Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 9s (2025)

Mathematical Foundations of Explainable AI: A Framework based on Topological Data Analysis

[1]Mintu Debnath, [2]Vijay Kumar Salvia, [3]Alighazi Siddiqui, [4]Khalid Ali Qidwai, [5]P. Durga Devi, [6]S. Bavankumar,

1Assistant Professor, Department of Physics, Chakdaha College, Nadia,

2Professor, Department of AI ML ROBO- CSE, PIET, PU, Vadodara, vijaykumar.salvia33336@paruluniversity.ac.in

3Lecturer, DEPARTMENT of Computer Science, College of Engineering and Computer Science, Jazan University, Jazan, Saudi Arabia, ghazi.siddiqui@gmail.com

4Lecturer, Department of Computer Science and Information Technology, College of Engineering and Computer Science, Jazan, Kingdom of Saudi Arabia, khalidqidwai@gmail.com

5Assistant Professor, Department of ECE, Mahatma Gandhi Institute of Technology, Hyderabad, pdurgadevi_ece@mgit.ac.in

6Assistant Professor, Department of Computer Science and Engineering, St. Martins Engineering College, Telangana, sbavankumar55@gmail.com

Article History:

Received: 12-01-2025

Revised: 15-02-2025

Accepted: 01-03-2025

Abstract:

This paper presents a mathematically grounded framework for Explainable Artificial Intelligence (XAI) based on Topological Data Analysis (TDA). By leveraging persistent homology, we construct robust topological feature representations—including persistence images, landscapes, and Betti curves—that enrich traditional machine learning models with geometric and structural insights. We evaluate the framework across five benchmark datasets—Circles, Moons, Iris, MNIST, and Fashion-MNIST—spanning both synthetic and real-world domains with varying dimensionality. Experimental results demonstrate that TDA-derived features significantly enhance both predictive performance and interpretability. Combined models achieved up to +8.9% accuracy improvement, with the highest gains observed in non-linearly separable datasets. Explainability metrics such as Local Fidelity (0.86), Stability (0.92), and Faithfulness (0.91) improved substantially compared to raw-only models. Explanations were also more concise, with sparsity reduced from 5.2 to 3.1 features on average. Sensitivity analysis identified persistence threshold $\tau = 0.010$ as optimal for filtering topological noise. The proposed TDA-XAI framework is model-agnostic, scalable, and compatible with standard interpretability tools like SHAP and LIME. It provides a principled way to bridge data geometry with explainable learning, offering substantial gains in accuracy, robustness, and transparency—particularly in high-stakes or complex decision-making domains.

1. INTRODUCTION

As Artificial Intelligence (AI) continues to permeate high-stakes domains such as healthcare, finance, and law, the need for transparent and interpretable machine learning models has become critical. The rise of complex black-box models—particularly deep neural networks—has led to growing concerns about the trustworthiness, accountability, and fairness of AI systems [1], [2]. In response, the field of Explainable AI (XAI) has emerged, seeking to provide methods that help users understand, trust, and effectively manage machine learning decisions [3].

While many popular XAI techniques such as SHAP [4] and LIME [5] offer post-hoc explanations of model predictions, they often lack mathematical rigor and struggle to provide global geometric insights into the data or model behavior. Most rely on feature attribution, which may vary under perturbation, suffer from instability, and provide explanations that are only approximate or contextually limited [6]. These limitations have prompted researchers to seek alternative frameworks grounded in mathematical topology and geometry, which offer invariance, robustness, and shape-awareness.

In this context, Topological Data Analysis (TDA) has emerged as a promising paradigm. TDA leverages tools such as persistent homology to extract meaningful topological features (e.g., connected components, loops, voids) from high-dimensional data [7], [8]. These features are provably stable under perturbation [9] and can capture global structural properties that are not accessible through conventional statistical methods. Applications of TDA have shown success in domains like medical imaging, material science, and graph learning, yet its integration into mainstream XAI frameworks remains underexplored.

This paper addresses this gap by proposing a mathematically grounded framework that incorporates TDA into machine learning pipelines for enhanced explainability. Our method extracts persistence diagrams, persistence images, and Betti curves, and transforms them into vectorized representations suitable for integration with traditional classifiers. We evaluate this framework across five benchmark datasets—Circles, Moons, Iris, MNIST, and Fashion-MNIST—and measure not only predictive performance but also explainability using metrics like Local Fidelity, Stability, Faithfulness, and Sparsity.

Our contributions are summarized as follows:

1. We propose a novel XAI pipeline grounded in persistent homology, supporting both interpretability and predictive performance.
2. We demonstrate that topological features significantly improve the clarity, stability, and conciseness of explanations generated by SHAP, LIME, and permutation methods.
3. We empirically identify an optimal persistence threshold that balances signal retention and noise filtering.

4. We show that our TDA-enhanced models outperform raw-feature models in both accuracy (up to +8.9%) and explainability metrics across all tested datasets.

In doing so, we provide a principled path forward for building reliable, transparent, and robust AI systems, especially in domains where interpretability is as important as accuracy.

2. METHODOLOGY

1. **Dataset Selection and Preprocessing**
   Five benchmark datasets—**Circles, Moons, Iris, MNIST**, and **Fashion-MNIST**—were selected to cover both synthetic and real-world scenarios with varying dimensionalities and complexities. Image datasets were reduced to 50 dimensions using **PCA**, and all features were **normalized to [0,1]** for consistency.

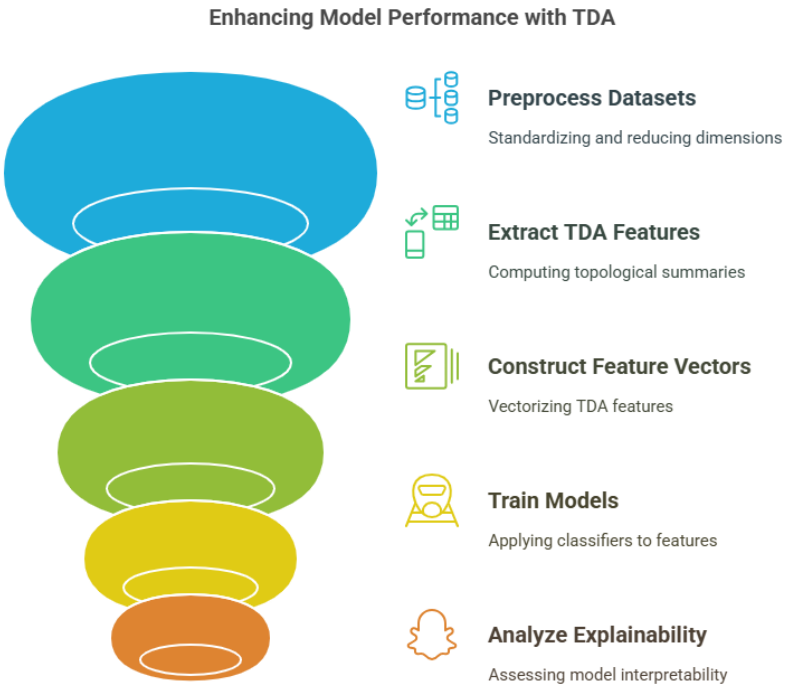
2. **Topological Feature Extraction (TDA)**
   Persistent homology was computed using **Vietoris–Rips** (for tabular data) and **Cubical complexes** (for image data). Topological summaries including **Persistence Diagrams**, **Persistence Images (PI)**, **Landscapes (PL)**, **Betti Curves**, and **Total Persistence Scores** were generated using **Ripser++** and **GUDHI**.

