A Study on Similarity Space Metrics as Predictors for k-Nearest Neighbor Classificator Performance

Eduardo Henrique Basilio de Carvalho
Departamento de Engenharia Eletrônica
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
eduardohbc@ufmg.br

***Abstract*—The k-Nearest Neighbor (k-NN) algorithm, while conceptually simple, is highly sensitive to hyperparameter choices, such as the number of neighbors 'k' and the distance metric. The standard method for tuning these hyperparameters, k-fold cross-validation, is computationally expensive. This paper posits that the performance of k-NN can be predicted by computationally efficient proxy metrics derived from the geometric and statistical properties of the data. We introduce a transformation from the original feature space to a class-aggregated similarity space using a sparse Radial Basis Function (RBF) kernel. Within this space, we propose and define a suite of novel metrics to quantify class separability, compactness, and overlap. The central thesis is that optimizing these proxy metrics via Bayesian optimization can provide a faster and more insightful alternative to the brute-force cross-validation approach for hyperparameter tuning, without sacrificing classification accuracy.**

***Index Terms*—k-Nearest Neighbor, similarity space, hyperparameter optimization, Bayesian optimization, classification, class separability, proxy metrics, RBF kernel**

I. Introduction

A. The k-Nearest Neighbor Algorithm: A Foundation of Non-Parametric Classification

The k-Nearest Neighbor (k-NN) algorithm stands as one of the most fundamental and intuitive methods in machine learning for classification and regression tasks [1], [2]. As a non-parametric, instance-based learning method, it makes no underlying assumptions about the distribution of the data, offering significant flexibility [1], [2]. The core principle of k-NN is that similar data points are likely to have similar outcomes. For classification, an unlabeled observation is assigned to the class most frequently represented among its 'k' nearest neighbors in the feature space [3], [4]. The algorithm's conceptual development traces back to the work of Evelyn Fix and Joseph Hodges in 1951, but it was the seminal 1967 paper, "Nearest Neighbor Pattern Classification" by Thomas Cover and Peter Hart, that provided its rigorous theoretical underpinning [5], [6]. Their work established a crucial property: for a large number of samples, the error rate of the simple 1-NN rule is bounded by at most twice the Bayes error rate—the theoretical minimum achievable error for a given data distribution [5], [7]. This result guarantees a level of performance relative to an optimal classifier, solidifying k-NN's position as a viable and theoretically sound method that remains relevant decades after its conception [8]–[10].

B. The Achilles' Heel: Sensitivity to Hyperparameters and Data Structure

Despite its conceptual simplicity, the practical application of k-NN reveals a significant sensitivity to several key factors that dictate its performance [3], [11]. The choice of 'k', the number of neighbors, represents a critical bias-variance trade-off. A small 'k' can lead to a model that is highly sensitive to noise and outliers, resulting in high variance and overfitting, while a large 'k' can oversmooth the decision boundary, leading to high bias and underfitting by ignoring local data patterns [3], [12]. Furthermore, the algorithm's effectiveness is contingent on the choice of distance metric (e.g., Euclidean, Manhattan) used to quantify "nearness" in the feature space [1], [4]. The performance can also be severely degraded by irrelevant features or differences in the scales of features, necessitating careful data preprocessing. This sensitivity is particularly acute in high-dimensional spaces, a phenomenon known as the "curse of dimensionality," where the concept of distance can become less meaningful as all points tend to become equidistant from one another [4], [11].

C. The Conventional Solution and Its Limitations: Cross-Validation

To address the challenge of hyperparameter selection, the standard and most robust methodology is k-fold cross-validation [12]–[14]. This technique involves partitioning the training data into 'k' subsets, or "folds." The model is then trained on 'k-1' folds and validated on the remaining held-out fold. This process is repeated 'k' times, with each fold serving as the validation set exactly once, and the performance metrics are averaged to produce a stable and reliable estimate of the model's generalization ability [13], [15]. While effective, k-fold cross-validation suffers from a major drawback: it is computationally intensive, often prohibitively so [11], [13]. For each combination of hyperparameters under consideration (e.g., for each pair of 'k' and a distance metric parameter), the entire process of 'k' model trainings and evaluations must be performed. This brute-force, black-box approach becomes impractical for large datasets or when exploring a wide range of hyperparameter values [16], [17].

