Automatic Recommendation of Distance Metric for k-Means Clustering: A Meta-Learning Approach

Mark Edward M. Gonzales*, Lorene C. Uy*, Jacob Adrianne L. Sy*, Macario O. Cordel II†
*Department of Software Technology, College of Computer Studies, De La Salle University, Manila, Philippines
†Department of Computer Technology, College of Computer Studies, De La Salle University, Manila, Philippines
{mark_gonzales, lorene_c_uy, jacob_adrianne_sy, macario.cordel}@dlsu.edu.ph

***Abstract*—The choice of distance metric impacts the clustering quality of centroid-based algorithms, such as k-means. Theoretical attempts to select the optimal metric entail deep domain knowledge while experimental approaches are resource-intensive. This paper seeks to address this problem by employing a meta-learning approach to automatically recommend a distance metric for k-means clustering that optimizes the Davies-Bouldin score. Three distance measures were considered: Chebyshev, Euclidean, and Manhattan. General, statistical, information-theoretic, structural, and complexity meta-features were extracted, and random forest was used to construct the meta-learning model; borderline SMOTE was applied to address class imbalance. The model registered an accuracy of 70.59%. Employing Shapley additive explanations, it was found that the mean of the sparsity of the attributes has the highest meta-feature importance. Feeding only the top 25 most important meta-features increased the accuracy to 71.57%. The main contribution of this paper is twofold: the construction of a meta-learning model for distance metric recommendation and a fine-grained analysis of the importance and effects of the meta-features on the model's output.**

***Index Terms*—Meta-learning, meta-features, k-means, clustering, distance metric, random forest.**

I. Introduction

A staple unsupervised task in machine learning is clustering; the segregation of data points into groups has been leveraged for several domain-specific tasks, such as grouping search results for information retrieval, detecting atmospheric trends in meteorology, analyzing genomes in biology, and identifying customer segments in business [1].

However, a problem with the current machine learning pipeline is that each stage involves a selection from a wide array of alternatives, such as the choice of the clustering algorithm [2], [3] and the number of clusters [4]. In the specific case of centroid-based clustering algorithms, such as the k-means algorithm, another important consideration is the choice of the distance measure or metric, as it is pivotal in determining the cluster assignment of a data point. The impact of the distance metric on the overall clustering quality has also been reported in several studies [5], [6], [7], [8], [9].

In practice, although the Euclidean or L_2 distance is the most widely used distance metric [10], it is not suitable for every problem. Specifically, in view of the "curse of high dimensionality," the performance of L_k norms (e.g., Manhattan and Euclidean distances) is sensitive to the value of k as the number of dimensions increases, with the Euclidean distance becoming less preferable both theoretically and empirically when applied to high-dimensional datasets [11]. Although the optimal distance measure for a specific clustering task may be selected through theoretical or experimental methods, the former approach hinges on deep domain expertise and understanding of the geometry of the dataset, while the latter demands a significant amount of time and resources [12].

A possible approach to address this is via meta-learning. Derived from the idea of "learning to learn," meta-learning is a subfield of machine learning that explores the automatic recommendation of algorithms. In the context of clustering, this idea has been applied to recommend not only the most suitable algorithm but also the number of clusters [4] and the cluster validity index [13].

However, there exists a gap in the use of meta-learning for the recommendation of distance measures. While this idea was explored by Zhu et al. [12], who published one of the earliest studies in this domain and built a meta-learning model that performed better compared to defaulting to fixed or arbitrarily chosen distance metrics, their study opens interesting directions for further improvement.

The present study seeks to augment the work of Zhu et al. [12] by exploring a wider set of meta-features combined from other works [14], [15]. In particular, this study investigates the use of *(i)* general, *(ii)* statistical, *(iii)* information-theoretic, *(iv)* structural, and *(v)* complexity meta-features in building a random forest model that automatically recommends a distance metric for k-means clustering. Additionally, it attempts to provide a fine-grained analysis of the importance and effects of the meta-features on the output of the meta-learning model.

The rest of this paper is organized as follows. Section II gives a brief survey of related works. The methodology is discussed in Section III. Section IV provides an evaluation and fine-grained analysis of the model's performance. Finally, the conclusion and some prospects for future research are presented in Section V.

II. Related Works

This section situates this study in the context of prior studies related to the impact of the distance metric on clustering tasks, as well as the application of meta-learning to clustering.

A. Distance Metric and Clustering

Data clustering is a well-utilized technique for organizing data into meaningful subgroups on the basis of their intrinsic properties. The similarity of data points is measured and

subsequently used in determining the formation of these subgroups [16]. Since similarity measurement is typically based on distance, choosing the appropriate distance metric plays an important role in clustering. However, determining the optimal distance measure for a particular task may be challenging due to the sheer number of options and their specificity to the geometry and characteristics of the dataset [17].

An empirical study conducted by Bora and Gupta [16] using Manhattan, Euclidean, cosine, and correlation distances as the distance metrics for clustering two small datasets using *k*-means. It was found that the use of correlation distance yielded the best clustering quality although its downside was the slower runtime performance compared to the less computationally expensive measures, such as Manhattan distance.

Meanwhile, in the work of Giancarlo et al. [18], experiments were conducted on six benchmark datasets for microarray data analysis, and nine distance metrics were evaluated in relation to four clustering algorithms: *k*-means, average link, complete link, and minimum spanning tree. Their results showed variations between the clustering qualities depending on the distance measure used. Their study also empirically demonstrated the suitability of Minkowski, cosine, and Pearson distances to microarray data clustering.

B. Meta-Learning for Clustering

The recent advancements in machine learning and the growing interest among ML specialists and practitioners have driven the development of meta-learning approaches to improve the performance of clustering tasks by optimizing the algorithm and parameter selection. Jilling and Alvarez [2] developed an algorithm recommendation system that leverages two artificial neural networks (ANNs) that analyze the distance-based and statistical meta-features of datasets and ranks the different clustering algorithms, which include average agglomerative, affinity propagation, Gaussian mixture, OPTICS, and *k*-means clustering based on accuracy and runtime; the first ANN predicts the accuracy while the second attempts to predict the runtime.

