Check for updates

Statistical Analysis and Data Mining: An ASA Data Science Journal

WILEY

RESEARCH ARTICLE OPEN ACCESS

An Integrated Gradient Boosting Machine Approach With Fisher's Discriminant Analysis: Empirical Assessment on Five Benchmark Datasets and Simulations

Giulio Enzo Donninelli[1] | Pietro Giorgio Lovaglio[1,2]

[1]Department of Statistics and Quantitative Methods, University of Milano-Bicocca, Milan, Italy | [2]CRISP Research Center, University of Milano-Bicocca, Milan, Italy

Correspondence: Pietro Giorgio Lovaglio (piergiorgio.lovaglio@unimib.it)

Received: 18 October 2025 | **Revised:** 16 September 2026 | **Accepted:** 18 September 2026

ABSTRACT

The aim of this study is to assess the impact of different dimensionality reduction techniques—specifically Principal Component Analysis (PCA) and Fisher's Linear Discriminant Analysis (LDA)—on the classification performance of Gradient Boosting Machines (GBM). While PCA is commonly employed to reduce feature dimensionality, often to speed up training or lower memory usage, its application is not standard in boosting pipelines. Moreover, to the best of our knowledge, no previous work has systematically explored whether LDA can enhance performance or is compatible with gradient boosting frameworks. In addition to evaluating the effects of PCA and LDA on the predictive performance of GBM, we introduce a novel integrated iterative algorithm that incorporates LDA-based feature transformation at each boosting step. Our analysis spans five classical benchmark datasets, representing diverse scenarios in terms of sample size, number of features, for both binary and multiclass targets. Results show that applying LDA to extract features—and especially using the proposed LDA-integrated GBM—improves classification accuracy compared to standard GBM, particularly in multiclass settings or in binary problems with high feature dimensionality. Finally, through a set of controlled simulations, we investigate how the performance of the integrated approach evolves with increasing feature dimensionality and varying levels of feature correlation and sample size.

1 | Introduction

Real-world datasets often exhibit high dimensionality, which can hinder the performance of machine learning models. A common strategy to address this challenge is dimensionality reduction [1]. This involves transforming high-dimensional data into a more compact and informative representation, ideally one that reflects the data's intrinsic dimensionality [2, 3].

Dimensionality reduction plays a critical role in machine learning by addressing the high dimensionality present in real-world data, thereby improving computational efficiency. By extracting the most informative features, it enables faster and more scalable algorithms—an essential advantage in real-time or resource-constrained applications. Beyond computational gains, dimensionality reduction also mitigates the effects of the curse of dimensionality, which can degrade model performance [4]. For example, classification trees are prone to overfitting when trained on data with many irrelevant features, often resulting in higher misclassification rates on test sets. In practice, dimensionality reduction not only reduces these risks but also facilitates the compression of high-dimensional data, ultimately supporting more accurate and efficient classification [5].

Among the various data reduction techniques, feature extraction (FE) focuses on transforming the original feature

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2026 The Author(s). Statistical Analysis and Data Mining published by Wiley Periodicals LLC.

space to generate new variables that form a more compact, lower-dimensional representation of the problem. Unlike unsupervised FE methods, supervised FE techniques aim not only to reduce dimensionality, but also to preserve class separability as much as possible [6]. Common examples of unsupervised linear feature extraction methods include Principal Component Analysis (PCA) and Independent Component Analysis (ICA), which do not take class labels into account. In contrast, supervised methods such as Fisher's Linear Discriminant Analysis (LDA) and Partial Least Squares (PLS) explicitly incorporate class information to enhance discriminative power [7].

Extending the first family of methods, manifold-based learning techniques aim to capture the nonlinear structure of the original data by uncovering its intrinsic local geometric properties. These approaches are based on the premise that the apparent high dimensionality of many datasets is often artificial [8]. Accordingly, they seek a lower-dimensional representation within a manifold—a topological space that is locally Euclidean—capable of preserving the underlying data structure. Widely used examples include Locally Linear Embedding (LLE) [9], Isometric Mapping (ISOMAP) [10], and Laplacian Eigenmaps (LE) [11]. Another nonlinear approach that has gained popularity in recent years is the Autoencoder, which functions as an unsupervised multilayer perceptron for dimensionality reduction. However, a major limitation of manifold learning methods is the lack of an explicit projection function, making them unable to handle new, unseen samples—commonly referred to as the out-of-sample problem [12]. This practical drawback helps explain the continued widespread use of linear feature extraction methods such as PCA, which offer both simplicity and robustness in real-world applications [13, 14]. A key limitation of PCA is that it identifies directions of maximum variance in the input space without regard to the target variable. As a result, the principal components may lack any discriminative power for classification tasks. Consistent with this, the literature has shown that supervised feature extraction methods generally outperform their unsupervised counterparts. Among these, LDA is perhaps the most widely used supervised technique and has been shown to yield superior performance compared to PCA across a range of machine learning algorithms. This advantage stems from LDA's objective: it seeks linear combinations of the original predictors that maximize the separation between class means while minimizing within-class variance. However, LDA comes with an inherent constraint: the number of extracted components is limited to $C-1$, where C is the number of classes. This is because LDA constructs a feature space optimized for discriminating among classes and thus only requires at most $C-1$ dimensions to achieve optimal class separation. As a result, LDA can be viewed as an aggressive form of dimensionality reduction, compressing the original p-dimensional predictor space into a space of at most $C-1$ dimensions. In the extreme case of a binary classification problem ($C=2$), LDA produces only a single discriminant component.

