JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES
www.journalimcms.org

ISSN (Online) : 2454 -7190 Vol.-21, No.-06, June (2026) pp 114-130 ISSN (Print) 0973-8975

ENHANCING IMBALANCED DATA CLASSIFICATION USING MULTI-CLASS MAHALANOBIS DATA TRANSFORMATION AND LOGISTIC REGRESSION

Aparna Shrivatsava[1], P. Raghu Vamsi[2]

[1]Department of Information Technology, JSS Academy of Technical Education, Noida, UP, India.

[2] Department of Computer Science and Engineering, Jaypee Institute of Information Technology, Sector 62, Noida, UP, India.

Email : [1]aparnashrivatsava1981@gmail.com, [2]prvonline@yahoo.co.in

Corresponding Author: **Aparna Shrivatsava**

https://doi.org/10.26782/jmcms.2026.06.00006

(Received: March 05, 2026; Revised: June 02, 2026; Accepted: June 11, 2026)

Abstract

Imbalanced datasets present significant challenges for standard classification algorithms and often lead to biased models that perform poorly on minority classes. To address this, this study proposes a framework combining adaptive data transformation, Multi-class Mahalanobis Distance (MMD) metric learning, and logistic regression. The MMD transformation optimizes the feature space by computing a shared covariance matrix to pull similar data points closer and to increase class separability. The proposed MMD (LR) method significantly outperformed existing techniques like Local Mahalanobis Distance Learning (LMDL) on various benchmark imbalanced datasets. MMD(LR) achieved an average performance gain of 6.70% in precision, 7.16% in F1-score, and 14.10% in Area Under the Curve (AUC). Notably, the model achieved a perfect 100% AUC on the Wine and Iris datasets, and a 99.69% AUC on the Breast Cancer dataset. It demonstrates its exceptional robustness and adaptability for classifying complex and imbalanced data.

Keywords: Classification, Imbalanced dataset, Logistic Regression, Machine Learning, Mahalanbois distance, Multivariate datasets.

I. Introduction

Imbalanced datasets feature one class that significantly outnumbers the others. This issue is prevalent in critical applications such as fraud detection, leading to challenges for traditional classifiers [II, XIV]. This imbalance causes classifiers to prioritize the majority class, which may result in the misclassification or neglect of the minority class. Compounding factors, including within-class imbalance, small sample sizes, class overlap, and noisy data, hinder model performance on unseen data, particularly with smaller datasets. Standard performance metrics, like accuracy, can be misleading under these conditions, and hence, there is a necessity for specialized

handling of imbalanced datasets for reliable classification. Data transformation alters raw data into a more analyzable format to address issues such as disparate scales and uneven distributions, making it crucial before analysis. Common transformation methods include scaling (e.g., min-max normalization, z-score normalization), advanced techniques (e.g., Pareto scaling, Vast scaling), and skewness correction (e.g., log, Box-Cox, Yeo-Johnson) [V, IX, III]. Additional methods like encoding, imputation, binning, and discretization enhance data quality and manage continuous data. Metric learning, aligned with data transformation, focuses on learning distance or similarity measures tailored to specific tasks [XI, XXVII]. The primary objective of these techniques is to optimize the separation of data points and enhance classification effectiveness. In contrast to transformation, metric learning often employs label information to guide the development of similarity functions to function as a supervised learning technique that accentuates class distinctions.

To this end, this study proposes a structured approach to improve imbalanced dataset classification by integrating data transformation, distance metric learning, and logistic regression. The method initiates with data transformation, followed by applying Multi-class Mahalanobis Distance learning to enhance class separability and emphasize minority distinctions [XVIII]. The transformed features are subsequently classified via logistic regression. Simulations on imbalanced benchmark datasets obtained from the UCI Machine Learning Repository were conducted to evaluate the proposed method's effectiveness using metrics including Precision, F1-score, and Area Under the Curve. Results indicated notable classification performance improvements compared to existing literature. The paper is structured as follows: Section 2 reviews related work, Section 3 outlines the methodology, Section 4 presents the simulation setup and results, and Section 5 concludes the study.

II. Related Work

Tsai et al. [XXX] developed a hybrid method for binary classification that combines data block construction, dimensionality reduction via Metric Learning for Kernel Regression, and ensemble learning with deep neural networks for addressing challenges such as overfitting and cost parameter settings. This system exhibited superior recall and overall performance compared to existing methods on various public datasets. Ashoorzadeh et al. [IV] proposed a classification method for local optimum issues in Large Margin Nearest Neighbor classifiers by defining data similarities with a cost function, employing a genetic algorithm for solution space reduction, and utilizing gradient descent for cost parameter optimization. It achieved the highest accuracy relative to k-NN and LMNN on benchmarks. Yin et al. [XXXIV] introduced a model for imbalanced data classification by incorporating sampling, data space construction, cost-sensitive learning, and ensemble learning components, with AWA to adjust class weights, outpacing state-of-the-art methods across 14 public datasets. Elmi et al. [XII] developed a dynamic selection technique for multi-classifier systems by utilizing a multi-label classifier during training for effective classifier identification. It demonstrated strong accuracy and outperformed numerous benchmarking techniques. Safi et al. [XXIV] introduced the Enhanced Tree Ensemble to address extreme class imbalance by generating synthetic minority class data to balance training and employing tree selection based on performance metrics,

outperforming traditional machine learning methods in error rate and precision. Zhu et al. [XXXVI] proposed a method merging hybrid resampling with fine cost-sensitive Support Vector Machine (FCSVM), using Mahalanobis distance to replace "pseudo-negative samples" and optimize class cost weights via RIME, achieving enhanced classification performance on imbalanced datasets. Karthikeyan et al. [XIX] formulated a classification approach combining chi-square feature filtering and Mahalanobis distance, reducing misclassification without data resampling and outperforming multiple state-of-the-art algorithms with better AUC scores. Sun et al. [XXVII] reviewed the class imbalance issue in data classification, assessing standard algorithms' challenges, categorizing solutions at data and algorithmic levels, and discussing evaluation metrics and complexities in multi-class scenarios. Gøttcke et al. [XVI] introduced kNN-BPP, optimizing recall through balancing internal prior class probabilities, demonstrating competitive performance to SMOTE with similar complexity to kNN. Qiao et al. [XXIII] presented LMNNB and LMNNB-E, integrating metric learning and ensemble learning for clearer class boundaries, with LMNNB minimizing distances and LMNNB-E enhancing performance via soft voting. Siddappa and Kampalappa [XXVI] introduced Local Mahalanobis Distance Learning (LMDL) for imbalanced data classification, learning a Mahalanobis distance metric for prototypes to capture local discriminative information, preserving data distribution, and showing superior metrics such as F-measure on datasets including E-coli and breast cancer.

