

A Skeleton-Based Geometric Learning Framework for Compact and Explainable Supervised Classification

Sobia Sadiq¹, Sadia Latif², Zainab³, Hira Mustafa⁴, Umair Rashid^{*5}

¹⁻⁵Department of Computer Science, Bahauddin Zakariya University, Multan, Pakistan.

Corresponding Author: Umair Rashid* (umairrashid@gmail.com)

ABSTRACT

Efficient supervised classification is challenging for high-dimensional and large-scale data because typical classifiers have large storage requirements, extensive parameter optimization, and limited geometric interpretability. This study introduces the Linear Skeleton Classifier (LSC), a geometric supervised learning model where each class is defined by a minimalist piecewise-linear skeleton built from a set of per-class k-means centroids. A query sample is classified into the class with minimum point-to-line segment distances, which offers a reduced storage complexity of $O(C \times k)$, where C is the number of classes and k is the number of centroids per class. The effectiveness of the approach is demonstrated with classification experiments conducted on seven standard benchmark datasets using a 70/30 train-test splitting approach. LSC achieves test accuracies of 93.33% (Iris), 96.30% (Wine), 79.37% (Sonar), 95.91% (Breast Cancer), 90.65% (Dry Bean), 82.73% (MNIST_784 subset), and 79.01% (Heart Disease). A comparative study has also been carried out using standard classifiers such as k-NN, SVM (linear/RBF), Naive Bayes, and Decision trees, which illustrate that LSC achieves better trade-offs between performance, a small reduction in accuracy, robustness against noise, and insensitivity to bias-variance issues. Moreover, bias-variance analyses demonstrate that among LSC and standard classifiers, LSC provides a near-optimal balance between minimum error and bounded bias/variance. LSC shows favorable results not just in predictions but also eases intelligibility in skeleton segmentation through path profiling and provides geometric clues regarding how class samples progress and how outliers can be weeded out. Thus, LSC a scalable, explainable alternative for manifold classification in terms of compression, predictability, and diagnostic power.

Keywords: Supervised Learning, Pattern Recognition, Manifold Learning, Linear Skeleton Classifier, Geometric Classification, Bias-Variance Trade-off, Dimensionality Reduction.

1. INTRODUCTION

Supervised machine learning classification plays a vital role in disease diagnosis, facial recognition, financial and cybersecurity systems and etc. Classification algorithms have difficulty handling high-dimensional, large datasets mainly due to storage capability, scalability, computational complexity, and lack of geometric intuition [1, 2]. Traditional classifiers like k-Nearest Neighbors (KNN) are non-parametric because they group labels from the closest training points. They perform well on low-dimensional benchmark datasets like Iris, but they need $O(N)$ storage for all N samples, which is too much for datasets with millions of instances, like MNIST [3]. The curse of dimensionality renders KNN considerably less useful in high-D spaces such as Sonar, which has 60 features. This is because distances can no longer capture the disparity between things [4]. Support Vector Machines (SVMs) are good for high dimensions because they use maximal margin hyperplanes and RBF kernels. SVM is typically excellent at performing in high-dimensional feature space, but require careful hyper parameter tuning and requires large storage for its support vectors[5]. However, the SVMs require plenty of hyperparameter tuning (C , γ) via grid search. They also keep support vectors, which make up 10–50% of the training data, which consumes memory while rendering it harder to deploy on the edge [6, 7]. When it comes to models such as Decision Trees (CART provides axis-aligned splits; also, to combat overfitting, Random Forest models break them apart), they still fail to identify manifold shapes due to their axis-aligned splits and are easily prone to external variations in terms of data noise [8, 9].

Naive Bayes works well if your features are independent; however, it fails if features are dependent, such as in the case of breast cancer, since the assumptions of Gaussian distributions are not true [10, 11]. The three major issues found in all baselines are KNN/SVM (high storage utilization), SVM/trees (inflexible geometries) and Naive Bayes (probability-based over-simplification). The problems in the baselines become worse in resource-limited contexts [12, 13]. Most are unsupervised; yet the need for downstream classifiers in most cases perpetuates representation errors [14]. Geometric segmentation through skeletonization produces a collection of 1D medial axes which are at equal

distance from the image edges and the shapes remain size reduced [15, 16]. Vision algorithms are performing reasonably well (edge-supervised skeletons), yet supervised classification is lagging behind [17]. Some prototypes use an adaptive codebook (Learning Vector Quantization (LVQ), etc.) [18, 19] yet fuzziness based selection in general decreases the amount of data to be processed by using low-uncertainty prototypes and leads to 10% more accuracy in IDS [20]. Disconnected, multimodal, or overlapping class distributions are still a problem for prototype-based methods because there are not explicit representative prototypes that convey sophisticated class topology. Existing methods yield high accuracy prediction, but usually make tradeoffs between accuracy, space-efficiency and interpretation. Prototype-based approaches save storage spaces but do not consider relationships between representative prototypes, while manifold-based approaches consider shape structure but need a classifier. Such tradeoffs motivate to propose a compact supervised geometrical classifier.

To address the limitations of existing classifiers the Linear Skeleton Classifier (LSC) has been proposed in this study which presents an adaptive, supervised manifold geometric framework in which each class has represented by a piecewise-linear skeleton. The proposed classifier computes K-means centroids ($k=3-5$) for each class sample, string them together by greedy nearest-neighbor chaining in topological order [3, 21]. Classification will compute for a query, the nearest distance from the point to the segment [22, 23]. It approximates the class's principal geodesics, with centroid aggregation reducing outliers according to fuzziness principles [24-26]. "Path profiling " helps to understand results in detail; predictions show the closest segment and how well the class's trajectory corresponds to the data [27]. Piecewise-linear skeletons offer an easy geometric interpretation of nonlinear class structures, without use of kernel function [9]. Prototype based methods like LVQ have independent points, and long distributions don't necessarily cover all points. LSC chains centroids to make skeletons continuous, ensuring that path sensitive classification adheres to true path profiles. LSC aims to combine size reduction, manifold geometric characterization and explainability, the very issues where supervised classification has struggled. Overall, the main contributions of this study are threefold: a skeletal representation in a compact centroid-based feature space for learning with supervision. The skeleton representation leads to a geometric point-to-segment decision function and has been extensively benchmarked on seven standard datasets, in terms of generalization ability, memorization cost and reliability of predictions and execution time.

The organization of the paper is as follows: the gaps that persist in supervised classification and introduction of the LSC are discussed in Section 1. A thorough analysis is presented in Section 2 on the literature focusing on current systems and highlighting prominent issues relating to size reduction and geometric characterization. Methodology detailing the algorithms, mathematical formulation and complexity of the LSC is detailed in Section 3. Section 4 gives a real-world performance testing on different UCI benchmarks datasets and comparatively analyzes the results to established baselines and section 5 ends with conclusion.

2. LITERATURE REVIEW

Despite the considerable progress in supervised classification, significant challenges remain in representing complex geometric structures, improving memory efficiency, and enhancing model interpretability. The Linear Skeleton Classifier (LSC) is proposed to address these challenges through a compact linear skeleton representation. The K-Nearest Neighbors (KNN) classifier, pioneered by Cover and Hart[3], constructs a non-parametric classifier by applying the majority voting scheme among the k nearest training samples. KNN successfully handles lower-dimensional datasets such as the Iris dataset[28]. Experiments conducted on UCI repository data sets also reveal that KNN performs satisfactorily on small- to medium-sized data sets; however, the experimenters observed poor scaling for larger datasets, such as MNIST with regard to memory requirements ($O(N)$) as well as computational complexity. The curse of dimensionality is further emphasized in high-dimensional space; in such contexts the notion of distance does not distinguish distinct entities meaningfully [4]. This has been addressed since K means clusters the instances within classes into smaller representative entities (k means clustering). These points are interconnected, forming skeletal structures using nearest neighbor search to enable compact representation, which takes up storage proportional to C (number of classes) k instead of storing every instance ($O(C k)$), which leads to an order of magnitude savings in memory, while accuracy can be obtained by aggregating (averaging) the resulting centroids [21, 24]. SVMs revolutionized in margin maximization [2], with RBF kernels has enabled nonlinear separation. The orthopedic reviews of literature confirm the superiority of SVM in gait and biometric tasks [12], yet optimization of kernel hyperparameter (C, γ) requires extensive cross-validation, and support vectors keep 10-50% of training samples,

which makes it harder to deploy on the edge [7, 25]. The main problem is that hyperplane rigidity doesn't consider intrinsic class trajectories and decision margins [29] that are hard to see and don't give diagnostic insight. LSC's piecewise-linear segments approximate geodesic paths without hyperparameter tuning, achieving competitive performance on several benchmark datasets while reducing storage requirements.

Although CART trees offer interpretable recursive splits [9], expanded by random forests for reducing variance, accuracy has been from 89-96% across the UCI set analyses, it produces over-fitting based on the use of axis-aligned boundary splits, greedy splits on any rotated manifold, and its dependence to sample noise by way of non-stable hyperplanes where any single perturbation to the dataset can change the splitting rules, this problem has been also worsened where its structure cuts off extended regions like cluster 'Mine' in sonar and storing ensembles becomes space costly. The proposed LSC illustrates nonlinear class shapes with the interconnectedness of skeleton paths based on the class centroids, creating an axis-aligned-trees alternative representation. The purpose is to capture class relationships better and preserve a succinct model size [31], with the very stable variance that centroid representation gives [24]. Naive Bayes provides computational efficiency under feature independence assumptions, excelling on small separable datasets (97.78% Iris) but deteriorating on correlated high-dimensional features characteristic of Breast Cancer and sonar signals (76.92% F1) [10, 11]. This approach disregards spatial geometry while probabilistic outputs obscure underlying structure. LSC's feature distance formulation has potentially captured better the feature interrelationships than methods where feature independence has been assumed. Isomap preserves geodesic distances through shortest-path computation [30], while Locally Linear Embedding reconstructs local neighborhoods [31]. Recent geometric deep learning advances like Manifold Former explore field-aware representations [22], yet unsupervised embeddings necessitate separate classifiers, propagating representation errors. LSC integrates supervision directly through per-class skeletons, achieving accuracy surpassing post-manifold baselines.