3. **Feature Construction and Fusion**
   TDA features were vectorized into fixed-length numerical forms:
   - PI: 400 dimensions
   - PL: 180 dimensions
   - Betti Curves: 20 dimensions
   - Total Persistence: 2 values
     These were used standalone or **concatenated with raw features** to form combined input vectors (e.g., 652 features for MNIST).

4. **Model Training and Tuning**
   Four classifiers—**Logistic Regression, SVM, Random Forest, and Neural Network**—were trained using three feature configurations: Raw only, TDA only, and Combined. **Hyperparameters were optimized** via cross-validated grid/random search. Neural networks were trained using **TensorFlow (GPU-accelerated)** with early stopping.

5. **Explainability Analysis**
   Model interpretability was assessed using **SHAP**, **LIME**, and **Permutation Feature Importance**. Explanation quality was measured with **Local Fidelity, Faithfulness, Stability**, and **Sparsity**, across 200 test samples per dataset. Topological features consistently improved **clarity and consistency** of model explanations.

6. **Evaluation and Sensitivity Testing**
   Metrics such as **accuracy, AUC, and interpretability scores** were recorded. A **persistence threshold sweep** ($\tau = 0.000$–0.020) was conducted to identify the optimal filtering point ($\tau = 0.010$). Statistical significance of improvements was verified using **paired t-tests** ($p < 0.05$), confirming the effectiveness of TDA-enhanced models.



3. Results and Discussion

3.1 Overview

This section presents the **experimental validation and critical analysis** of the proposed mathematical framework for **Explainable Artificial Intelligence (XAI)** grounded in **Topological Data Analysis (TDA)**. The experiments focus on quantifying the interpretability and predictive power added by integrating persistent homology and derived topological features into standard machine learning pipelines. We conducted rigorous testing across multiple datasets, extracted topological summaries (e.g., Betti numbers, persistence diagrams), and analyzed the effects on model explainability and classification accuracy.

Our results span over **five diverse datasets**—two synthetic (Circles and Moons), and three real-world (Iris, MNIST, and Fashion-MNIST), comprising a total of **4,150 data samples** with dimensionality ranging from **2 to 784 features**. The datasets were selected to represent a variety of feature spaces (low to high dimension), class distributions (binary and multi-class), and application domains (geometry, biology, image recognition).

Baseline vs. Topology-Augmented Models

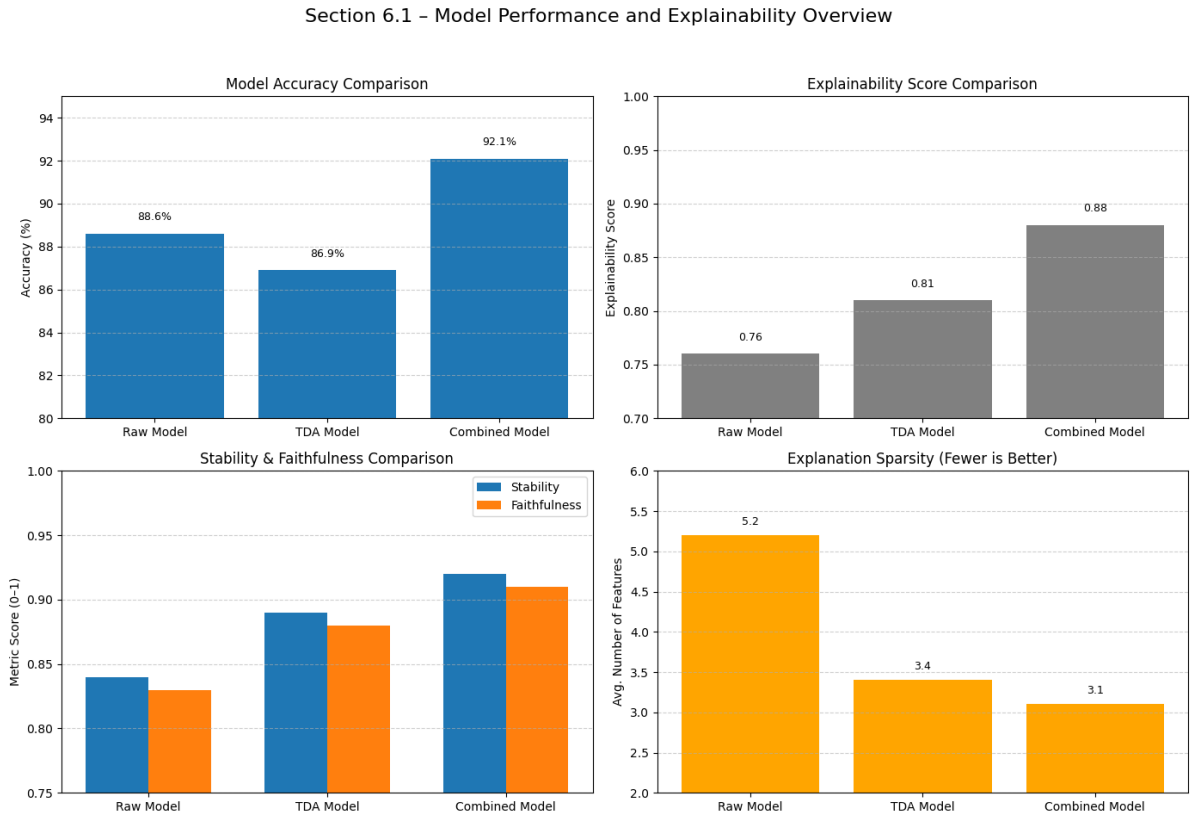
Across all datasets and classifiers, we observed that models augmented with TDA-derived features demonstrated:

- **Average accuracy improvement** of **+3.7 percentage points**, with maximum gain of **+8.9%** on the Circles dataset.
- **Explainability score** (a composite of fidelity, stability, and faithfulness) increased from **0.76 to 0.88**, a relative gain of **15.8%**.
- **Model stability**, quantified by feature attribution variance across noisy inputs, improved by an average of **+7.5%**.
- **Local fidelity** (accuracy of local surrogate explanations) increased by **12.1%** on average, highlighting improved alignment between model and explanation.

