

A Study on Similarity Space Metrics as Predictors for k-Nearest Neighbor Classifier 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 alternative to the brute-force cross-validation approach for hyperparameter tuning. However, our experimental results on 15 datasets indicate that while the proposed metrics are computationally faster, they do not consistently match the classification accuracy of the baseline k-NN tuned with traditional cross-validation. This suggests a trade-off between computational efficiency and predictive performance, where the direct optimization of accuracy remains superior for achieving the best results.

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:

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

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 $\mathcal{C} = \{c_1, c_2, \dots, c_p\}$ 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 :

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

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, adapted from the work of Menezes et al. [72], 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:

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

This formula, originally proposed as a dissimilarity function for optimizing RBF kernel widths in SVMs [72], combines the Euclidean distance between the centroids with their cosine similarity. The intuition, as described by Menezes et al., is that multiplying the Euclidean distance by the cosine similarity addresses a limitation of using Euclidean distance alone, where the metric would not converge to zero for very small kernel widths, potentially creating spurious local maxima. The final score in our implementation 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 also aligns with broader 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 is based on the geometric intersection of the class convex hulls [27], [28]. Counterintuitively, instead of minimizing this overlap to achieve maximum separability, the objective is to *maximize* the volume of the intersection. The rationale is that a controlled degree of overlap can serve as a regularizer, preventing the model from creating overly rigid decision boundaries that are brittle and overfit to the training data. By rewarding a larger intersection volume, the optimization is guided towards solutions that exhibit better generalization, under the assumption that some class ambiguity in the similarity space is characteristic of a more robust model. The computational basis for this metric lies in algorithms for intersecting convex polyhedra [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 (μ_V) and standard deviation (σ_V) of these volumes:

$$\text{Final Score} = (\mu_V - \sigma_V) \cdot (1 - f_k)$$

The intuition behind this metric is that maximizing the volume of the class convex hulls can lead to better generalization. While compact clusters are often desired, overly small volumes may indicate overfitting, where the class representations are too specific to the training data and thus harder to distinguish. This metric, therefore, rewards solutions where classes occupy a larger, more robust region in the similarity space. The final score is designed to maximize the mean volume (μ_V) while minimizing the standard deviation of the volumes (σ_V), promoting class representations that are both large and uniformly sized. This concept is related to methods in anomaly detection and classification that use the spatial extent of data distributions to define class boundaries [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:

$$\text{Final Score} = ((\mu_{\text{between}} \cdot \mu_{\text{within}}) - (\sigma_{\text{between}} \cdot \sigma_{\text{within}})) \cdot (1 - f_h) \cdot (1 - f_k) \quad (4)$$

This metric diverges from traditional separability measures like Fisher's Linear Discriminant Analysis (LDA), which aim to maximize the distance between classes while minimizing the spread within them [34]–[36]. Instead, the "Spread" metric seeks a representation where classes are not only well-separated (high μ_{between}) but also internally spread out (high μ_{within}). The objective is to maximize the product of the mean inter-class and intra-class distances, rewarding data representations where points are well distinguished. The metric also favors consistency by penalizing high variance in both distance distributions ($\sigma_{\text{between}}, \sigma_{\text{within}}$), promoting a uniform spread across the 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:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (5)$$

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 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.

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]

V. RESULTS

The experimental results indicate a trade-off between computational efficiency and classification accuracy. While the proposed similarity space metrics generally lead to faster training times, they do not consistently achieve the same level of accuracy as the baseline k-NN classifier tuned with 5-fold cross-validation. A summary of the mean accuracy for each metric across the 15 datasets is presented in Table II, where the custom models are indicated as I through VI, corresponding to the six proposed metrics, as follows:

- I: n-Volume of the Intersection of the Convex Hulls
- II: n-Volume of the Convex Hulls
- III: Dissimilarity
- IV: Cosine Similarity between the Centroids' Hyperplane and the Opposite Unit Hyperplane
- V: Silhouette
- VI: Spread

The baseline model, optimized directly for accuracy, generally achieves the highest performance. However, some of the proposed metrics show competitive results on specific datasets. For instance, the **Silhouette** metric performs on par with the baseline on the **Car Evaluation** dataset, and **Opposite hyperplane** is competitive on the **Heart Disease (Statlog)** dataset.