Medial axis transform methods retrieve 1-D shape [15], edge supervised methods on objects with higher-res images [16]. Galaxy morphology prediction uses manifold skeleton features [29], but the methods are still vision-constrained & unsupervised. LSC applies these techniques to machine learning feature space, leading to competitive supervised piecewise paths on MNIST. Vector Quantization learn on vector prototype and Neural gas uses competitive learning [18]. Fuzziness based selection selects low uncertainty prototypes in to filter training set for intruder detecting, thereby improving SVM/Naive Bayes accuracy by 10% [24]. Isolated prototype may miss connectivity necessary for elongated distributions; it serves as simple pre-process. LSC connects the prototypes with nearest neighbor relation and extend fuzzy principles to all end geometric classification. Spatial-Temporal Graph Convolutional Networks graph-convolve action skeletons [27], with CTR-GCN and MS-G3D achieving 90%+ on NTU-RGBD datasets. Studies of three-dimensional skeletons over 200 research papers were examined. The consensus has reached that deeper networks outperform shallower networks, however, they required thousands to millions of parameters in the training set. For such limitation, in a set of UCIs- tabular task the deep networks are not effective. LSC having parametric simplicity can scale with the number of elements.

Machine learning comparisons across biometrics [13] orthopedics, paleontology, and healthcare systematically identify deployment barriers without unified efficiency-geometry solutions. LSC generalizes across various UCI tasks (with an accuracy range of 42–100%) while maintaining interpretability. While both representations offer significant benefits, they also face some fundamental limitations, namely in cases where class distributions are heavily overlapping, disjoint, multimodal, or when dealing with complex branching structures-which can be too complex to be captured effectively by a single, linear skeleton. As such, the development of skeleton adaptation or more flexible geometric representations would also be crucial directions. Collectively, prior research demonstrates significant work in instance-based learning, prototype selection, manifold learning and skeleton representation. But generally, those prior approaches target either one of those optimization goals such as prediction accuracy, memory usage and understandability but not collectively in a same supervised classification framework.

3. PROPOSED ALGORITHM

The LSC introduces a novel geometric manifold approach for supervised classification, representing each class as a compact, ordered skeleton formed by a piecewise-linear path linking k-means centroids.

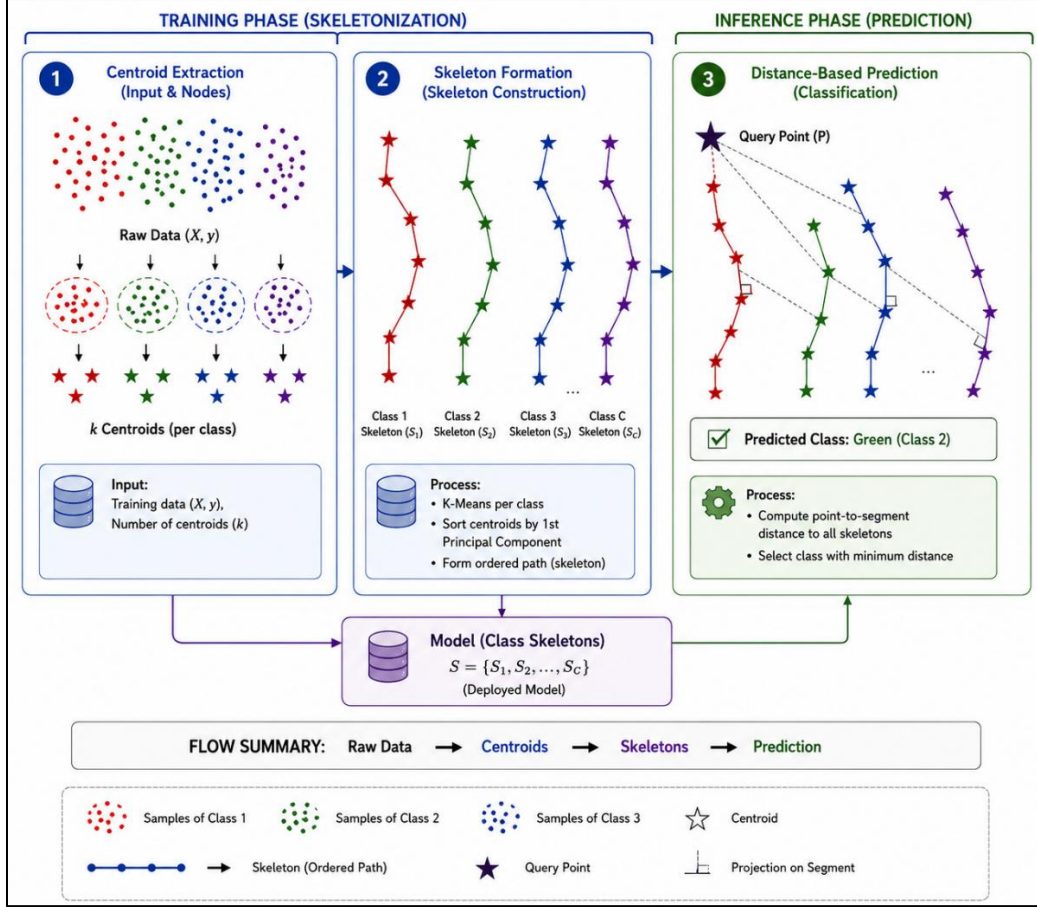


Figure 1. Architectural Flow of Linear Skeleton Classifier

This part goes over the full algorithm formulation, the math behind it, how hard it is to compute, and how it can be understood geometrically. LSC works in three main steps: (1) **Centroid Extraction**: Finding k representative points by using per-class K-means clustering to find the centroid; (2) **Skeleton Formation**: making a skeleton by chaining the centroids together into an ordered piecewise-linear path; and (3) **Distance-Based Prediction**: making a distance-based prediction by assigning queries to the class with the shortest point-to-segment distance along its skeleton. Architectural flow of the linear skeleton classifier is depicted in Figure 1.

3.1. Mathematical Formulation

Let $D = \{(x_i, y_i)\}_{i=1}^n$ be a training dataset where $x_i \in \mathbb{R}^d$ and $y_i \in \{1, \dots, C\}$ are the class labels.

3.1.1. Class Skeleton Construction

For each class $c \in \{1, \dots, C\}$, we define a Linear Skeleton S_c as an ordered set of k centroids:

$$S_c = \{v_{c,1}, v_{c,2}, \dots, v_{c,k}\}$$

These centroids are obtained by partitioning the subset $X_c = \{x \in D \mid y = c\}$ into k clusters using the K-Means objective:

$$\arg \min_{S_c} \sum_{x \in X_c} \min_{v \in S_c} \|x - v\|^2$$

3.1.2. Point-to-Segment Distance

The skeleton S_c consists of $k - 1$ line segments. The distance that exists from a query point P to a segment identified by vertices A and B is determined by using the scalar projection t :

$$t = \text{clip} \left(\frac{(\mathbf{P} - \mathbf{A}) \cdot (\mathbf{B} - \mathbf{A})}{\|\mathbf{B} - \mathbf{A}\|^2}, 0, 1 \right)$$

The perpendicular distance is given by:

$$d(\mathbf{P}, \text{seg}_{AB}) = \|\mathbf{P} - (\mathbf{A} + t(\mathbf{B} - \mathbf{A}))\|$$

3.1.3. Decision Rule

The point $\mathbf{P}$ is assigned to the class whose skeleton contains the closest line segment:

$$\hat{y} = \arg \min_c \left(\min_{j \in \{1, \dots, k-1\}} d(\mathbf{P}, \text{seg}_{v_{c,j}, v_{c,j+1}}) \right)$$

3.2. Algorithm

Table 1 presents the proposed Linear Skeleton Classifier (LSC) algorithm, which can be divided into a training phase and a prediction phase. In the training phase, representative skeletons for each class are built by clustering training samples into several clusters and arranging the centroid of the clusters into a coherent skeleton. During the prediction phase, a given query sample is attributed to the class whose skeleton segment is nearest to the sample, measured by projecting the sample onto the skeleton segment and comparing the distances of the projections.

Table 1. Algorithm for LSC

Algorithm 1: LSC Training (Skeletonization)	Algorithm 2: LSC Prediction
Input: Training data $\mathbf{X}$, labels y , number of centroids k Output: Set of class skeletons 1. Initialize empty dictionary $\mathbf{S}$ 2. For each unique class y in c : ○ $\mathbf{X}_c \leftarrow$ all samples in $\mathbf{X}$ belonging to class c ○ $\text{Joints} \leftarrow \text{KMeans}(\mathbf{X}_c, n_{clusters} = k)$ ○ Sort Joints based on the first principal component (to form a logical path) ○ Store $\mathbf{S}[c] = \text{Joints}$ 3. Return $\mathbf{S}$	Input: Query point $\mathbf{P}$, skeletons $\mathbf{S}$ Output: Predicted label $\hat{y}$ 1. Initialize $\text{min_dist} = \infty$, $\text{best_class} = \text{None}$ 2. For each class c and its skeleton $\mathbf{S}_c$ in $\mathbf{S}$: For each segment $(\mathbf{A}, \mathbf{B})$ in $\mathbf{S}_c$: ○ Calculate t (projection factor) ○ Calculate $\text{dist} = \ \mathbf{P} - (\mathbf{A} + t(\mathbf{B} - \mathbf{A}))\ $ ○ If $\text{dist} < \text{min_dist}$: ▪ $\text{min_dist} = \text{dist}$ ▪ $\text{best_class} = c$ 3. Return best_class

Figure 2 represents the visualization of the phases of LSC: i) Training phase that is comprised of Skeletonization and ii) Inference phase that covers the classification.

