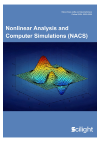
Scilight

Nonlinear Analysis and Computer Simulations
https://www.sciltp.com/journals/nacs



Article

From Correlations to Manifolds: A Geometric Approach to Classifications

İnan Ünal [1,*] and Özal Yıldırım [2]

[1] Department of Computer Engineering, Munzur University, Tunceli 62000, Türkiye
[2] Department of Software Engineering, Fırat University, Elazığ 23000, Türkiye
* Correspondence: inanunal@munzur.edu.tr

How To Cite: Ünal, İ.; Yıldırım, Ö. From Correlations to Manifolds: A Geometric Approach to Classifications. *Nonlinear Analysis and Computer Simulations* **2026**, *1*(1), 4.

Received: 9 November 2025
Revised: 17 December 2025
Accepted: 29 December 2025
Published: 13 January 2026

Abstract: This study addresses a machine learning application based on Riemannian geometry. The main objective is to demonstrate how classical learning algorithms, particularly Support Vector Machine (SVM), can be adapted for non-Euclidean data structures. The UCI Wine dataset is used as a real dataset. Instead of directly applying the algorithm to raw feature vectors, correlation matrices are obtained from these vectors through class-consistent windows, defining points on the SPD manifold. These matrices are then projected onto the tangent space around the Fréchet mean using the Riemannian logarithm map and provided as input to a Euclidean-form SVM classifier. Thus, the integration of manifold-based representation with classical methods is achieved, and its contribution to classification performance is examined.

Keywords: SPD manifolds; support vector machine; Riemann machine learning

2020 MSC: 53Z99; 68T05

1. Introduction

In the fields of machine learning and pattern recognition, Support Vector Machines (SVMs) are among the most widely used classification methods. The success of SVMs largely depends on the representation of the data and the distance metric used. In the traditional approach, data is represented as vectors in Euclidean space, and the separation between classes is achieved through hyperplanes. However, in some applications, particularly those involving biomedical signals, image recognition, and statistical correlation-based features, data is naturally defined on non-Euclidean (Riemannian) manifolds. In such cases, Symmetric Positive Definite (SPD) matrices or covariance-based features are considered on Riemannian manifolds, and Riemannian distances or log-maps for tangent space reduction are employed instead of classical Euclidean metrics.

In the machine learning literature, Support Vector Machines (SVMs) stand out as an effective method for classifying high-dimensional data [1,2]. However, much real-world data is naturally structured in non-Euclidean forms, particularly when represented by Symmetric Positive Definite (SPD) matrices. To process such data, Riemannian geometry-based approaches have been developed [3–6]. In recent years, Riemannian geometry-based methods have been successfully applied in brain-computer interface (BCI) and EEG signal classification, yielding superior results compared to classical methods [7,8]. Additionally, statistical analyses performed on SPD matrices have been reported to make significant contributions in biomedical fields such as diffusion tensor imaging (DTI) [9]. These studies demonstrate that methods aligned with the natural geometric structure of the data are both theoretically robust and practically successful.

Recently, Riemannian geometry-based methods have gained significant attention in machine learning tasks that involve structured data such as covariance matrices. Particularly in brain–computer interface (BCI) applications, symmetric positive definite (SPD) matrices defined on Riemannian manifolds have been shown to yield state-of-the-art classification performance by capturing the intrinsic geometry of the data [7,10]. Beyond BCI, manifold-aware representations have also been successfully employed in image classification through region covariance descriptors [11], as well as in visual tracking problems using geometry-preserving dimensionality reduction techniques [12]. Moreover, applications in signal processing further highlight the broad utility and robustness of Riemannian learning frameworks in handling non-Euclidean data distributions [13]. Collectively, these studies underline the potential of SPD manifold learning as a principled and effective alternative to classical Euclidean methods for structured data analysis.

In this study, the performance of the SVM classifier on the same dataset is compared using Euclidean representations and Riemannian-based non-Euclidean representations. Thus, the advantages offered by different geometric representations in terms of classification accuracy, computational cost, and interpretability are systematically examined. The paper is organized as follows: the second section provides the theoretical background of Euclidean and Riemannian data representations, the third section presents the methods used, the fourth section includes experimental results and comparisons, and the final section discusses general evaluations and future work.

2. Definition and Properties of SPD Manifold

In this section, the definition and properties of the SPD manifold will be presented, along with a discussion on how data is represented on SPD manifolds.