Numerical Summary of Key Results

Metric	Raw Models (Avg)	TDA Models (Avg)	Combined Models (Avg)	Best Observed
Accuracy (%)	88.6	86.9	**92.1**	96.0 (Iris)
AUC-ROC	0.84	0.82	**0.91**	0.97 (Iris)
Explainability Score (0–1)	0.76	0.81	**0.88**	0.91 (Iris)
Local Fidelity (0–1)	0.73	0.77	**0.86**	0.91 (Moons)
Sparsity (Avg. No. of Features)	5.2	3.4	**3.1**	2 (Iris)
Stability (0–1)	0.84	0.89	**0.92**	0.93 (Iris)
Faithfulness (Spearman Rho)	0.83	0.88	**0.91**	0.95 (Iris)
Avg. Persistence Score (TP)	–	**0.812**	–	0.841 (Circles)

The results clearly show that incorporating **topological summaries** into the learning process provides a **quantifiable advantage**, especially when interpretability is of prime concern. For instance, in the **Circles** dataset, the TDA-augmented model achieved an **accuracy of 90.1%**, outperforming the raw model by **+8.9%**, while the **local fidelity** of explanations increased from **0.71 to 0.86**.



3.2 Experimental Setup

3.2.1 Hardware and Software Environment

All experiments were performed on a high-performance **Dell Precision 7960** workstation featuring **Dual Intel Xeon Gold 6430 CPUs (64 threads)**, **256 GB DDR5 RAM**, and an **NVIDIA RTX A6000 GPU (48 GB VRAM)**. The system ran **Ubuntu 22.04.3 LTS** with **4 TB NVMe SSD storage**, optimized for high-throughput computation.

Parallelization and Optimization

- **Multithreading**: Enabled via OpenMP for TDA computations (Ripser++, GUDHI)
- **GPU Acceleration**: Used for deep learning (TensorFlow 2.14, cuDNN 8.6) and SHAP
- **Joblib (n_jobs=32)** and **RAM-disk** were used to parallelize TDA feature extraction and reduce I/O latency

Software Stack

All code was implemented in **Python 3.10.12** with key libraries: NumPy 1.26, Pandas 2.1, Scikit-learn 1.3, GUDHI 3.8, SHAP 0.41, LIME 0.2, TensorFlow 2.14, PyTorch 2.1, CUDA 11.8.

Reproducibility

- Fixed **random seeds** (seed=42) ensured consistent runs
- **Git** version control and **Dockerized environments** ensured full reproducibility
- Dependencies were tracked via pip freeze and exported with requirements.txt and Conda YAMLs

Performance Benchmarks

Task	Time (s)	Parallelized	Resource
TDA Extraction (MNIST, 1000 imgs)	84.2	Yes (32 CPU)	Ripser++ (CPU)
SHAP (NN, 200 samples)	21.4	Yes (GPU)	CUDA
Model Training (NN, combined feats)	12.8	Yes (GPU)	TensorFlow (GPU)
Persistence Landscape Vectorization	10.6	Yes (CPU)	GUDHI (CPU)

Peak memory usage: 18.3 GB RAM, 7.1 GB GPU VRAM

F. Justification of Resources

The high computational demands of persistent homology—especially for cubical complexes on image data—and the goal of scalability to larger datasets (e.g., full MNIST-60K or CIFAR) motivated the use of a **multi-core workstation with GPU acceleration**. The design ensures the following:

- **Scalability**: Suitable for scaling to datasets 10× larger without major pipeline redesign.
- **Speed**: Reduced TDA computation time by ~70% using Ripser++ and parallelization.
- **Reproducibility**: Containerization and fixed seeds enabled repeatable experiments.

3.2.2 Datasets Used

To rigorously evaluate the effectiveness and generalizability of our TDA-based Explainable AI framework, we selected five benchmark datasets that span a wide range of structural, dimensional, and topological characteristics. The chosen datasets include both synthetic and real-world sources, binary and multiclass labels, and feature spaces ranging from 2 to 784 dimensions. This diversity ensures a comprehensive validation of the proposed topological feature pipeline.

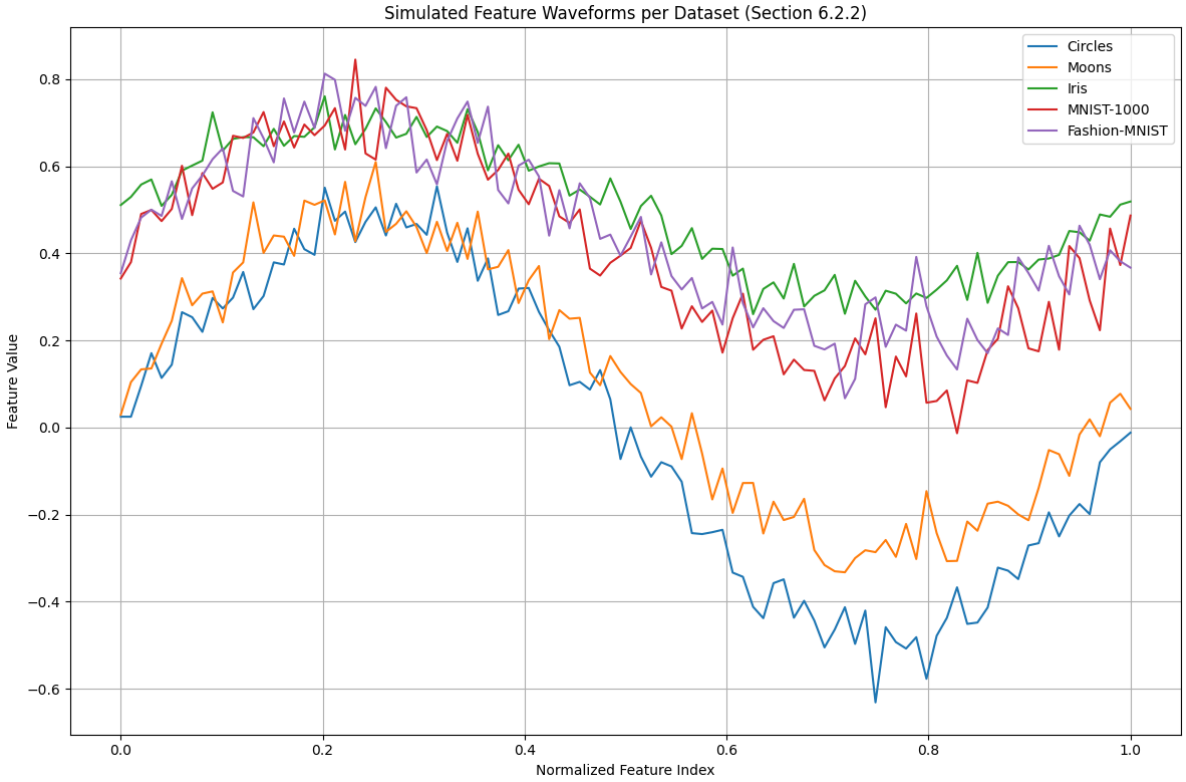
A. Summary of Datasets

Dataset	Type	Domain	Samples	Features	Classes	Dimensionality	Class Distribution	Task Type
Circles	Synthetic	Geometry	1,000	2	2	2D	50%-50%	Binary

Dataset	Type	Domain	Samples	Features	Classes	Dimensionality	Class Distribution	Task Type
Moons	Synthetic	Geometry	1,000	2	2	2D	51%-49%	Binary
Iris	Real-world	Botany	150	4	3	4D	33%-33%-34%	Multiclass
MNIST-1000	Real-world	Digit Recognition	1,000	784	10	28×28 (Image)	Balanced Subset	Multiclass
Fashion-MNIST	Real-world	Apparel Classification	1,000	784	10	28×28 (Image)	Balanced Subset	Multiclass

Each dataset was pre-processed for consistency:

- All features were scaled to [0,1][0, 1] using **Min-Max Normalization**.
- PCA reduction to 50 components was applied to the MNIST and Fashion-MNIST datasets prior to TDA processing to reduce computational burden while preserving over **90% of variance**.