III. Proposed Methodology

The objective of the proposed methodology is to enhance classification in imbalanced datasets. The process is detailed in Fig 1. Initially, Imbalanced multivariate tabular datasets undergo pre-processing, which includes identifying outliers, addressing missing data, and scaling features based on data distribution. Subsequently, datasets are split into training sets using stratified sampling to maintain class ratios. To improve feature distinction, Multi-class Mahalanobis Distance (MMD) metric learning is applied to training data. The *fit()* function calculates class-specific parameters, such as the mean vector and covariance matrix, using training data and labels. These parameters transform both datasets into new features based on MMD values via the *transform()* function. The transformed training data is used to train a Logistic Regression (LR) model. It is selected for its stability and interpretability, and is also effective in handling imbalanced datasets with methods like class weighting. The trained model classifies MMD-transformed test data, and classification performance is assessed using metrics including precision, F1-score, and area under the Receiver Operating Characteristic (AUC) score for both datasets. Algorithm 1 outlines the procedure, with a detailed explanation in subsequent sections.

Data pre-processing: The first step of the proposed methodology is to utilize a data pre-processing sequence to prepare the datasets for model training and evaluation. This sequence involves the identification of outliers, the assessment of data normality, and the application of feature scaling techniques [IV, V].

- Outliers within each feature were identified using the Inter-Quartile Range (IQR) method. The IQR is defined as the difference between the third quartile (Q3) and the first quartile (Q1), expressed as IQR=Q3−Q1. Data points that fell outside the

calculated range of [Q1−1.5×IQR, Q3+1.5×IQR] were classified as outliers and were subsequently removed from the dataset.

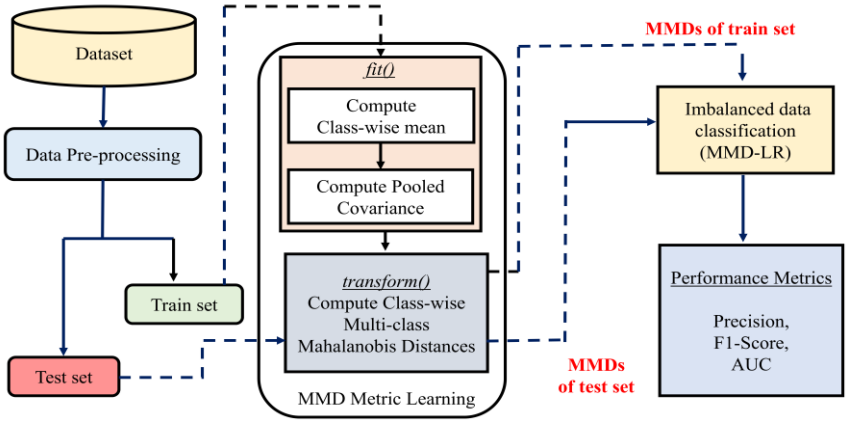


Fig. 1. Methodology of the proposed work.

- The Shapiro-Wilk test was performed to determine if the distributions of the features significantly differed from a normal distribution. For a given dataset $X = x_1, x_2, \ldots, x_n$, the Shapiro-Wilk test statistic, W, was calculated using the following formula:

$$W = \frac{\left(\sum_{i=1}^{n} a_i(x_i)\right)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \quad (1)$$

  In this equation, a_i represents precomputed constants derived from the expected values of order statistics for a normally distributed dataset, $x_{(i)}$ denotes the ordered sample values, and $\bar{x}$; is the sample mean. A low value of W indicates a departure from normality.
- The outcomes of the outlier detection and normality tests informed the selection of a suitable scaling method for each feature. Robust scaling, which is less sensitive to the presence of outliers, was applied to features where outliers were detected. Standard scaling was used for features whose distributions approximated a normal distribution. Conversely, min-max scaling was employed for features that did not follow a normal distribution. These transformations were implemented to ensure that all features contributed equitably to the classification process. This prevented features with larger numerical ranges from unduly influencing the learning process.

Multi-class Mahalanobis Distance (MMD) Metric Learning: Mahalanobis distance (MD) [XVIII] is a metric used to measure the distance between a data point and a distribution. It is a promising approach for identifying outliers in multivariate data because it accounts for correlations between variables. Given a multivariate dataset $X \in R^{n \times d}$ where n is the number of data points (rows), and d is the number of variables (columns), the MD computation starts by calculating the mean $\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$ of each variable (column) by averaging the values for that variable across all data points. Next, the covariance matrix $S = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T$ is computed to capture how the variables vary together. Finally, the MD between a point x and a distribution with mean μ and covariance matrix S is computed by

$$MD = \sqrt{(x_i - \mu)S^{-1}(x_i - \mu)^T} \quad (2)$$

The MD and its squared values are closely linked to the Gaussian and Chi-squared distributions, respectively. Data points with a higher MD lie further from the mean and are less likely to belong to the Gaussian distribution. On the other hand, the squared MD follows a Chi-squared distribution with degrees of freedom equal to the number of features (or variables) d. This is because the MD measures deviations in terms of variance and covariance, similar to how the Chi-squared statistic works in hypothesis testing for variance. As an extension of MD, computation of MMDs is the key aspect of metric learning in the proposed method. These MMDs represent refined measures of separation between individual data points and the central tendency of their respective classes, considering the specific covariance structure of each class. The inverse of the covariance matrix for each class distribution effectively captures the correlations between features and the inherent structure present within that class. By mapping data samples into a new feature space based on these calculated distances, the classification task becomes more capable of distinguishing between classes, especially when the number of samples in each class is uneven. Let the input dataset consist of N samples in a d-dimensional feature space, i.e., $X \in R^{N\times d}$, with corresponding class labels $y \in \{1,2,\dots,K\}^N$, where K is the number of unique classes. Algorithm 1 presents the procedure for MMD transformation. The MMD transformation involves the following steps:

Step 1: Class mean calculation: For each class $k = 1, 2, \dots, K$, compute the class-conditional mean vector:

$$\mu_k = \frac{1}{N_k}\sum_{x_i:y_i=k} x_i \quad (3)$$

Where, N_k is the number of samples belonging to class k. These K mean vectors are arranged into a mean matrix $M = [\mu_1, \mu_2, \dots, \mu_K] \in R^{d\times K}$.

Step 2: Class covariance matrix computation and regularization: For each class k, estimate the covariance matrix:

$$\Sigma_k = \frac{1}{N_k - 1}\sum_{x_i:y_i=k}(x_i - \mu_k)(x_i - \mu_k)^{\top}. \quad (4)$$

To ensure positive-definiteness and numerical stability (especially in high-dimensional or small-sample regimes), a shrinkage regularization is applied [VIII, VII, VI]:

$$\Sigma_k \leftarrow \Sigma_k + \epsilon I, \quad \epsilon > 0 \quad (\text{e.g., } \epsilon = 10^{-6}), \quad (5)$$

where I is the $d \times d$ identity matrix.

Step 3: Multi-class covariance estimation: To ensure that the global covariance estimates are not destabilized by the low-covariance matrices of the sparsely sampled minority classes, the multi-class covariance matrix is calculated as the frequency-weighted average of the regularized class-specific covariance estimates.

$$\Sigma_m = \sum_{k=1}^{K}\frac{n_k}{n}\Sigma_k \quad (6)$$

Its inverse, Σ_m^{-1}, defines a global metric that balances varying class spreads and orientations. For unequal sample sizes, equal-weight averaging falsely assumes identical estimation confidence, allowing the high-variance, ill-conditioned covariance matrices of minority classes to distort the overall geometry. By weighting each class covariance by its relative frequency, the pooled estimate acts as a maximum likelihood estimator by naturally suppressing minority-class noise. Furthermore, this approach is hybridized with Tikhonov shrinkage regularization ($\Sigma_k + \epsilon I$, Equation 5) [VIII, VII, VI]. While equal-weighting over-represents minority noise and pure shrinkage ignores class skewness, the proposed method optimally resolves both: Tikhonov regularization guarantees numerical stability for small samples, while frequency weighting ensures a statistically robust MMD metric.

Step 4: Data transformation to MMD Features: For each sample x_i, compute its Mahalanobis distance to every class mean using the shared multi-class inverse covariance:

$$d_{i,k} = \sqrt{(x_i - \mu_k)^\top \Sigma_m^{-1} (x_i - \mu_k)}, \quad k = 1, \dots, K. \tag{7}$$

These K distances form a new feature vector:

$$z_i = \left[d_{i,1}, d_{i,2}, \dots, d_{i,K}\right]^\top \in R^K. \tag{8}$$

The transformed dataset becomes $Z \in R^{N \times K}$, where each row is the MMD metric representation of the corresponding original sample. Under the assumption that samples within class k follow a multivariate Gaussian distribution $x \sim \mathcal{N}(\mu_k, \Sigma_k)$, and if $\Sigma_k \approx \Sigma_m$ (a reasonable approximation in many practical cases), the squared MD to the true class mean follows a chi-squared distribution:

$$(x_i - \mu_k)^\top \Sigma_m^{-1} (x_i - \mu_k) \sim \chi_d^2 \quad (\text{when } y_i = k). \tag{9}$$

Taking the square root yields equation (7), which follows a chi-distribution with d degrees of freedom. For large d, this is well-approximated by a normal distribution (via the Wilson–Hilferty or central limit theorem arguments) [XXXII, XXXV], supporting the subsequent application of discriminant analysis in the transformed space. This MMD metric learning-based transformation can be used as a pre-processing step for classical discriminant classifiers due to a significant reduction in dimensionality from the original feature dimension d to the number of classes K (typically $K \ll d$). This overcomes the curse of dimensionality and enhances computational efficiency in high-dimensional domains. The second feature is the enhancement of class separability by mapping each sample into a new space, where instances from the same class are typically located in close proximity to their corresponding centroids and far away from competing classes' centroids, resulting in more linear or quadratically separable structures. By utilizing a pooled covariance Σ_m instead of class-specific inverses, the transformation is more resistant to small-sample instability and heterogeneity in classes. It avoids the potential for extreme distortions when individual class covariances are estimated poorly or almost singly for discrimination and imbalanced data classification.

While dimensionality reduction inherently involves some loss of the original feature space geometry, the MMD transformation is specifically designed to preserve the

discriminative boundaries necessary for parametric classification, rather than generating naive spherical clusters. To quantify the geometric shift, we evaluated the global Silhouette coefficient before and after transformation as shown in Fig 2. Fig 2 (a) presents the data distribution and Silhouette score [XXXI] with original data, and Fig 2 (b) presents the data distribution and Silhouette score after MMD transformation on the highly imbalanced Yeast dataset.

Algorithm 1: Multi-class Mahalanobis Distance (MMD) metric calculation

Input:

$X \in R^{n \times d}$: Input dataset matrix (n samples, d features).

$y \in \{C_1, C_2, \ldots, C_k\}$: Corresponding class labels (k unique classes).

λ: Small positive scalar for regularization (e.g., $1e-06$).

I_d: Identity matrix of dimension $d \times d$.