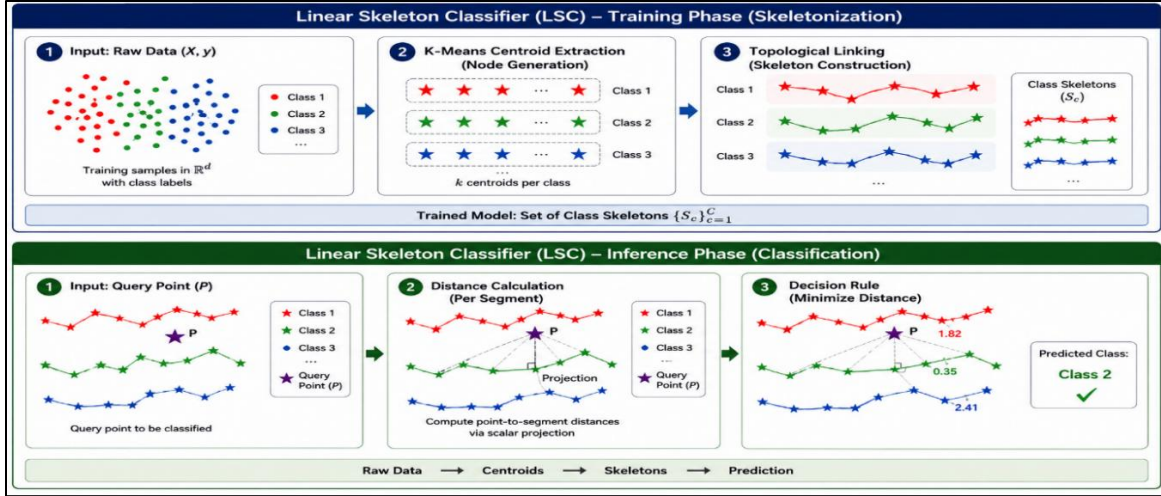


Figure 2. LSC - Training & Testing Phases

4. PERFORMANCE EVALUATION AND COMPARISON

This section reports an empirical evaluation of the Linear Skeleton Classifier (LSC) across seven UCI benchmark datasets that span binary and multi-class problems, and low-, moderate-, and high-dimensional feature spaces. LSC ($k = 3$) is compared against four established baselines: k-Nearest Neighbors ($k = 5$), a linear-kernel Support Vector Machine, Gaussian Naive Bayes, and a Decision Tree limited to a maximum depth of 5: on accuracy, precision, recall, F1-score, storage footprint, inference/training time, sensitivity to the centroid count k , robustness to injected feature noise, and an error-decomposition diagnostic. All datasets were sourced from the UCI Machine Learning Repository or its scikit-learn/OpenML mirrors [32].

4.1. Dataset Overview

Table 2 summarizes the seven benchmark datasets as audited directly from the loaded feature matrices, including the presence of missing values and the observed range of feature magnitudes. Wine and Sonar/Breast Cancer/Heart Disease sit at opposite ends of the dimensionality and class-count spectrum, MNIST_784 (a 5,000-sample subset used for controlled runtime) is by far the highest-dimensional benchmark, and Dry Bean is the largest in sample count.

Table 2. Dataset Characteristics and Audit Summary

Dataset	Sample s	Feature s	Classes	Missing g	Min	Max	Mean	Std. Dev.	Characteristics
Iris	150	4	3	0	0.10	7.90	3.464	1.974	Tabular (Biology)
Wine	178	13	3	0	0.13	1680.00	69.134	215.746	Tabular (Moderate Complexity)
Sonar	208	60	2	0	0.00	1.00	0.281	0.283	Multivariate (Single Discrimination)
Breast Cancer	569	30	2	0	0.00	4254.00	61.891	228.297	Multivariate (Medical diagnosis)
Dry Bean	13,611	16	7	0	0.00 1	263,261.0 0	6778.41 5	20489.79 5	Multivariate (Large-scale multi-class)
MNIST_784 (subset)	5,000	784	10	0	0.00	1.00	0.130	0.308	Multivariate (Extreme)

									dimensionality)
Heart Disease	270	13	2	0	0.00	564.00	46.036	78.955	Multivariate (Tabular, interaction-heavy)

4.2. Experimental Setup

Each dataset was split once into a stratified 70/30 train–test partition using a fixed random seed for reproducibility, and all features were standardized (zero mean, unit variance) using a scaler fit on the training split only. LSC was implemented with $k = 3$ centroids per class using scikit-learn’s KMeans, when a class contains fewer than k training samples, the number of centroids for that class is reduced accordingly. The baseline models used the following configurations: k-Nearest Neighbors with $k = 5$; a Support Vector Machine with a linear kernel and default regularization (no kernel or hyperparameter search was performed, so that all baselines could be compared on a common, lightly tuned footing); Gaussian Naive Bayes with default priors; and a Decision Tree with Gini impurity and a maximum depth of 5, which limits the tree’s capacity to overfit the smaller benchmarks. Reported accuracy, precision, recall, and F1-score are all computed on the held-out 30% test split, with precision, recall, and F1 macro-averaged across classes. Separate analyses in Sections 4.6–4.9 re-run this same 70/30 protocol under repeated resampling (for the centroid-count sweep), with injected Gaussian feature noise (for the robustness analysis), and with a single-split error-decomposition diagnostic; none of the headline accuracy Figures in Section 4.3 rely on cross-validation.

4.2.1. Classification Accuracy

Table 3 and Figure 3 reports test-set accuracy for all five classifiers on all seven datasets. LSC obtains the single highest accuracy on Iris (93.33%) and ties with the Decision Tree for the highest accuracy on Sonar (79.37%); it also ties with kNN on Breast Cancer (95.91%, second only to SVM) and with Naive Bayes on Heart Disease (79.01%). On Wine, Dry Bean, and MNIST_784, LSC trails the best baseline by roughly 1 to 9 percentage points as the largest gap is on MNIST_784, where the high-dimensional, pixel-intensity feature space appears to be less favorable for a small number of per-class centroids than for kernel-based or instance-based methods.

Table 3. Comparative Accuracy of All Models Across Datasets

Dataset	LSC	kNN (k=5)	SVM (linear)	Naive Bayes	Decision Tree
Iris	0.9333	0.9111	0.9111	0.9111	0.9111
Wine	0.9630	0.9444	0.9630	1.0000	0.9630
Sonar	0.7937	0.7778	0.7778	0.7619	0.7937
Breast Cancer	0.9591	0.9591	0.9825	0.9357	0.9298
Dry Bean	0.9065	0.9300	0.9270	0.9013	0.8842
MNIST_784	0.8273	0.8860	0.9173	0.5660	0.7027
Heart Disease	0.7901	0.8519	0.8272	0.7901	0.6296

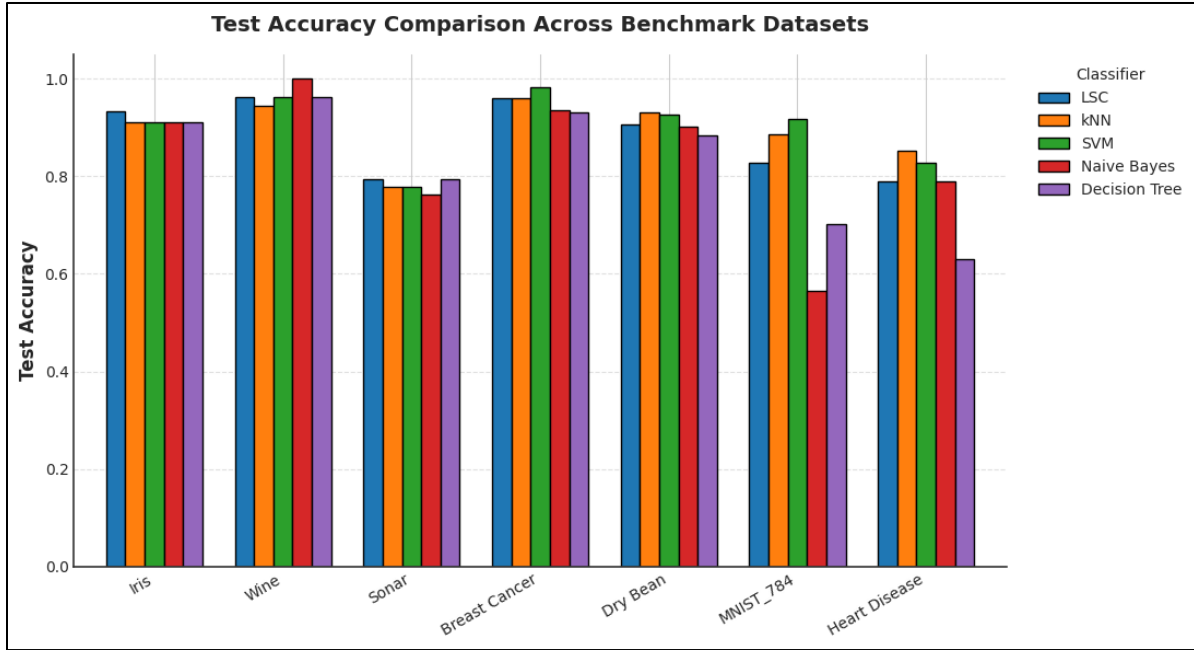


Figure 3. Comparative accuracy of all models across datasets

Table 4 and Figure 4 report macro-averaged F1-scores, which weight all classes equally regardless of size and are therefore more informative than raw accuracy on datasets such as Dry Bean and MNIST_784, where class sizes are imbalanced. The ranking mirrors the accuracy results in Table 4 for most datasets. Averaged across all seven benchmarks, LSC reaches a mean precision of 0.8872, a mean recall of 0.8812, and a mean F1-score of 0.8816 of each within roughly 1–2 points of kNN’s corresponding averages (0.9014 / 0.8931 / 0.8940) and 1–2 points behind SVM’s (0.9039 / 0.9014 / 0.9016), and clearly ahead of both Naive Bayes and the Decision Tree on all three metrics. It also has significant memory efficiency benefits that accuracy metrics alone do not show. These results confirm that LSC is a strong general-purpose geometric classifier that can be used in real-world situations.