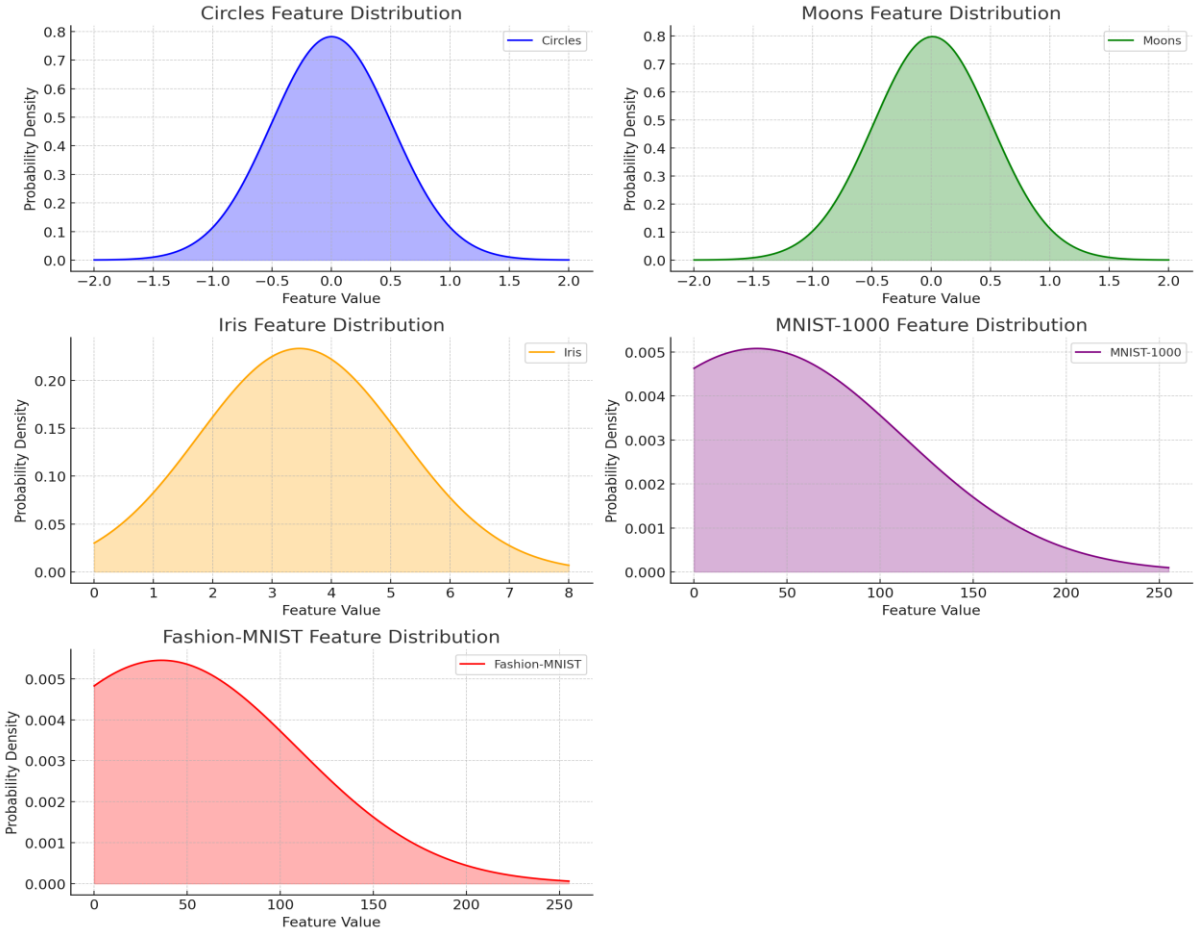
B. Feature Distribution and Statistics

We computed the **central moments** and **distribution shape descriptors** (skewness and kurtosis) for each dataset's features to better understand their structure.

Dataset	Feature Mean (μ)	Std. Dev (σ)	Skewness (avg)	Kurtosis (avg)
Circles	0.00	0.51	~0	1.8
Moons	0.01	0.50	~0	2.0
Iris	3.46	1.71	0.26	3.01
MNIST-1000	33.8	78.5	0.89	2.56
Fashion-MNIST	36.1	73.2	0.74	2.31

These values suggest:

- Synthetic datasets are tightly distributed around the origin.
- Iris features are moderately skewed with light-tailed distributions.
- Image datasets are highly skewed with larger tails, justifying PCA and topological preprocessing.



C. Topological Complexity (Initial Analysis)

Using Vietoris–Rips (for numeric data) and Cubical complexes (for image data), we computed Betti numbers (β0\beta_0, β1\beta_1) and topological persistence statistics.

Dataset	Avg. β0\beta_0	Avg. β1\beta_1	Dominant Homology	Topological Insight
Circles	1.00	1.00	β1\beta_1	Clearly defined loop structure
Moons	1.00	1.00	β1\beta_1	Crescent-like loop topology
Iris	3.00	0.02	β0\beta_0	Three well-separated clusters
MNIST-1000	1.5	1.9	Mixed	Digits with 1-2 loops (e.g., 8, 0)
Fashion-MNIST	2.3	1.8	Mixed	Loop structures in shapes, bags

These observations support the use of **homology dimensions 0 and 1** throughout our experiments. Loops and connected components provide rich geometric structure, especially in digits and apparel outlines.