In empirical studies comparing different feature extraction (FE) strategies, the classifiers used are often limited to traditional models such as logistic regression, classification trees, k-nearest neighbors (KNN), or Naïve Bayes. However, these models are rarely applied in real-world classification tasks due to their relatively limited performance compared to ensemble methods like Random Forest (RF) or Gradient Boosting Machines (GBM) [15, 16]. GBM is an iterative algorithm that combines multiple weak learners—typically shallow classification trees—into a strong predictive model. Each weak learner is trained to minimize the loss function of the previous ensemble using gradient descent. The learners are built sequentially, with each iteration aiming to model the residuals (errors) of the previous stage. This process continues iteratively, where each new learner corrects the mistakes of its predecessors, progressively improving overall model accuracy.

Extreme Gradient Boosting (XGBoost, Chen and Guestrin [15]) is a widely used implementation of GBM, known for its speed and strong performance in both applied settings and machine learning competitions. XGBoost incorporates regularization techniques and additional strategies—such as random selection of features for each weak learner—to mitigate overfitting. However, studies on preprocessing for GBM and XGBoost remain limited, typically focusing on standard procedures such as missing value imputation, outlier handling, and managing high collinearity or near-zero variance in input features [15, 17].

Linear Programming Boosting (LPBoost; [18]) and FilterBoost [19] are two variants of classical boosting that primarily modify how weak learners are selected and weighted. LPBoost reformulates boosting as a global optimization problem, closely related to margin-based methods such as support vector machines, with the goal of finding an optimal and often sparse combination of weak learners rather than adding them in a purely greedy, sequential manner as in Gradient Boosting Machines (GBM). FilterBoost, by contrast, is designed with scalability in mind: it introduces a filtering or sampling mechanism that allows the algorithm to operate on subsets of the training data at each iteration, thereby reducing computational cost while maintaining predictive performance. In both cases, however, the representation of the data remains unchanged. The algorithms operate directly on the original input features, without introducing new feature transformations beyond standard preprocessing steps. The key differences with respect to GBM therefore lie in the learning strategy rather than in the feature space.

Boosting with Kernel Projections [20] introduces a more substantial departure from the classical framework by incorporating a transformation of the input space prior to or within the boosting procedure. The central idea is to map the original features into a higher-dimensional or more expressive space through a kernel-based transformation, enabling the construction of weak learners that can capture more complex, nonlinear patterns. The boosting process then proceeds in this transformed space, iteratively combining weak learners to improve predictive accuracy on the target. Conceptually, this approach combines the representational power of kernel methods with the sequential error-correcting nature of boosting.

From the perspective of feature representation, this third approach differs fundamentally from both GBM and its variants such as LPBoost and FilterBoost. The transformed features are not simple linear combinations of the original inputs but arise from a kernel-induced mapping that may be implicit and high-dimensional. Importantly, this mapping is typically not constructed in a supervised manner with respect to the target; it is

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

defined a priori by the choice of kernel function. However, the boosting procedure introduces a form of indirect supervision by progressively selecting and emphasizing those components of the transformed space that are most useful for predicting the target. In this sense, feature construction occurs in two stages: an initial, largely unsupervised expansion of the feature space, followed by a supervised selection driven by the boosting iterations. More advanced variants further blur this distinction by adapting or combining kernels during training, thereby allowing the representation itself to become partially target-dependent. This dual mechanism distinguishes kernel-based boosting approaches from classical GBM, where both representation and learning are typically confined to the original feature space.

While not standard, the use of PCA prior to gradient boosting has not been extensively studied in the literature and is primarily mentioned in the context of machine learning competitions (such as the KDD Cup or Kaggle) as a heuristic to reduce feature dimensionality and, in some cases, to accelerate training or reduce memory consumption [21]. In contrast, to the best of our knowledge, no existing study has explored the use of LDA in conjunction with GBM. Specifically, there is no established practice or empirical evidence suggesting that LDA enhances performance or is compatible with gradient boosting models. In this respect, it is perhaps unsurprising that LDA has not been incorporated as a preprocessing step in widely used automated machine learning (AutoML) frameworks [22].

The aim of this paper is twofold. First, it investigates the impact of feature extraction via PCA and LDA on the classification accuracy of the GBM algorithm. Second, in contrast to common approaches that apply machine learning models to a fixed set of preprocessed features, we propose an integrated iterative method that leverages LDA-based feature transformation at each stage of the GBM training process. This integration acknowledges the iterative nature of GBM, where the target variable is updated at each step as the residual of the previous weak learner. The resulting algorithm, which we refer to as LdaBoost, dynamically recalculates LDA projections throughout boosting iterations. The rationale behind this approach lies in the complementary strengths of LDA and GBM. While LDA projects data onto a lower-dimensional linear subspace that maximally separates the classes, the resulting projections serve as informative inputs for GBM, which constructs nonlinear decision boundaries through ensembles of shallow trees. For instance, in a three-class problem, LDA maps observations into a two-dimensional space where the classes form compact and well-separated clusters—structures that GBM can efficiently capture using combinations of simple, rectangular tree splits. This synergy is particularly beneficial in complex classification tasks where linear separability is insufficient on its own. To address both research objectives, we conduct a series of experiments on several classical benchmark datasets that cover a variety of scenarios in terms of sample size, feature dimensionality, and target type (binary and multiclass). We compare the classification performance of standard GBM with and without dimensionality reduction, applying PCA, LDA, and the proposed LdaBoost algorithm. The results show that incorporating LDA-based features substantially enhances GBM accuracy, with the integrated LdaBoost method yielding the most consistent improvements over the baseline. In addition, we explore how the performance of LdaBoost varies with key data characteristics, specifically the number of observations, input features and their average correlation. Using simulations, we examine these effects systematically, offering insights into the conditions under which LdaBoost is most beneficial also in comparison with the optimal Bayes classifier. To the best of our knowledge, this is the first study to investigate the integration of LDA within a gradient boosting framework.