Pimentel and De Carvalho [3] also implemented another clustering algorithm recommendation system focusing on ten algorithms: average agglomerative clustering, complete agglomerative clustering, fuzzy *c*-means, Gaussian mixture with diagonal matrix, Gaussian mixture with full matrix, kernel *k*-means, *k*-means, *k*-medoids, mini-batch *k*-means, and ward agglomerative clustering. These algorithms were further ranked on a meta-level using *k*-medoids, *k*-means, and multivariate fuzzy *c*-means in light of statistical, evaluation, and correlation-based meta-features.

Noting that the number of clusters is another important parameter in classification tasks, Pimentel and De Carvalho [4] developed a meta-learning approach to determine the optimal number of clusters given statistical, clustering-based, distance-based, evaluation-based, correlation-based, and density-based meta-features. To this end, they compared different regression models such as k-nearest neighbors, support vector machine with a radial basis function kernel, and random forest.

Zhu et al. [12] proposed an automatic distance metric recommendation algorithm for *k*-means and CURE (clustering using representatives). They extracted a total of 41 statistical, information-theoretic, structural, and distance-based meta-features extracted from 199 OpenML datasets and considered nine distance metrics as candidates: Manhattan, Euclidean, Chebyshev, standardized Euclidean, Canberra, Mahalanobis, cosine, adjusted cosine, and Pearson correlation distance.

Two meta-learning models were constructed: one for predicting the optimal distance measure (using a random forest classifier) and another for ranking the distance measures (using k-nearest neighbors). Although the former registered F1 scores between 30% and 40% and the latter recorded a Spearman's rank correlation coefficient of only 0.2, the model achieved a mean recommendation accuracy that is significantly higher compared to fixed or randomly selected distance metric selection. This corpus of studies presents ample opportunities for the exploration of a meta-learning approach, particularly towards the improvement of automatic distance metric recommendation.

III. Methodology

This section discusses the methodology of the study, from data collection and preprocessing, data labeling, and meta-feature extraction to model construction. An overview of the entire process is shown in Figure 1.

A. Data Collection and Preprocessing

A total of 340 datasets were collected from multiple sources:

- 195 from the collection of datasets published by Pimentel [19] in OpenML [20]
- 60 from the University of California Irvine (UCI) Machine Learning Repository [21]
- 85 from Kaggle [22]

The OpenML datasets have already been used in previous meta-learning studies related to clustering [2], [23], [12], [4]. Meanwhile, the UCI Machine Learning Repository and Kaggle datasets were chosen from among those tagged by these platforms as suitable for clustering. Although clustering is an unsupervised task, the datasets were restricted to those with ground-truth assignments with the intention of using the number of labels as the basis for the number of clusters; this approach is consistent with other meta-learning studies [23].

The size of this study's collection of datasets also presents an improvement over those in most previous studies on meta-learning in relation to clustering (Table I).

TABLE I
SIZE OF DATASET COLLECTIONS IN EXISTING META-LEARNING STUDIES RELATED TO CLUSTERING

Study	Dataset Size
Pimentel & De Carvalho [3]	57
Jilling & Alvarez [2]	135
Zhu et al. [12]	199
Pimentel & De Carvalho [23]	219
Pimentel [4]	219
Present Study	**340**

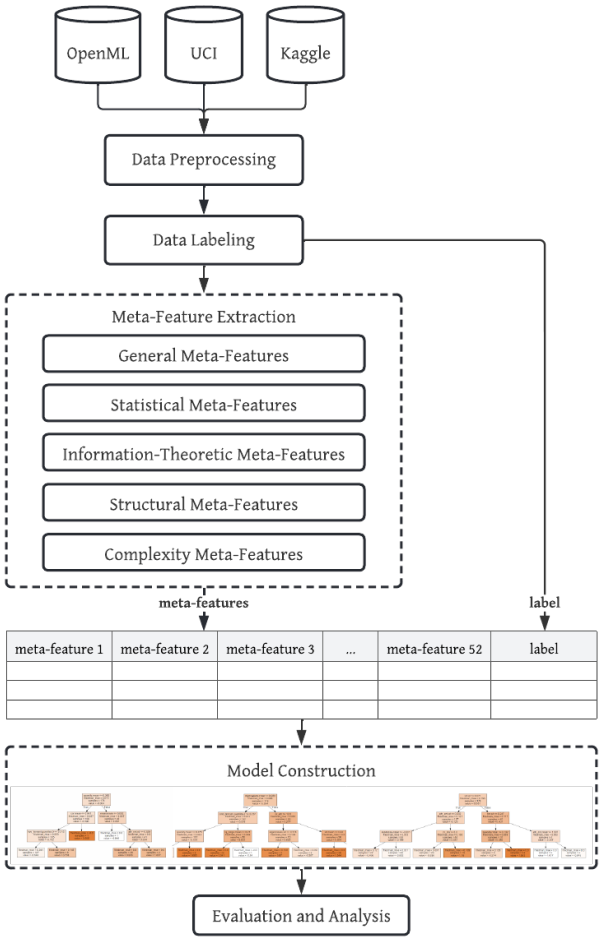


Fig. 1. Overview of the Methodology. The methodology of the study consists of data collection and preprocessing, data labeling, meta-feature extraction, and model construction. The built model is then evaluated and subjected to fine-grained analysis.

For the data preprocessing, imputation was done using either the mean (for numerical data) or mode (for categorical data). Categorical data were then converted to numerical data via one-hot encoding. Consistent with the methodology of other meta-learning studies on clustering [2], [4], [12], [23], the values in the datasets were scaled to the interval $[0, 1]$ via min-max normalization.

B. Data Labeling

k-means clustering, coupled with a grid search over selected distance measures, was performed to label each dataset in the collection with the distance metric that optimizes the Davies-Bouldin score (DBS); ties were broken by giving preference to the metric with the lower runtime. The number of clusters was decided based on the number of ground-truth labels.

Although the datasets in the collection have ground-truth assignments, clustering is an unsupervised task, and the ground-truth assignments may not be available in most real-world use cases [24], these considerations motivated the usage of an internal validation index, specifically DBS. Defining similarity as the ratio of the intracluster to intercluster distances, DBS measures the average similarity of a cluster with its most similar cluster [25]. Therefore, lower values are indicative of higher clustering quality.