The set of symmetric positive definite matrices, denoted as $\mathcal{S}_p^+$, is a subset of symmetric matrices in $\mathbb{R}^{p\times p}$ and is defined by the following equivalent expressions:

$$\mathcal{S}_p^+ = \left\{C \in \mathbb{R}^{p\times p} \,\middle|\, C = C^\top,\ x^\top C x > 0\ \forall x \in \mathbb{R}^p\right\}$$

or

$$\mathcal{S}_p^+ = \left\{C \in \mathbb{R}^{p\times p} \,\middle|\, C = C^\top,\ \lambda_{\min}(C) > 0\right\}.$$

This structure is a Riemannian manifold in the differential geometric sense and is referred to as the SPD manifold [14,15]. Each point on this manifold, i.e., each $C \in \mathcal{S}_p^+$, can be considered along with its tangent space $\mathcal{T}_C\mathcal{S}_p^+$. The tangent space is isomorphic to the space of symmetric matrices, $\mathcal{S}_p$. One of the most commonly used metrics on the SPD manifold is the Affine-Invariant Riemannian Metric (AIRM). The distance between two SPD matrices $C_1, C_2 \in \mathcal{S}_p^n$ is defined as:

$$d_R(C_1, C_2) = \left\|\log\left(C_1^{-\frac{1}{2}} C_2 C_1^{-\frac{1}{2}}\right)\right\|_F,$$

where $\log(\cdot)$ denotes the matrix logarithm and $\|\cdot\|_F$ is the Frobenius norm. Additionally, the inner product defined by AIRM is given by:

$$\langle W_1, W_2\rangle_C = \mathrm{Tr}\left(C^{-1}W_1C^{-1}W_2\right).$$

From this, the projection of an SPD matrix Ω onto the tangent space at a reference point C (logarithm map) is defined as (Figure 1):

$$\log_C(\Omega) = C^{1/2}\log\left(C^{-1/2}\Omega C^{-1/2}\right)C^{1/2}.$$

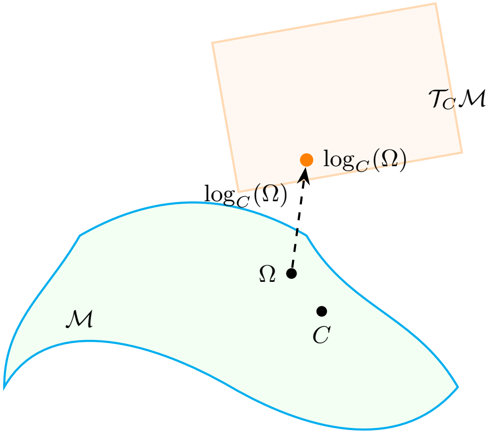


Figure 1. Projection of a point Ω on an SPD manifold onto the tangent space at reference point C using the logarithm map.

The natural Riemannian metric on the SPD manifold provides the foundation for geodesics and derivative calculations. In many applications, data is represented through covariance structures. For example, the covariance matrix for a data matrix $X \in \mathbb{R}^{d\times T}$ containing d features and T time samples is calculated as:

$$C = \frac{1}{T-1}(X-\mu)(X-\mu)^T, \quad \mu = \frac{1}{T}\sum_{i=1}^{T} x_i.$$

This matrix C is symmetric and positive definite, thus it lies on the $\mathcal{S}_p^d$ manifold. In this context, the covariance matrices of the data can be considered as points on the SPD manifold.

The classical arithmetic mean is not valid on the SPD manifold because the Riemannian distance is used instead of the Euclidean distance. Therefore, the mean of data points is defined as the point that minimizes the sum of distances on the manifold. This is called the Fréchet mean. For a set $\{C_i\}_{i=1}^N \subset \mathcal{S}_p^n$, the Fréchet mean is expressed as:

$$\bar{C} = \arg\min_{C\in\mathcal{S}_p^n} \sum_{i=1}^{N} d_R^2(C, C_i),$$

where $d_R(\cdot,\cdot)$ is the Riemannian distance (Figure 2). The Fréchet mean is generally not found in closed form but is computed using iterative methods. This point can be considered the most suitable center representing the data on the manifold and serves as a reference point in applications such as classification.

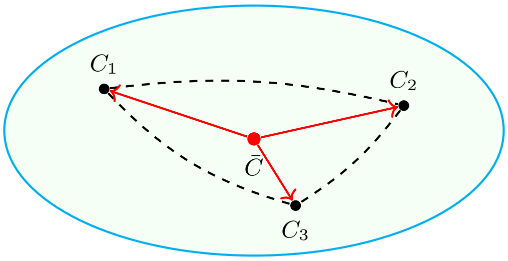


Figure 2. Fréchet mean $\bar{C}$ of points C_1, C_2, C_3 on the SPD manifold based on Riemannian distances.