Output:

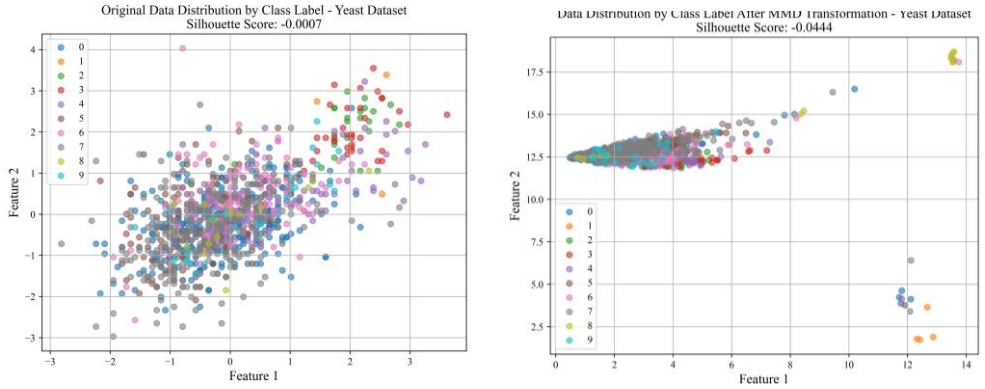
$D \in R^{n \times k}$: Transformed feature matrix.

Procedure (MMD Transformation):

1. **Calculate class mean vectors (μ_i):**
   1.1. For each class C_i ($i = 1$ to k), compute the mean vector $\mu_i \in R^d$. n_i is the number of samples in C_i.
   $$\mu_i = \frac{1}{n_i} \sum_{x \in C_i} x$$
2. **Calculate regularized class covariance matrices (Σ_i'):**
   2.1. For each class C_i, compute the covariance matrix Σ_i.
   $$\Sigma_i = \frac{1}{n_i - 1} \sum_{x \in C_i} (x - \mu_i)(x - \mu_i)^\top$$
   2.2. Apply Tikhonov regularization:
   $$\Sigma_i' = \Sigma_i + \lambda I_d$$
3. **Estimate multi-class covariance matrix (Σ_m):**
   3.1. Calculate the multi-class covariance matrix Σ_m as the average of all regularized class covariance matrices. n is the total number of samples. n_i is the number of samples of class i.
   $$\Sigma_m = \frac{n_i}{n} \sum_{i=1}^{k} \Sigma_i'$$
   3.2. Compute the inverse of the multi-class covariance matrix, Σ_m^{-1}.
4. **Transform data into MMD features:**
   4.1. For every sample $x \in X$ and for each class mean μ_i, compute the MMD, $d_i(x)$, using the inverse of the multi-class covariance matrix Σ_m^{-1}.
   $$d_i(x) = \sqrt{(x - \mu_i)^\top \Sigma_m^{-1} (x - \mu_i)}$$
5. **Form transformed feature matrix D:**
   5.1. Organize all calculated MMDs into the feature matrix $D \in R^{n \times k}$, where D_{ij} is the MMD of the i-th sample (x_i) to the j-th class mean (μ_j).
   $$D_{ij} = d_j(x_i)$$
6. **Return D**

The original feature space exhibited a Silhouette score of -0.0007, indicating extreme baseline class overlap. Following the MMD transformation, the score shifted to -0.0444. It suggests the transformed space remains tightly intermixed; the

improvement in the classifier's predictive performance is due to the theoretical advantage. The MMD transformation mapped the data into a functional feature space where the underlying classes become highly linearly separable.



(a) (b)

Fig. 2. Data distribution of the Yeast dataset before and after MMD transformation

By normalizing the feature space with respect to the pooled covariance matrix (Σ_m), the mapping successfully retains and aligns the specific discriminative variance required by the logistic regression hyperplanes by overriding the limitations of traditional distance metrics.

Logistic Regression: Logistic regression (LR) [XXV, XXXI] is a statistical model employed for classification tasks. Algorithm 2 shows the procedure for classifying MMD-transformed data with LR. LR estimates the probability that a given input belongs to a specific category. Given a feature vector $x \in \mathbb{R}^d$, and model parameters consisting of a weight vector $w \in \mathbb{R}^d$ and a bias term $b \in \mathbb{R}$, the predicted probability is computed as:

$$P(y = 1 \mid \mathbf{x}) = \sigma(\mathbf{w}^{\mathsf{T}}\mathbf{x} + b) = \frac{1}{1+e^{-(\mathbf{w}^{\mathsf{T}}\mathbf{x}+b)}} \tag{10}$$

The model's parameters are learned by minimizing the log-loss, also known as cross-entropy loss, which quantifies the discrepancy between the actual class labels and the predicted probabilities. For binary classification, the loss function is defined as:

$$\mathcal{L}(\boldsymbol{w}, b) = -\frac{1}{n}\sum_{i=1}^{n} \alpha_{y_i}[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)] \tag{11}$$

where $\hat{y}_i$ represents the predicted probability for the i-th sample, and $y_i \in \{0, 1\}$ is the corresponding true label, and α_{y_i} is the class weight, computed inversely proportional to class frequencies. This weighting ensures the optimization gradient is not dominated by the majority class. We implemented this using the class_weight='balanced' option available in the scikit-learn library for the LR implementation. In the context of multi-class classification, logistic regression extends its capability by using the SoftMax function instead of the sigmoid function. This generalization is referred to as multinomial logistic regression. For a problem with K distinct classes, the model calculates the probability of a sample x belonging to class j as follows:

Algorithm 2: Multi-class Mahalanobis Distance-based LR (MMD (LR))

Input:
X_{train}, y_{train}: Training data and labels.
X_{test}: Test data.
Output:
y_{pred}: Predicted labels for X_{test}.