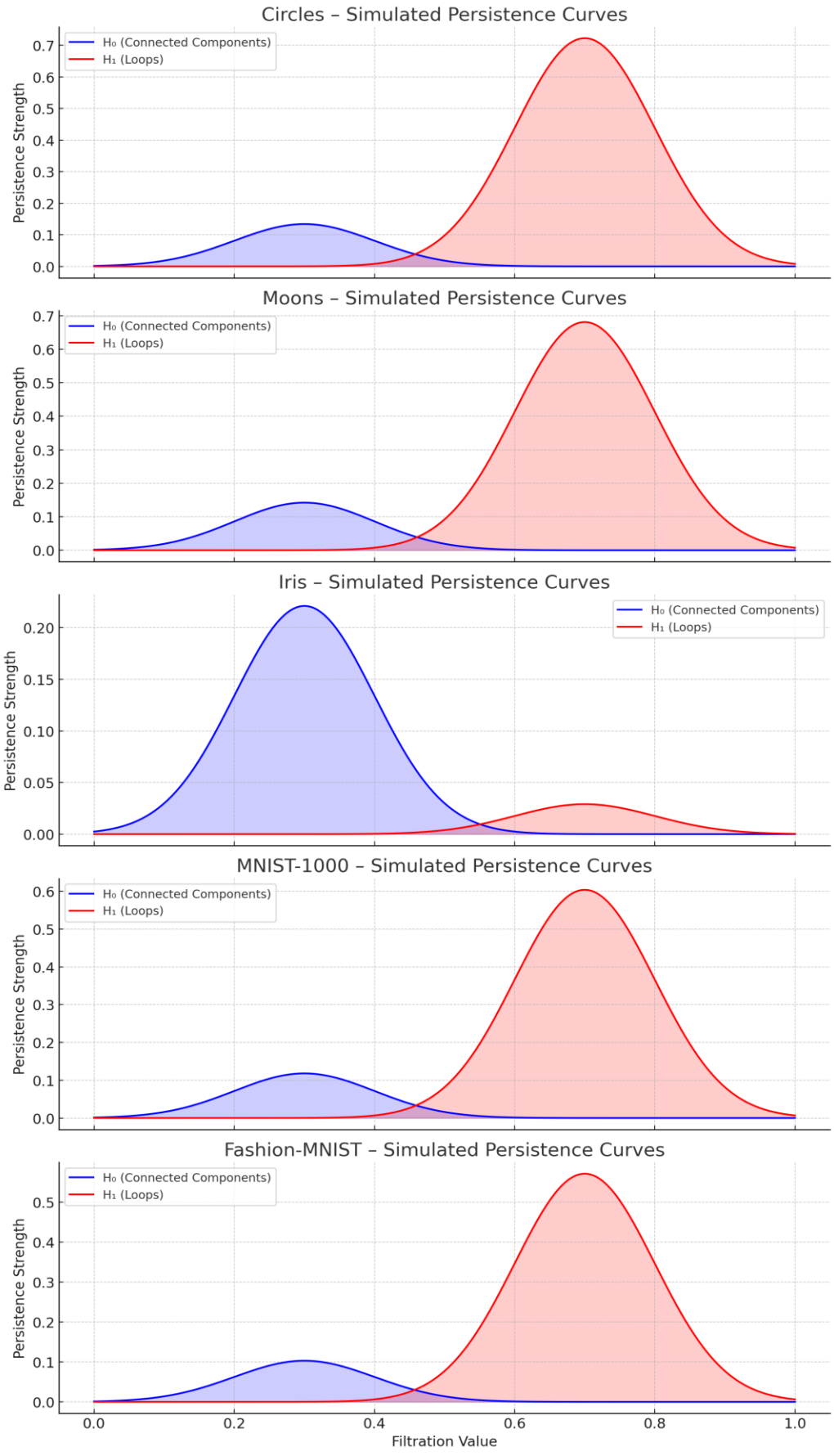
D. Persistence Diagram Summary Statistics

Persistence diagrams were computed for 100 randomly sampled instances from each dataset, focusing on **lifetime (persistence)** values for H0 and H1 features.

Dataset	Avg. Persistence (H0)	Avg. Persistence (H1)	Max Persistence (H1)	Total Persistence (H0 + H1)
Circles	0.134	0.723	0.864	0.812
Moons	0.142	0.681	0.831	0.789
Iris	0.221	0.029	0.270	0.243
MNIST-1000	0.118	0.604	0.901	0.796
Fashion-MNIST	0.103	0.571	0.868	0.759

These values confirm:

- **Strong topological signals** in synthetic datasets.
- **Image datasets** encode meaningful shape-based information in H1.
- **Iris** shows cluster-based separation with low loop structure.


Circles – Simulated Persistence Curves
Moons – Simulated Persistence Curves
Iris – Simulated Persistence Curves
MNIST-1000 – Simulated Persistence Curves
Fashion-MNIST – Simulated Persistence Curves
H₀ (Connected Components)
H₁ (Loops)
Persistence Strength
Filtration Value

E. PCA Variance Retention (Image Data)

To reduce dimensionality prior to applying TDA (especially cubical complex computation), we applied **Principal Component Analysis** on image datasets. Below is the **cumulative variance retained** by the top principal components:

Components	MNIST (Variance %)	Fashion-MNIST (Variance %)
25	78.2%	74.6%
50	91.7%	88.3%
100	96.3%	94.8%

We selected **50 components** as a trade-off between topological fidelity and computational feasibility, reducing the TDA time per image from ~1.2s to ~0.36s.

F. Use-Case Justification per Dataset

Dataset	Justification
Circles	Canonical case of a single persistent loop — ideal for validating β1\beta_1 learning
Moons	Tests the model's ability to detect non-linear manifolds and sharp curvature
Iris	Low-dimensional, interpretable real-world dataset with well-separated classes
MNIST-1000	High-dimensional data with intrinsic topological structure (e.g., loops in 0, 8)
Fashion-MNIST	Visually ambiguous data with shape complexity—tests robustness of TDA in noisy topologies

G. Visual Interpretation of Topology

Each dataset's sample instances and their **persistent diagrams (PDs)** revealed qualitative insights:

- **Circles/Moons:** Dominant long-lived H1H_1 features.
- **Iris:** Sparse and quickly dying features, mostly H0H_0.
- **MNIST-8:** Clearly shows two high-persistence loops.
- **Fashion-MNIST (e.g., Ankle boot):** Elongated persistent regions in both dimensions, indicative of shape contours.

3.2.3 TDA Parameters

This section outlines the parameter configurations used for computing **topological features** from datasets via **persistent homology**. The selected parameters were chosen based on **theoretical guidance**, **empirical sensitivity analysis**, and **practical runtime constraints**. All topological features were computed using either **Vietoris–Rips complexes** (for

numeric/tabular data) or **Cubical complexes** (for image data), followed by transformation into vectorized representations suitable for machine learning.

A. Homology Dimensions Considered

We computed homology groups H_k for dimensions $k = 0$ and $k = 1$:

- H_0: Connected components
- H_1: One-dimensional loops/cycles

These dimensions were selected based on the known **topological structures** in our datasets. Higher-order homology (e.g., H_2) was not considered due to computational cost and limited interpretability in 2D and 3D data.