The remainder of the paper is structured as follows: Section 2 reviews related work on feature extraction in machine learning, with a focus on PCA and LDA. Section 3 introduces the LdaBoost algorithm in detail. Section 4 describes the analysis framework and datasets used. Section 5 presents the main empirical findings, including results from the simulation study. Finally, Section 6 concludes the paper by summarizing the contributions and outlining directions for future research.

2 | Related Work

Many authors have attempted to evaluate how feature extraction methods perform in classification tasks. Zebari et al. [14] provide an excellent review in this regard. The authors reviewed 21 feature extraction methods from the literature. Among them, eight methods relied on PCA, while seven were based on Deep Neural Network (DNN) and Convolutional Neural Network (CNN) tools, some of which were integrated with autoencoders or PCA. Other approaches aimed at dimensionality reduction used specific techniques for extracting features from machinery signals, such as Fourier transform frequency spectrum, envelope analysis, and local mean decomposition. Additionally, methods specific to image analysis adopted texture feature extraction and wavelet transformations.

Most of the reviewed approaches are unsupervised FE methods, with some exceptions that integrate PCA or autoencoders with supervised DNN and CNN architectures used as wrapper feature extraction tools. Specifically, both semi-supervised algorithms select the most optimal features (neurons from hidden layers or image features from convolutional layers) based on the performance of the learning algorithm aimed at predicting the target variable. Here, PCA was not used to extract the initial features for DNN and CNN but rather to reduce the dimensionality of the extracted features (neurons) from both neural network tools. The findings demonstrated the efficacy of semi-supervised dimensionality reduction techniques in improving the performance of machine learning algorithms compared to unsupervised tools (essentially PCA). However, no study has used LDA as a feature extraction method.

In the above-cited review, the SVM algorithm was the most frequently used classifier in conjunction with feature extraction, followed by neural networks, KNN, and LDA. The conclusion is that optimized PCA (within a semi-supervised approach) achieves better performance in terms of accuracy, computational time, and the number of reduced features [14, 23]. Apart from a few exceptions, however, classifiers used in the literature rarely include tree-based ensembles such as Random Forest (RF) or Bagging, and none of them incorporate GBM.

As mentioned, only a few published studies have adopted LDA as a feature extraction method. Subasi [24] classified epileptic

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

seizures using EEG signals and an SVM classifier. The classification rate with LDA feature extraction was the highest (100%), followed by ICA (99.5%), while PCA achieved the lowest correct classification rate (98.75%). All FE strategies performed better than classification without feature extraction (98%). Moreover, the proposed FE strategies outperformed the overall classification accuracy obtained using a Mixture of Experts model (94.5%) and an Artificial Neural Network (93.2%).

Leo et al. [25], using a chemical sensor array dataset, fitted an SVM classifier with dimensionality reduction methods PCA and LDA. The authors demonstrated that LDA better separated the data to be classified, resulting in higher accuracy.

Data from Human Activity Recognition (HAR), aimed at identifying human activity based on attributes measured by a smartphone's accelerometer and gyroscope, were also used to assess machine learning tools in conjunction with feature preprocessing. In this field, using the Human Activity Recognition Using Smartphones Data Set (UCI Machine Learning Repository), Luštrek and Kaluža [26] evaluated various algorithms (including C4.5 decision trees) using cross-validated classification accuracy.

Cann [27], applying the C4.5 classification decision tree algorithm with LDA, found that such feature preprocessing increased accuracy by 1.61% compared to using the original inputs.

Pechenizkiy et al. [28] proposed a novel feature extraction method combining two steps: first extracting principal components and then deriving discriminant components from the resulting PCA feature space. However, this combined approach introduces some arbitrariness, as the discriminant components (scores) depend on the initial number of extracted principal components. Yang [29] recommends using more PCA dimensions to improve the performance of discriminant components in the second step.

These authors assessed KNN, Naïve Bayes, and C4.5 decision tree algorithms on 21 datasets. Although specific improvements depended on the dataset, combining LDA and PCA generally showed an overall increase in accuracy compared to using PCA or LDA alone, though the improvement over LDA was less pronounced. Li [30] applied the same combined approach for face recognition using the nearest neighbor classifier and observed improved results compared to PCA alone.

Liu et al. [31], analyzing numerous public image datasets, developed a method for feature extraction based on combining discriminant analysis with the low-rank representation (LRR) of the original data samples. In particular, LDA was used as a discriminative constraint term within the dimensionality reduction loss function, which was minimized with respect to the lowest-rank representation of the original data. A projection matrix was simultaneously learned for dimensionality reduction and class separation. The global subspace structure of the original samples was first captured by LRR, and then the discriminative constraint term was used to maximize the between-class distances and minimize the within-class distances in the low-dimensional projection space.

The authors demonstrated that the proposed strategy achieved better recognition accuracy compared to state-of-the-art feature extraction methods. Although the proposed supervised FE method was specifically designed for image processing, this approach illustrates that integrating discriminant analysis with supervised classifiers can be beneficial in terms of predictive accuracy.

3 | Methods

Given N training data $\big(x_1, y_1(c)\big), \ldots, \big(x_N, y_N(c)\big)$, where $x_i \in \mathbb{R}^p$ are the normalized original features and $y_i(c)$ are the labels of the $c = 1, \ldots, C$ target classes.

PCA maps the feature space $x_i \in \mathbb{R}^p$ to a lower-dimensional space $t_i \in \mathbb{R}^r$ $(r \leq p)$ through a linear transformation:

$$t_i = P^\top x_i$$

(Principal component scores), where these new r variables are uncorrelated.