Procedure (MMD (LR)):
1. **MMD Parameter Estimation (Fit Phase):**
   1.1. Apply the MMD transformation (Algorithm 1, Steps 1-4) using the computed parameters (μ_i, Σ_m) to convert X_{train} and X_{test} into MMD feature matrices, $\boldsymbol{D'}_{\boldsymbol{train}}$ and $\boldsymbol{D'}_{\boldsymbol{test}}$.
2. **Train Logistic Regression (LR) Classifier:**
   2.1. Train a standard LR classifier using $\boldsymbol{D'}_{\boldsymbol{train}}$ using the LBFGS optimizer.
3. **Predict Classification:**
   3.1. Use the trained LR model to classify $\boldsymbol{D'}_{\boldsymbol{test}}$, yielding the predicted labels y_{pred}.

$$P(y = j \mid \boldsymbol{x}) = \frac{\exp(\boldsymbol{w_j}^{\top}\boldsymbol{x} + b_j)}{\sum_{k=1}^{K} \exp(\boldsymbol{w_k}^{\top}\boldsymbol{x} + b_k)} \quad \text{for } j = 1, \ldots, K \tag{12}$$

This formulation yields a probability distribution across K classes for selecting the class with the highest probability as the model's prediction. To efficiently train the model, particularly with numerous features or classes, an optimization algorithm is necessary. The LBFGS (Limited-memory Broyden–Fletcher–Goldfarb–Shanno) is a commonly employed optimizer since it approximates the inverse of the Hessian matrix rather than computing it directly. This approximation enhances memory efficiency in high-dimensional feature spaces. The "lbfgs" solver is well-suited for multinomial logistic regression as it can manage large datasets and support simultaneous optimization over multiple classes. It often achieves faster convergence and greater stability compared to traditional gradient descent methods when the number of classes exceeds two. This study utilized the "lbfgs" solver for performance evaluation, with a detailed simulation study presented in the next section.

IV. Simulation and Results

Simulation setup: The MMD (LR) method was tested on standard datasets from the UCI repository. These datasets are also considered for the evaluation of models. The details of the datasets considered for study are presented in Table 1 with the details of the total number of instances, feature count, number of classes, the partitioning of data into training and test sets, and Imbalance Ratio (IR). Based on the IR values, datasets were grouped into three categories: balanced (IR $\leq$ 1.15), moderately imbalanced (1.15 < IR $\leq$ 3.5), or highly imbalanced (IR > 3.5). The simulation experiments were implemented using Python, utilizing the scikit-learn library for all machine learning tasks, including preprocessing, model training, and performance evaluation. Each dataset was pre-processed according to the procedure outlined in Section III, followed by a 75:25 stratified train-test split to ensure representative class distributions.

Table 1: Details of datasets considered for study

Dataset Name	Total Instances	Feature count	Number of Classes	Training Set Size	Testing Set Size	IR
Breast cancer [XXXIII]	569	30	2	426	143	1.68
Diabetes [XVII]	768	8	2	576	192	1.87
E-coli [XXII]	318	5	5	238	80	7.15
Glass [XV]	214	9	6	160	54	8.44
Iris [XIII]	150	4	3	112	38	1
Wine [I]	178	13	3	133	45	1.48
Yeast [XXI]	1484	8	10	1113	371	92.6

Performance metrics: The performance of the proposed MMD transformation when used with LR was evaluated by utilizing the values from the confusion matrix, specifically True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN). From these values, several metrics were calculated, such as precision, F1-score, and AUC. The formulas for each of these metrics are presented below, as referenced in (13) and (14) [X].

$$Precision = \frac{TP}{TP+FP} \tag{13}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{14}$$

Precision indicates the proportion of predicted positive cases that were actually positive. The F1-score, a balanced metric, represents the harmonic mean of precision and recall, providing a single measure of performance. Further, the area under the curve (AUC) is derived from the ROC curve, which plots the true positive rate ($TPR = \frac{TP}{TP+FN}$) and the false positive rate ($FPR = \frac{FP}{FP+TN}$) at different thresholds.

Results Analysis: Fig 3 depicts the performance of the MMD (LR) on seven imbalanced datasets considered for the study. It can be observed from the figure that the MMD (LR) demonstrated its strong and excellent ability to classify data on seven standard datasets. MMD (LR) observed perfect classification on the Iris dataset with a score of 100% on all measures. This highlights its potential for handling data that is well-organized. Similarly, very high scores were obtained on the Breast Cancer and Wine datasets, with accuracy levels of 98.60% and 97.78%, respectively. Their AUC values were also close to 100%. It indicates a very good ability to distinguish between classes. MMD (LR) also performed well on datasets with moderate complexity, such as E-coli and Glass, by achieving accuracy scores of 88.75% and 70.37%, respectively, in addition to high AUC values. This suggests that MMD (LR) can handle class imbalance effectively and generalizes well across different classes. Further, even on more difficult and imbalanced datasets like Diabetes and Yeast, MMD (LR) showed competitive performance where traditional methods often struggle. It achieved an accuracy value of 72.40% and 60.38%, and notably high AUC values of 81.80% and 83.72%, respectively (Table 2 and Fig 3). It can be observed that this high performance was achieved because the data points in the data sets are clearly separated after MMD transformation. Thus, LR is able to effectively carry out the classification task.

Table 2: Performance comparison of MMD (LR) and LMDL

S.no.	Dataset	Method	Precision	F-measure	AUC
1	Breast cancer	LMDL	93	92	91.064
2		MMD (LR)	**98.63**	**98.6**	**99.69**
3	Diabetes	LMDL	76	75	71.38
4		MMD (LR)	71.7	71.88	**81.8**
5	E-coli	LMDL	80	79	83.64
6		MMD (LR)	**89.2**	**88.73**	**97.78**
7	Glass	LMDL	67	68	100
8		MMD (LR)	**72.22**	**70.23**	89.75
9	Iris	LMDL	98	97	90.62
10		MMD (LR)	**100**	**100**	**100**
11	Wine	LMDL	72	71	67.35
12		MMD (LR)	**97.89**	**97.76**	**100**
13	Yeast	LMDL	57	55	50
14		MMD (LR)	**60.23**	**59.93**	**83.72**