Formally, given k clusters, let C_i refer to the i^{th} cluster, c_i be the center of C_i, n_i be the number of data points in C_i, and $d(x, y)$ be the distance between x and y. The DBS is given by Expression 1.

$$\frac{1}{k}\sum_{i}\max_{j,j\neq i}\left\{\frac{\frac{1}{n_i}\sum_{x\in C_i} d(x,c_i)+\frac{1}{n_j}\sum_{x\in C_j} d(x,c_j)}{d(c_i,c_j)}\right\} \quad (1)$$

Following the methodology in the work of Zhu et al. [12], eight distance measures were initially considered (standardized Euclidean distance was excluded since the data values were already scaled as part of the preprocessing). However, these resulted in severe class imbalance as shown in Table II. To mitigate this, the set of distance measures was restricted to only the top three metrics with the most number of instances, and the datasets were relabeled. The final number of datasets under each distance metric is also reported in Table II.

TABLE II
NUMBER OF DATASETS UNDER EACH DISTANCE METRIC

Distance Metric	Before Relabeling	After Relabeling
Chebyshev	123 (36.18%)	139 (40.88%)
Euclidean	88 (25.88%)	122 (35.89%)
Manhattan	57 (16.76%)	79 (23.23%)
Canberra	20 (5.88%)	–
Cosine	17 (5.00%)	–
Mahalanobis	17 (5.00%)	–
Pearson	9 (2.65%)	–
Adjusted Cosine	9 (2.65%)	–

Let x_i and x_j be two observations in the dataset $\{a_{n\times m}\}$, where n is the number of observations and m is the number of features. The formulae for Chebyshev (L_∞ norm), Euclidean (L_2 norm), and Manhattan (L_1 norm) distances are given in Equations 2, 3, and 4, respectively.

$$\text{distance}_{\text{chebyshev}}(x_i, x_j) = \max_{1\leqslant k\leqslant m}\{|a_{ik} - a_{jk}|\} \quad (2)$$

$$\text{distance}_{\text{euclidean}}(x_i, x_j) = \sqrt{\sum_{k=1}^{m}(a_{ik} - a_{jk})^2} \quad (3)$$

$$\text{distance}_{\text{manhattan}}(x_i, x_j) = \sum_{k=1}^{m}|a_{ik} - a_{jk}| \quad (4)$$

C. Meta-Feature Extraction

Meta-features from previous works on meta-learning [12], [14], [15] were selected based on their applicability to unsupervised tasks. In total, 52 meta-features were extracted from each of the datasets, as enumerated in Table III; these are classified into five categories:

- *General.* These meta-features describe the dimensionality and size of the dataset [14].

TABLE III
META-FEATURES EXTRACTED*

Category	No. of Meta-Features	Meta-Features	
		Abbreviation	Description
General	5	Attr-to-Inst Ratio	Ratio between the number of attributes and instances [26]
		Inst-to-Attr Ratio	Ratio between the number of instances and attributes [27]
		Num Attr	Number of attributes [28]
		Num Binary Attr	Number of binary attributes [28]
		Num Instances	Number of instances [28]
Statistical	32	Canonical Corr †	Canonical correlations of data [29]
		Correlation †	Absolute value of the correlation of distinct dataset column pairs [30]
		Covariance †	Absolute value of the covariance of distinct dataset attribute pairs [30]
		Eigenvalues †	Eigenvalues of covariance matrix from dataset [31]
		IQ Range †	Interquartile range (IQR) of each attribute [32]
		Kurtosis †	Kurtosis of each attribute [28]
		Median Abs Dev †	Median Absolute Deviation (MAD) adjusted by a factor [31]
		Mean †	Mean value of each attribute [33]
		Median †	Median value from each attribute [33]
		Num Correlated Attr	Number of distinct highly correlated pair of attributes [34]
		Num Outliers	Number of attributes with at least one outlier value [35]
		SD †	Standard deviation of each attribute [33]
		Skewness †	Skewness for each attribute [28]
		Sparsity †	(Possibly normalized) sparsity metric‡ for each attribute [34]
		Trimmed Mean †	Trimmed mean of each attribute [33]
		Variance †	Variance of each attribute [30]
Information-Theoretic	2	Concentration Coeff †	Concentration coefficient of each pair of distinct attributes [36]
		Shannon's Entropy †	Shannon's entropy for each predictive attribute [28]
Structural	10	1-Itemset Min, Q1, Q2, Q3, Max	Minimum, first quartile, second quartile, third quartile, and maximum of one itemset meta-feature [37]
		2-Itemset Min, Q1, Q2, Q3, Max	Minimum, first quartile, second quartile, third quartile, and maximum of two itemset meta-feature [37]
Complexity	3	Ave Num Feat per PCA Dim	Average number of features per PCA dimension [38]
		Ave Num PCA Dim per Point	Average number of PCA dimensions per points [38]
		PCA-to-Orig Dim Ratio	Ratio of the PCA dimension to the original dimension [38]

* The meta-feature descriptions are taken from the API documentation [39] of the package PyMFE [40]

† The mean and standard deviation (abbreviated as SD) of these meta-features across all attributes were extracted.

‡ Let n be the number of instances in the dataset and $\phi(a)$ be the number of distinct values under the attribute a. The sparsity metric is given by $\frac{1}{n-1}\left(\frac{n}{\phi(a)}-1\right)$. It is measure of the degree of discreteness [14].

- *Statistical.* These meta-features capture characteristics related to feature interdependence, normality, degree of discreteness, and noisiness [14].
- *Information-Theoretic.* These meta-features quantify feature informativeness and interdependence [14], [30].
- *Structural.* These meta-features capture patterns, statistics, and correlation information from the frequencies of k-itemsets [37].
- *Complexity.* These meta-features pertain to attributes that are related to the principal component analysis (PCA) dimensions [38].

The resulting dataset created from representing each dataset in the collection as a vector of meta-features and labeling it with the optimal distance metric is henceforth referred to as the *meta-feature dataset*.

D. Model Construction

Framing the recommendation of the optimal distance metric as a multiclass classification problem posits the vector of meta-features as the input and the distance metric as the output.

The meta-feature dataset was subjected to a 70%-30% stratified train-test split, resulting in the training and test sets having sizes of 238 and 102, respectively. Following Zhu et al. [12], the meta-features were fed to a random forest classifier (henceforth referred to as the *meta-learning model*). Random forest is an ensemble method that employs bootstrap aggregation (bagging) to combine multiple decision trees. As such, it is considered robust to noise and overfitting [41] and known to perform well on small datasets [42], thus providing a preliminary justification for the suitability of this machine learning algorithm to the present task.