The PCA criterion aims to maximize the sum of variances (trace) of the retained r principal component scores, for example, $\text{trace}\big(P^\top SP\big)$, where $P^\top SP$ is the covariance matrix of the principal component scores and S is the covariance matrix of the original features $S = X^\top X$. This is achieved by selecting the r largest eigenvalues of S, and the corresponding eigenvectors $P_1, \ldots, P_r$ form the optimal transformation matrix:

$$P = \big[P_1 \mid \ldots \mid P_r\big].$$

LDA maps the feature space $x_i \in \mathbb{R}^p$ to a lower-dimensional space $q_i \in \mathbb{R}^r$ $(r < p)$ through a linear transformation:

$$q_i = G^\top x_i.$$

Discriminant scores are weighted linear combinations of the predictors. The weights are estimated to maximize the differences between class mean discriminant scores and are obtained as generalized eigenvectors that maximize the ratio of the between-class sum of squares (S_B) to the within-class sum of squares (S_W). The former evaluates the spread of the class means (μ_c) around the global mean (μ), whereas the latter evaluates the spread of the data (x_i) around the class mean (μ_c).

For transformed data, the (transformed) within-class and between-class sums of squares are defined respectively as $G^\top S_W G$ and $G^\top S_B G$. The LDA optimization problem requires minimizing the transformed within-class scatter while maximizing the transformed between-class scatter. The optimal transformation G can be obtained by maximizing the Fisher criterion:

$$\frac{\mid G^\top S_B G \mid}{\mid G^\top S_W G \mid}$$

by selecting the r largest eigenvalues of $S_W^{-1} S_B$ and the corresponding eigenvectors $G_1, \ldots, G_r$, which form the optimal transformation matrix:

$$G = \big[G_1 \mid \ldots \mid G_r\big].$$

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

In LDA, feature weights depend not only on the separation between class means but also on the within-class variability. Features that show large differences between class means and low variance within each class receive higher weights, whereas features with similar class means or high within-class variability tend to have smaller weights.

Hence, LDA provides new N training data $(q_1, y_1(c))$, $\ldots, (q_N, y_N(c))$, where $q_i \in \mathbb{R}^r$ are the input r variables retained after LDA computation (discriminant components).

Concerning the Gradient Boosting classifier, we use the classical GBM [32] implemented in the Python library `GradientBoostingClassifier` within Scikit-learn.

GBM builds an additive model in a forward stage-wise fashion using classification trees. In each stage (`n_estimators`), trees are fit on the negative gradient of the loss function, which is either the binary log-loss (binomial deviance) or the multiclass log-loss (the multinomial negative log-likelihood loss function) for multi-class classification with `n_classes` mutually exclusive classes.

The GBM is an additive model whose prediction $\hat{y}_i$ for a given input x_i is defined as:

$$\hat{y}_i = F_M(x_i) = \sum_{m=1}^{M} h_m(x_i) \tag{1}$$

where h_m is estimated by the m-th tree, and M is the maximum number of trees used. GBM proceeds in a greedy fashion:

$$F_m(x) = F_{m-1}(x) + h_m(x),$$

where at iteration m, a newly added tree h_m is fitted to minimize a sum of losses L_m, given the previous ensemble $F_{m-1}(x)$:

$$h_m = \arg\min_h L_m = \arg\min_h \sum_{i=1}^{n} \ell\big(y_i, F_{m-1}(x_i) + h(x_i)\big). \tag{2}$$

Moreover, GBM allows a regularization strategy that scales the contribution of each weak learner by a constant factor ν (`learning_rate`):

$$F_m(x) = F_{m-1}(x) + \nu h_m(x).$$

At each iteration, the estimator $h_m(x)$ is fitted to predict the negative gradients of the samples.

The iterative minimization of the loss function is achieved by Gradient Descent, using all instances in each iteration or a subsample of observations for fitting the individual trees (Stochastic or Batch Gradient Descent).

The main tuning parameters for GBM are: the learning rate (`learning_rate`), the maximum number of trees (`n_estimators`), the maximum number of final nodes in each tree (`max_depth`), and the minimum number of observations in final nodes (`min_samples_leaf`).

Additionally, one could also tune the minibatch percentage, that is, the proportion of observations used in each stochastic gradient descent iteration (`subsample`), which we kept fixed at the default value of 0.6.

3.1 | An Integrated LdaBoosting Algorithm

The key insight behind integrating LDA within the GBM algorithm — referred to as **LdaBoost** — is to dynamically leverage the discriminative power of LDA at each iteration of the boosting procedure. Traditionally, LDA is employed as a pre-processing step to reduce feature dimensionality by projecting the original feature space onto a subspace optimized for class separability. However, this static approach neglects the iterative nature of GBM, wherein the target changes at each step as the model seeks to correct residual errors from the preceding weak learner.

In contrast, LdaBoost iteratively updates the feature projections using LDA based on the residuals produced at each boosting round, thereby continuously refining the feature space in alignment with the evolving predictions. Specifically, at each iteration, LdaBoost calculates class-specific residuals, determines a corresponding transformed feature space via LDA tailored explicitly to these residuals, and subsequently trains individual trees on this optimized subspace. This iterative integration not only capitalizes on LDA's capacity to enhance linear class separation but also synergistically complements GBM's nonlinear modeling strengths, resulting in improved classification accuracy.

The pseudocode, illustrated in Algorithm 1, summarizes the LdaBoost procedure, clarifying the step-by-step algorithmic implementation of this integrated approach, as follows:

- **Initialization**. One-hot encode labels Y (also for binary target) and estimate empirical class priors $\hat{\pi}_c$; initialize logits with $F_c^{(0)} = \log \hat{\pi}_c$ so boosting starts from the log prior odds shared by all samples.
- **Step 1 — Iteration-specific LDA transform**. At $m = 0$, fit LDA on true labels y to get projection $G^{(0)}$ and discriminant scores $Q^{(0)}$ (each $q_i = G^{(0)\top} x_i$) in an $r \le C - 1$ subspace that maximizes between-class separation. For $m > 0$, form current probabilities $P^{(m-1)}$, gradients $g^{(m)} = Y - P^{(m-1)}$, and induced labels $z_i^{(m)} = \arg\max_c g_{i,c}^{(m)}$ (where $g_{i,c}^{(m)}$ is the element for the c-th class of the i-th observation). Refit LDA on $(X, z^{(m)})$ to obtain the new projection matrix $G^{(m)}$; if $z^{(m)}$ is assigned to fewer than two distinct classes, reuse $G^{(m-1)}$.
- **Step 2 — Pseudo-residuals**. Recompute $g^{(m)} = Y - P^{(m-1)}$; these are the negative gradients of the multinomial deviance with respect to the logits and provide the working response for the boosted trees.
- **Step 3 — Class-wise weak learners**. Train one regression tree per class c to predict the corresponding gradient column $g_{\cdot,c}^{(m)}$ from the current discriminant scores $Q^{(m)}$; using the LDA subspace concentrates each tree on directions most informative for class separation at round m.
- **Step 4 — Stagewise update**. Update additive logits:

$$F_{i,c}^{(m)} = F_{i,c}^{(m-1)} + \nu\, h_{m,c}\left(Q_i^{(m)}\right),$$

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

ALGORITHM 1 | LdaBoost (Multiclass).

Input: Training data $(\mathbf{x}_i, y_i)_{i=1}^N$, classes $\{1, \ldots, C\}$, rounds M, learning rate ν
Output: Additive logits $F^{(M)}$ and per-round projections $\{G^{(m)}\}_{m=0}^{M-1}$
Init: One-hot $Y \in \{0,1\}^{N\times C}$, priors $\hat{\pi}_c$, $F_{i,c}^{(0)} \leftarrow \log \hat{\pi}_c$, $P^{(0)} = \mathrm{softmax}(F^{(0)})$. Fit $G^{(0)}$ via LDA on (X, y); set $Q^{(0)} = XG^{(0)}$.
for $m = 1$ **to** M **do**
  $g^{(m)} \leftarrow Y - P^{(m-1)}$; `// negative gradients of multinomial deviance`
  $z_i^{(m)} \leftarrow \arg\max_c g_{i,c}^{(m)}$ for each i;
  if $z^{(m)}$ *has at least two classes* **then**
    Refit $G^{(m)}$ via LDA on $(X, z^{(m)})$;
  else
    $G^{(m)} \leftarrow G^{(m-1)}$;
  $Q^{(m)} \leftarrow XG^{(m)}$;
  for $c = 1$ **to** C **do**
    Fit regression tree $h_{m,c} : Q^{(m)} \to g_{\cdot,c}^{(m)}$;
  $F_{\cdot,c}^{(m)} \leftarrow F_{\cdot,c}^{(m-1)} + \nu\, h_{m,c}(Q^{(m)})$ for all c;
  $P^{(m)} \leftarrow \mathrm{softmax}(F^{(m)})$.
Predict: Apply $\{G^{(m)}\}$ and $\{h_{m,c}\}$ sequentially to a new $\mathbf{x}$; return $\arg\max_c F_c^{(M)}$.

where the learning rate ν (shrinkage) scales each correction, as in standard gradient boosting.

- **Class prediction**. Start from $F^{(0)}$; for each stage, apply stored $G^{(m)}$ to a new x to get q, add $\nu\, h_{m,c}(q)$ to each class logit, then apply the softmax to obtain probabilities and choose the class with the maximum value.

In the Discussion section, we further justify the proposed algorithm and its underlying rationale, particularly the pseudo-labeling phase, by drawing on existing literature. We also suggest an alternative pseudo-labeling procedure for future studies.

3.1.1 | Computational Scalability

Let n be the number of samples, p the number of features, C the number of classes, and M the number of boosting rounds. A one-shot LDA + GBM pipeline fits LDA once and then trains M boosting stages on fixed transformed features. By contrast, LdaBoost recomputes LDA at each round.

Denoting by $C_{\mathrm{LDA}}(n, p, C)$ the cost of one LDA fit and by $C_{\mathrm{tree}}(n, C)$ the cost of one class-wise boosting stage, the overall costs are:

$$\text{LDA + GBM} : C_{\mathrm{LDA}}(n, p, C) + M\, C_{\mathrm{tree}}(n, C),$$
$$\text{LdaBoost} : M\, C_{\mathrm{LDA}}(n, p, C) + M\, C_{\mathrm{tree}}(n, C).$$

Thus, the additional overhead of LdaBoost is approximately $(M-1)C_{\mathrm{LDA}}(n, p, C)$ relative to one-shot LDA preprocessing. In practice, this trade-off is more visible in high-dimensional settings, where repeated scatter-matrix computations dominate runtime, and is much smaller when p is modest.

4 | Dataset and Empirical Strategy

From an empirical perspective, the assessment of algorithms (GBM in PCA/LDA framework) can be summarized as follows:

- **Step 1**. The features were normalized to have values in the range $[0, 1]$.
- **Step 2**. Features were extracted using the PCA (principal components) and LDA (discriminant components) algorithms to reduce dimensionality. For situations where both training and test sets are available, both datasets were transformed using PCA and LDA. For the test set, the principal and discriminant scores were calculated using the training weights.
- **Step 3**. The classification process for each dataset was carried out using GBM-based classification on:
  - normalized original features;
  - PCA-based features (**PCA + GBM**);
  - LDA-based features (**LDA + GBM**).
- **Step 4**. The proposed **LdaBoost** model was also fitted for each dataset.

4.1 | Datasets

The Human Activity Recognition Using Smartphones dataset (UCI Machine Learning Repository). The dataset has 6 balanced class labels representing the following conditions: 1 WALKING, 2 WALKING_UPSTAIRS, 3 WALKING_DOWNSTAIRS, 4 SITTING, 5 STANDING, 6 LAYING.

The dataset contains 10,299 observations and 561 features recorded from a waist-mounted smartphone with inertial sensors. The data from the UCI Machine Learning Repository are already normalized (within $[-1, 1]$) and reported in a 0–1 range.