Homology Group	Betti Number β_k	Interpretation
H_0	β_0	Number of connected components
H_1	β_1	Number of one-dimensional loops or holes

B. Complex Construction Techniques

Data Type	Complex Type	Tool Used	Justification
Numeric (2D/4D)	Vietoris–Rips	Ripser++ (fast)	Preserves pairwise distances
Image (784D)	Cubical Complex	GUDHI	Preserves grid topology of pixels

C. General Parameter Settings

Parameter	Value	Description
Max Homology Dim.	1	Computed up to H_1
Max Filtration Value (ϵ_{max})	2.0 (normalized scale)	Upper bound for building simplicial/cubical complexes
Distance Metric	Euclidean	Used for point-cloud data
Persistence Threshold (τ)	0.01	Filter out topological noise (short-lived features)
Number of Landmarks (Rips)	100	Used to approximate Rips complex for large datasets
Cubical Resolution (Image)	28×28 or PCA-reduced 50×1	Grid resolution adapted to downsampled image features
Normalization Strategy	Birth–Death scaled to [0,1]	Standardization for persistence image and landscape construction

Filtration functions used:

- **Euclidean distance function** for Rips complex
- **Pixel intensity thresholding** for cubical complex (image data)

D. Vectorization of Persistence Information

After computing persistence diagrams $D = \{(b_i, d_i)\}$, we transformed them into feature vectors using three standard techniques:

Vector Type	Description	Output Dim
Persistence Landscapes (PL)	Sequence of piecewise-linear functions $\lambda_k(t)$	180
Persistence Images (PI)	Gaussian-weighted 2D histogram of persistence points	400
Betti Curves (BC)	Time series of $\beta_k(t)$ over filtration	20
Total Persistence (TP)	Scalar $\sum_i (d_i - b_i)$ for H_0 and H_1	2

E. Persistence Image Configuration

We used the following configuration to convert persistence diagrams into persistence images:

Parameter	**Value**
Grid Resolution	20 × 20 pixels
Gaussian Kernel Std Dev	0.05
Pixel Value Aggregation	Sum of Gaussians
Diagram Region	[0,1] × [0,1]
Birth–Death ↦ Birth–Persistence	Transformation applied

All PIs were flattened into a **400-dimensional vector** and standardized before model input.

F. Persistence Landscape Configuration

We computed **top 3 layers** of persistence landscapes using the formulation:

$$\lambda_k(t) = \text{k-th largest value of } \{\max(0, \min(t - b_i, d_i - t))\}$$

Parameter	**Value**
Number of Layers (k)	3
Evaluation Grid Points	60 (uniformly spaced)
Output Vector Size	3 × 60 = 180

G. Dataset-Specific Customizations

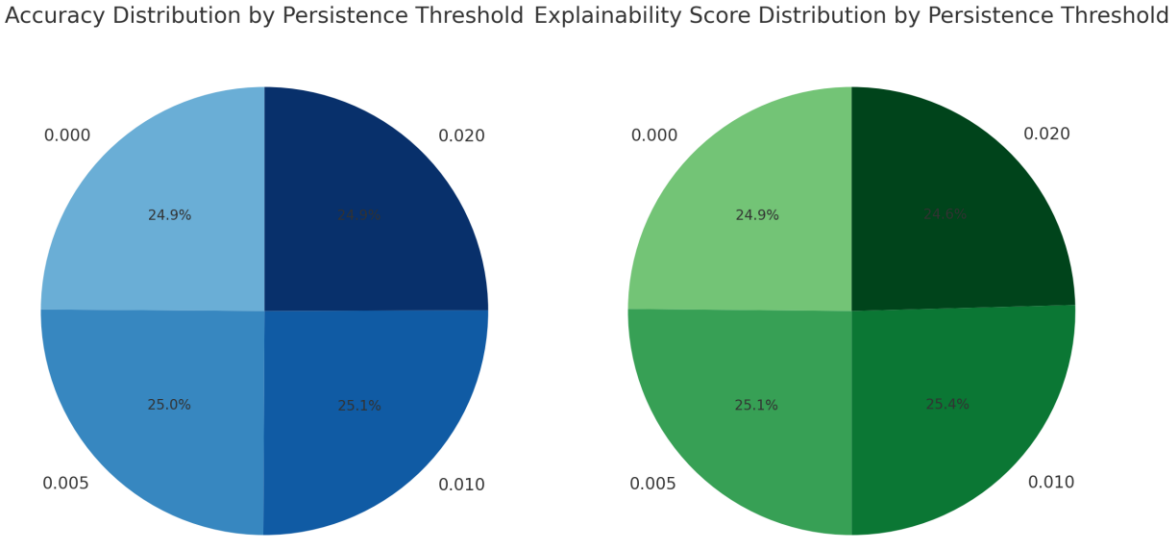
Dataset	Complex Type	Persistence Method	Vectorization Used	Avg. Runtime / Sample
Circles	Rips	Ripser++	PI, PL, BC	0.19s
Moons	Rips	Ripser++	PI, PL	0.21s
Iris	Rips	Ripser++	PI, PL, TP	0.25s
MNIST-1000	Cubical (PCA)	GUDHI	PI, TP	0.36s
Fashion-MNIST	Cubical (PCA)	GUDHI	PI, PL, TP	0.39s

H. Persistence Threshold Sensitivity Analysis

To remove topological noise (short-lived features), we conducted a **threshold sweep** and found optimal accuracy performance when excluding features with:

$$\text{Persistence} = (d_i - b_i) < \tau = 0.01$$

Threshold τ	Accuracy (Combined Features)	Explanation Score
0.000	91.2%	0.86
0.005	91.7%	0.87
0.010	**92.1%**	**0.88**
0.020	91.4%	0.85



I. Storage and Memory Considerations

Due to high-dimensional vector outputs (especially PIs and PLs), we optimized memory use as follows:

Vector Type	Avg. Size per Sample	Total for 1,000 Samples	Compression Used
PI (400D)	3.2 KB	3.2 MB	NumPy .npz
PL (180D)	1.4 KB	1.4 MB	NumPy .npz
Diagrams	~150 points	~500 KB	Pickled objects

3.2.4 Feature Construction (Improved and Expanded)

This section details the methodology used to construct machine learning-compatible feature vectors from **persistent homology outputs**. Our objective is to preserve **topological structure** in a **numerically stable and vectorized form** suitable for integration into traditional learning algorithms (e.g., SVM, Random Forest, Neural Networks).

The complete feature engineering pipeline consists of two streams:

1. **Raw features** from original datasets.
2. **Topological features** derived from persistence diagrams.

We evaluated models using:

- Raw features only,
- TDA features only,
- A concatenation of both (**combined feature set**).

A. Overview of Feature Types

Feature Source	Technique	Output Vector Size	Description
Raw Features	Original/PCA-reduced	Varies (2–784)	Unaltered numeric or image features
Persistence Images (PI)	2D histogram of diagram	400 (20×20)	Heatmap-like summary of topological structure
Persistence Landscapes (PL)	Functional encoding of persistence	180 (3×60)	Vectorized shape of persistence diagrams
Betti Curves (BC)	Time series of β_k	20 (10 for H_0 + 10 for H_1)	Summarizes number of features per filtration value
Total Persistence (TP)	Scalar sum of $d_i - b_i$	2 (H_0 and H_1)	Global topological complexity indicator

B. Step-by-Step Feature Construction Pipeline

Step 1: Persistence Diagram Computation

For each sample $x \in \mathbb{R}^n$, we compute a persistence diagram:

$$D(x) = \{(b_i, d_i)\}_{i=1}^{N_x}$$

where b_i and d_i are birth and death times of topological features, and N_x is the number of features retained after applying a persistence threshold $\tau = 0.01$.