D. Thesis Statement: Predicting Performance Through Geometric Proxies

The central thesis of this paper is that the computational burden of hyperparameter optimization for k-NN can be substantially mitigated by using computationally efficient proxies that correlate strongly with classification performance. We posit that metrics derived from the geometric and statistical properties of data classes, when transformed into a carefully constructed "similarity space," can serve as effective predictors of k-NN accuracy. This work introduces and evaluates a suite of novel metrics designed to quantify class separability, compactness, and overlap. By optimizing the k-NN hyperparameters to maximize these proxy metrics instead of the cross-validation score, we aim to develop a faster, more insightful, and equally effective method for hyperparameter tuning. This approach moves away from the black-box treatment of the model and instead leverages the intrinsic geometric nature of the k-NN algorithm itself, addressing the paradox where a conceptually simple algorithm requires a computationally complex and "unintelligent" optimization procedure.

II. From Feature Space to Similarity Space: A Kernel-Based Transformation

To analyze the geometric relationships between classes, we first transform the data from its original feature space into a "similarity space." This transformation is designed to create a new representation where the spatial arrangement of data points directly reflects their relationship to the different classes, making geometric separability metrics more meaningful.

A. The Role of the Radial Basis Function (RBF) Kernel

The core of this transformation is the Gaussian Radial Basis Function (RBF) kernel, a widely used similarity function in kernelized learning algorithms [18], [19]. The RBF kernel measures the similarity between two vectors, $\mathbf{u}$ and $\mathbf{v}$, and is defined by the equation:

$$K(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{\|\mathbf{u} - \mathbf{v}\|^2}{2h^2}\right) \tag{1}$$

where $\|\mathbf{u} - \mathbf{v}\|$ represents the Euclidean distance, and the bandwidth parameter h controls the width of the kernel, determining how quickly similarity decays with distance [18], [20]. The function's value ranges from 1 (for identical vectors) to 0 (for infinitely distant vectors), providing an intuitive measure of similarity. The use of RBFs was first formulated in the context of neural networks in a seminal 1988 paper by Broomhead and Lowe [21], [22]. While their work focused on RBFs as activation functions in a network, this study employs the RBF as a standalone kernel function, a concept popularized by methods like Support Vector Machines (SVMs) to handle non-linear data relationships by implicitly mapping data to a higher-dimensional space [18], [19]. This mapping allows complex, non-linear class structures in the original feature space to be represented in a way that is more amenable to linear or geometric analysis.

B. Sparse Kernel Matrix Construction

A key feature of our implementation is the use of a *sparse* RBF kernel matrix. Given a set of samples to be evaluated and a set of reference samples, a full kernel matrix would compute the similarity between every pair of points. For large datasets, this is computationally infeasible. Instead, for each sample in the evaluation set, we compute its similarity only to the 'k' closest points in the reference set [20]. All other similarity values are set to zero. This approach not only makes the computation tractable but also aligns the transformation process with the local nature of the k-NN algorithm itself, focusing only on the most relevant neighbors.

C. Class-Aggregated Similarity Space Transformation

The final step in creating the similarity space is to aggregate the similarity scores based on class labels. Let K be the sparse $n \times m$ similarity matrix, where n is the number of evaluated samples and m is the number of reference samples. Let $\mathbf{y}$ be the vector of class labels for the m reference samples, and let $\mathcal{C} = \{c_1, c_2, \ldots, c_p\}$ be the set of p unique classes. We compute a new $n \times p$ matrix, Q, which represents the data in the similarity space. An element Q_{ik} of this matrix is defined as the sum of similarities from the i-th evaluated sample to all reference samples belonging to class c_k:

$$Q_{ik} = \sum_{j=1}^{m} K_{ij} \cdot \mathbb{I}(y_j = c_k) \tag{2}$$

where $\mathbb{I}(\cdot)$ is the indicator function [20]. This operation can be expressed as the matrix product $Q = KY$, where Y is an $m \times p$ binary matrix indicating class membership.