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Data were split into training and test sets. Both test accuracy and cross-validated accuracy were evaluated.

Iris dataset (UCI Machine Learning Repository). The dataset consists of 50 samples from each of three species of Iris (*Iris setosa*, *Iris virginica*, and *Iris versicolor*). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters. Features were normalized. Data were not split, and the cross-validated accuracy was evaluated.

Sonar dataset (UCI Machine Learning Repository). The data contain patterns obtained by bouncing sonar signals off a metal cylinder at various angles and under various conditions. Each pattern is a set of numbers in the range 0.0 to 1.0. Each number represents the energy within a particular frequency band, integrated over a certain period of time. The label associated with each record is "R" if the object is a rock and "M" if it is a mine (metal cylinder). The data have 60 features and 208 observations. Features are already normalized. Data were not split, and the cross-validated accuracy was evaluated.

Yeast dataset (Kaggle Repository). The Yeast dataset consists of a protein–protein interaction network. The target has 10 unbalanced classes, indicating the localization site of the protein (e.g., cytosolic or cytoskeletal, nuclear, extracellular, etc.). The data contain 8 features and 1484 observations. Features were normalized. Data were not split, and the cross-validated accuracy was evaluated.

Rainfall Dataset (Kaggle Repository). This dataset collects a time series of 2919 observations (days) regarding weather conditions, meteorological variables, and rainfall. The target (rainfall) is binary, and there are 11 features summarizing atmospheric conditions (humidity, sunshine, wind direction, wind speed, etc.). Features were normalized. Data were split, and both test accuracy and cross-validated accuracy were evaluated.

4.2 | Tuning Strategy

Firstly, using normalized inputs, we fit a classical GBM with `GradientBoostingClassifier` of Scikit-Learn Python library. We tune, using 10-fold cross-validation, the following parameters: `max_depth`, `n_estimators`, `min_samples_leaf`, and `learning_rate`, for each method (free tuning). Specifically, for PCA + GBM, we also tune the optimal number of principal components (maximizing the cross-validated accuracy), whereas for both LDA + GBM and LdaBoost, we retain the maximum number of discriminant scores (#classes −1). Moreover, we use an alternative strategy as robustness check. To evaluate how the performance of different algorithms depends solely on the selected features, we fix the four hyperparameters of GBM at their optimal values and apply them unchanged to PCA + GBM, LDA + GBM, and LdaBoost. Results were described in the Appendix rather than in the main text, to avoid making the discussion too heavy and to allow us to focus more on the free-tuning approach.

For each dataset, algorithms were compared using 10-fold cross-validated accuracy, and for the HAR and Rainfall datasets, also by test accuracy (for HAR we use the test set available from UCI Machine Learning). For the HAR dataset, we also evaluate the macro F1 score as an accuracy measure for a more complete assessment. Lastly, we performed pairwise comparisons using the Wilcoxon rank-sum test with Holm correction for multiple testing to assess whether the distributions of cross-validated accuracy scores were statistically indistinguishable across methods, or whether the distribution associated with a given method tended to yield systematically higher values than that of another method.

4.3 | LdaBoost Simulations

To comprehensively evaluate the robustness and performance of the proposed LdaBoost algorithm, we conducted a series of simulations to explore how classification accuracy varied across different scenarios. Specifically, we systematically assessed the influence of three critical factors: the number of observations, the number of features and the degree of correlation among these features.

In these simulations, synthetic datasets of 1000 and 10,000 observations were generated using a multivariate normal distribution, $X \sim \mathcal{N}(\mu_y, \Sigma)$, with mean vectors μ_y set distinctly for each class and type of target (see below). The covariance matrix Σ was generated using a symmetric structure with diagonal elements equal to 1, and off-diagonal elements sampled from a truncated normal distribution with a mean of 0.5 and a standard deviation of 1, ensuring correlations within a realistic range between −1 and 1.

Additionally, correlation levels among features were manipulated through the covariance matrix, simulating conditions ranging from weak to strong correlations. Particularly, the mean correlation (ρ) of non-diagonal elements of Σ was considered, and representative values of $\rho = 0.0, 0.2, 0.5$ and 0.7 were simulated.

To construct the target variable Y, two distinct scenarios were considered:

- **Binary scenario**: Two distinct class-specific mean vectors, $\mu_0 = 0$ and $\mu_1 = 1$, were established. Observations were generated from multivariate densities (f) as follows:

$$f(X|Y=0) \sim \mathcal{N}(0, \Sigma), \quad f(X|Y=1) \sim \mathcal{N}(1, \Sigma)$$

  The class labels were determined based on which Gaussian distribution each observation originated from.

- **Multiclass scenario**: Multiple distinct mean vectors $\mu_c = c$ were carefully defined to maximize nonlinear separability. Observations were generated as:

$$f(X|Y=c) \sim \mathcal{N}(\mu_c, \Sigma), \quad \mu_c = c, \quad c = 1, 2, \ldots, C$$

  Each observation's class label was directly assigned according to the Gaussian distribution from which it was drawn. In simulation, for multiclass target we use both $C = 3$ and $C = 5$.