Each SPD matrix C_i can be mapped to the tangent space around the Fréchet mean $\bar{C}$ using the logarithm map:

$$\log_{\bar{C}}(C_i) = \bar{C}^{\frac{1}{2}} \log\left(\bar{C}^{-\frac{1}{2}} C_i \bar{C}^{-\frac{1}{2}}\right) \bar{C}^{\frac{1}{2}}.$$

Thus, the data is reduced to Euclidean space and can be used with classical classifiers (e.g., SVM).

There are different classification methods developed based on the manifold's own geometry. One such method is the Minimum Distance to Mean (MDM) method. MDM is a non-parametric classifier that operates on symmetric positive definite (SPD) manifolds. The core idea of MDM is to determine a center for each class on the manifold and classify test samples based on their Riemannian distance to these centers. The Fréchet mean is computed for each class on the manifold. A test sample X is compared to the class means μ_c and assigned to the nearest center:

$$\hat{y} = \arg\min_c \; d_{\mathcal{M}}(X, \mu_c),$$

where $d_{\mathcal{M}}(\cdot,\cdot)$ denotes the Riemannian distance and μ_c is the Fréchet mean for class c [16].

Figure 3 represents the Fréchet means (μ_1, μ_2, μ_3) of three different classes on the SPD manifold and the position of a test point X to be classified. When comparing the geodesic distances between each mean and X, the shortest distance is to μ_2. Therefore, X is assigned to class 2.

This approach, unlike classical learning algorithms:

- Does not require model training,
- Operates directly on the manifold geometry,
- Can make more reliable decisions in high-dimensional and low-sample data scenarios.

The MDM method is a classification technique that respects the geometric structure and is intuitively easy to understand. It stands out as a robust and interpretable alternative, especially in fields such as signal processing, brain-computer interfaces, and statistical learning where symmetric positive definite (SPD) matrices are encountered [8,13].

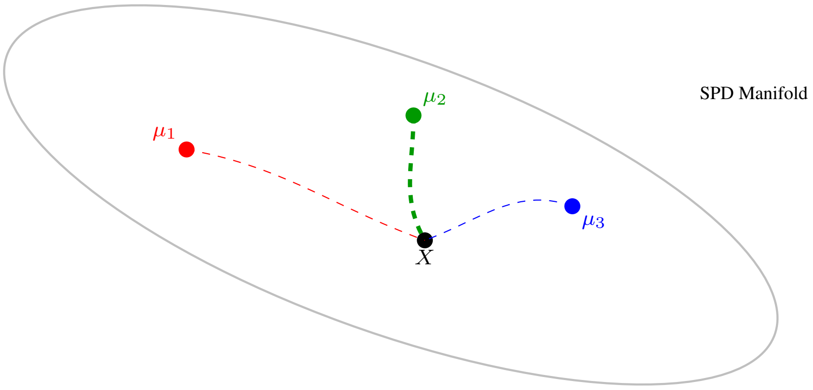


Figure 3. X is assigned to μ_2 by the shortest distance (green curve).

3. Materials and Methods

In this study, the dataset used is the UCI Wine dataset, which is widely preferred in the fields of machine learning and pattern recognition [17]. The dataset contains a total of 178 samples, each defined by 13 numerical features representing different chemical properties. The samples belong to three distinct classes (different types of wine). The aim of the study is to compare classification performances by representing the same data both in Euclidean space using classical methods and on a non-Euclidean manifold. Therefore, the data has been represented in two different forms: Euclidean vectors and correlation matrices on the SPD manifold.

In the Euclidean approach, each observation is treated as a vector in $\mathbb{R}^{13}$ space and directly fed into the SVM algorithm. At this stage, both linear and RBF kernel SVM models were applied to test the discriminative power between classes. However, in some data types, especially when statistical dependencies and covariance relationships are strong, this representation may be insufficient. Therefore, the same dataset was also represented on the SPD manifold using correlation matrices.

For the SPD representation, correlation matrices were first created from the scaled data while maintaining class consistency. Sliding windows were used for this purpose, with each window defined to contain 10 observations and consisting only of samples from the same class. A correlation matrix was calculated for each window, and a small positive constant ($10^{-3}I$) was added to each matrix to enhance numerical stability. Thus, the resulting matrices, due to their symmetric and positive definite properties, were considered as points on the $\mathcal{S}_p^n$ SPD manifold. This process resulted in a new dataset defined on the manifold, labeled, and preserving class information.