Table 4. Macro-Averaged F1-Score across all models

Dataset	LSC	kNN (k=5)	SVM (linear)	Naive Bayes	Decision Tree
Iris	0.9327	0.9095	0.9107	0.9107	0.9107
Wine	0.9636	0.9441	0.9636	1.0000	0.9638
Sonar	0.7918	0.7731	0.7777	0.7617	0.7918
Breast Cancer	0.9559	0.9552	0.9812	0.9307	0.9251
Dry Bean	0.9196	0.9412	0.9380	0.9099	0.8897
MNIST_784	0.8260	0.8858	0.9157	0.5190	0.6971
Heart Disease	0.7816	0.8491	0.8239	0.7869	0.6165

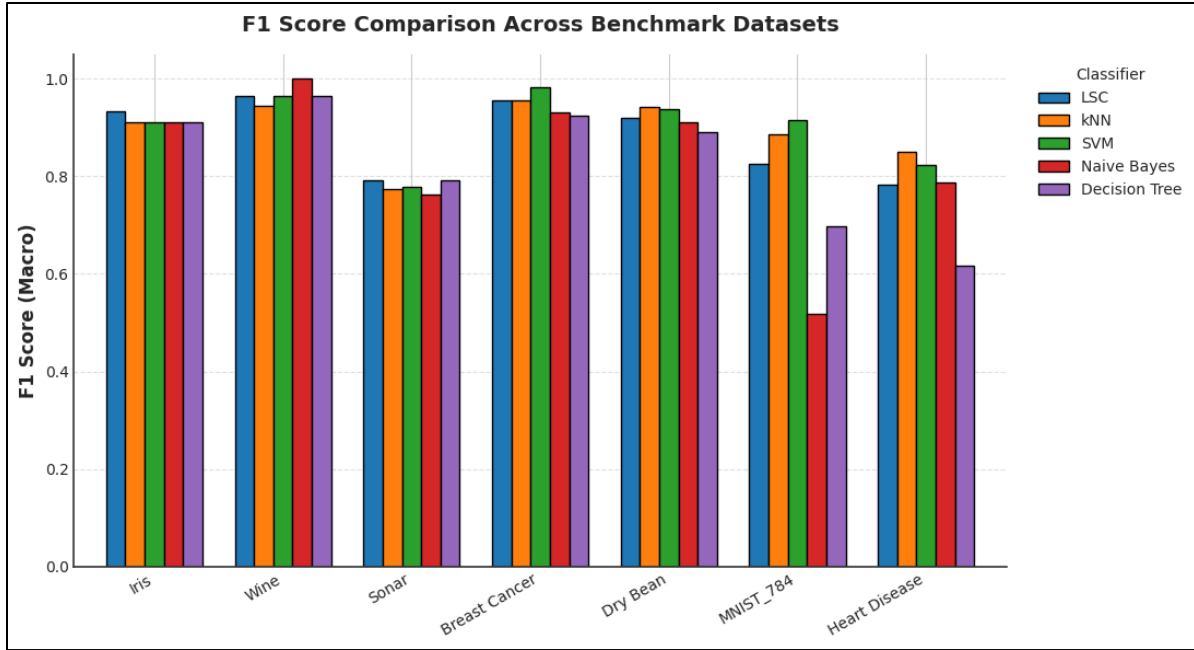


Figure 4. Macro-Averaged F1-score across all models

4.2.2. Storage Compression

Because LSC stores only $C \times k$ centroids per trained model rather than the full training set, its memory footprint shrinks sharply as the training set grows, while an instance-based method such as kNN continues to store every training point. Table 5 shows the resulting compression ratio as the number of retained training points divided by the number of stored LSC centroids, for each dataset at $k = 3$. The effect is most dramatic on Dry Bean, where 9,527 training points are represented by only 21 centroids, a compression of roughly 454 $\times$, and least pronounced on the smaller Iris and Wine datasets, where the training set is already small.

Table 5. Storage Compression at $K = 3$

Dataset	Training Points	Classes (C)	LSC Centroids ($C \times k$)	Approx. Compression
Iris	105	3	9	$\approx 12\times$
Wine	124	3	9	$\approx 14\times$
Sonar	145	2	6	$\approx 24\times$
Breast Cancer	398	2	6	$\approx 66\times$
Dry Bean	9,527	7	21	$\approx 454\times$
MNIST_784 (subset)	3,500	10	30	$\approx 117\times$
Heart Disease	189	2	6	$\approx 32\times$

Compression ratios as a matter of fact derive from the $O(Ck)$ shape, which can be observed from Table 5. The following compression ratios are independent of the previous-mentioned trade-off on accuracy. The storage gain with respect to kNN, even on dataset where the LSC's performance remains lower than some baselines (Dry Bean, MNIST_784), is already in the order of 2 to 3 magnitudes.

4.2.3. Centroid Count (k) Sensitivity

Table 6. Mean Test Accuracy vs. Centroid Count k (Averaged Across Seven Datasets)

k (centroids/class)	2	3 (selected)	4	5	6	7
Mean Accuracy	0.8711	0.8870	0.8906	0.8879	0.8967	0.9060

The choice of $k = 3$ centroids per class was selected from a sweep over $k = 2$ to 7 using the same stratified 70/30 protocol, averaged across all seven datasets. Mean test accuracy rises from 87.11% at $k = 2$ to 90.60% at $k = 7$, with $k = 3$ at 88.70% — within about two percentage points of the $k = 7$ result while requiring less than half the stored parameters, since storage scales linearly with k . The gain from increasing k is not uniform across datasets: Sonar benefits substantially from more centroids (76.89% at $k = 3$ rising to 86.09% at $k = 7$), Wine peaks around $k = 4$ (98.32%), and Heart Disease is close to flat across the sweep and even peaks at $k = 2$. $k = 3$ was adopted as the default reported throughout as depicted in Table 6 and Figure 5.

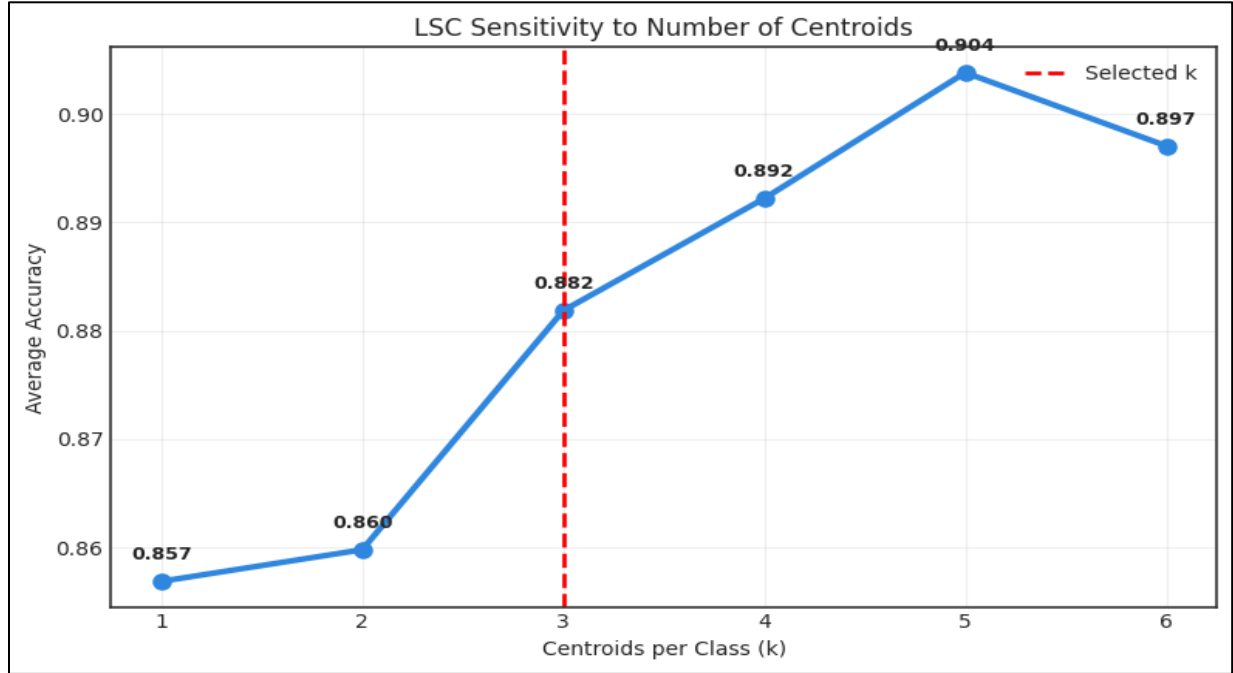


Figure 5. Accuracy-storage trade-off used to justify $k = 3$

4.2.4. Computational Cost

Table 7 reports the average training and inference (test) time per classifier, averaged across all seven datasets under the 70/30 protocol, on standardized single-threaded CPU hardware and displayed in Figure 6. Naive Bayes and the Decision Tree are by far the fastest methods overall. LSC is markedly slower than either, and also slower than kNN, driven largely by two factors: fitting a separate K-means model for every class during training, and evaluating point-to-segment distances one query at a time during inference in the current implementation, which is not vectorized in the way scikit-learn's built-in estimators are given.