Finally, hyperparameter tuning was conducted via grid search with five-fold stratified cross-validation to optimize the model's accuracy (which, in the case of multiclass classification tasks, is equivalent to the micro-F1). To attempt to address the problem of data imbalance, class balancing was performed within each fold; three techniques were explored to this end: *(i)* synthetic minority oversampling technique (SMOTE), *(ii)* borderline SMOTE, and *(iii)* adaptive synthetic algorithm (ADASYN).

SMOTE is a technique utilized to address class imbalance by augmenting the minority class with synthetic data that aids in adding new information to the model [43]; however, outliers in the minority class that appear in the majority class may be oversampled. To address this concern, borderline SMOTE disregards normal and noisy points of the minority class, and

only generates synthetic data based on the data points that can be found in the border between classes [44]. ADASYN provides a generic framework that uses a weighted distribution according to the difficulty of learning for each minority class, reducing the bias from class imbalance and allowing the decision boundary to shift towards the difficult examples [45].

IV. Results and Analysis

This section presents the results of the evaluation of the model's performance alongside a fine-grained analysis of the importance and effects of the meta-features on its output.

A. Model Evaluation

The models were evaluated based on their accuracy (micro-F1) and macro- and micro-averaged F1, precision, and recall. Accuracy gives equal weight to each observation. On the other hand, the macro-averaged metrics give equal weight to each class (distance metric). Meanwhile, the weighted metrics adjust the evaluation according to the number of instances under each class. The performance of the built meta-learning models are presented in Table IV.

TABLE IV
Performance of the Meta-Learning Models

	SMOTE	Borderline SMOTE	ADASYN
Accuracy (Micro-F1)	63.73%	**70.59%**	65.69%
Macro-F1	60.29%	**67.86%**	63.01%
Macro-Precision	60.78%	**67.95%**	63.06%
Macro-Recall	60.32%	**67.92%**	63.10%
Weighted F1	63.52%	**70.35%**	65.43%
Weighted Precision	63.90%	**70.29%**	65.34%
Weighted Recall	63.73%	**70.59%**	65.69%

The hyperparameter space for tuning the random forest meta-learning model is as follows (the optimal hyperparameters are given in bold):

- *Number of trees:* 10, **50**, 100, 150
- *Splitting criterion:* **Gini impurity**, Information entropy
- *Maximum depth:* 5, **15**, 25, 25
- *Minimum number of samples to be leaf node:* 1, 2, **3**, 4
- *Minimum number of samples to split internal node:* 1, **2**, 3, 4
- *Number of features to consider at each split:* **$\log_2$ of number of features**, square root of number of features
- *Warm start:* **True**, False
- *Minimum impurity decrease:* **0.0**, 0.5, 1.0
- *Complexity parameter α for minimal cost-complexity pruning:* **0.0**, 0.5, 1.0

B. Feature Importance

Shapley Additive Explanations (SHAP) [46], a game-theoretic and model-agnostic approach for interpreting the outputs of a machine learning model, was used to analyze the importance of the features. SHAP addresses the challenge of interpreting an ensemble model by approximating it using a linear explanation model. Formally, given M simplified input features, the simplified input vector $z' \in \{0,1\}^M$, and the Shapley value ϕ_j for the j^{th} feature, the explanation model $g(z')$ is provided in Equation 5.

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z_j^{'} \tag{5}$$

The formula for computing the Shapley values can be found in Shapley's original paper [47]. The sign of the Shapley value and, by extension, the SHAP value denotes the direction of the feature's contribution, i.e., a positive value indicates a positive impact while a negative value indicates a negative impact on the prediction. Hence, the global importance of the j^{th} feature (denoted as I_j) is the mean of the absolute values of the SHAP value per feature across the dataset.

Given a dataset of size n, Equation 6 presents its mathematical formulation.

$$I_j = \frac{1}{N} \sum_{j=1}^{N} \mid \phi_j^{(i)} \mid \tag{6}$$

Figure 2 plots the top meta-features in decreasing order of global feature importance (denoted as I). The top five meta-features are the mean of the sparsity (I = 0.0789), number of binary features (I = 0.0633), mean of Shannon's entropy (I = 0.0557), standard deviation of the variance (I = 0.0467), and standard deviation of the eigenvalues (I = 0.0451). Among the top 15 meta-features, one is categorized under structural (third quartile of the one-itemset frequencies), one under general (number of binary attributes), and one under information-theoretic (mean of Shannon's entropy); the remaining 12 are statistical meta-features.

On the other hand, the five meta-features with the lowest global feature importance are the minimum among the one-itemset frequencies (I = 0.0068), mean of the canonical correlations (I = 0.0074), maximum among the one-itemset frequencies (I = 0.0074), standard deviation of the canonical correlations (I = 0.0077), and first quartile of the two-itemset frequencies (I = 0.0081).

A fine-grained analysis was performed by limiting the domain of the computation of the SHAP values to a specific class (distance metric), as seen in Figures 3 to 5.

The five most important meta-features for predicting Chebyshev distance are the mean of sparsity (I = 0.0431), number of binary attributes (I = 0.0311), mean of Shannon's entropy (I = 0.0261), standard deviation of the variance (I = 0.0240), and third quartile of the one-itemset frequencies (I = 0.0200). For Euclidean distance, these are the standard deviation of the eigenvalues (I = 0.0209), mean of the skewness (I = 0.0209), mean of the sparsity (I = 0.0190), number of binary attributes (I = 0.0169), and mean of Shannon's entropy (I = 0.0150). Lastly, for Manhattan distance, these are the mean of the sparsity (I = 0.0239), mean of the skewness (I = 0.0176), number of binary attributes (I = 0.0167), number of correlated attributes (I = 0.0159), and mean of the kurtosis (I = 0.0145).