To better understand the structural properties of SPD matrices, one of the samples was visualized using a heatmap (Figure 4). The heatmap intuitively reveals the symmetric structure and positive definiteness of the matrix, while also allowing positive and negative correlations between features to be distinguished by color tones.

For the application of manifold-based learning methods, it is not sufficient to use SPD matrices directly. Therefore, the data was transformed into the tangent space using the Fréchet mean as a reference on the manifold.

In the final stage, both Euclidean vector representations and SPD manifold-based tangent space representations were used in the SVM classifier. The performances of both approaches were compared in terms of accuracy, computation time, and discriminative power between classes. For performance evaluation, the k-fold cross-validation method (mostly $k = 10$) was applied. This systematic comparison revealed the relationship between geometric data representation and classification performance in detail.

The high-dimensional vector representations projected onto the tangent space were reduced to two dimensions using Principal Component Analysis (PCA) and visualized (Figure 5). This visualization allowed samples separated by class labels to be shown in different colors, thus visually examining how well the manifold-based representation preserved class separation. The distribution of points obtained in the PCA plane provides an intuitive interpretation of the discriminative power between classes.

After projection, the obtained vector representations were divided into training and test sets, and classification was performed using an RBF kernel SVM model. The training/test ratio was generally set to 70/30. Model parameters were learned using the training data, and the accuracy of predictions made on the test data was calculated. This stage provides a critical evaluation to measure the contribution of SPD manifold-based representations to classification performance compared to classical Euclidean representations.

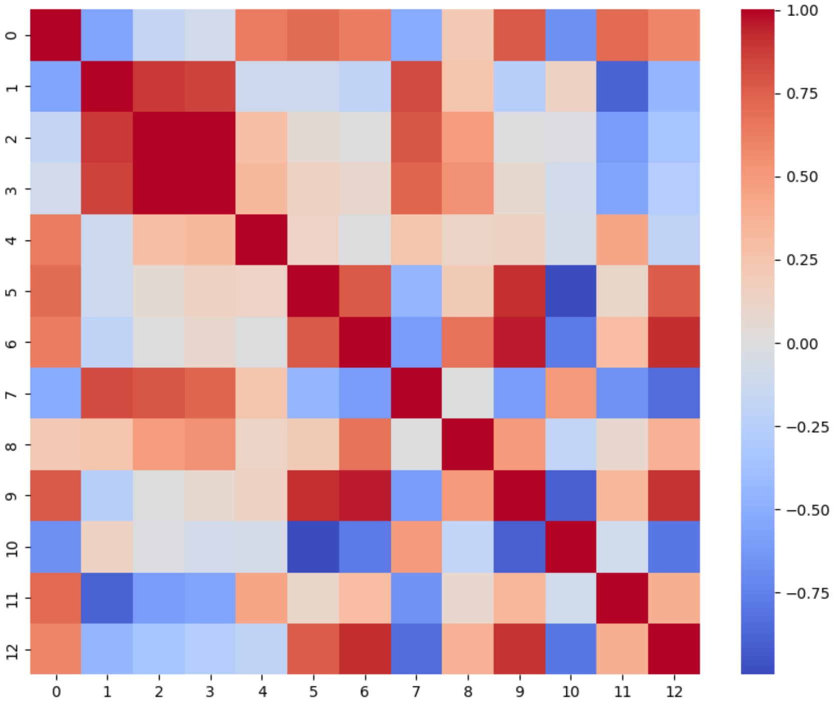


Figure 4. Heatmap visualization of the first SPD matrix.