These simulated distributions provide a moderate to high class overlapping in the feature space, thus avoiding linear separability or clear class separation. Moreover, the use of normal likelihoods makes the Bayes classifier—based on

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

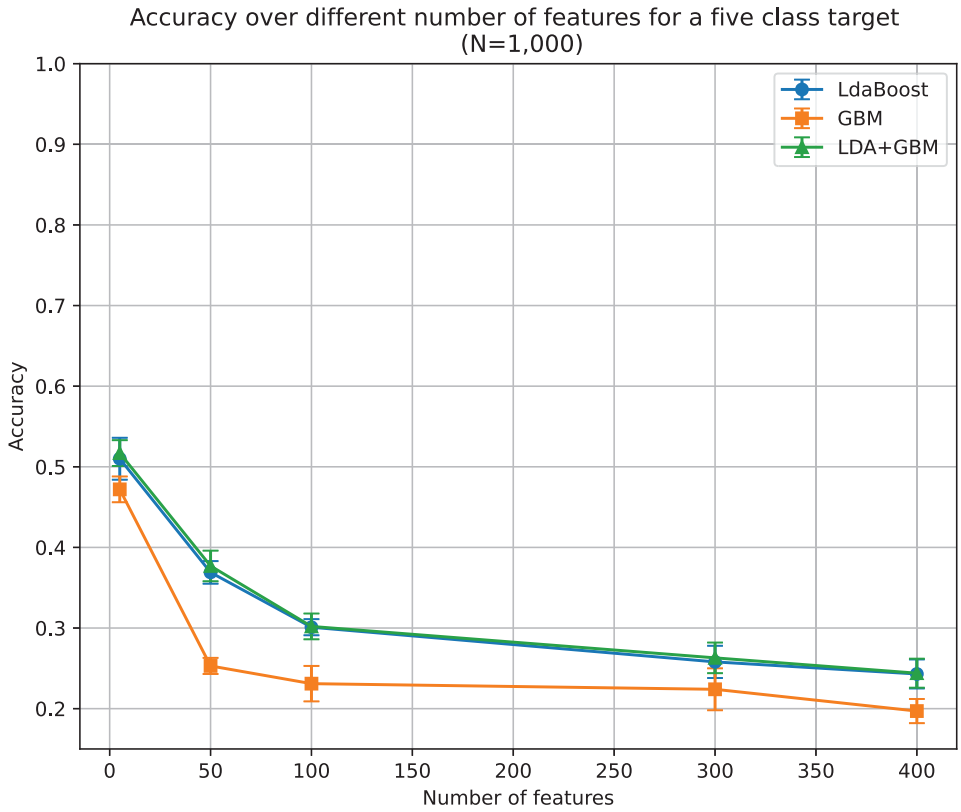


FIGURE 1 | Comparison of the different pipelines (multiclass target) in a medium-correlation scenario ($\rho \approx 0.5$) and smallest N.

Bayes' Theorem—computationally feasible. The Bayes classifier (assuming equal priors) was fitted to each simulated dataset, and its resulting accuracy—representing the theoretical upper bound or ideal performance achievable—was used as a benchmark when evaluating LdaBoost simulations. The number of features was systematically varied at distinct levels (5, 50, 100, 300, and 400) to represent scenarios ranging from low to very high dimensionality.

For each scenario (sample size, number of features and ρ) we generate random samples (realizations) from the assumed distribution in order to exploit Monte Carlo variability of accuracies (with standard deviation, SD). In the analysis of the real dataset, SD denotes the cross-validated standard deviation, while accuracy, unless otherwise specified, refers to cross-validated accuracy. Two complementary simulation campaigns were performed. The first compared GBM, PCA + GBM, LDA + GBM, and LdaBoost (Figures 1 and 2 and A1–A3). The second evaluates LdaBoost across scenarios, benchmarking it against the Bayes classifier (Figures 3–6). This design highlights how closely the proposed method can approach the theoretical upper bound across heterogeneous data-generating processes.

4.4 | Results

4.4.1 | HAR

In the HAR dataset (Table 1), 10-fold cross-validation shows that the baseline GBM reaches an accuracy of 0.995 (SD = 0.002) when using the full feature set. PCA + GBM drops to 0.946 (SD = 0.006), whereas LDA + GBM and LdaBoost—both relying on five discriminant components instead of the original 561 predictors—achieve 0.982 (SD = 0.004) and 0.978 (SD = 0.006), respectively.

On the held-out test set, LDA + GBM attains the highest accuracy (0.9528), closely followed by LdaBoost (0.9511), while GBM and PCA + GBM reach 0.9400 and 0.9053.

From a computational perspective, HAR highlights an enormous acceleration of one-shot LDA preprocessing in front of GBM: LDA + GBM needs 10.25 s versus 675.07 s for GBM and 652.62 s for PCA + GBM, corresponding to about 65.9× and 63.7× speedups, respectively. This gain comes with only a small cross-validated accuracy reduction versus full-feature GBM (0.978 vs. 0.995), while preserving better external validation accuracy (0.9528 vs. 0.9400).

Luštrek and Kaluža [26] assess eight algorithms (C4.5 decision trees, RIPPER decision rules, Naive Bayes, 3-Nearest Neighbors, Support Vector Machine (SVM), Random Forest, Bagging, and Adaboost M1 boosting) and found that the highest cross-validated classification accuracy is achieved by Support Vector Machine (0.977) and Adaboost (0.977). Both LDA + GBM and LdaBoost match or exceed that benchmark while using drastically fewer features, underscoring the benefit of integrating discriminant components within boosting frameworks.

Macro-F1 results mirror the accuracy rankings: GBM (F1 = 0.995, SD = 0.002), PCA + GBM (F1 = 0.949, SD = 0.009), LDA + GBM (F1 = 0.983, SD = 0.004), and LdaBoost (F1 = 0.980, SD = 0.006).

