0330
sample4	0.0155	0.0155	0.0119	0.0155	0.0119	0.0155	0.0210	0.0220
sample5	0.0093	0.0033	0.0033	0.0093	0.0033	0.0135	0.0250	0.0210
sample6	0.0212	0.0032	0.0032	0.0212	0.0032	0.0212	0.0390	0.0520
sample7	0.0036	0.0110	0.0110	0.0036	0.0110	0.0110	0.0170	0.0640
sample8	0.0138	0.0138	0.0138	0.0138	0.0138	0.0138	0.0290	0.0170
sample9	0.0111	0.0094	0.0041	0.0111	0.0041	0.0041	0.0150	0.0760
sample10	0.0061	0.0061	0.0077	0.0061	0.0077	0.0077	0.0060	0.0150
sample11	0.0036	0.0036	0.0036	0.0036	0.0036	0.0036	0.0110	0.0070
sample12	0.0185	0.0185	0.0019	0.0185	0.0019	0.0185	0.0050	0.0130
sample13	0.0203	0.0301	0.0301	0.0203	0.0301	0.0203	0.0030	0.0090
Average-MSE	0.0144	0.0161	0.0139	**0.0119**	0.0120	0.0146	0.0162	0.0290
STD.	0.0089	0.0125	0.0132	**0.0089**	0.0106	0.0087	0.0109	0.0227
T-test	0.0862	0.1353	0.0892	0.0251*	0.0601	0.0925		

In contrast, the results of Case 2 reveal a markedly different behavior. Unlike the TFT-LCD dataset, where the benefits of attribute extension were primarily confined to SVR, the second case demonstrates that cluster-validity–driven attribute extension can outperform both SVR and BPNN. Every validity index yields lower prediction errors than the raw SVR and raw BPNN models, and in several instances even surpasses linear regression. Notably, while C-index is still the best performer with SVR (average MSE = 0.0114, STD = 0.0081), the DUN & H-index combination provides the largest improvement for BPNN, achieving an average MSE of 0.0119, lower than the baseline BPNN error of 0.0290, and is the only index to reach statistical significance ($p = 0.0251$). It is implicit that the effectiveness of attribute extension is not uniform across datasets but deeply dependent on the underlying geometry of the data. Compactness-oriented validity indices like C-index tend to work best with margin-based or regularized learners such as SVR, while separation-oriented indices like DUN & H-index may be more effective for high-capacity nonlinear models such as NN when the data exhibit clear regime separation.

The dataset of Case 2 contains clearer or more separable substructures, allowing cluster-validation indices to identify partitions that meaningfully reflect the intrinsic organization of the data. These clusters generate extended attributes that are smoother, less noisy, and highly informative, so strengthening the predictive models. As a result, the additional feature dimensions do not introduce harmful variance, as they do in the small-sample neural network setting of Case 1; instead, they reduce nonlinear complexity and enable BPNN to generalize more effectively. Consequently, both SVR and BPNN benefit from the internal structure revealed through cluster validity.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Luu-Ly Tran conducted the research experiments, wrote the final manuscript; Chih-Chieh Chang provided idea and conducted methodology; Hsiang-An Yu collected and analyzed the data, wrote the first draft. All authors had approved the final version.

REFERENCES

[1] L.-S. Lin, Y.-S. Lin, D.-C. Li, S. C. Hu, C.-I. Huang, "Enhancing prediction accuracy for high-dimensional small-sample-size microarray data cancer by combining Chebyshev interpolation with new dual-net GAN," *Appl. Soft Comput.*, vol. 171, 112826, 2025. doi: 10.1016/j.asoc.2025.112826

[2] D.-C. Li and C.-W. Liu, "Extending attribute information for small data set classification," *IEEE Trans. Knowl. Data Eng*., vol. 24, no. 3, pp. 452–464, 2012. doi: 10.1109/TKDE.2010.254

[3] N. Jain, A. Olmo, S. Sengupta, L. Manikonda, and S. Kambhampati, "Imperfect ImaGANation: Implications of GANs exacerbating biases on facial data augmentation and snapchat face lenses," *Artif. Intell*., vol. 304, 103652, 2022. doi: 10.1016/j.artint.2021.103652

[4] I. M. Alkhawaldeh, I. Albalkhi, and A. J. Naswhan, "Challenges and limitations of synthetic minority oversampling techniques in machine learning," *World J. Methodol.*, vol. 13, no. 5, pp. 373–378, Dec. 2023. doi: 10.5662/wjm.v13.i5.373

[5] D.-C. Li, C.-C. Chang, C.-W. Liu, and W.-C. Chen, "A new approach for manufacturing forecast problems with insufficient data: The case of TFT–LCDs," *J. Intell. Manuf*., vol. 24, pp. 225–233, 2013. doi: 10.1007/s10845-011-0577-6