The resulting matrix Q provides the final representation. Each row of Q is a p-dimensional vector where each component quantifies the sample's total similarity to a specific class. This transformation serves as a crucial conceptual bridge. It takes data that may be non-linearly separable in its raw form and projects it into a new, low-dimensional space where the axes themselves represent class affinity. In this new space, the geometric arrangement of points—their clustering, separation, and overlap—is directly related to their classification potential, thereby justifying the application of the geometric metrics described in the following section.

III. A Suite of Novel Metrics for Quantifying Class Separability

The core contribution of this work is a suite of six novel metrics designed to quantify different aspects of class separability and compactness within the similarity space. These metrics are intended to serve as computationally efficient proxies for k-NN classification accuracy. Each metric is derived from established principles in pattern recognition, computational geometry, and statistics, but is formulated to capture specific geometric properties of the class distributions in the transformed space.

A. Dissimilarity: A Hybrid Distance-Direction Metric

This metric computes a scalar value representing the separability between classes by considering both the distance and the directional alignment of their centroids in the similarity space. For each class, a centroid vector $\mathbf{v}_k$ is calculated as the arithmetic mean of all sample vectors in that class. Then, for each unique pair of class centroids $(\mathbf{v}_i, \mathbf{v}_j)$, a pairwise dissimilarity score is computed as:

$$D_{ij} = \|\mathbf{v}_i - \mathbf{v}_j\| \cdot \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\|\|\mathbf{v}_j\|} \tag{3}$$

This formula combines the Euclidean distance between the centroids with their cosine similarity [20]. The intuition is that well-separated classes should have centroids that are not only far apart (large Euclidean distance) but also point in different directions within the similarity space (high cosine similarity, as vectors are non-negative). The final score is derived from the mean ($\mu_{\mathcal{D}}$) and standard deviation ($\sigma_{\mathcal{D}}$) of all pairwise dissimilarities, adjusted by factors related to the hyperparameters being tuned. This approach of combining magnitude and direction is motivated by research into "direction-aware" distance metrics, which have been shown to be more robust than Euclidean distance alone in high-dimensional spaces where magnitude can be misleading [23]–[25].

B. n-Volume of the Intersection of the Convex Hulls

This metric quantifies class overlap by measuring the volume of the geometric intersection of the class convex hulls in the N-dimensional similarity space. A convex hull is defined as the smallest convex set that encloses all points of a given class [26]. The intersection of the convex hulls for all classes, $\mathcal{I} = \bigcap_{k=1}^{p} \text{conv}(\mathcal{C}_k)$, represents the region of the similarity space where points cannot be unambiguously assigned to a single class based on their convex hull membership. The final score is the N-dimensional volume of this intersection region [20].

This metric operationalizes a fundamental concept from geometry and learning theory: two sets of points are linearly separable if and only if their convex hulls do not intersect [27], [28]. While this theorem provides a binary condition (intersect or not), measuring the *volume* of the intersection transforms it into a continuous, quantitative heuristic for class separability. A volume of zero corresponds to perfect linear separability, while a larger volume indicates a greater degree of class overlap and confusion, which is expected to correlate negatively with k-NN performance. Algorithms for computing the intersection of convex polyhedra provide the computational basis for this metric [29], [30].

C. n-Volume of the Convex Hulls

In contrast to measuring overlap, this metric focuses on the compactness and uniformity of the individual class distributions. For each class, the N-dimensional volume of its convex hull is computed. The final score is then calculated from the mean ($\mu_{\mathcal{V}}$) and standard deviation ($\sigma_{\mathcal{V}}$) of these volumes:

$$\text{Final Score} = (\mu_{\mathcal{V}} - \sigma_{\mathcal{V}}) \cdot (1 - f_k)$$

The intuition here is that an ideal class structure for k-NN would consist of tight, compact clusters. A small mean volume ($\mu_{\mathcal{V}}$) indicates that classes, on average, occupy a small region in the similarity space. A small standard deviation ($\sigma_{\mathcal{V}}$) indicates that all classes are similarly compact. The metric thus favors class distributions that are consistently and tightly clustered. This approach draws on work in classification and anomaly detection where the volume of a convex hull is used to characterize the spatial extent of a data distribution [31]–[33].