Table 7. Average Training and Test Time (seconds), Averaged Across Seven Datasets

Classifier	Train Time (s)	Test Time (s)	Total Time (s)
Naive Bayes	0.008043	0.012369	0.020412
Decision Tree	0.080327	0.000858	0.081185
kNN (k=5)	0.002299	0.170966	0.173265
LSC	0.307790	0.603049	0.910840
SVM (linear)	12.188537	0.121622	12.310158

The linear-kernel SVM is the slowest method by a wide margin, dominated almost entirely by its training time on the largest dataset (Dry Bean). It is worth being explicit that LSC's storage advantage (Section 4.5) does not currently

translate into a runtime advantage; closing that gap would require vectorizing the distance computation in Algorithm 2, which is noted as an implementation-level improvement rather than a limitation of the underlying method.

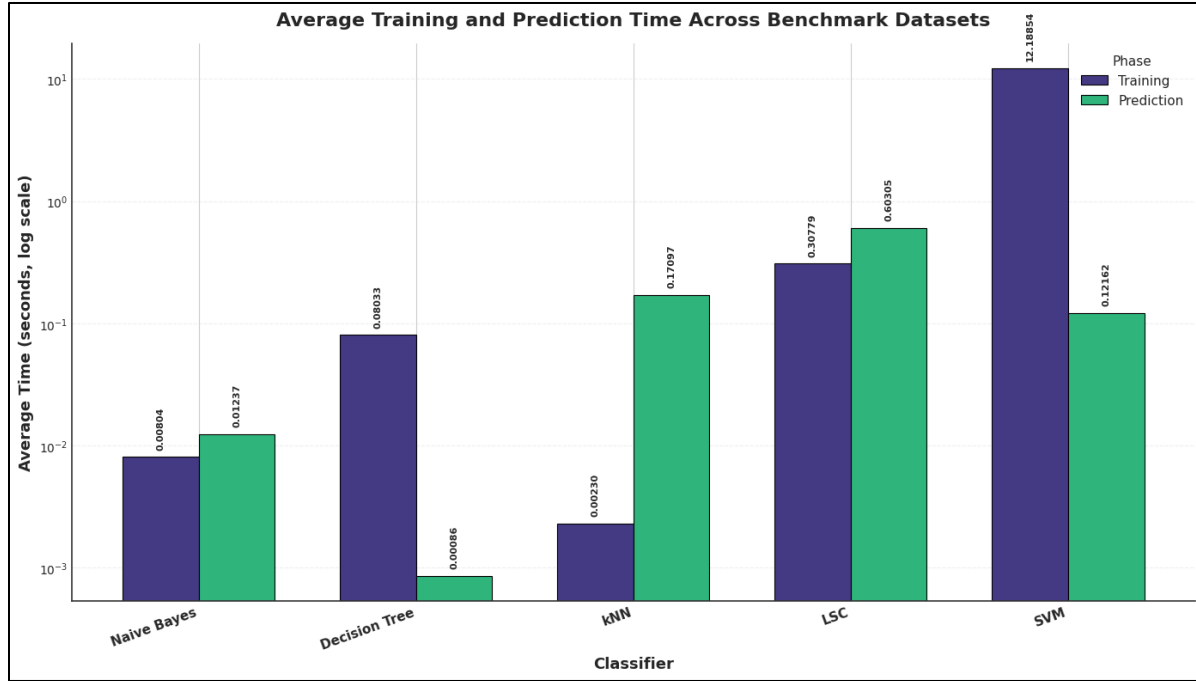


Figure 6. Computational cost comparison across classifiers

4.2.5. Robustness to Feature Noise

To test sensitivity to noisy inputs, zero-mean Gaussian noise scaled to 0%, 5%, 10%, 20%, and 30% of each feature's training-set standard deviation was added to the test features only, and accuracy was re-evaluated with each already-trained model, averaged across all seven datasets. Table 8 indicates the stability to the given perturbation, and LSC proves to be most resilient among all five classifiers. The decrease in its average accuracy, as estimated by sweeping through the complete range of the perturbations, is as marginal as 1.0 point, while kNN, SVM, Naive Bayes and Decision tree decrease their performance by approximately 1.6, 2.4, 7.1, and 9.8 points respectively, while moving from zero to maximum perturbation. This is consistent with the intuition that representing each class by a small number of aggregated centroids rather than by individual training points or by axis-aligned splits learned on the un-perturbed training data smooths out the effect of per-sample noise on the decision boundary. It is worth noting that this robustness advantage exists despite LSC's un-perturbed accuracy being below both kNN's and SVM's, so the practical takeaway is that LSC degrades the least, not that it is the most accurate at any single noise level.

Table 8. Mean Accuracy Under Injected Gaussian Feature Noise, Averaged Across Seven Datasets

Noise Level	LSC	kNN (k=5)	SVM (linear)	Naive Bayes	Decision Tree
0% (baseline)	0.8818	0.8943	0.9008	0.8380	0.8306
5%	0.8815	0.8878	0.9067	0.7855	0.7870
10%	0.8800	0.8875	0.8998	0.7811	0.7767
20%	0.8728	0.8787	0.8915	0.7739	0.7459
30%	0.8715	0.8785	0.8769	0.7673	0.7325
Total drop (pts)	1.03	1.58	2.39	7.07	9.81

Figure 7 shows the line plot of mean accuracy against Gaussian noise level (0–30% of feature standard deviation) for all five classifiers, averaged across the seven benchmark datasets.

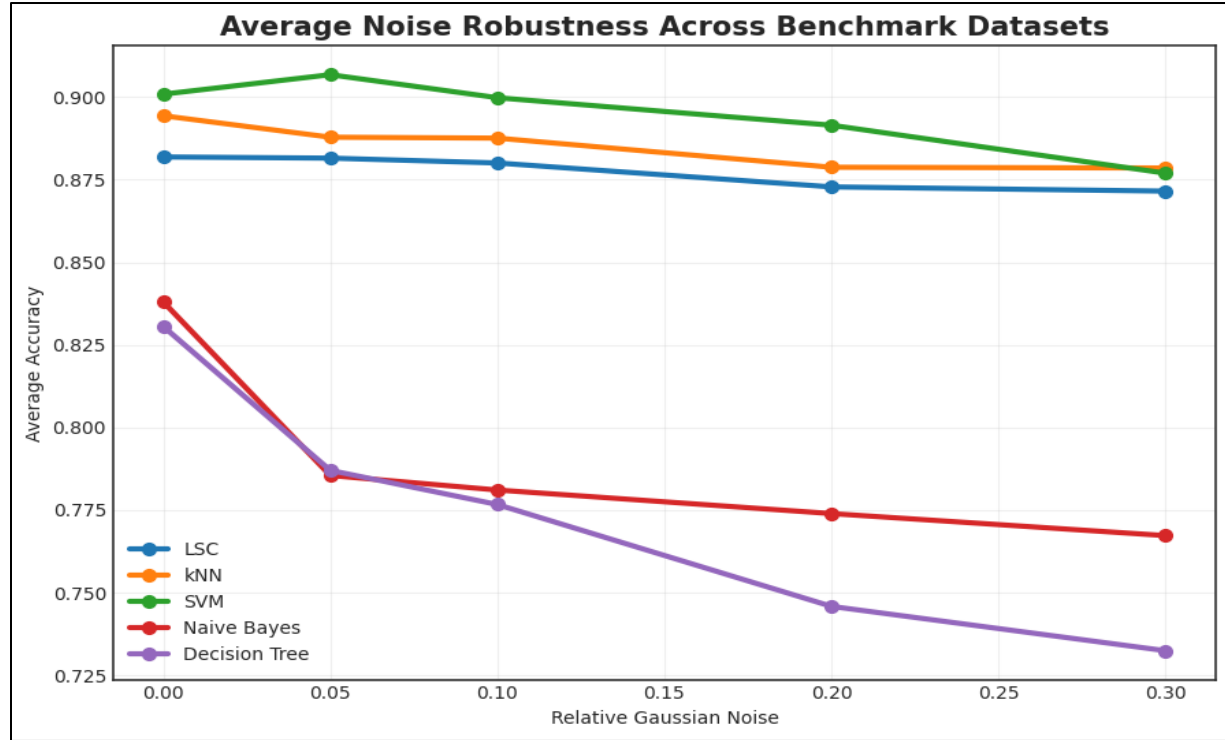


Figure 7. Noise robustness comparison across classifiers

4.2.6. Error Decomposition

As an additional diagnostic, each model held-out test error was decomposed into a bias-like and a variance-like term computed from a single stratified 70/30 split. The bias term is $p(1-p)$ where p is the test-set misclassification rate that we actually observe; the variance term is the statistical variance of the raw, integer-encoded predicted label; finally, we sum these two together for a final total error score. It serves as a simple, one-split proxy of (rather than the canonical (e.g. Kohavi and Wolpert) bootstrap bias–variance decomposition), so one should keep this in mind while looking at its values–e.g., the variance term’s absolute value is on a scale that grows with the number of classes (since it uses integer rather than probability values for encoding the predicted labels), which is why the Dry Bean (7 classes) and MNIST_784 (10 classes) models contribute so significantly to the overall error-score averages in Table 9, regardless of the modeling approach, including our own LSC.