Therefore, it can be observed that, aside from their high global contribution, the mean of the sparsity and the number

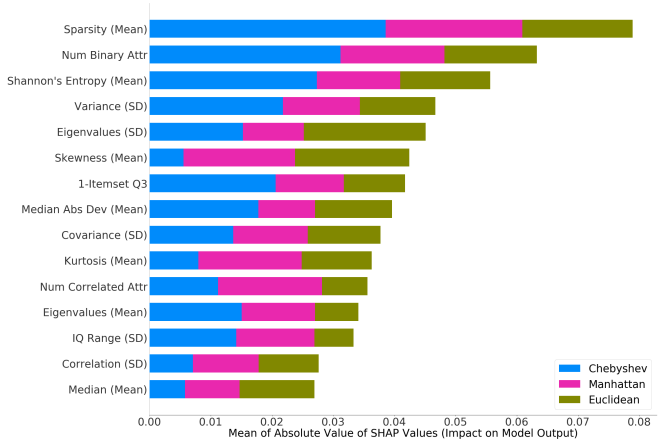


Fig. 2. Top 15 Meta-Features with the Highest Global Feature Importance. The length of each color-coded segment corresponds to the absolute magnitude of the feature's importance relative to each class. For instance, the mean of the skewness does not contribute to the prediction of Chebyshev distance as much as it does for the other two distance measure.

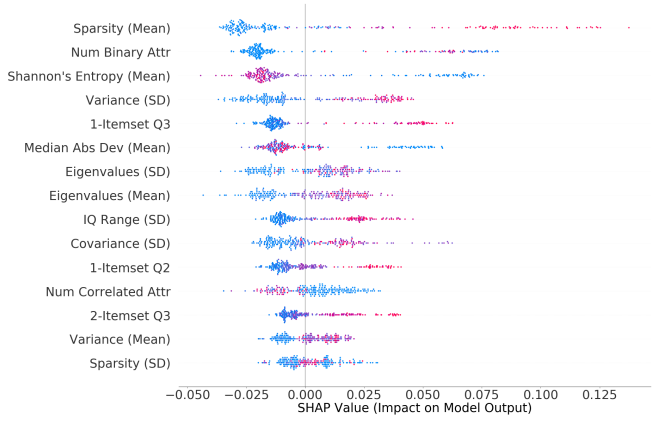


Fig. 3. Top 15 Meta-Features with the Highest Feature Importance Relative to the Prediction of Chebyshev Distance. Each point corresponds to the SHAP value of a meta-feature for an instance. Overlapping points are jittered vertically, providing some insight into the distribution of the SHAP values. The color indicates the (actual) value of the meta-feature; blue corresponds to lower values while red corresponds to higher values.

of binary attributes are also consistently among the top five meta-features with the highest importance in relation to each of the distance measures.

C. Feature Effects

The beeswarm visualizations in Figures 3 to 5 already provide preliminary insights on the relationship between the magnitudes of the (actual) values of the meta-features and their contribution to the likelihood of a prediction (as quantified via the SHAP values). Following Molnar [48], dependence plots were generated to corroborate and investigate these trends.

As seen in Figure 6, the dependence plots suggest that higher values of the mean of the sparsity, standard deviation of the standard deviation, and third quartile of the two-itemset frequencies relate to a higher probability of the built meta-learning model classifying an instance under Chebyshev distance. On the contrary, higher values of the mean of Shannon's entropy were found to contribute negatively to this probability.

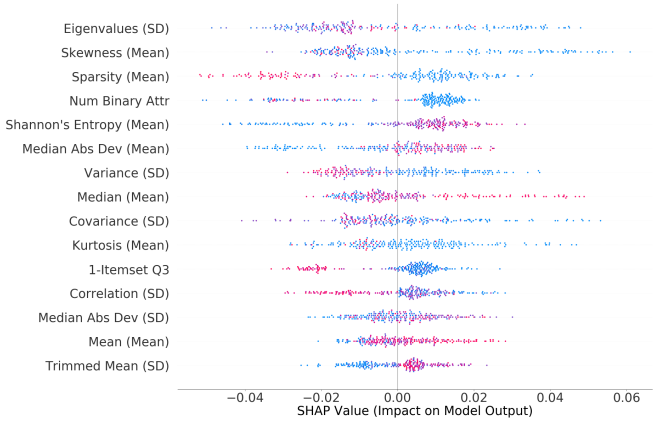


Fig. 4. Top 15 Meta-Features with the Highest Feature Importance Relative to the Prediction of Euclidean Distance. Refer to the caption of Figure 3 for an explanation of the plot.

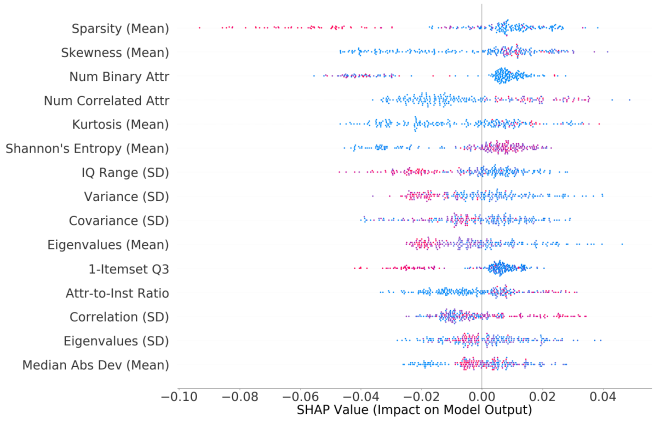


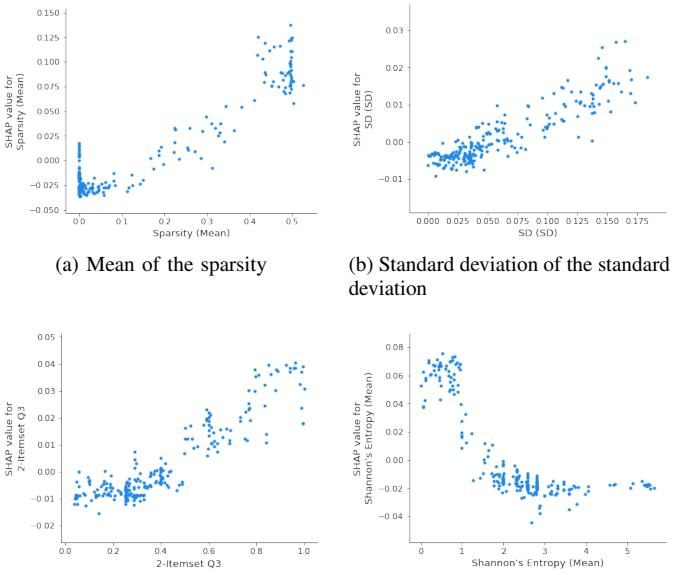
Fig. 5. Top 15 Meta-Features with the Highest Feature Importance Relative to the Prediction of Manhattan Distance. Refer to the caption of Figure 3 for an explanation of the plot.