D. Spread: A Statistical View on Inter- and Intra-Class Distances

This metric provides a statistical measure of class separability by comparing the distributions of distances between points within the same class (intra-class) to distances between points in different classes (inter-class). After computing all pairwise Euclidean distances in the similarity space, they are partitioned into these two sets. The mean and standard deviation are calculated for both the within-class distances ($\mu_{\text{within}}, \sigma_{\text{within}}$) and the between-class distances ($\mu_{\text{between}}, \sigma_{\text{between}}$). The final score is a function of these four statistics:

$$\text{Final Score} = ((\mu_{\text{between}} \cdot \mu_{\text{within}}) - (\sigma_{\text{between}} \cdot \sigma_{\text{within}})) \cdot (1 - f_h) \cdot (1 - f_k)$$

This metric is rooted in the foundational principles of pattern recognition, most notably Fisher's Linear Discriminant Analysis (LDA), which seeks a projection that maximizes the ratio of between-class scatter to within-class scatter [34]. Good separability is achieved when between-class distances are large and within-class distances are small [35]–[37]. The "Spread" metric extends this classic idea by also rewarding consistency (low standard deviation) in these distance distributions, penalizing scenarios where separability is erratic across the feature space.

E. Silhouette: Adapting a Clustering Metric for Supervised Evaluation

This metric adapts the well-known Silhouette score, traditionally used for unsupervised cluster validation, to the supervised context of evaluating class structure. The standard silhouette score for a single point i is given by:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{4}$$

where $a(i)$ is the mean distance from point i to other points in its own cluster (a measure of cohesion), and $b(i)$ is the mean distance from point i to points in the nearest neighboring cluster (a measure of separation) [38]. The score ranges from -1 to +1, with higher values indicating a better fit. This metric was introduced by Peter J. Rousseeuw in 1987 as a graphical aid for interpreting and validating cluster analysis [38]–[40].

In this paper, we repurpose this unsupervised tool for a supervised task. We treat each ground-truth class as a "cluster" and compute the silhouette score for every point in the similarity space. This provides a measure of how well-defined the natural class groupings are. A high average score suggests that the classes are inherently compact and well-separated, a condition favorable for k-NN classification. The final score is a custom formula that combines the mean (μ_s) and standard deviation (σ_s) of the individual silhouette scores, rewarding a high mean while penalizing high variance.

F. Cosine Similarity between the Centroids' Hyperplane and the Opposite Unit Hyperplane

This metric quantifies the geometric orientation of the class separation. It measures the parallelism between two specific hyperplanes in the N-dimensional similarity space. The first, the "Centroid Hyperplane," is the hyperplane defined by the set of p class centroids. The second is a fixed "Reference Hyperplane" defined by the equation $\sum_{i=1}^{N} x_i = 0$. The degree of parallelism is measured by the absolute cosine similarity of their normal vectors, $\mathbf{n}_{\text{centroid}}$ and $\mathbf{n}_{\text{opposite}}$. The final score is inversely related to this similarity:

$$\text{Final Score} = (1 - |\mathbf{n}_{\text{centroid}} \cdot \mathbf{n}_{\text{opposite}}|) \cdot (1 - f_k)$$

The intuition is that a high degree of parallelism between the hyperplane formed by the class centroids and the reference hyperplane indicates a degenerate or overfitted state where class distinctions are collapsing along a single axis of similarity. The metric penalizes this condition. A low cosine similarity (and thus a high final score) indicates that the centroid hyperplane is not parallel to the reference plane, suggesting a more robust and non-degenerate separation of the classes.

IV. Methodology and Experimental Protocol

To rigorously evaluate the proposed similarity space metrics as predictors of k-NN performance, a comprehensive experimental protocol was designed. This protocol ensures a fair comparison between models optimized using the novel metrics and a baseline model optimized using a conventional, state-of-the-art approach.

A. Baseline Model

The benchmark for this study is a standard k-NN classifier. The hyperparameters of this model—specifically, the number of neighbors ('k') and the RBF kernel bandwidth ('h') used in the similarity transformation—are tuned by directly optimizing for the 5-fold cross-validation accuracy on the training data [20]. This represents the "gold standard" but computationally expensive method that is commonly used in practice to achieve the best possible performance from a k-NN model.