A. Computational Efficiency: Training and Inference Times

In terms of computational cost, the proposed metrics consistently show lower training and prediction times compared to the baseline, which relies on the more expensive 5-fold cross-validation. This confirms the central hypothesis that using proxy metrics can significantly reduce the computational burden of hyperparameter optimization. Tables III and IV provide a detailed breakdown of the mean training and inference times (in seconds) for each model across all datasets.

The data clearly shows that the baseline model is the most computationally expensive to train, often by a significant margin, especially for larger datasets like Mushroom, Phoneme,

and Spambase. The proxy metrics offer a much faster alternative for the optimization phase. Inference times are generally low across all models, but the baseline still tends to be slightly slower, though the differences are less pronounced than in training. This suggests that for applications where computational resources are a major constraint, these metrics could be a viable alternative, provided that the potential drop in accuracy is acceptable.

VI. CONCLUSION

This study investigated the viability of using a suite of novel similarity space metrics as computationally efficient proxies for k-NN classification performance. The experimental results on 15 diverse datasets demonstrate that while these metrics can significantly reduce the computational burden of hyperparameter optimization compared to traditional k-fold cross-validation, this efficiency comes at the cost of classification accuracy. The baseline k-NN classifier, tuned by directly optimizing for accuracy, consistently outperformed the models tuned with the proposed proxy metrics. This indicates that while the geometric and statistical properties captured by these metrics are relevant to class separability, they do not serve as a perfect substitute for direct performance evaluation.

The analysis of computational cost, however, confirms the primary advantage of the proposed metrics. The significant reduction in training time makes them a compelling option for scenarios with limited computational budgets or where rapid model prototyping is required.

Future research could explore several avenues. First, a deeper analysis of the cases where certain metrics perform well could reveal more about the relationship between data characteristics and the effectiveness of specific proxies. Second, hybrid optimization strategies that combine the speed of the proposed metrics with the robustness of cross-validation could be developed. For instance, the proxy metrics could be used to quickly narrow down the hyperparameter search space, followed by a more focused cross-validation on the most promising candidates. Finally, the anomalous results observed for some datasets highlight the importance of robust

TABLE II
MEAN ACCURACY OF K-NN WITH DIFFERENT METRICS

Dataset	Baseline (accuracy)	I	II	III	IV	V	VI
Banknote Authentication	0.942	0.887	0.887	0.887	0.881	0.887	0.887
Breast Cancer	0.947	0.887	0.899	0.910	0.913	0.910	0.887
Car Evaluation	0.850	0.700	0.824	0.701	0.700	0.850	0.701
Credit Germany (Statlog) ¹	0.725	0.000	0.000	0.000	0.703	0.708	0.000
Diabetes (Pima)	0.751	0.699	0.699	0.723	0.734	0.723	0.723
Heart Disease (Statlog)	0.577	0.570	0.570	0.574	0.587	0.574	0.580
Ionosphere	0.860	0.717	0.717	0.780	0.780	0.746	0.717
Iris ²	0.000	0.826	0.820	0.833	0.813	0.846	0.833
Monk's Problems (monk2)	0.754	0.722	0.747	0.747	0.752	0.747	0.747
Mushroom	1.000	0.892	0.892	0.892	0.991	0.996	0.915
Phoneme	0.809	0.775	0.775	0.775	0.745	0.775	0.775
Spambase	0.885	0.756	0.756	0.788	0.796	0.756	0.755
Titanic	0.779	0.742	0.742	0.773	0.754	0.773	0.773
Wine	0.983	0.961	0.960	0.961	0.971	0.977	0.972
Yeast	0.584	N/A	0.529	0.507	0.507	0.505	N/A

¹ Models with zero accuracy failed to complete the cross-validation.