[6] D.-C. Li, C.-S. Wu, T.-I. Tsai, and Y.-S. Lin, "Using mega-trend-diffusion and artificial samples in small data set learning for early flexible manufacturing system scheduling knowledge," *Comput. Oper. Res.*, vol. 34, no. 4, pp. 966–982, 2007. doi: 10.1016/j.cor.2005.05.019

[7] Y. Cai, H. Che, B. Pan, M.-F. Leung, C. Liu, and S. Wen, "Projected cross-view learning for unbalanced incomplete multi-view clustering," *Inf. Fusion*, vol. 105, 102245, 2024. doi: 10.1016/j.inffus.2024.102245

[8] F. Zorriassatine and J. D. T. Tannock, "A review of neural networks for statistical process control," *J. Intell. Manuf*., vol. 9, no. 3, pp. 209–224, 1998. doi: 10.1023/A:1008818817588

[9] H. Arslan and M. Toz, "Hybrid FCM-WOA data clustering algorithm," in *Proc. 26th Signal Process. Commun. Appl. Conf. (SIU)*, 2018, pp. 1–4. doi: 10.1109/SIU.2018.8404531

[10] C.-C. Peng, C.-J. Tsai, T.-Y. Chang, J.-Y. Yeh, and P.-W. Hua, "A new approach to generate diversified clusters for small data sets," *Appl. Soft Comput*., vol. 95, 106564, 2020. doi: 10.1016/j.asoc.2020.106564

[11] D. Radjenović, M. Heričko, R. Torkar, and A. Živkovič, "Software fault prediction metrics: A systematic literature review," *Inf. Softw. Technol*., vol. 55, no. 8, pp. 1397–1418, 2013. doi: 10.1016/j.infsof.2013.02.009

[12] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, no. 1, 2016, pp. 1888–1894. doi: 10.1609/aaai.v30i1.10249

[13] D. M. McNeish and J. R. Harring, "Clustered data with small sample sizes: Comparing the performance of model-based and design-based approaches," *Commun. Stat. Simul. Comput*., vol. 46, no. 2, pp. 855–869, 2017, doi: 10.1080/03610918.2014.983648

[14] H. Yu, Q.-F. Wang, and J.-Y. Shi, "Data augmentation generated by generative adversarial network for small sample datasets clustering," *Neural Process. Lett.*, vol. 55, pp. 1–20, 2023. doi: 10.1007/s11063-023-11315-z

[15] K. R. Žalik and B. Žalik, "Validity index for clusters of different sizes and densities," *Pattern Recognit. Lett.*, vol. 32, no. 2, pp. 221–234, 2011. doi: 10.1016/j.patrec.2010.08.007

[16] H. Yu, Q.-F. Wang, and J.-Y. Shi, "Data augmentation generated by generative adversarial network for small sample datasets clustering," *Neural Process. Lett*., vol. 55, pp. 1–20, 2023. doi: 10.1007/s11063-023-11315-z

[17] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010. doi: 10.1016/j.patrec.2009.09.011

[18] A. Adolfsson, M. Ackerman, and N. C. Brownstein, "To cluster, or not to cluster: An analysis of clusterability methods," *Pattern Recognit.*, vol. 88, pp. 13–26, 2019. doi: 10.1016/j.patcog.2018.10.026

[19] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Stat. Theory Methods*, vol. 3, no. 1, pp. 1–27, 1974. doi: 10.1080/03610927408827101

[20] J. C. Dunn, "Well-separated clusters and optimal fuzzy partitions," *J. Cybern.*, vol. 4, no. 1, pp. 95–104, 1974. doi: 10.1080/01969727408546059

[21] J. Li, D. Zhu, and C. Li, "Comparative analysis of BPNN, SVR, LSTM, random forest, and LSTM-SVR for conditional simulation of non-Gaussian measured fluctuating wind pressures," *Mech. Syst. Signal Process*., vol. 178, 109285, 2022. doi: 10.1016/j.ymssp.2022.109285

[22] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell*., vol. PAMI-1, no. 2, pp. 224–227, 1979. doi: 10.1109/TPAMI.1979.4766909

[23] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, pp. 159–179, 1985. doi: 10.1007/BF02294245

[24] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math*., vol. 20, pp. 53–65, 1987. doi: 10.1016/0377-0427(87)90125-7

[25] W. J. Krzanowski and Y. T. Lai, "A criterion for determining the number of groups in a data set using sum-of-squares clustering," *Biometrics*, vol. 44, no. 1, pp. 23–34, 1988. doi: 10.2307/2531893

[26] F. Iglesias, T. Zseby, and A. Zimek, "Absolute cluster validity," *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 42, no. 9, pp. 2096–2112, 2020. doi: 10.1109/TPAMI.2019.2912970

[27] M. Naderipour, M. H. Fazel Zarandi, and S. Bastani, "A fuzzy cluster-validity index based on the topology structure and node attribute in complex networks," *Expert Syst. Appl.*, vol. 187, 115913, 2022. doi: 10.1016/j.eswa.2021.115913