B. Bayesian Optimization

For all models, including the baseline, hyperparameter tuning is conducted using Bayesian optimization. This choice is crucial for ensuring a fair comparison of the optimization processes. Bayesian optimization is a sequential, model-based approach for finding the extremum of expensive-to-evaluate, black-box functions [44]–[46]. The method works by building a probabilistic surrogate model (typically a Gaussian Process) of the objective function (e.g., cross-validation accuracy or one of the proposed metrics). It then uses an acquisition function, such as Expected Improvement (EI), to intelligently select the next set of hyperparameters to evaluate, balancing exploration of the search space with exploitation of promising regions [47], [48].

This approach is rooted in the foundational work of Jonas Mockus, who introduced the Bayesian framework for optimization and the EI principle in the 1970s [47], [49]. Its modern application was significantly advanced by Jones et al. in their 1998 paper on Efficient Global Optimization (EGO), which established the use of Gaussian Processes as the surrogate model [50]–[52]. While the baseline model uses Bayesian optimization to maximize 5-fold cross-validation accuracy, the six experimental models use it to maximize one of the six proposed similarity space metrics over the training set. To ensure a fair comparison of computational efficiency, the optimization process for every model is limited to a maximum of 10 iterations [20].

C. Experimental Protocol

The overall performance of each of the seven models (one baseline, six metric-based) is assessed using a 10-fold cross-validation scheme for the entire experiment. For each of the 10 folds, the dataset is partitioned into a training set (90%) and a test set (10%). Each of the seven models is then trained on the training portion; this training phase includes the internal 10-iteration Bayesian optimization loop to find the best hyperparameters according to that model's specific objective function. Once trained and optimized, the model's performance is evaluated on the held-out test set. The final results are reported as the mean and standard deviation of accuracy, total training time, and inference time, aggregated across the 10 outer folds. This nested validation structure ensures a robust and unbiased evaluation of how well each optimization strategy generalizes to unseen data.

D. Datasets

To ensure the generalizability of our findings, the experiments are conducted on a diverse collection of 15 publicly available datasets. These datasets vary widely in the number of samples, features, and classes, representing a range of challenges for classification algorithms. The datasets, sourced primarily from the UCI Machine Learning Repository, OpenML, and Kaggle, are summarized in Table I. Categorical features in datasets like Car Evaluation and Mushroom were converted to a numerical representation using one-hot encoding, resulting in an expanded feature space as noted in the table.

V. Results

The empirical evaluation of the proposed similarity space metrics against the baseline cross-validation approach is forthcoming. The experiments outlined in the methodology section have been designed but not yet executed. This section,

TABLE I
DATASETS USED IN THE STUDY

Dataset Name	Samples	Features	Classes	Source / Citation
Banknote Authentication	1372	4	2	Dua, D. & Graff, C. (2017) [53], [54]
Breast Cancer (Wisconsin)	569	30	2	Dua, D. & Graff, C. (2017) [54], [55]
Car Evaluation	1728	21	4	Bohanec, M. & Rajkovic, V. (1990) via UCI [54], [56]
Credit Germany (Statlog)	1000	74	2	Dua, D. & Graff, C. (2017) [54], [57]
Diabetes (Pima)	768	8	2	Dua, D. & Graff, C. (2017) [54], [58]
Heart Disease (Statlog)	270	13	2	Dua, D. & Graff, C. (2017) [54], [59]
Ionosphere	351	34	2	Sigillito, V. (1989) via UCI [54], [60]
Iris	150	4	3	Fisher, R.A. (1936) via UCI [54], [61]
Monk's Problems (monk2)	432	6	2	Thrun, S.B., et al. (1991) via UCI [54], [62]
Mushroom	8124	105	2	Schlimmer, J.S. (1987) via UCI [54], [63]
Phoneme	5404	5	2	OpenML (ID: 1489) [64]
Spambase	4601	57	2	Hopkins, M., et al. (1999) via UCI [65], [66]
Titanic	1309	13	2	Kaggle / Vanderbilt University [67]
Wine	178	13	3	Aeberhard, S. & Forina, M. (1991) via UCI [68], [69]
Yeast	1484	8	10	Nakai, K. (1996) via UCI [70], [71]

therefore, serves as a blueprint for the analysis that will be conducted once the experimental data are available. The primary goal of the analysis will be to determine whether the proposed metrics can serve as reliable and efficient proxies for k-NN performance.