Figure 7 shows that higher values of the third quartile of the one-itemset frequencies and the third quartile of the two-itemset frequencies are associated with a decrease in the likelihood of classifying an instance under Euclidean distance.

The same negative relationship can be observed between higher values of the mean of the eigenvalues and the likelihood of classifying an instance under Manhattan distance. Meanwhile, a positive relationship exists between higher values of the mean of the canonical correlations and the said prediction probability (Figure 8).

D. Feature Ablation

To determine how the removal of features would affect the performance of the built meta-learning model, ablation experiments were conducted, with the meta-features ranked based on their global feature importance number (average of the absolute values of the SHAP values) and the number of features incrementally reduced by one by removing the least important features.



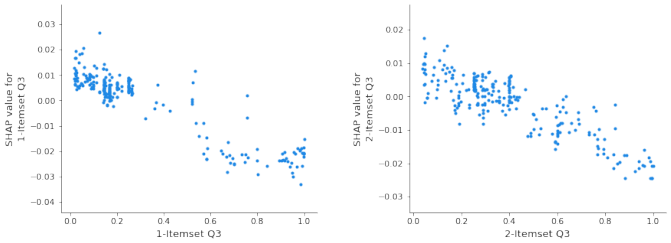
(a) Mean of the sparsity

(b) Standard deviation of the standard deviation

(c) Third quartile of the two-itemset frequencies

(d) Mean of Shannon's entropy

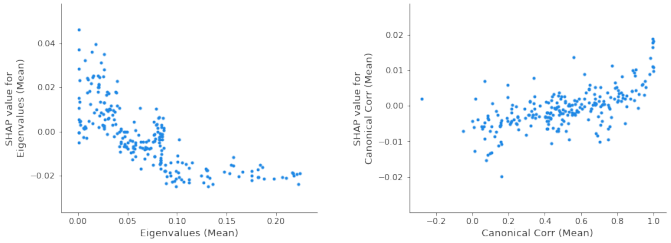
Fig. 6. SHAP Dependence Plots of Selected Meta-Features for the Prediction of Chebyshev Distance



(a) Third quartile of the one-itemset frequencies

(b) Third quartile of the two-itemset frequencies

Fig. 7. SHAP Dependence Plots of Selected Meta-Features for the Prediction of Euclidean Distance



(a) Mean of the eigenvalues

(b) Mean of the canonical correlations

Fig. 8. SHAP Dependence Plots of Selected Meta-Features for the Prediction of Manhattan Distance

The highest mean validation accuracy after performing five-fold stratified cross-validation was achieved by feeding only the top 25 meta-features to the built meta-learning model, corresponding to almost half of the entire meta-feature set. The performance of this model on the test set is reported in Table V.

TABLE V
PERFORMANCE OF THE META-LEARNING MODEL AFTER FEATURE ABLATION

	All 52 Meta-Features	Top 25 Meta-Features
Accuracy (Micro-F1)	70.59%	**71.57%**
Macro-F1	67.86%	**68.06%**
Macro-Precision	67.95%	**68.77%**
Macro-Recall	**67.92%**	**67.92%**
Weighted F1	70.35%	**70.92%**
Weighted Precision	70.29%	**70.71%**
Weighted Recall	70.59%	**71.57%**

E. Error Analysis

As seen in the confusion matrix (Figure 9), most of the misclassifications were instances under Manhattan distance that were classified under Euclidean distance. While Borderline SMOTE was applied to generate synthetic samples in an attempt to address the problem of class imbalance, this result is reflective of the underrepresentation of Manhattan distance in the dataset.

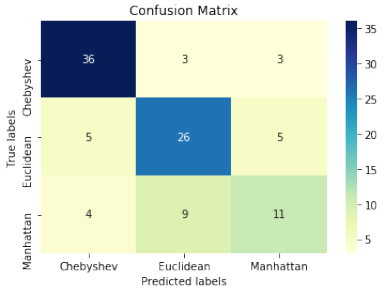


Fig. 9. Confusion Matrix of the Model. The F1 scores on Chebyshev, Euclidean, and Manhattan distances are 82.76%, 70.27%, and 51.16%, respectively. The precision scores are 80.00%, 68.42%, and 57.89%, respectively. The recall scores are 85.71%, 72.22%, and 45.83%, respectively.

Consistent with the game-theoretic foundations of SHAP, it is possible to localize the interpretation of the model's output to a single instance and maintain consistency with the global interpretation by modeling prediction as a game where each meta-feature either increases or decreases the likelihood of a prediction [48]. This framing of the SHAP values was exploited to analyze the misclassifications in the test set.

In particular, the SHAP explanation model suggests that the attributes driving some instances to be incorrectly predicted under Chebyshev distance are high values of the mean of the sparsity and standard deviation of the variance, which are also the meta-features with the highest and fourth-highest global importance, respectively. Additionally, low values of the mean of the kurtosis may be associated with some instances

under Manhattan distance being erroneously classified under Chebyshev distance.

Meanwhile, it was found that low values of the mean of the skewness may have pushed some of the misclassified data points to be predicted under Euclidean distance. Moreover, low standard deviations of the correlation and the eigenvalues may be related to some instances under Chebyshev distance being misclassified under Euclidean distance. For data points under Manhattan distance that were incorrectly predicted to be under Euclidean distance (which constitute the bulk of the model's confusion), the driving attributes include low values of the mean of the sparsity and low standard deviations of the variance and interquartile range.

Finally, low values of the average number of features per PCA dimension (a complexity meta-feature) may have pushed some instances to be erroneously classified under Manhattan distance. Low values of the mean of the sparsity may also be associated with some instances under Chebyshev distance being misclassified under Manhattan distance. Low standard deviations of the variance and high standard deviations of the correlation were found to have contributed to some data points under Euclidean distance to be incorrectly predicted under Manhattan distance.

F. Hypothesis Testing

To further evaluate the performance of the built meta-learning model, its mean recommendation accuracy (RA) was compared with the mean RA values if fixed and randomly chosen distance measures were selected. The RA compares the clustering quality relative to the best- and worst-performing distance measures.