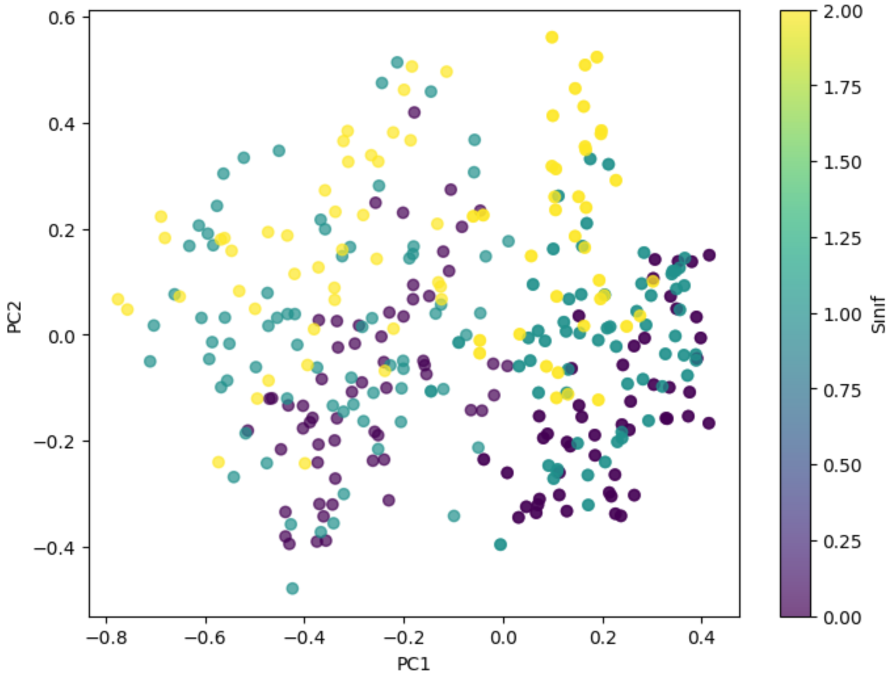


Figure 5. 2D projection in the tangent space using PCA. Colors represent classes.

Classification performance was analyzed not only by accuracy but also using the confusion matrix (Figure 6). The confusion matrix reveals the number of correctly and incorrectly classified samples for each class at the cellular level, clearly showing which classes the model performs well on and which classes it tends to misclassify. Thus, the overall performance of the SVM classifier, as well as its discriminative power on a per-class basis, was evaluated in detail.

In this study, in addition to Euclidean-based classifiers, the non-parametric Minimum Distance to Mean (MDM) method, which operates directly on the manifold, was also applied. The implementation of the method was carried out in the following steps:

(1) Calculation of Class Centers: For each class in the training data, the Fréchet mean of the relevant SPD matrices was calculated based on the Riemannian distance. This mean was used as the center of the class on the manifold.

(2) Test Phase: Each SPD matrix in the test data was compared to all class centers based on the Riemannian distance.

(3) Decision Rule: The test sample was assigned to the class with the shortest geodesic distance:

$$\hat{y} = \arg\min_{c} \ d_{\mathcal{M}}(X, \mu_c)$$

(4) Performance Evaluation: Accuracy, confusion matrix, and class-based metrics (precision, recall, f1-score) were calculated.

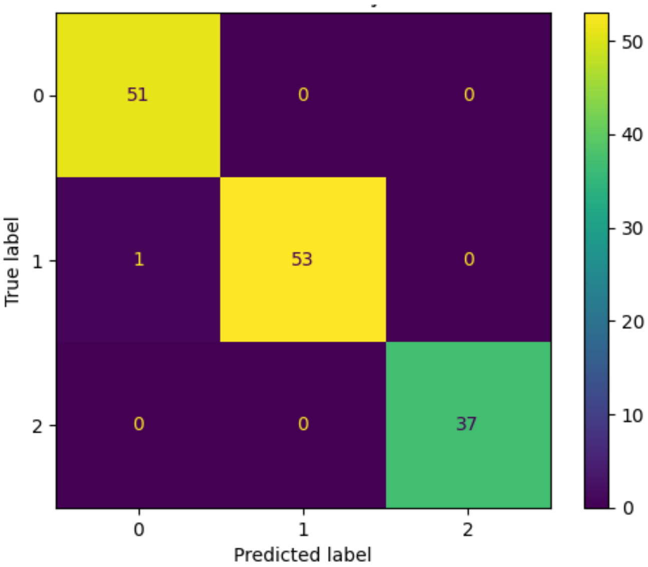


Figure 6. Confusion matrix of classification results.

In this approach, no model training was performed; only the calculation of class centers was sufficient. Thus, MDM offers a fast classification opportunity with low computational cost while being a method that directly respects the manifold structure.

4. Comparative Analysis of Classical and Riemannian Classifiers

In this section, three different classifier approaches were tested on the same dataset: Euclidean SVM, Riemannian SVM, and Riemannian MDM. The aim is to compare classical methods with manifold-based approaches in terms of both accuracy and interpretability.

Upon examining Table 1, it is observed that the highest accuracy of 99.30% is achieved by Riemannian SVM. Euclidean SVM demonstrated a highly successful performance with 98.15% accuracy on raw data, but considering the manifold geometry provided an additional contribution. Although the Riemannian MDM method yielded a slightly lower result of 96.62% compared to other methods, it offers a strong alternative with its simple and fast structure that does not require model training. This comparison shows that geometric representations play a significant role in improving classification success.

Table 1. Comparison of accuracy for classical (Euclidean) and Riemannian classifiers.