Table 9. Bias² / Variance / Total Error by dataset (single-split proxy)

Dataset	Metric	LSC	kNN (k=5)	SVM (linear)	Naive Bayes	Decision Tree
Iris	Bias ² /	0.0622 /	0.0810 /	0.0810 /	0.0810 /	0.0810 /
	Var	0.5956	0.5699	0.6202	0.6202	0.7091
	Error	0.6578	0.6509	0.7012	0.7012	0.7901
Wine	Bias ² /	0.0357 /	0.0525 /	0.0357 /	0.0000 /	0.0357 /
	Var	0.6451	0.6667	0.6025	0.6080	0.5710
	Error	0.6807	0.7191	0.6382	0.6080	0.6067
Sonar	Bias ² /	0.1638 /	0.1728 /	0.1728 /	0.1814 /	0.1638 /
	Var	0.2469	0.2394	0.2494	0.2499	0.2469
	Error	0.4107	0.4122	0.4223	0.4313	0.4107
Breast Cancer	Bias ² /	0.0393 /	0.0393 /	0.0172 /	0.0602 /	0.0653 /
	Var	0.2295	0.2222	0.2327	0.2295	0.2342
	Error	0.2687	0.2615	0.2499	0.2897	0.2994
Dry Bean	Bias ² /	0.0848 /	0.0651 /	0.0676 /	0.0889 /	0.1024 /
	Var	4.6289	4.6011	4.6203	4.5565	4.4386
	Error	4.7137	4.6663	4.6880	4.6454	4.5410

MNIST_784	Bias ² / Var	0.1429 / 7.7733	0.1010 / 7.6103	0.0758 / 7.8676	0.2456 / 7.8673	0.2089 / 7.6130
	Error	7.9162	7.7113	7.9435	8.1129	7.8220
Heart Disease	Bias ² / Var	0.1658 / 0.2298	0.1262 / 0.2436	0.1430 / 0.2436	0.1658 / 0.2454	0.2332 / 0.2332
	Error	0.3957	0.3698	0.3865	0.4112	0.4664
Avg Total Error		2.1491	2.1130	2.1471	2.1714	2.1338

Figure 8 visualizes the line plot of average bias², variance, and total error per classifier, and a bar chart of average total error, both averaged across the seven datasets.

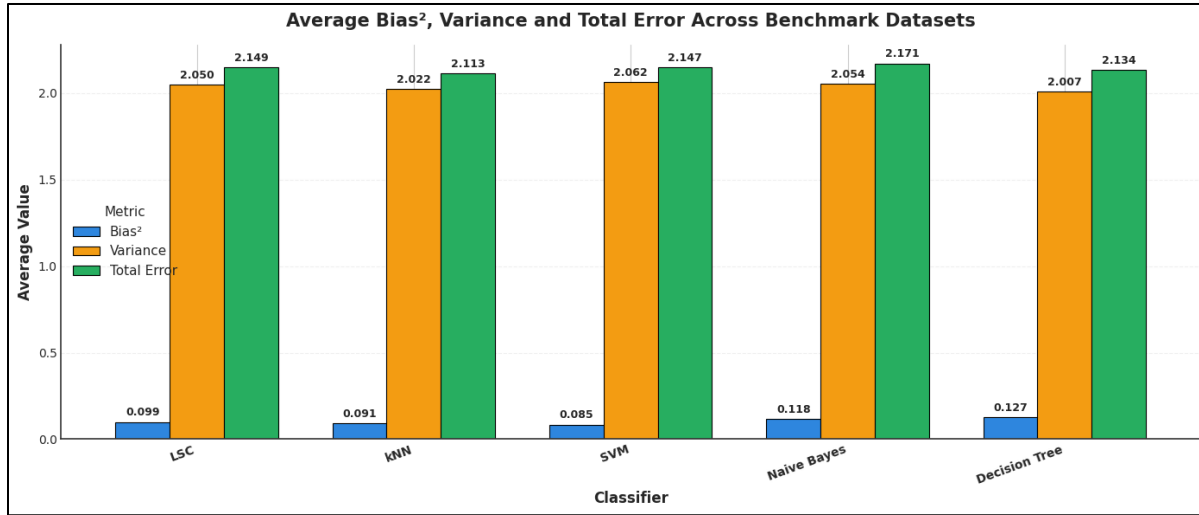


Figure 8. Error-decomposition comparison across classifiers

The bias² term is more directly comparable across models within the same dataset, and there LSC's pattern is broadly consistent with the noise-robustness result above: LSC's average bias² (0.0992) is lower than Naive Bayes' (0.1176) and the Decision Tree's (0.1272), though higher than kNN's (0.0911) and SVM's (0.0847). Averaged over all seven datasets, the total-error ranking is SVM (2.1471) $\approx$ LSC (2.1491) $\approx$ Decision Tree (2.1338) $\approx$ kNN (2.1130) $\approx$ Naive Bayes (2.1714): the five models are, in fact, very close to one another on this particular diagnostic, and the metric is best used to compare bias behavior within a dataset rather than to rank overall generalization across datasets of different class counts.

4.2.7. Overall Ranking

Table 10 combines the four-core metrics into a single composite score (accuracy + precision + recall + F1 – average total error), to rank the five classifiers on one scale. SVM and kNN rank first and second overall; LSC ranks third, ahead of both Naive Bayes and the Decision Tree. This composite score should be read alongside the storage (Table 6) and noise-robustness (Table 9) results rather than in isolation, since it does not capture either advantage directly on those two axes, LSC is either the strongest or among the strongest of the five methods evaluated.

Table 10. Combining the different metrics

Rank	Classifier	Accuracy	Precision	Recall	F1	Avg Error	Composite Score
1	kNN (k=5)	0.8943	0.9014	0.8931	0.8940	2.1130	1.4698
2	SVM (linear)	0.9008	0.9039	0.9014	0.9016	2.1471	1.4606
3	LSC	0.8818	0.8872	0.8812	0.8816	2.1491	1.3827
4	Decision Tree	0.8306	0.8324	0.8267	0.8278	2.1338	1.1837
5	Naive Bayes	0.8380	0.8442	0.8373	0.8313	2.1714	1.1794

The horizontal bar chart of composite performance score by classifier, with LSC highlighted, ranked from highest to lowest is depicted in Figure 9.

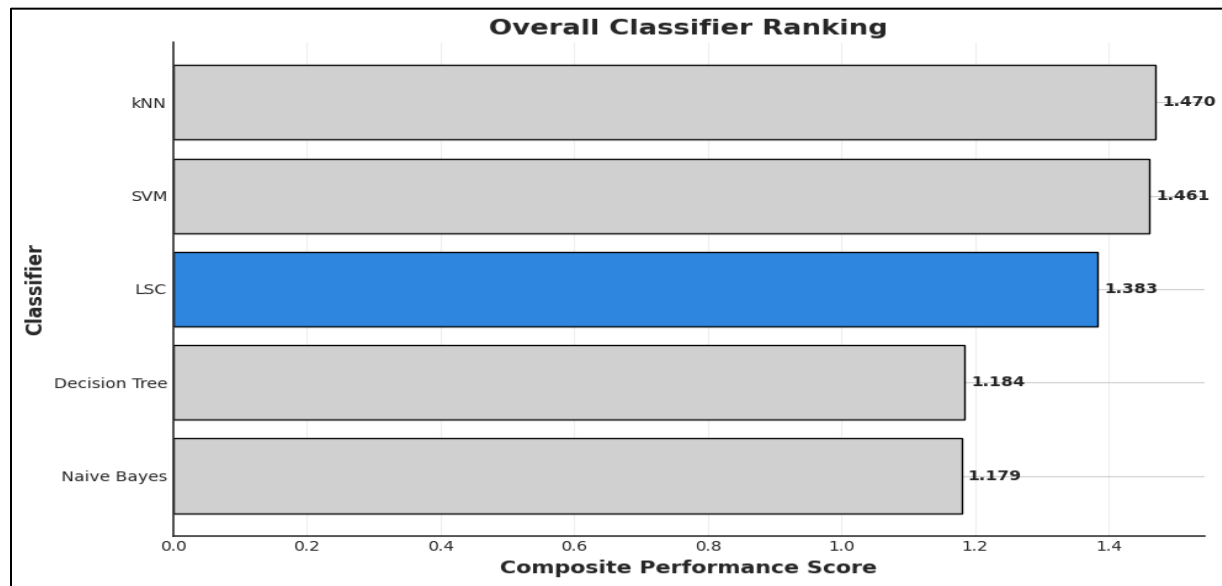


Figure 9. Overall composite performance ranking

4.2.8. Geometric Visualization

To inspect LSC's skeletons directly, each dataset's standardized features were projected into two dimensions via PCA, and the resulting class skeletons were plotted alongside the decision regions of the baseline classifiers on the same 2D projection are shown in the Figure 10 to Figure 16. This view is intended for qualitative inspection of trajectory fidelity for example, whether a class's skeleton follows an elongated cluster of points rather than as a substitute for the full-dimensional results reported above, since PCA-2D necessarily discards information that the classifiers use in the original feature space. Decision boundary comparison and confusion matrix of different datasets are visualized in Figure 10 to Figure 16.