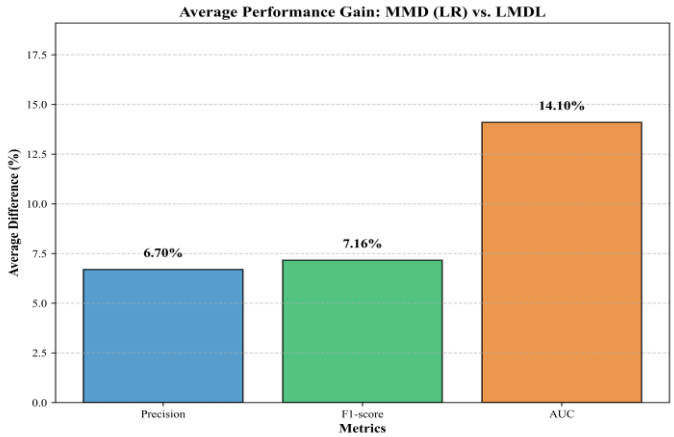
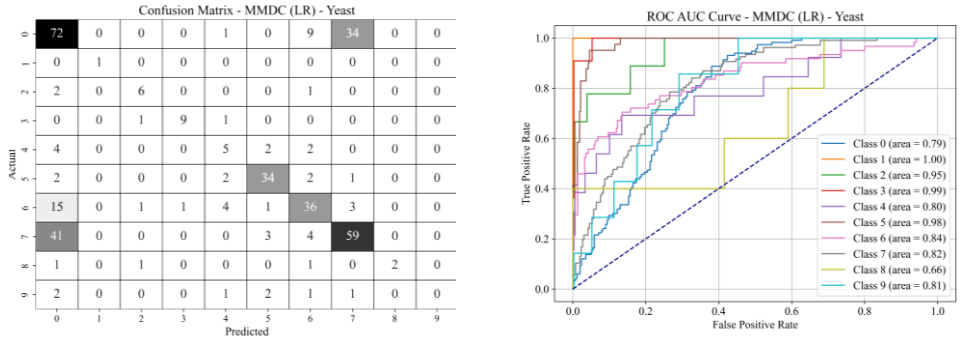


Fig. 3. Average performance gain MMD (LR) vs. LMDL across considered datasets.



(a) (b)

Fig 4. (a) Confusion matrix and (b) Class-wise ROC AUC of MMD (LR) on Yeast dataset

For the assessment of the model's resilience to extreme skewness, metrics of the minority class have been isolated. On the highly imbalanced yeast dataset (IR = 92.6), the MMD (LR) efficiently detected often ignored cases with 90% precision and 81.81% recall for class 3 and perfect 100 percent accuracy for extreme cases for classes 1 and 8 (Fig 4 (a) and 4 (b)). In addition, the performance increase is directly correlated to the imbalance ratio (IR) of the dataset. While low-imbalance datasets (IR < 2) showed the expected improvement in AUC, high-imbalance datasets such as E-coli (IR = 7.15) and Yeast (Table 1 and Fig 5) showed massive increases in AUC of 14.14% and 33.72%, respectively. (a) 5. This architectural advantage is due to the frequency-weighted covariance aggregation of the framework. Standard methods will struggle because inverting the matrix of a small sample size (e.g., <15 samples) exponentially increases the estimation errors and distorts the decision boundary. In MMD (LR), frequency weighting acts as a structural buffer; since minority covariances contribute only a small amount to the global pooled matrix Σ_m, local estimates do not corrupt the inverse covariance (Σ_m^{-1}). Therefore, instances are mapped using a stable and very reliable global metric, which avoids geometry distortions in extreme imbalances.

Performance comparison with existing work: The classification performance of the MMD (LR) was evaluated against the LMDL [XXVI] method across seven benchmark datasets. It can be observed from Table 2 that MMD (LR) significantly outperformed LMDL. On the Breast Cancer dataset, MMD (LR) performed much better than LMDL. Accuracy improved from 92.3% to 98.60%, and AUC increased significantly from 91.064% to 99.69%. This means MMD (LR) is good at classifying both the larger and smaller groups in the data. On the Diabetes dataset, LMDL had a slightly better AUC, i.e., 81.80% vs. 71.38%. This suggests that MMD (LR) is better at providing a balanced classification, especially when the classes are imbalanced. In the E-coli dataset, MMD (LR) did better than LMDL on all performance metrics. AUC improved greatly from

83.64% to 97.78%. The F-measure also showed improvements. This confirms that the MMD (LR) method is effective with unevenly distributed data.

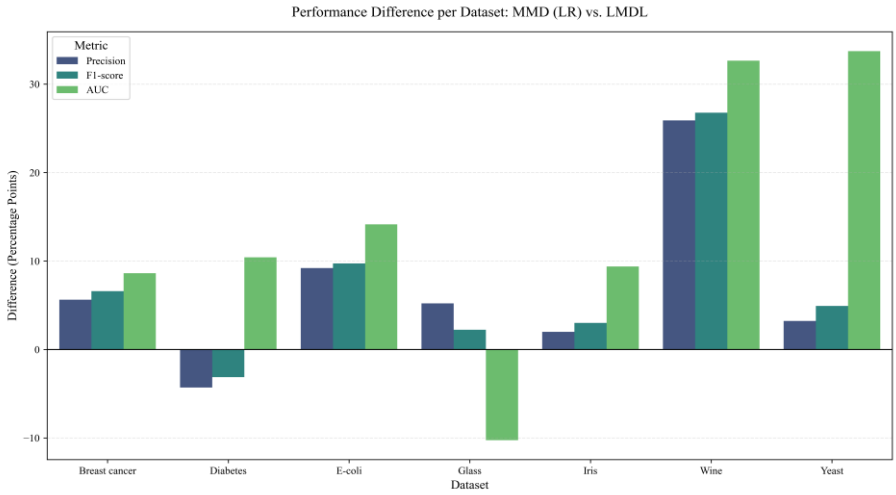


Fig. 5. Performance difference per dataset MMD (LR) vs. LMDL.