Formally, given a dataset, let DBS_{rec} refer to the DBS if the distance metric recommended by the built model is selected and DBS_{best} and DBS_{worst} refer to the DBS values if the best- and worst-performing distance metrics are selected. Following Zhu et al. [12], the recommendation accuracy RA on this dataset is given by Equation 7.

$$RA = \frac{DBS_{\text{rec}} - DBS_{\text{worst}}}{DBS_{\text{best}} - DBS_{\text{worst}}} \tag{7}$$

As reported in Table VI, using the recommendation of the built model registered the highest mean RA at 0.8360, followed by fixing the distance measure to Euclidean distance at 0.5082. To determine whether the differences between the mean RA values of the five distance metric selection methods are statistically significant, the Scott-Knott effect size difference test [49], [50], which partitions the mean RA values into distinct groups with non-negligible difference via hierarchical clustering, was employed. This statistical test avoids the problem of subset overlapping present in procedures such as Tukey's and Duncan's, where a treatment may be simultaneously classified under two or more groups [51].

Figure 10 shows the results of the Scott-Knott effect size difference test. The mean recommendation accuracy of using the built meta-learning model is significantly different from the mean recommendation accuracies of utilizing fixed or random distance metric selection methods, which may be taken as indicative of the effectiveness of the built meta-learning model.

TABLE VI
MEAN RECOMMENDATION ACCURACIES OF DISTANCE METRIC SELECTION METHODS

Distance Metric Selection	Mean Recommendation Accuracy
Recommended	**0.8360**
Fixed – Chebyshev	0.4702
Fixed – Euclidean	0.5746
Fixed – Manhattan	0.4556
Random	0.5215

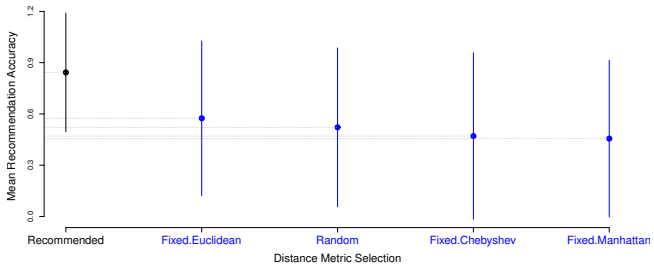


Fig. 10. Results of the Scott-Knott Effect Size Difference Test. The dot marks the mean, and the length of each line is inversely related to the stability of the distance metric selection method. Lines that share the same color indicate that the difference between the means is negligible; inversely, lines that have different colors indicate that the difference is statistically significant.

V. CONCLUSION

This study explored the use of a meta-learning approach for the automatic recommendation of a distance metric for k-means clustering that optimizes the Davies-Bouldin score (an internal clustering validation index). In particular, five categories of meta-features were considered: general, statistical, information-theoretic, structural, and complexity. The built model registered an accuracy of 70.59%.

The fine-grained analysis using SHAP showed that the mean of the sparsity registered the highest globally, as well as for the prediction of Chebyshev and Manhattan distances. For the prediction of Euclidean distance, the standard deviation of the eigenvalues has the highest feature importance. Limiting the meta-feature set to only the top 25 most important meta-features resulted in a slight increase to the accuracy $(+0.98\%)$. Although the prediction of the minority class (Manhattan distance) posed a difficulty to the meta-learning model despite the application of borderline SMOTE, its overall mean recommendation accuracy is significantly higher compared to defaulting to fixed or randomly chosen distance measures.

With the increased interest in automated machine learning (AutoML), future works may focus on extending the meta-learning approach in this research to other clustering algorithms (e.g., agglomerative clustering and *k*-medoids) and validation indices (e.g., c-scatter separability criterion, Dunn index, and normalized mutual information). Theoretical studies may also be done to examine and possibly corroborate the empirically observed patterns between the values of certain meta-features and the predicted optimal distance measures.

REFERENCES

[1] P. N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining*. Pearson, 2018.

[2] A. Jilling and M. Alvarez, "Optimizing recommendations for clustering algorithms using meta-learning," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–10.

[3] B. A. Pimentel and A. C. P. L. F. de Carvalho, "Unsupervised meta-learning for clustering algorithm recommendation," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

[4] B. Pimentel and A. Carvalho, "A meta-learning approach for recommending the number of clusters for clustering algorithms," *Knowledge-Based Systems*, vol. 195, p. 105682, 02 2020.

[5] E. Xing, M. Jordan, S. J. Russell, and A. Ng, "Distance metric learning with application to clustering with side-information," in *Advances in Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, Eds., vol. 15. MIT Press, 2002.

[6] J. L. Suárez, S. García, and F. Herrera, "A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges," *Neurocomputing*, vol. 425, pp. 300–322, 2021.

[7] R. Qaddoura, H. Faris, and I. Aljarah, "An efficient clustering algorithm based on the k-nearest neighbors with an indexing ratio," *Int. J. Mach. Learn. Cybern.*, vol. 11, pp. 675–714, 2020.

[8] D. Usman and S. Sani, "Performance evaluation of similarity measures for k-means clustering algorithm," *Bayero Journal of Pure and Applied Sciences*, vol. 12, no. 2, pp. 144–148, 2020.

[9] M. K. Gupta and P. Chandra, "Effects of similarity/distance metrics on k-means algorithm with respect to its applications in iot and multimedia: A review," *Multimedia Tools and Applications*, pp. 1–26, 2021.

[10] R. Shahid, S. Bertazzon, M. L. Knudtson, and W. A. Ghali, "Comparison of distance measures in spatial analytical modeling for health service planning," *BMC Health Services Research*, vol. 9, no. 1, pp. 1–14, 2009.

[11] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *International conference on database theory*. Springer, 2001, pp. 420–434.

[12] X. Zhu, Y. Li, J. Wang, T. Zheng, and J. Fu, "Automatic recommendation of a distance measure for clustering algorithms," *ACM Transactions on Knowledge Discovery from Data*, vol. 15, no. 1, December 2020.

[13] S. Muravyov, A. Filchenkov, P. Brazdil, J. Vanschoren, F. Hutter, and H. Hoos, "Meta-learning system for automated clustering." in *AutoML@ PKDD/ECML*, 2017, pp. 99–101.