The analysis will be structured around three key research questions:

1) **Performance Equivalence:** Can optimizing for the proposed metrics yield classification accuracy comparable to that achieved by optimizing directly for cross-validation accuracy?
2) **Computational Efficiency:** Do the proposed metrics offer a significant reduction in the computational time required for hyperparameter optimization compared to the baseline?
3) **Predictive Validity:** Is there a strong statistical correlation between the values of the proposed metrics and the actual classification performance of the k-NN model?

To address these questions, the forthcoming results will be presented and analyzed as follows. First, a comprehensive comparison of the mean and standard deviation of test accuracy for all seven models across the 15 datasets will be conducted to assess performance and stability. Second, a detailed analysis of the training times will quantify the computational speed-up offered by the metric-based approaches. Finally, a correlation analysis will be performed to directly measure the strength of the relationship between each proposed metric and the resulting classification accuracy, providing evidence for their validity as performance predictors. A synthesis of these findings will determine which, if any, of the metrics provide the best trade-off between computational cost and predictive power.

REFERENCES

[1] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[2] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.

[3] J. Mockus, "On Bayesian methods for seeking the extremum," in *Optimization Techniques IFIP Technical Conference, Novosibirsk, July 1–7, 1974*, G. I. Marchuk, Ed. Berlin, Heidelberg: Springer, 1975, pp. 400–404.

[4] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[5] E. Fix and J. L. Hodges, "Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties," USAF School of Aviation Medicine, Randolph Field, Texas, Project 21-49-004, Report 4, Feb. 1951.

[6] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[7] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[8] E. Y. Boateng, J. Otoo, and D. A. Abaye, "Basic tenets of classification algorithms K-nearest-neighbor, support vector machine, random forest and neural network: a review," *Journal of Data Analysis and Information Processing*, vol. 8, no. 4, pp. 341–357, 2020.

[9] A. Singh, N. Thakur, and A. Sharma, "A Review on Analysis of K-Nearest Neighbor Classification Machine Learning Algorithms based on Supervised Learning," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 10, no. 5, pp. 191–196, 2021.

[10] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[11] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is 'nearest neighbor' meaningful?," in *International Conference on Database Theory*, 1999, pp. 217–235.

[12] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*. New York: Springer, 2013.

[13] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, pp. 40–79, 2010.

[14] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 36, no. 2, pp. 111–133, 1974.

[15] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, 1995, vol. 2, pp. 1137–1145.

[16] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, pp. 40–79, 2010.

[17] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," in *Encyclopedia of Database Systems*, L. Liu and M. T. Özsu, Eds. Boston, MA: Springer US, 2009, pp. 532–538.

[18] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.

[19] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[20] This study's methodology.

[21] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.

[22] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.

[23] X. Gu, P. P. Angelov, D. Kangin, and J. C. Principe, "A new type of distance metric and its use for clustering," *Evolving Systems*, vol. 8, no. 3, pp. 167–177, Sep. 2017.

[24] X. Gu, P. P. Angelov, D. Kangin, and J. C. Principe, "A new type of distance metric and its use for clustering," *Evolving Systems*, vol. 8, no. 3, pp. 167–177, Sep. 2017.

[25] 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*, 2001, pp. 420–434.

[26] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd ed. Berlin, Heidelberg: Springer-Verlag, 2008.

[27] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[28] G. M. Ziegler, *Lectures on Polytopes*. New York: Springer-Verlag, 1995.

[29] B. Chazelle, "An optimal algorithm for intersecting three-dimensional convex polyhedra," *SIAM Journal on Computing*, vol. 22, no. 6, pp. 1271–1288, 1993.

[30] J. O'Rourke, *Computational Geometry in C*, 2nd ed. Cambridge, UK: Cambridge University Press, 1998.

[31] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Machine Learning*, vol. 54, no. 1, pp. 45–66, 2004.