Table-3: Statistical significance test results for MMD (LR)

S.no.	Metric	N	W-Statistic	*p*-value	Decision
1	AUC	7	25	0.039063	Reject H0
2	F1-score	7	25	0.039063	Reject H0
3	Precision	7	25	0.039063	Reject H0

For the Glass dataset, MMD (LR) had better Precision and F-measure. LMDL had a perfect AUC of 100%, but MMD (LR)'s AUC of 89.75%, along with better performance on specific classes, which still indicates better predictive quality. The Iris dataset showed almost perfect performance for both methods, but MMD (LR) achieved 100% on all performance measures, with no classification errors. LMDL, while already good with 90.62 AUC, was not as perfect as MMD (LR). A significant improvement was seen on the Wine dataset, where MMD (LR) improved Accuracy from 71.11% to 97.78% and AUC from 67.35% to a perfect 100%. The large improvements in F1-score also supported the robustness of MMD (LR) for more complex classification tasks. Finally, on the Yeast dataset, which is known to have very imbalanced classes, MMD (LR) outperformed LMDL on all measures. The AUC increased significantly from 50% to 83.72%. This shows that MMD metric usage greatly improved the classifier's ability to distinguish between the classes. It can be concluded from the results that MMD (LR) consistently performed better than the baseline LMDL on most datasets and performance measures. The improvements are especially noticeable on datasets where the classes are imbalanced, which is a situation where traditional methods often struggle to perform well on all classes. However, it can be seen from Fig 3 that MMD (LR) has shown an average difference in improvement of 6.7%, 7.16%, and 14.1% in precision, f1-score, and AUC, compared to LMDL. But when the results per dataset are seen (see Fig 5), MMD (LR) is found to be inferior to LMDL in precision and f1-

score on the Diabetes dataset and in AUC on the Glass dataset. To this end, to evaluate the MMD (LR) is broadly significant to apply on imbalanced datasets, a statistical significance test was conducted.

Statistical significance test: The Wilcoxon signed-rank test [XXIX] was conducted for statistical validation of the results of MMD (LR) vs. LMDL. This test is a non-parametric statistical test frequently used to compare two related paired samples. It can be used as a non-parametric alternative to the paired t-test. This test can be used when the difference between paired observations is not normally distributed or when the data is ordinal [XX]. This test can be conducted when paired data (before and after measurement) or two models are compared on the same dataset. This paired data difference, in many cases, cannot guarantee normal distribution. Hence, to make the test robust to outliers and avoid the strict assumptions of a parametric test, the goal is to test if the median of differences is zero or not. Based on this, the hypothesis formulation is as follows:

H0 (Null): The median of differences is zero (i.e., no significant difference)
H1 (Alternate):

- Two-sided: Median difference $\neq 0$
- One-sided: Median difference > 0
- One-sided: Median difference < 0

The statistical difference is determined using the p-value. If the p-value is below a chosen significance level (e.g., 0.05), the result is considered statistically significant, leading to rejection of the null hypothesis. Conversely, if the p-value is greater than 0.05, the result is considered statistically insignificant, providing strong evidence for the null hypothesis. To validate the performance of MMD-LR against LMDL, a one-sided Wilcoxon signed-rank test (N=7) was conducted across performance metrics, Precision, F1-score, and AUC. Table 3 shows the test results. The test revealed that MMD-LR achieved a statistically significant improvement over the LMDL in terms of AUC, Precision, and F1-score ($p < 0.05$). These improvements in AUC, F1-score, and Precision were found marginally non-significant at the 5% level of significance. The statistical test results show that the MMD-based method is robust, adaptable, and generalizes well for supervised classification tasks.

V. Conclusions and Future Work

This study addressed the challenge of imbalanced data classification by proposing the MMD (LR) framework. This approach combines data pre-processing, Multi-class Mahalanobis Distance metric learning, and logistic regression to enhance class separability and reduce classifier bias toward majority classes. MMD(LR) was evaluated on seven UCI benchmark datasets. MMD (LR) significantly outperformed the baseline LMDL method. The model achieved an average performance gain of 6.70% in precision, 7.16% in F1-score, and a remarkable 14.10% increase in Area Under the Curve (AUC). MMD (LR) reached a perfect 100% AUC on the Iris and Wine datasets, and improved the AUC on the highly imbalanced Yeast dataset from 50% to 83.72%. Statistical significance tests further confirmed the robustness of these improvements. Ultimately, the MMD (LR) method provides a highly effective, adaptable, and interpretable solution for complex imbalanced classification tasks.

Despite these significant performance increases, the study has some theoretical limitations in terms of the geometric properties of the MMD transformation. First, while the transformation improves overall linear separation, it maps data purely on the basis of weighted-pooled covariance and centroid distances. As a result, complex local neighborhood topologies and non-linear manifold structures that were present in the original high-dimensional space may not be fully preserved. Moreover, the precise quantification of information loss by means of a mutual information or non-linear reconstruction analysis is not measured in the present framework. In future work, we will focus on integrating topological data analysis (TDA) and regularization of local neighborhood integrity into MMD to explicitly bound information loss and to ensure that local minority geometry is preserved under extreme skewness.

Conflict of Interest:

There was no relevant conflict of interest regarding this paper.

References

I. Aeberhard, S., and M. Forina. *Wine Dataset*. UCI Machine Learning Repository, 1992. 10.24432/C5PC7J.

II. Altalhan, M., A. Algarni, and M. Turki-Hadj Alouane. "Imbalanced Data Problem in Machine Learning: A Review." *IEEE Access*, 2025. 10.1109/access.2025.3531662.

III. Shrivastava, A., and P. R. Vamsi. "Improving Anomaly Classification using Combined Data Transformation and Machine Learning Methods." *International Journal of Performability Engineering*, vol. 20, 2024, p. 68. 10.23940/ijpe.24.02.p2.6880.

IV. Ashoorzadeh, A., A. T. Eshlaghy, and M. A. Kazemi. "A Novel Classification Method: A Hybrid Approach Based on Large Margin Nearest Neighbor Classifier." *Journal of Computational Robotics*, vol. 17, 2024, pp. 17–34.