Method	Accuracy (%)
Euclidean SVM (Raw Data)	98.15
Riemannian SVM (Tangent Space)	99.30
Riemannian MDM (Manifold)	96.62

Euclidean SVM was directly applied to normalized raw data. The class-based metrics obtained are presented in Table 2, and the confusion matrix is shown in Figure 7.

As seen in Table 2, Euclidean SVM classification generally shows high success. Class 0 was correctly classified in all instances, while Class 1 had minor errors but maintained a recall value of 1.00. For Class 2, the recall value dropped to 0.93, indicating some samples were confused with other classes. When examining the average values, the precision, recall, and F1-score metrics are at 0.98, indicating that SVM performs very strongly with Euclidean representation.

Table 2. Euclidean SVM classification report.

Class	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	19
1	0.95	1.00	0.98	21
2	1.00	0.93	0.96	14
Average	0.98	0.98	0.98	54

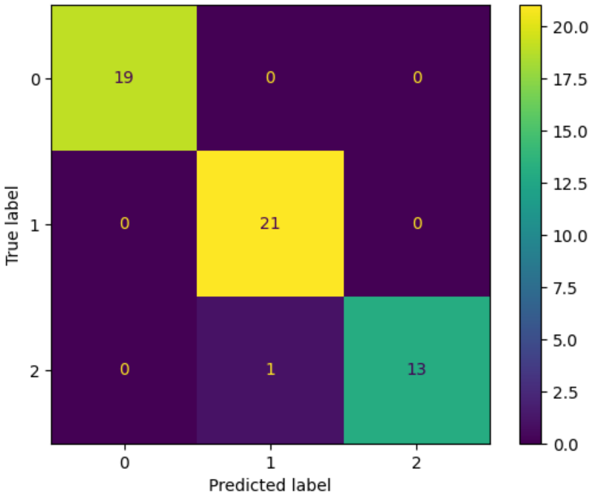


Figure 7. Confusion matrix for Euclidean SVM.

Riemannian SVM was applied after embedding the data into the SPD manifold and projecting it onto the tangent space using the logarithm map. The results are presented in Table 3 and Figure 8.

Table 3. Riemannian SVM classification report.

Class	Precision	Recall	F1-Score	Support
0	0.98	1.00	0.99	51
1	1.00	0.98	0.99	54
2	1.00	1.00	1.00	37
Average	0.99	0.99	0.99	142

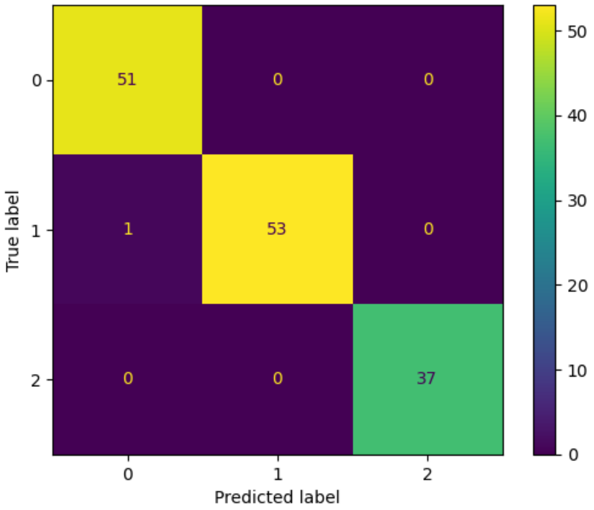


Figure 8. Confusion matrix for Riemannian SVM.

Upon examining Table 3, it is observed that Riemannian SVM achieves very high success across all classes. For Class 2, precision, recall, and F1-score values are 1.00, indicating error-free classification. Although there are minor differences in Class 0 and Class 1, the metrics are above 0.98. When looking at the average values, precision, recall, and F1-score are at 0.99, indicating that Riemannian representations significantly improve SVM performance.

Riemannian MDM is a method that operates directly on the manifold and does not require model training. Class-based metrics are presented in Table 4, and the confusion matrix is shown in Figure 9.

Table 4. Riemannian MDM classification report.