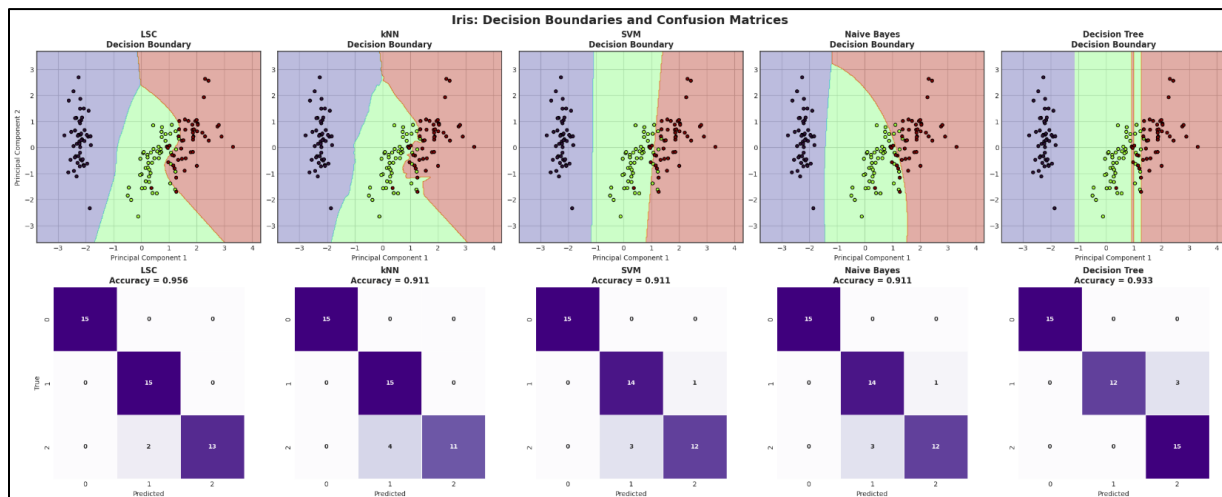


Figure 10. PCA-2D Decision Boundary Comparison and Confusion Matrix — Iris

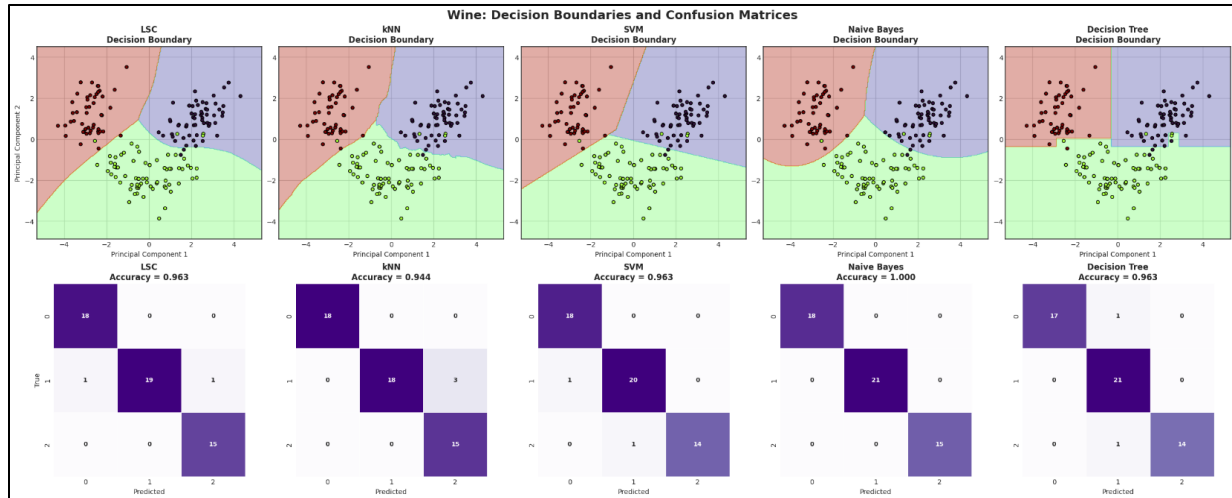


Figure 11. PCA-2D Decision Boundary Comparison and Confusion Matrix — Wine

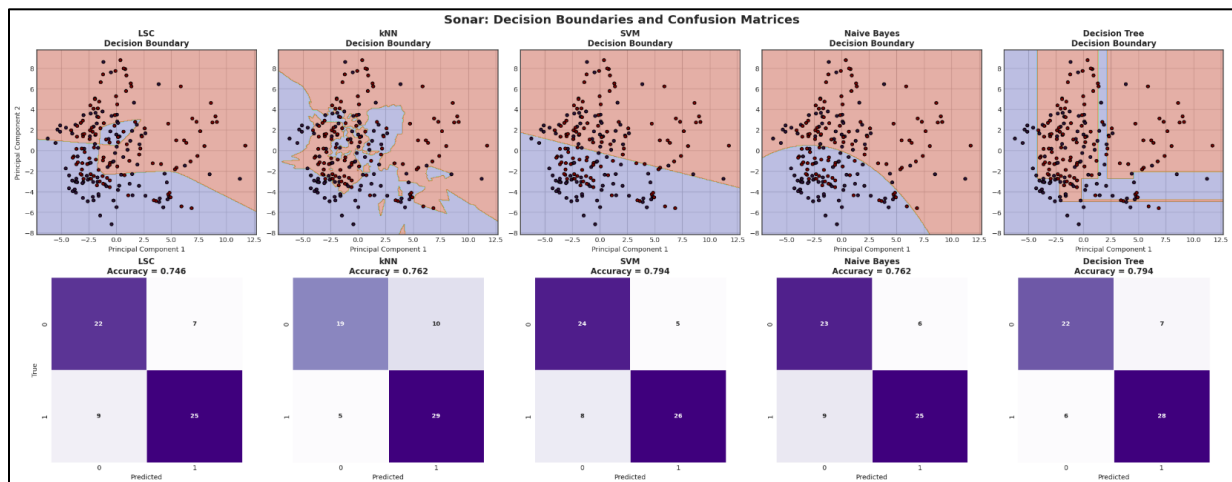


Figure 12. PCA-2D Decision Boundary Comparison and Confusion Matrix — Sonar

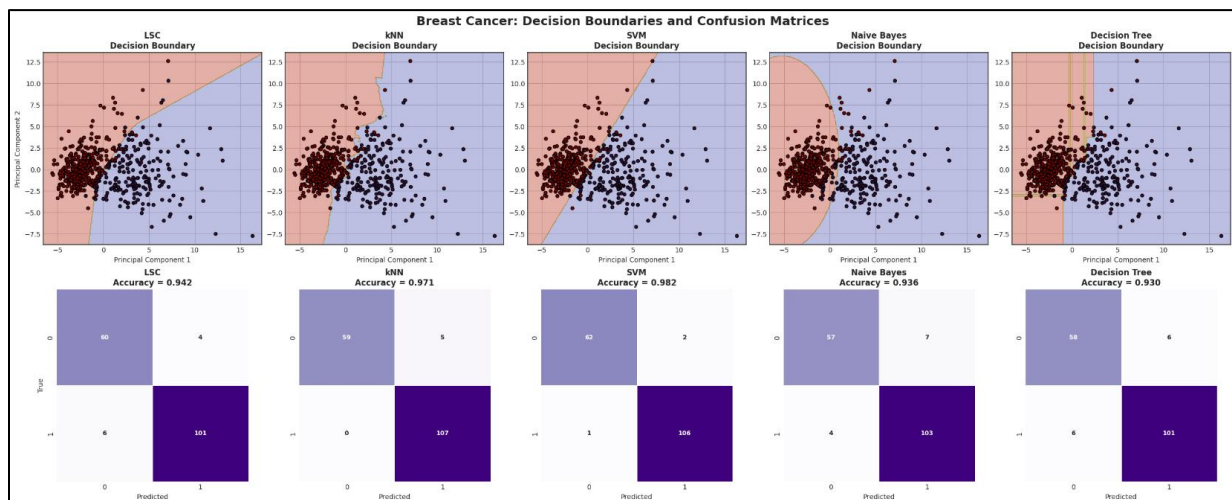


Figure 13. PCA-2D Decision Boundary Comparison and Confusion Matrix — Breast Cancer