[14] J. Vanschoren, "Meta-learning," in *Automated Machine Learning*. Springer, Cham, 2019, pp. 35–61.

[15] E. Alcobaça, F. Siqueira, A. Rivolli, L. P. F. Garcia, J. T. Oliva, A. C. de Carvalho *et al.*, "Mfe: Towards reproducible meta-feature extraction." *J. Mach. Learn. Res.*, vol. 21, no. 111, pp. 1–5, 2020.

[16] D. Bora and D. Gupta, "Effect of different distance measures on the performance of k-means algorithm: An experimental study in matlab," vol. 5, 05 2014.

[17] C. Liu, T. Hu, Y. Ge, and H. Xiong, "Which distance metric is right: An evolutionary k-means view," in *Proceedings of the 2012 SIAM international conference on data mining*. SIAM, 2012, pp. 907–918.

[18] R. Giancarlo, G. Lo Bosco, and L. Pinello, "Distance functions, clustering algorithms and microarray data analysis," in *International Conference on Learning and Intelligent Optimization*. Springer, 2010, pp. 125–138.

[19] B. Pimentel, "Datasets." [Online]. Available: https://www.openml.org/s/88/data Accessed: Apr. 2, 2022.

[20] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo, "OpenML: Networked science in machine learning," *SIGKDD Explor. Newsl.*, vol. 15, no. 2, p. 49–60, Jun 2014.

[21] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml Accessed: Apr. 2, 2022.

[22] "Kaggle." [Online]. Available: https://www.kaggle.com Accessed: Jul. 17, 2022.

[23] B. A. Pimentel and A. C. P. L. F. de Carvalho, "Statistical versus distance-based meta-features for clustering algorithm recommendation using meta-learning," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.

[24] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *2010 IEEE International Conference on Data Mining*, 2010, pp. 911–916.

[25] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.

[26] A. Kalousis and T. Theoharis, "Noemon: Design, implementation and performance results of an intelligent assistant for classifier selection," *Intelligent Data Analysis*, vol. 3, no. 5, pp. 319–337, 1999.

[27] P. Kuba, P. Brazdil, C. Soares, A. Woznica *et al.*, "Exploiting sampling and meta-learning for parameter setting for support vector machines," 2002.

[28] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, "Machine learning, neural and statistical classification," 1994.

[29] A. Kalousis, "Algorithm selection via meta-learning," Ph.D. dissertation, University of Geneva, 2002.

[30] C. Castiello, G. Castellano, and A. M. Fanelli, "Meta-data: Characterization of input features for meta-learning," in *International Conference on Modeling Decisions for Artificial Intelligence*. Springer, 2005, pp. 457–468.

[31] S. Ali and K. A. Smith, "On learning algorithm selection for classification," *Applied Soft Computing*, vol. 6, no. 2, pp. 119–138, 2006.

[32] S. Ali and K. A. Smith-Miles, "A meta-learning approach to automatic kernel selection for support vector machines," *Neurocomputing*, vol. 70, no. 1-3, pp. 173–186, 2006.

[33] R. Engels and C. Theusinger, "Using a data metric for preprocessing advice for data mining applications." in *ECAI*, vol. 98. Citeseer, 1998, pp. 23–28.

[34] M. A. Salama, A. E. Hassanien, and K. Revett, "Employment of neural network and rough set in meta-learning," *Memetic Computing*, vol. 5, no. 3, pp. 165–177, 2013.

[35] C. Köpf and I. Iglezakis, "Combination of task description strategies and case base properties for meta-learning," in *Proceedings of the 2nd international workshop on integration and collaboration aspects of data mining, decision support and meta-learning*, 2002, pp. 65–76.

[36] K. Alexandros and H. Melanie, "Model selection via meta-learning: a comparative study," *International Journal on Artificial Intelligence Tools*, vol. 10, no. 04, pp. 525–554, 2001.

[37] Q. Song, G. Wang, and C. Wang, "Automatic recommendation of classification algorithms based on data set characteristics," *Pattern recognition*, vol. 45, no. 7, pp. 2672–2689, 2012.

[38] A. Lorena, L. P. Garcia, J. Lehmann, M. de Souto, and T. Ho, "How complex is your classification problem?: A survey on measuring classification complexity," *ACM Computing Surveys*, vol. 52, pp. 1–34, 09 2019.

[39] E. Alcobaça and F. Siqueira, "Meta-feature description table." [Online]. Available: https://pymfe.readthedocs.io/en/latest/auto_pages/meta_features_description.html Accessed: Jul. 17, 2022.

[40] E. Alcobaça, F. Siqueira, A. Rivolli, L. P. F. Garcia, J. T. Oliva, and A. C. P. L. F. de Carvalho, "Mfe: Towards reproducible meta-feature extraction," *Journal of Machine Learning Research*, vol. 21, no. 111, pp. 1–5, 2020. [Online]. Available: http://jmlr.org/papers/v21/19-348.html

[41] K. Fawagreh, M. M. Gaber, and E. Elyan, "Random forests: from early developments to recent advancements," *Systems Science & Control Engineering*, vol. 2, no. 1, pp. 602–609, 2014.

[42] M. Ibrahim and M. J. Carman, "Improving scalability and performance of random forest based learning-to-rank algorithms by aggressive subsampling." in *Proceedings of the 12th Australasian Data Mining Conference*, 2014, pp. 91–99.

[43] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[44] H. Han, W. Y. Wang, and B. H. Mao, "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning," in *International conference on intelligent computing*. Springer, 2005, pp. 878–887.

[45] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE, 2008, pp. 1322–1328.

[46] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[47] L. S. Shapley, *17. A Value for n-Person Games*. Princeton University Press, 2016, pp. 307–318.

[48] C. Molnar, *Interpretable Machine Learning*, 2nd ed., 2022.

[49] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto, "An empirical comparison of model validation techniques for defect prediction models," *IEEE Transactions on Software Engineering*, vol. 43, no. 1, pp. 1–18, 2017.
[50] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto, "The impact of automated parameter optimization on defect prediction models," *IEEE Transactions on Software Engineering*, vol. 45, no. 7, pp. 683–711, 2019.
[51] A. Scott and M. Knott, "A cluster analysis method for grouping means in the analysis of variance," *Biometrics*, vol. 30, p. 507, 1974.