Class	Precision	Recall	F1-Score	Support
0	0.98	0.94	0.96	51
1	0.98	0.94	0.96	54
2	0.90	1.00	0.95	37
Average	0.96	0.96	0.96	142

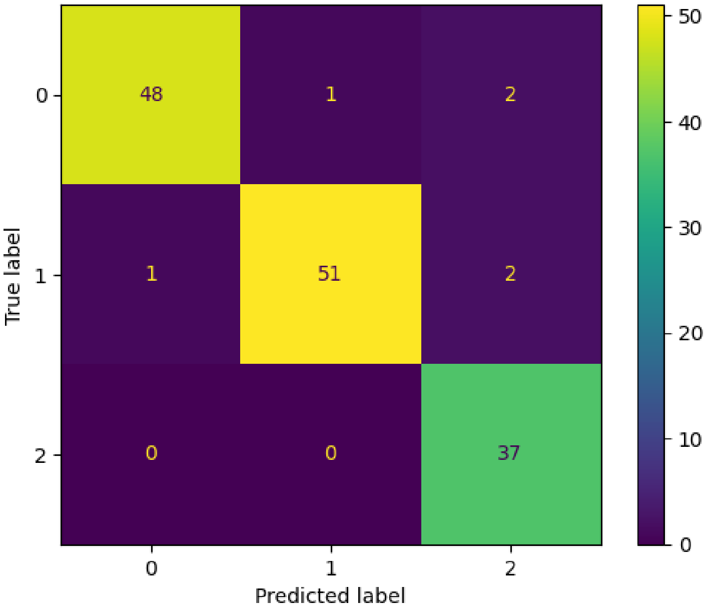


Figure 9. Confusion matrix for Riemannian MDM.

According to the results in Table 4, the Riemannian MDM classifier generally achieves high accuracy. For Class 0 and Class 1, precision values are 0.98, while recall values are 0.94, indicating some samples were misclassified. For Class 2, the recall value is 1.00, meaning all samples were correctly predicted, but the precision value of 0.90 indicates a tendency to confuse with other classes.

The average metrics (precision, recall, F1-score) are at 0.96. These results show that the MDM method, despite its simplicity and lack of training, offers a strong alternative (Table 5).

Table 5. Theoretical comparison table.

Feature	Euclidean SVM	Riemannian SVM	Riemannian MDM
Data Space	$\mathbb{R}^n$	Tangent Space	Manifold
Model Training	Required	Required	None
Speed	High	Medium	High
Interpretability	Medium	Medium	High

Table 6 presents a comparative overview of the overall experimental performance of the three methods. Riemannian SVM achieved the highest success with 99.30% accuracy and an F1-score of 0.99. Although Euclidean SVM achieved a very strong result with 98.15% accuracy on raw data, it lagged behind the Riemannian approach, which considers geometric structure. Riemannian MDM, despite its lower performance with 96.62% accuracy and an F1-score of 0.96, stands out with its untrained and fast structure.

Table 6. Experimental performance table.

Method	Accuracy (%)	F1-Score (Average)
Euclidean SVM	98.15	0.98
Riemannian SVM	99.30	0.99
Riemannian MDM	96.62	0.96

Table 7 shows the approximate computation times of the three methods. Euclidean SVM has the lowest training and testing times, making it the fastest method. Riemannian SVM incurs higher costs during the training phase due to the logarithm map and tangent space transformations, and requires a moderate amount of time during the testing phase. Riemannian MDM has no training time since there is no model training, but the test time is slightly longer than Euclidean SVM due to the calculation of the Fréchet mean. These results indicate that computational costs are also an important factor in method selection, alongside accuracy performance.

Table 7. Computation times (approximate).

Stage	Euclidean SVM	Riemannian SVM	Riemannian MDM
Training	∼10 ms	∼60 ms	None
Test	∼1 ms	∼5 ms	∼8 ms

These comparisons show that while Riemannian SVM provides the best performance in terms of accuracy, MDM stands out with its untrained and fast structure, and Euclidean SVM offers a practical solution with high performance even on raw data.

All Riemannian operations on the SPD manifold—including geodesic computations, logarithmic and exponential maps, and Fréchet mean estimation—were implemented using the geomstats Python library [18]. This library provides a modular and extensible framework for differential geometric computations and statistical learning on manifolds.

In particular, the affine-invariant metric on the space of $d \times d$ symmetric positive definite (SPD) matrices was used throughout the experiments. Geodesic distances and mappings were computed using the built-in SPD manifold class provided in geomstats.geometry.spd_matrices. All experiments were performed using version 2.3 of the library in a Python 3.10 environment.

5. Results and Discussion

In this study, Euclidean SVM, Riemannian SVM, and Riemannian MDM methods were applied to the UCI Wine dataset, and their performances were compared. The findings reveal that embedding data into the SPD manifold in a manner suitable to its geometric structure enhances classification success.

As a result of the experimental analyses:

- Although Euclidean SVM achieves high accuracy (98.1%), it disregards geometric structure.
- Riemannian SVM, with tangent space transformation, achieved the highest success with 99.3% accuracy.
- Riemannian MDM, despite being an untrained and fast method, offered a strong alternative with 96.6% accuracy.