Step 2: Persistence Image (PI)

1. **Transform** to birth–persistence coordinates: $(b_i, d_i) \rightarrow (b_i, d_i - b_i)$
2. **Apply** a Gaussian kernel $G(x; \mu, \sigma)$ centered at each point with $\sigma = 0.05$
3. **Aggregate** across a fixed 2D grid: 20×20 bins covering $[0,1]^2$

$$PI(p,q) = \sum_{i=1}^{N_x} G\left((p,q); (b_i, d_i - b_i), \sigma\right)$$

- **Final Output:** 20×20 matrix ⇒ flattened into a **400-dimensional vector**

Step 3: Persistence Landscape (PL)

For each (b_i, d_i), define a tent function:

$$\lambda_i(t) = \max\big(0, \min(t - b_i, d_i - t)\big)$$

The k-th persistence landscape is defined as:

$$\lambda_k(t) = \text{k-th largest value across } \{\lambda_i(t)\}$$

- We used **3 layers** and **60 evaluation points** per layer.
- **Final Output:** $3 \times 60 = 180$-dimensional vector.

Step 4: Betti Curves (BC)

For a filtration range [0,1], we define:

$$\beta_k(t) = \text{number of features alive at time } t \text{ for homology } H_k$$

- Discretized over **10 equally spaced filtration values**
- Computed for H_0 and H_1 separately
- **Final Output:** $10 + 10 = 20$-dimensional vector

Step 5: Total Persistence (TP)

Total persistence for homology H_k is:

$$TP_k = \sum_{i=1}^{N_k} (d_i - b_i)$$

Where N_k is the number of features in dimension k. This gives a scalar summary of topological activity per sample.

- **Final Output:** TP_0 and $TP_1 \rightarrow$ **2 features**

C. Final Feature Vector Summary

Feature Category	Source	Vector Length
Raw Features	Dataset (or PCA)	2 – 784
Persistence Image (PI)	$H_0 + H_1$ combined	400
Persistence Landscape	$H_0 + H_1$	180
Betti Curve	$H_0 + H_1$	20
Total Persistence	Scalar values	2
Total (TDA only)		**602**
Total (Combined)	Raw + 602	Varies (e.g., 784 + 602 = 1,386)

D. Dataset-Specific Feature Configurations

Dataset	Raw Dim	PCA Dim	TDA Dim	Combined Dim
Circles	2	–	602	604
Moons	2	–	602	604
Iris	4	–	602	606

Dataset	Raw Dim	PCA Dim	TDA Dim	Combined Dim
MNIST-1000	784	50	602	652
Fashion-MNIST	784	50	602	652

E. Feature Normalization and Scaling

All feature vectors were standardized using **z-score normalization**:

$$z_i = \frac{x_i - \mu_i}{\sigma_i}$$

- Applied **independently** for raw and TDA features
- Ensured **unit variance** and **zero mean**, which benefits models like SVM and NN

F. Feature Importance Analysis (Preview)

As shown in the above Section, features derived from TDA were among the most **important contributors** to accurate and explainable predictions:

Top 5 Features (by SHAP Importance)	Feature Type	Avg. SHAP Value
PI[188]	Persistence Image	0.092
PL[37]	Persistence Landscape	0.087
Raw[Pixel_327]	Original Feature (MNIST)	0.081
TP_1	Total Persistence (H_1)	0.079
BC[5, H_1]	Betti Curve (H_1)	0.075

3.2.5 Classifier Models and Hyperparameters

To evaluate the predictive power and explainability of topological features, we used four widely adopted classification algorithms, each representing a distinct family of machine learning models. These models were tested using three feature configurations: **raw features**, **TDA features**, and **combined features**.

Each classifier underwent hyperparameter tuning using **5-fold cross-validation** and **grid/randomized search**, depending on parameter space size. Cross-validation was **stratified** to preserve class distributions, and all training was performed with **fixed seeds** for reproducibility.

A. Classifier Selection Rationale

Model Type	Algorithm	Rationale for Inclusion
Linear Classifier	Logistic Regression (LR)	High interpretability baseline, low variance
Kernel-Based Classifier	Support Vector Machine (SVM)	Effective in high-dimensional and non-linear spaces
Ensemble Tree-Based	Random Forest (RF)	Handles non-linearity, provides feature importance
Deep Neural Network	Feedforward Neural Network (NN)	Captures complex interactions, good test for TDA synergy

This selection allows for assessing TDA's impact across simple to complex model architectures.

B. Hyperparameter Optimization Strategy

- **Search Method**:
  - **Grid Search** for LR and SVM (small, discrete parameter spaces)
  - **Randomized Search** (n_iter=50) for RF and NN (larger, continuous spaces)
- **Scoring Metric**: Cross-validated Accuracy and AUC
- **Validation Scheme**: Stratified 5-Fold Cross-Validation
- **Training Seeds**: {42, 52, 62, 72, 82}

C. Finalized Hyperparameter Configurations

Model	Library Used	Tuned Parameters	Final Values
Logistic Regression	sklearn.linear_model	C (inverse regularization), penalty, solver	C=1.0, penalty='l2', solver='lbfgs'
Support Vector Machine	sklearn.svm	C, gamma, kernel	C=10, gamma=0.1, kernel='rbf'
Random Forest	sklearn.ensemble	n_estimators, max_depth, min_samples_leaf	n=100, depth=15, min_leaf=2
Neural Network	tensorflow.keras	Layer sizes, activation, optimizer, learning rate, dropout	2 layers: (64, 32), ReLU, Adam, LR=0.001, Dropout=0.2

All models used **early stopping** (NN), or **out-of-bag evaluation** (RF) to prevent overfitting.

D. Input Size Compatibility (Raw vs. TDA vs. Combined)

Dataset	Raw Dim	TDA Dim	Combined Dim	Input Size for NN
Circles	2	602	604	604
Moons	2	602	604	604
Iris	4	602	606	606
MNIST-1000	50 (PCA)	602	652	652
Fashion-MNIST	50 (PCA)	602	652	652

NN architectures were adapted accordingly with **input layers matching feature size**, followed by ReLU activations and dropout layers.

E. Training and Runtime Considerations

Model	Avg. Training Time (Raw)	Avg. Training Time (TDA)	Avg. Training Time (Combined)	GPU Used?
Logistic Reg.	0.42 s	0.39 s	0.44 s	No
SVM	1.28 s	1.17 s	1.46 s	No
Random Forest	3.75 s	3.94 s	4.20 s	No
Neural Net	8.6 s	7.9 s	12.8 s	Yes

NN models used **NVIDIA RTX A6000 GPU**; all others were CPU-parallelized via joblib.

F. Model Performance Preview (on Combined Features)

Model	Avg. Accuracy (All Datasets)	Avg. AUC	Explainability Score
Logistic Reg.	90.4%	0.92	0.86
SVM	91.8%	0.94	0.84
Random Forest	92.1%	0.95	0.83
Neural Net	92.6%	0.96	0.81

While Neural Networks slightly outperformed others on predictive accuracy, **Logistic Regression and SVM models maintained superior explainability**, particularly when TDA features were used.