[28] X. Duan, Y. Ma, Y. Zhou, H. Huang, and B. Wang, "A novel cluster validity index based on augmented non-shared nearest neighbors," *Expert Syst. Appl.*, vol. 223, 119784, 2023. doi: 10.1016/j.eswa.2023.119784

[29] D.-C. Li and C.-W. Liu, "A neural network weight determination model designed uniquely for small data set learning," *Expert Syst. Appl*., vol. 36, no. 6, pp. 9853–9858, 2009. doi: 10.1016/j.eswa.2009.02.004

[30] E. Carrizosa, B. Martín-Barragán, and D. Romero Morales, "Binarized support vector machines," *INFORMS J. Comput.*, vol. 22, no. 1, pp. 154–167, 2010. doi: 10.1287/ijoc.1090.0346

[31] C.-C. Chang, C.-H. Lu, M.-Y. Chang, C.-E. Shen, Y.-C. Ho, and C.-Y. Shen, "Learning to augment graphs: Machine learning-based social network intervention with self-supervision," *IEEE Trans. Comput. Soc. Syst*., vol. 11, no. 3, pp. 3286–3298, 2024. doi: 10.1109/TCSS.2023.3340230

[32] D. F. Cook, J. G. Massey, and R. E. Shannon, "A neural network to predict particleboard manufacturing process parameters," *Forest Sci*., vol. 37, no. 5, pp. 1463–1478, 1991. doi: 10.1093/forestscience/37.5.1463

[33] J. Li, D. Zhu, and C. Li, "Comparative analysis of BPNN, SVR, LSTM, random forest, and LSTM-SVR for conditional simulation of non-Gaussian measured fluctuating wind pressures," *Mech. Syst. Signal Process.*, vol. 178, 109285, 2022. doi: 10.1016/j.ymssp.2022.109285

[34] C.-H. Wang, W. Kuo, and H. Bensmail, "Detection and classification of defect patterns on semiconductor wafers," *IIE Trans.*, vol. 38, no. 12, pp. 1059–1068, 2006. doi: 10.1080/07408170600733236

[35] S.-I. Amari and S. Wu, "Improving support vector machine classifiers by modifying kernel functions," *Neural Netw*., vol. 12, no. 6, pp. 783–789, 1999. doi: 10.1016/S0893-6080(99)00085-2

[36] P. Bouboulis, S. Theodoridis, C. Mavroforakis, and L. Evaggelatou-Dalla, "Complex support vector machines for regression and quaternary classification," *IEEE Trans. Neural Netw. Learn. Syst*., vol. 26, no. 6, pp. 1260–1274, 2015. doi: 10.1109/TNNLS.2014.2368568

[37] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Adv. Neural Inf.*

Process. Syst. (NeurIPS), vol. 9, 1996, pp. 155–161. doi: 10.5555/2998981.2999003

[38] W. Wu, A.-D. Li, X.-H. He, R. Ma, H.-B. Liu, and J.-K. Lv, "A comparison of support vector machines, artificial neural network and classification tree for identifying soil texture classes in southwest China," *Comput. Electron. Agric*., vol. 144, pp. 86–93, 2018. doi: 10.1016/j.compag.2017.11.037

[39] A. Çelikyılmaz and I. B. Türkşen, "Fuzzy functions with support vector machines," *Inf. Sci*., vol. 177, no. 23, pp. 5163–5177, 2007. doi: 10.1016/j.ins.2007.06.022

[40] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986. doi: 10.1038/323533a0

[41] A. Nithya, A. Appathurai, N. Venkatadri, D. R. Ramji, and C. A. Palagan, "Kidney disease detection and segmentation using artificial neural network and multi-kernel k-means clustering for ultrasound images," *Measurement*, vol. 149, 106952, 2020. doi: 10.1016/j.measurement.2019.106952

[42] J. MacQueen, "Classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab*., 1967, pp. 281–297.

[43] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.

[44] L. J. Hubert and J. R. Levin, "A general statistical framework for assessing categorical clustering in free recall," *Psychol. Bull*., vol. 83, no. 6, pp. 1072–1080, 1976. doi: 10.1037/0033-2909.83.6.1072

[45] J. A. Hartigan, *Clustering Algorithms*, New York, NY, USA: Wiley, 1975.

[46] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1650–1654, 2002. doi: 10.1109/TPAMI.2002.1114856

[47] G. Stegmayer, D. H. Milone, L. Kamenetzky, M. G. López, and F. Carrari, "A biologically inspired validity measure for comparison of clustering methods over metabolic data sets," *IEEE/ACM Trans. Comput. Biol. Bioinf*., vol. 9, no. 3, pp. 706–716, 2012. doi: 10.1109/TCBB.2012.10

[48] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res*., vol. 16, no. 1, pp. 321–357, Jun. 2002. doi: 10.5555/1622407.1622416

Copyright © 2026 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).