The confusion matrices and class-based metrics show that Riemannian SVM best preserves inter-class discrimination. In terms of computation times, Euclidean SVM is the fastest, MDM is practical with its untrained structure, and Riemannian SVM, although more costly, is the method that provides the best performance.

In conclusion, this study demonstrates that manifold-based classification methods provide more accurate and meaningful results compared to classical methods, especially for covariance or correlation-based data. In the future, testing these approaches on different datasets, improving computational efficiency, and integrating them with deep learning architectures stand out as potential research areas.

Author Contributions

İ.Ü.: conceptualization, methodology, software, data curation, writing—original draft preparation, visualization, investigation; Ö.Y.: supervision, software, validation; writing—reviewing and editing. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The dataset used in this study is publicly available from the UCI Machine Learning Repository. Specifically, the Wine dataset can be accessed at the following link: https://archive.ics.uci.edu/ml/datasets/wine (accessed on 7 November 2025). No new data were generated during the study.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable feedback and constructive suggestions, which helped improve the quality of the manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

During the preparation of this work, the authors used the LLMs (ChatGPT, Gemini etc) for English proofreading and code assistance. The authors take full responsibility for the content and accuracy of the final manuscript.

References

1. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297.
2. Burges, C.J.C. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.* **1998**, *2*, 121–167.
3. Pennec, X.; Fillard, P.; Ayache, N. A Riemannian framework for tensor computing. *Int. J. Comput. Vis.* **2006**, *66*, 41–66.
4. Arsigny, V.; Fillard, P.; Pennec, X.; et al. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM J. Matrix Anal. Appl.* **2007**, *29*, 328–347.
5. Bhatia, R. *Positive Definite Matrices*; Princeton University Press: Princeton, NJ, USA, 2009.
6. Amari, S.; Nagaoka, H. *Methods of Information Geometry*; AMS and Oxford University Press: Oxford, UK, 2007.
7. Congedo, M.; Barachant, A.; Bhatia, R. Riemannian geometry for EEG-based brain–computer interfaces; a primer and a review. *Brain-Comput. Interfaces* **2017**, *4*, 155–174.
8. Barachant, A.; Bonnet, S.; Congedo, M.; et al. Multiclass brain-computer interface classification by Riemannian geometry. *IEEE Trans. Biomed. Eng.* **2012**, *59*, 920–928.
9. Dryden, I.L.; Koloydenko, A.; Zhou, D. Non-Euclidean statistics for covariance matrices, with applications to diffusion tensor imaging. *Ann. Appl. Stat.* **2009**, *3*, 1102–1123.
10. Barachant, A.; Bonnet, S.; Congedo, M.; et al. Classification of covariance matrices using a Riemannian-based kernel for BCI applications. *Neurocomputing* **2013**, *112*, 172–178.
11. Tuzel, O.; Porikli, F.; Meer, P. Region Covariance: A Fast Descriptor for Detection and Classification. In Proceedings of the European Conference on Computer Vision (ECCV), Graz, Austria, 7–13 May 2006; pp. 589–600.
12. Harandi, M.T.; Salzmann, M.; Hartley, R. From Manifold to Manifold: Geometry-Aware Dimensionality Reduction for SPD Matrices. In Proceedings of the European Conference on Computer Vision (ECCV), Florence, Italy, 7–13 October 2012; pp. 17–32.
13. Yger, F.; Berar, M.; Lotte, F. Riemannian Approaches in Brain-Computer Interfaces: A Review. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2017**, 25, 1753–1762.
14. Moakher, M.; Batchelor, P.G. Symmetric Positive-Definite Matrices: From Geometry to Applications and Visualization. In *Visualization and Processing of Tensor Fields*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 285–298,
15. Schiavon, J. Differential Geometry of Symmetric and Positive Definite Matrices for Statistical Applications. Ph.D. Thesis, Università degli Studi di Padova, Padua, Italy, 2022.
16. Congedo, M.; Rodrigues, P.L.C.; Jutten, C. The Riemannian Minimum Distance to Means Field Classifier. In Proceedings of the BCI 2019—8th International Brain-Computer Interface Conference, Graz, Austria, 16–20 September 2019.

17. Dua, D.; Graff, C. *UCI Machine Learning Repository*; University of California, Irvine, School of Information and Computer Sciences: Irvine, CA, USA, 2019.
18. Miolane, N.; Guigui, N.; Le Brigant, A.; et al. Geomstats: A Python package for Riemannian geometry in machine learning. *J. Mach. Learn. Res.* **2020**, *21*, 1–9.