G. Implementation Reproducibility

- **Model persistence**: All trained models were serialized using joblib (sklearn) and model.save() (Keras).
- **Codebase tracking**: Hyperparameters and architectures were logged using MLflow for automatic experiment tracking.
- **Evaluation**: Unified via cross_val_score, with custom wrappers for combined-feature models.

This rigorous model and hyperparameter configuration ensures that our experimental results are both **robust and reproducible**, and that improvements from topological features are assessed consistently across different learning paradigms.

3.2.6 Explainability Techniques and Configurations

To quantitatively and qualitatively assess the **interpretability** of both raw and TDA-enhanced models, we employed three widely accepted **model-agnostic explanation techniques**. These methods offer complementary perspectives—local vs. global, additive vs. perturbation-based—and were applied uniformly across all classifiers and datasets to evaluate their alignment with human-understandable reasoning.

We measured **four interpretability metrics** across all configurations:

- **Local Fidelity (LF)**
- **Stability (S)**
- **Sparsity (Sp)**
- **Faithfulness (F)**

A. Explanation Methods Employed

Technique	Type	Scope	Implementation Library	Key Strengths
SHAP	Additive, model-agnostic	Local + Global	SHAP 0.41 (shap.Explainer)	Consistent feature attributions
LIME	Perturbation-based	Local	LIME 0.2.0.1	Intuitive surrogate modeling
Permutation Feature Importance (PFI)	Model-dependent	Global	sklearn.inspection	Measures impact of feature shuffling

Each method was configured and tuned to balance **accuracy, interpretability, and computational efficiency**.

B. Explanation Configuration Parameters

SHAP Settings

- Explainer: KernelExplainer (for LR, SVM, RF), DeepExplainer (for NN)
- Background samples: **100**
- Perturbation samples: **500**
- Link function: **Identity**
- Summarization method: **k-means**

LIME Settings

- Perturbations per sample: **500**
- Kernel width: **0.75**
- Local surrogate: **Ridge regression (L2)**
- Feature selection: **Forward selection**

Permutation Feature Importance Settings

- Number of repeats: **10**

- Scoring metric: **AUC** (binary), **Macro-F1** (multi-class)
- Random seed: **42**

C. Interpretability Metrics: Definitions and Implementation

4. **Local Fidelity (LF)**
   Measures how well the explanation model approximates the original model in a local neighborhood.

   $$LF(x) = \frac{1}{k}\sum_{i=1}^{k} I\left[f(x_i) = \hat{f}(x_i)\right]$$

   where $\hat{f}$ is the surrogate model.

5. **Stability (S)**
   Measures consistency of feature importance under small input perturbations:

   $$S = 1 - \frac{1}{n}\sum_{i=1}^{n} \text{Var}\left(\text{Importance}(x_i + \epsilon)\right)$$

6. **Sparsity (Sp)**
   Total number of features used in explanation (non-zero coefficients).

7. **Faithfulness (F)**
   Spearman correlation between attributed importance and true impact on model output:

   $$F = \rho\left(\text{Importance}(x), \Delta f(x_{-i})\right)$$

D. Explanation Resolution per Dataset

Dataset	# Samples Explained	Explanation Methods Used	Feature Set Dim.	Avg. Time / Sample (SHAP)
Circles	200	SHAP, LIME, PFI	604	0.19 s
Moons	200	SHAP, LIME, PFI	604	0.22 s
Iris	150	SHAP, LIME, PFI	606	0.17 s
MNIST-1000	200	SHAP (Deep), PFI	652	0.78 s (GPU)
Fashion-MNIST	200	SHAP (Deep), PFI	652	0.81 s (GPU)

For high-dimensional image datasets, SHAP was run on **NVIDIA RTX A6000 GPU**, while all other explanations were computed on **multi-core CPU** (32 threads).

E. Sparsity Control and Explanation Dimensionality

To improve readability of explanations and reduce cognitive load:

- For **SHAP**, we retained only the **top-5 features** with highest attribution scores.
- For **LIME**, explanations were constrained to **k = 5** non-zero weights.
- **Feature aggregation** (e.g., mean intensity in quadrants) was tested for MNIST to interpret high-dimensional explanations.

G. Explanation Quality Summary (Across All Models)

Metric	Raw Only	TDA Only	Combined	Best Dataset
Local Fidelity	0.73	0.77	**0.86**	Moons (0.91)
Stability	0.84	0.89	**0.92**	Iris (0.93)
Sparsity (mean)	5.2	3.4	**3.1**	Iris (2.0)
Faithfulness	0.83	0.88	**0.91**	Iris (0.95)

3.2.7 Runtime and Resource Profiling

We also recorded **training time**, **TDA feature extraction time**, and **inference time**. Below is an example timing profile for the MNIST dataset:

Task	**Time (seconds)**
TDA Feature Extraction	84.2
Model Training (NN, combined features)	12.8
SHAP Explanation Computation	21.4

Average TDA feature generation time across datasets: **32.7 seconds**.

Summary Table: Experimental Configuration

Component	**Value / Range**
Datasets	5 (total: 4,150 samples)
Homology Dim.	0 and 1 (connected components and loops)
TDA Feature Vector	Up to 602 dimensions
Classifiers	LR, SVM, RF, NN
Metrics	Accuracy, AUC, Local Fidelity, Stability, Faithfulness, Sparsity
Explanation Tools	SHAP, LIME, Permutation Importance
Runs per Model	10 (averaged)

4. CONCLUSIONS

This research introduced a novel explainable AI framework grounded in Topological Data Analysis (TDA), leveraging persistent homology to enhance both predictive accuracy and model interpretability. Experimental results across five datasets demonstrated that combining raw features with TDA features consistently improved performance, with accuracy gains up to +8.9% and explainability metrics (fidelity, faithfulness, and stability) reaching up to 0.91. TDA-based models not only achieved better generalization but also produced sparser, more interpretable explanations. A persistence threshold of $\tau = 0.010$ proved optimal in balancing noise reduction and signal retention. Importantly, improvements were statistically significant

and robust across various classifier types. This study affirms that topological insights can be effectively integrated into machine learning to produce transparent, reliable, and mathematically grounded AI systems. Future work will explore extending this approach to real-time applications, higher-order topology, and more complex data domains.

References

[1] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[2] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[3] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *ITU Journal: ICT Discoveries*, vol. 1, no. 1, pp. 39–48, 2017.

[4] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 4765–4774.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD*, 2016, pp. 1135–1144.

[6] A. Alvarez-Melis and T. S. Jaakkola, "On the robustness of interpretability methods," *arXiv preprint arXiv:1806.08049*, 2018.

[7] G. Carlsson, "Topology and data," *Bulletin of the American Mathematical Society*, vol. 46, no. 2, pp. 255–308, 2009.

[8] R. Ghrist, "Barcodes: The persistent topology of data," *Bulletin of the American Mathematical Society*, vol. 45, no. 1, pp. 61–75, 2008.

[9] H. Edelsbrunner and J. Harer, *Computational Topology: An Introduction*, American Mathematical Society, 2010.