[32] G. I. Nalbantov, J. C. Bioch, and F. C. A. Groen, "Nearest convex hull classification," *Pattern Recognition*, vol. 40, no. 4, pp. 1342–1353, Apr. 2007.

[33] F. Ecer, "A novel anomaly detection method based on convex hull," *Engineering Science and Technology, an International Journal*, vol. 24, no. 5, pp. 1148–1156, 2021.

[34] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[35] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. San Diego, CA: Academic Press, 1990.

[36] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley-Interscience, 2000.

[37] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proceedings of the 33rd International Conference on Machine Learning*, 2016, pp. 507–516.

[38] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, Nov. 1987.

[39] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ: John Wiley & Sons, 2005.

[40] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, Nov. 1987.

[41] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[42] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2002.

[43] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge, UK: Cambridge University Press, 2000.

[44] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2016.

[45] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 2951–2959.

[46] P. I. Frazier, "A tutorial on Bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.

[47] J. Mockus, "On the application of Bayesian methods for seeking the extremum," in *Towards Global Optimization 2*, L. C. W. Dixon and G. P. Szegö, Eds. Amsterdam: North-Holland, 1978, pp. 117–129.

[48] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[49] J. Mockus, "On Bayesian methods for seeking the extremum," in *Optimization Techniques IFIP Technical Conference, Novosibirsk, July 1–7, 1974*, G. I. Marchuk, Ed. Berlin, Heidelberg: Springer, 1975, pp. 400–404.

[50] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[51] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[52] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[53] V. Lohweg, "Banknote Authentication Data Set," UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/banknote+authentication

[54] D. Dua and C. Graff, "UCI Machine Learning Repository," Irvine, CA: University of California, School of Information and Computer Sciences, 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[55] W. H. Wolberg, W. N. Street, and O. L. Mangasarian, "Breast Cancer Wisconsin (Diagnostic) Data Set," UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

[56] M. Bohanec and V. Rajkovic, "Expert system for decision making," *Sistemica*, vol. 1, no. 1, pp. 145–157, 1990.

[57] H. Hofmann, "Statlog (German Credit Data) Data Set," UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

[58] "Pima Indians Diabetes Database," National Institute of Diabetes and Digestive and Kidney Diseases. Sourced from UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes

[59] "Statlog (Heart) Data Set," UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/statlog+(heart)

[60] V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker, "Classification of radar returns from the ionosphere using neural networks," *Johns Hopkins APL Technical Digest*, vol. 10, no. 3, pp. 262–266, 1989.

[61] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[62] S. B. Thrun et al., "The MONK's Problems: A performance comparison of different learning algorithms," Technical Report CS-CMU-91-197, Carnegie Mellon University, Dec. 1991.

[63] J. S. Schlimmer, "Concept acquisition through representational adjustment," Ph.D. dissertation, Dept. of Information and Computer Science, University of California, Irvine, 1987.

[64] "Phoneme Dataset," OpenML, ID 1489. [Online]. Available: https://www.openml.org/d/1489

[65] M. Hopkins, E. Reeber, G. Forman, and J. Suermondt, "Spambase Dataset," UCI Machine Learning Repository, 1999. [Online]. doi: 10.24432/C53G6X.

[66] L. F. Cranor and B. A. LaMacchia, "Spam!," *Communications of the ACM*, vol. 41, no. 8, pp. 74–83, Aug. 1998.

[67] "Titanic: Machine Learning from Disaster," Kaggle. [Online]. Available: https://www.kaggle.com/c/titanic/data. (Original data from Dept. of Biostatistics, Vanderbilt University).

[68] S. Aeberhard and M. Forina, "Wine Dataset," UCI Machine Learning Repository, 1991. [Online]. doi: 10.24432/C5PC7J.

[69] S. Aeberhard, D. Coomans, and O. de Vel, "Comparative analysis of statistical pattern recognition methods in high dimensional settings," *Pattern Recognition*, vol. 27, no. 8, pp. 1065–1077, 1994.

[70] K. Nakai, "Yeast Dataset," UCI Machine Learning Repository, 1996. [Online]. doi: 10.24432/C5KG68.

[71] K. Nakai and P. Horton, "A new method for predicting sorting signals in proteins from their amino acid sequences," in *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, 1996, pp. 146–155.