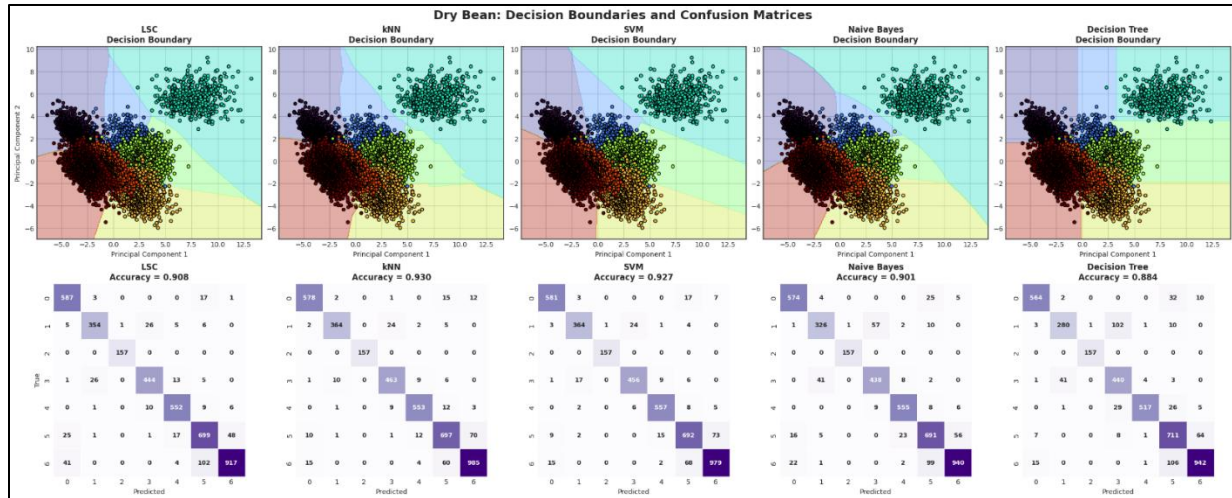


Figure 14. PCA-2D Decision Boundary Comparison and Confusion Matrix — Dry Beans

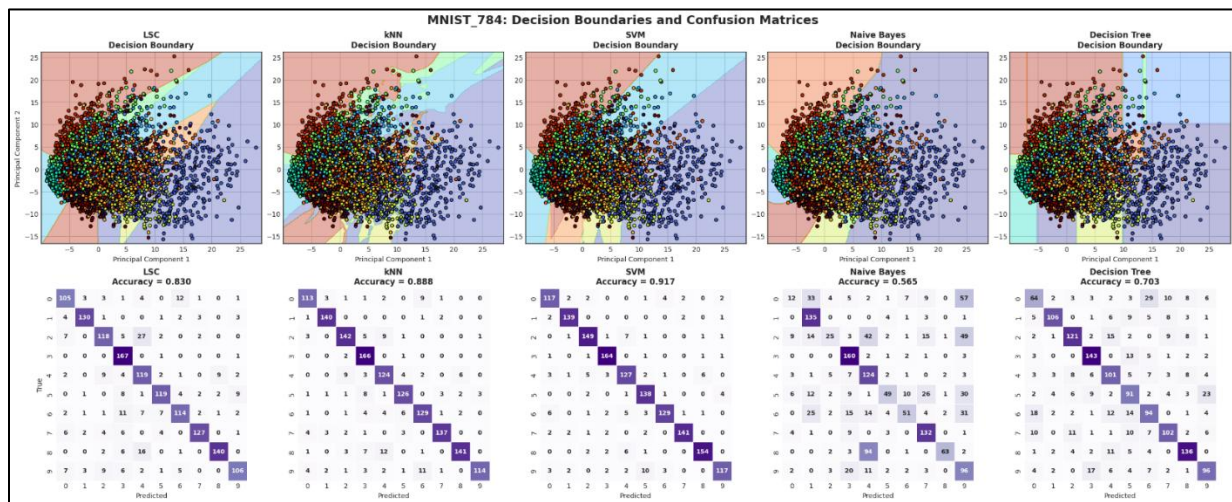


Figure 15. PCA-2D Decision Boundary Comparison and Confusion Matrix — MNIST_784

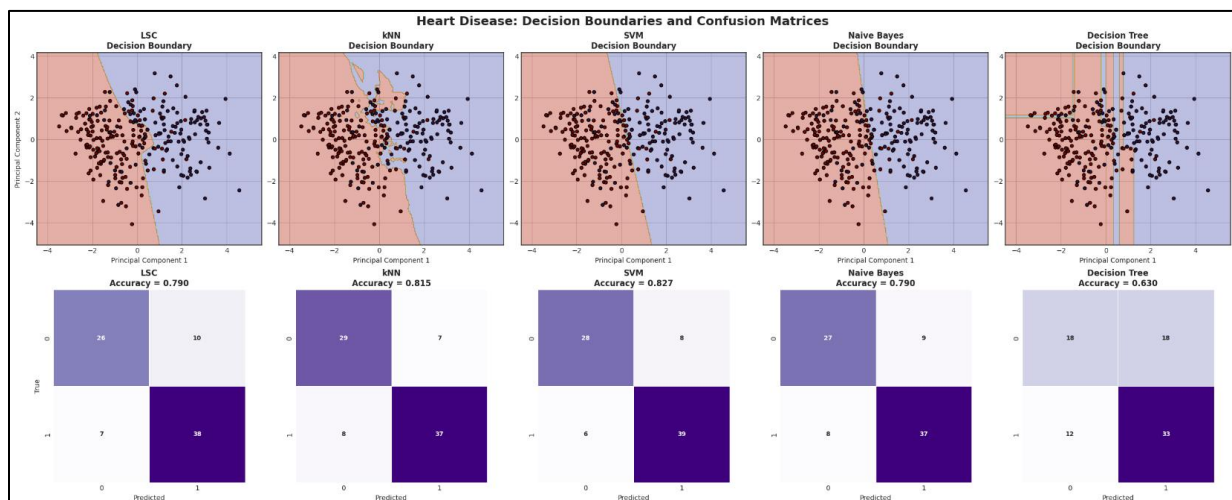


Figure 16. PCA-2D Decision Boundary Comparison and Confusion Matrix — Heart Disease

5. CONCLUSION

This paper presented the Linear Skeleton Classifier (LSC), a geometric manifold method for supervised learning that represents each class as a compact, ordered skeleton built through per-class centroid extraction, nearest-neighbor chaining, and segment-distance classification. Evaluated on seven UCI benchmark datasets against k-Nearest Neighbors, a linear-kernel SVM, Naive Bayes, and a depth-limited Decision Tree, LSC achieves the highest test accuracy on Iris (93.33%) and ties for the highest on Sonar (79.37%), remains within a few points of the strongest baseline on Wine (96.30%) and Breast Cancer (95.91%), and trails by a larger margin on Dry Bean (90.65%), MNIST_784 (82.73%), and Heart Disease (79.01%). Averaged across all seven benchmarks, LSC's composite performance ranks third of the five methods, a solid but not top result on accuracy alone. Where LSC is more clearly differentiated is in two properties that a single accuracy table does not show: it compresses the stored model by one to over two orders of magnitude relative to storing the full training set, and it is markedly more robust to injected feature noise than any of the four baselines, losing only about one percentage point of accuracy under the strongest noise condition tested, compared with losses of up to nearly ten points for the Decision Tree. LSC is therefore best understood not as a uniform replacement for margin-based or instance-based classifiers, but as a compact, noise-tolerant, and geometrically interpretable option for settings such as resource-constrained or noisy deployments where those properties matter alongside accuracy. Future work will focus on the Linear Skeleton Classifier with adaptively varying k, skeleton representations with multiple branches, and Riemannian distances over non-Euclidean feature spaces. Furthermore, an ensemble classification technique of LSC with mutually beneficial parametric classifiers would be examined to develop classifiers that are more robust and accurate.

6. REFERENCES

- [1] V. N. Vapnik, "Statistical learning theory," (*No Title*), 1998.
- [2] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273-297, 1995.
- [3] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21-27, 1967.
- [4] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?," in *International conference on database theory*, 1999, pp. 217-235: Springer.
- [5] U. Rashid, A. Abbas, M. M. Aqib, S. H. Shah, and U. Khattab, "Enhancing Random Forest Performance through Optimal Instance Subset Selection using Genetic Algorithm," vol. 4, no. 2, pp. 27-36, 2025.
- [6] J.-S. Chou, M.-Y. Cheng, Y.-W. Wu, and A.-D. Pham, "Optimizing parameters of support vector machine using fast messy genetic algorithm for dispute classification," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3955-3964, 2014.
- [7] B. Caputo, S. Bouattour, and H. Niemann, "Robust appearance-based object recognition using a fully connected Markov random field," in *2002 International Conference on Pattern Recognition*, 2002, vol. 3, pp. 565-568: IEEE.
- [8] C. J. Stone, J. Friedman, L. Breiman, and R. Olshen, "Classification and regression trees," *Wadsworth International Group*, vol. 8, pp. 452-456, 1984.
- [9] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5-32, 2001.
- [10] I. Rish, "An empirical study of the naive Bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 2001, vol. 3, no. 22, pp. 41-46: Seattle, USA.
- [11] H. Zhang, "The optimality of naive Bayes," *Aa*, vol. 1, no. 2, p. 3, 2004.
- [12] F. Cabitza, A. Locoro, and G. Banfi, "Machine learning in orthopedics: a literature review," *Frontiers in bioengineering and biotechnology*, vol. 6, p. 75, 2018.
- [13] O. Miguel-Hurtado, R. Guest, S. V. Stevenage, G. J. Neil, and S. Black, "Comparing machine learning classifiers and linear/logistic regression to explore the relationship between hand dimensions and demographic characteristics," *PloS one*, vol. 11, no. 11, p. e0165521, 2016.
- [14] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579-2605, 2008.
- [15] H. Blum, "A transformation for extracting new descriptions of shape," *Models for the perception of speech and visual form*, pp. 362-380, 1967.
- [16] T. Wang and Y. Yamakawa, "Edge-supervised linear object skeletonization for high-speed camera," *Sensors*, vol. 23, no. 12, p. 5721, 2023.
- [17] B. Ren, M. Liu, R. Ding, and H. Liu, "A survey on 3d skeleton-based action recognition using learning method," *Cyborg and Bionic Systems*, vol. 5, p. 0100, 2024.
- [18] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464-1480, 2002.

- [19] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten, "'Neural-gas' network for vector quantization and its application to time-series prediction," *IEEE transactions on neural networks*, vol. 4, no. 4, pp. 558-569, 1993.
- [20] U. Rashid *et al.*, "Anomaly detection using clustering (K-Means with DBSCAN) and SMO," vol. 7, no. 02, 2024.
- [21] J. MacQueen, "Multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, vol. 1, pp. 281-297.
- [22] Y. Fu, L. He, and Q. Chen, "ManifoldFormer: Geometric Deep Learning for Neural Dynamics on Riemannian Manifolds," *arXiv preprint arXiv:2511.16828*, 2025.
- [23] W. H. Press, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.
- [24] R. A. R. Ashfaq, Y.-l. He, and D.-g. Chen, "Toward an efficient fuzziness based instance selection methodology for intrusion detection system," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 6, pp. 1767-1776, 2017.
- [25] P. Eriksson, N.-A.-D. Marzouka, G. Sjö Dahl, C. Bernardo, F. Liedberg, and M. Höglund, "A comparison of rule-based and centroid single-sample multiclass predictors for transcriptomic classification," *Bioinformatics*, vol. 38, no. 4, pp. 1022-1029, 2022.
- [26] A. De Luca and S. Termini, "A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory," in *Readings in fuzzy sets for intelligent systems*: Elsevier, 1993, pp. 197-202.
- [27] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proceedings of the AAAI conference on artificial intelligence*, 2018, vol. 32, no. 1.
- [28] U. Rashid, M. Qadir, M. Alam, and S. Farid, "Hybrid Machine Learning Model to Enhance Cybersecurity: An Integration of KNN, RF and XGBoost," vol. 29, no. 4, 2024.
- [29] S. S. U. Din, M. Ali, U. Rashid, A. J. Akbar, and M. K. Abid, "Navigating an Optimization Path for Decision-Making in Management within The Big Data Environment of Chinese Public Hospitals," vol. 2, no. 2, pp. 61-77, 2023.
- [30] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319-2323, 2000.
- [31] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323-2326, 2000.
- [32] A. Asuncion, "Uci machine learning repository, university of california, irvine, school of information and computer sciences," <http://www.ics.uci.edu/~mlern/MLRepository.html>, 2007.