V. Blum, L., M. Elgendi, and C. Menon. "Impact of Box-Cox Transformation on Machine-Learning Algorithms." *Frontiers in Artificial Intelligence*, vol. 5, 2022. 10.3389/frai.2022.877569.

VI. Boudt, K., P. J. Rousseeuw, S. Vanduffel, and T. Verdonck. "The Minimum Regularized Covariance Determinant Estimator." *Statistics and Computing*, vol. 30, 2019, pp. 113–128. 10.1007/s11222-019-09869-x.

VII. Chen, Y., A. Wiesel, Y. C. Eldar, and A. O. Hero. "Shrinkage Algorithms for MMSE Covariance Estimation." vol. 58, 2010, pp. 5016–5029. 10.1109/TSP.2010.2053029.

VIII. Dai, D., J. Pan, and Y. Liang. "Regularized Estimation of the Mahalanobis Distance Based on Modified Cholesky Decomposition." vol. 8, 2022, pp. 559–573. 10.1080/23737484.2022.2107961.

IX. de Amorim, L. B., G. D. Cavalcanti, and R. M. Cruz. "The Choice of Scaling Technique Matters for Classification Performance." *Applied Soft Computing*, vol. 133, 2023, article 109924.

X. de la Cruz Huayanay, A., J. L. Bazán, and C. M. Russo. "Performance of Evaluation Metrics for Classification in Imbalanced Data." *Computational Statistics*, vol. 40, 2025, pp. 1447–1473. 10.1007/s00180-024-01539-5.

XI. de Vazelhes, W., C. J. Carey, Y. Tang, N. Vauquier, and A. Bellet. "metric-learn: Metric Learning Algorithms in Python." *Journal of Machine Learning Research*, vol. 21, 2020, pp. 1–6.

XII. Elmi, J., M. Eftekhari, A. Mehrpooya, and M. R. Ravari. "A Novel Framework Based on the Multi Label Classification for Dynamic Selection of Classifiers." *International Journal of Machine Learning and Cybernetics*, vol. 14, 2023, pp. 2137–2154. 10.1007/s13042-022-01751-z.

XIII. Fisher, R. A. *Iris Dataset*. UCI Machine Learning Repository, 1936. 10.24432/C56C76.

XIV. Gao, X., D. Xie, Y. Zhang, et al. "A Comprehensive Survey on Imbalanced Data Learning." *arXiv*, 2025, arXiv:2502.08960.

XV. German, B. *Glass Identification Dataset*. UCI Machine Learning Repository, 1987. 10.24432/C5WW2P.

XVI. Kahn, M. *Diabetes Dataset*. UCI Machine Learning Repository, 1990, 10.24432/C5T59G.

XVII. Kamoi, R., and K. Kobayashi. "Why is the Mahalanobis Distance Effective for Anomaly Detection?" *arXiv*, 2020.

XVIII. Nakai, K. *Yeast Dataset*. UCI Machine Learning Repository, 1991. 10.24432/C5KG68.

XIX. Nakai, K. *Ecoli Dataset*. UCI Machine Learning Repository, 1996. 10.24432/C5388M.

XX. Qiao, S., et al. "LMNNB: Two-in-One Imbalanced Classification Approach by Combining Metric Learning and Ensemble Learning." 10.1007/s10489-021-02901-6.

XXI. Safi, S. K., and S. Gul. "An Enhanced Tree Ensemble for Classification in the Presence of Extreme Class Imbalance." *Mathematics*, vol. 12, 2024, article 3243. 10.3390/math12203243.

XXII. Saran, N. A., and F. Nar. "Fast Binary Logistic Regression." *PeerJ Computer Science*, vol. 11, 2025, e2579. 10.7717/peerj-cs.2579.

XXIII. Siddappa, N. G., and T. Kampalappa. "Imbalance Data Classification Using Local Mahalanobis Distance Learning Based on Nearest Neighbor." *SN Computer Science*, vol. 1, 2020, article 76. 10.1007/s42979-020-0085-x.

XXIV. Suárez, J. L., S. García, and F. Herrera. "A Tutorial on Distance Metric Learning: Mathematical Foundations, Algorithms, Experimental Analysis, Prospects and Challenges." *Neurocomputing*, vol. 425, 2021, pp. 300–322. 10.1016/j.neucom.2020.08.017.

XXV. Sun, Y., A. K. Wong, and M. S. Kamel. "Classification of Imbalanced Data: A Review." *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, 2009, pp. 687–719. 10.1142/S0218001409007326.

XXVI. Thakkar, B. "Continuous Variable Analyses: Student's t-test, Mann–Whitney U Test, Wilcoxon Signed-Rank Test." *Translational Cardiology*, 2025, pp. 165–167. 10.1016/B978-0-323-91790-2.00052-6.

XXVII. Wang, L., M. Han, X. Li, N. Zhang, and H. Cheng. "Review of Classification Methods on Unbalanced Data Sets." *IEEE Access*, vol. 9, 2021, pp. 62719–62744. 10.1109/ACCESS.2021.3074243.

XXVIII. Wilson, E. B. "The Distribution of Chi-Square." *Proceedings of the National Academy of Sciences*, vol. 17, no. 12, 1931, pp. 684–688.

XXIX. Wolberg, W. *Breast Cancer Wisconsin (Original) Dataset*. UCI Machine Learning Repository, 1990. 10.24432/C5HP4Z.

XXX. Zhu, B., X. Jing, L. Qiu, and R. Li. "An Imbalanced Data Classification Method Based on Hybrid Resampling and Fine Cost Sensitive Support Vector Machine." *Computers, Materials & Continua*, vol. 79, 2024, pp. 3979–3997. 10.32604/cmc.2024.048062.

XXXI. Hu, Lianyu, et al. "Interpretable Clustering: A Survey." *ACM Computing Surveys*, vol. 58, no. 8, 2026, pp. 1–21. 10.1145/3789495.