² An unknown issue caused the baseline model to return zero accuracy on the Iris dataset, likely due to a data preprocessing error or an issue with the k-NN implementation.

TABLE III
MEAN TRAINING TIME (SECONDS) OF K-NN WITH DIFFERENT METRICS

Dataset	Baseline	I	II	III	IV	V	VI
Banknote Auth.	4.599	1.334	1.603	1.659	1.599	1.585	1.568
Breast Cancer	3.841	0.485	0.383	0.409	0.385	0.503	0.516
Car Evaluation	9.923	2.816	3.707	2.663	3.568	3.173	3.236
Credit Germany	14.604	3.980	3.923	3.945	3.872	4.006	4.067
Diabetes (Pima)	4.760	1.158	1.110	0.660	1.027	0.737	0.765
Heart Disease	3.152	1.762	0.618	0.268	0.373	0.291	0.246
Ionosphere	10.724	1.365	1.666	1.425	1.284	1.414	1.511
Iris	5.783	0.204	0.321	0.306	0.213	0.262	0.249
Monk's Problems	8.205	0.569	0.553	0.385	0.424	0.507	0.561
Mushroom	67.514	56.455	55.503	56.398	57.148	66.105	67.027
Phoneme	48.991	37.407	31.158	31.648	36.572	46.664	41.788
Spambase	38.910	42.578	43.507	43.606	42.967	51.477	51.589
Titanic	8.908	3.320	2.280	2.585	2.129	2.769	2.776
Wine	5.859	0.302	0.581	0.532	0.326	0.438	0.466
Yeast	10.429	N/A	N/A	3.882	2.389	3.916	3.954

TABLE IV
MEAN INFERENCE TIME (SECONDS) OF K-NN WITH DIFFERENT METRICS

Dataset	Baseline	I	II	III	IV	V	VI
Banknote Auth.	0.00347	0.0100	0.0124	0.0160	0.0246	0.0108	0.0107
Breast Cancer	0.00257	0.00377	0.00337	0.00360	0.00385	0.00330	0.00390
Car Evaluation	0.0470	0.0366	0.0172	0.0229	0.0420	0.0380	0.0238
Credit Germany	0.298	0.149	0.165	0.153	0.161	0.154	0.165
Diabetes (Pima)	0.00339	0.0110	0.0138	0.00554	0.0120	0.00530	0.00560
Heart Disease	0.00243	0.00254	0.00230	0.00238	0.00240	0.00245	0.00248
Ionosphere	0.286	0.081	0.093	0.070	0.082	0.0611	0.101
Iris	0.00234	0.00210	0.00194	0.00232	0.00170	0.00180	0.00185
Monk's Problems	0.00300	0.00530	0.00310	0.00301	0.00650	0.00395	0.00320
Mushroom	0.655	0.780	0.807	0.788	0.954	0.594	0.537
Phoneme	0.0472	0.290	0.268	0.282	0.571	0.330	0.312
Spambase	0.343	0.594	0.700	0.696	0.702	0.685	0.687
Titanic	0.0122	0.036	0.0186	0.021	0.0304	0.0184	0.0175
Wine	0.00271	0.0030	0.0060	0.0047	0.00275	0.0042	0.00350
Yeast	0.0075	N/A	N/A	0.0296	0.0193	0.026	0.022

experimental design and suggest that further investigation into the stability and reliability of these metrics is warranted.

In conclusion, while the proposed similarity space metrics do not fully replace the need for traditional cross-validation, they offer a valuable set of tools for practitioners who need to balance the trade-off between computational cost and classification performance in the context of k-NN hyperparameter tuning.

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-validated 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. Refaellizadeh, 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+\(Diagnostic\)](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+\(german+credit+data\)](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+\(heart\)](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.
- [72] M. V. F. Menezes, L. C. B. Torres, and A. P. Braga, "Width optimization of RBF kernels for binary classification of support vector machines: A density estimation-based approach," *Pattern Recognition Letters*, vol. 128, pp. 1-7, 2019.