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

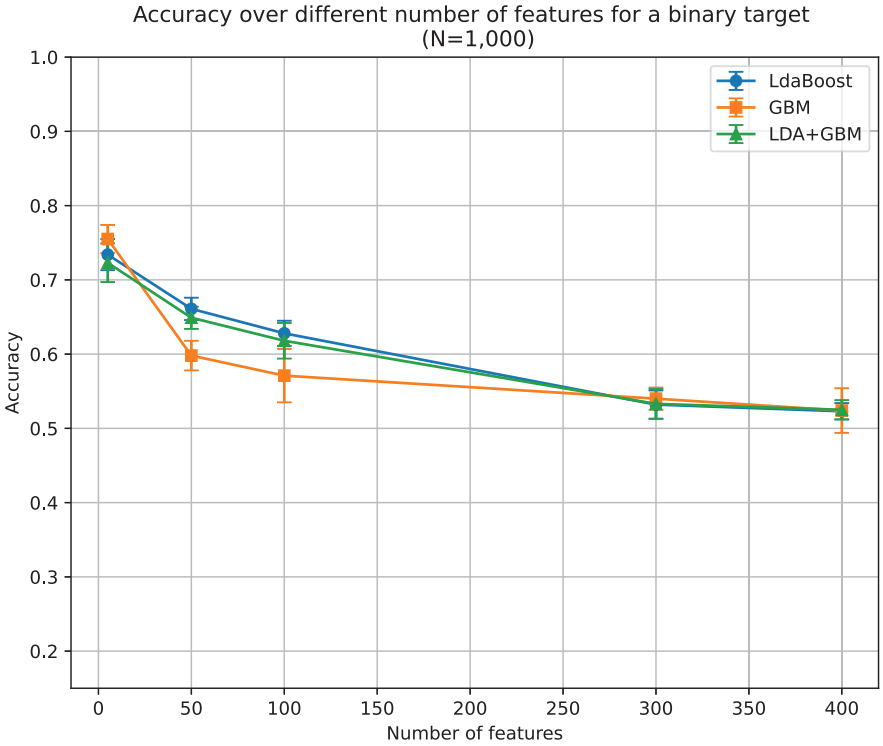


FIGURE 2 | Comparison of the different pipelines (binary target) in a medium-correlation scenario ($\rho \approx 0.5$) and smallest N.

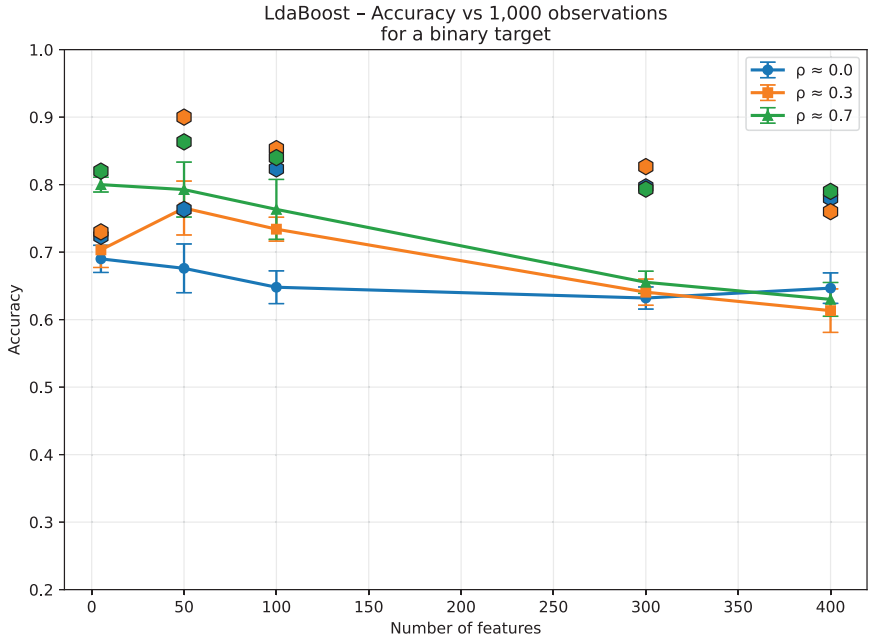


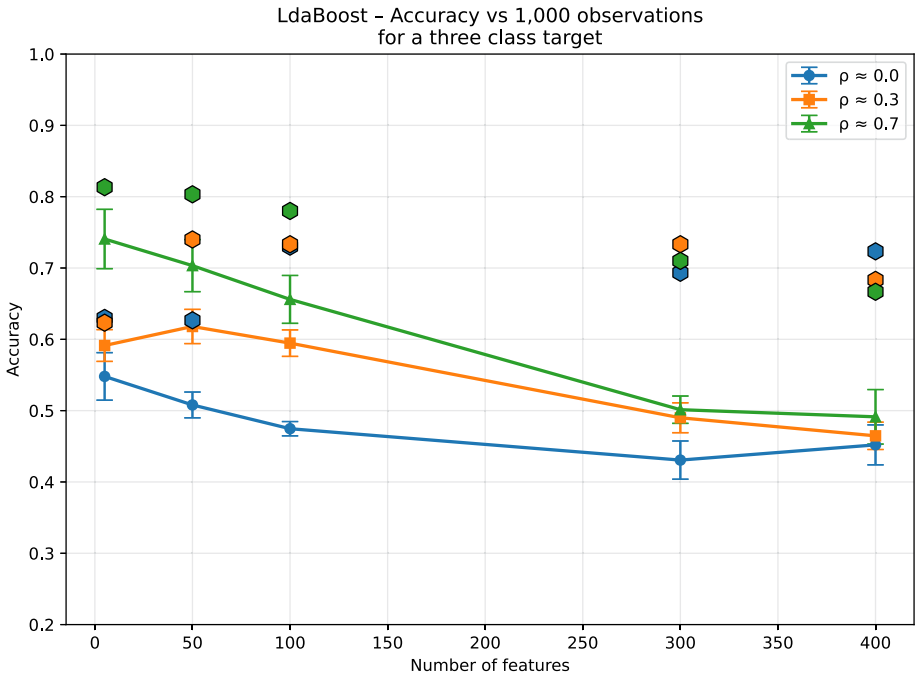
FIGURE 3 | LdaBoost accuracy (binary target) by correlation levels (ρ) and smallest N: Points indicate the accuracy of the Bayes classifier.

The HAR result is also informative as a failure mode for aggressive supervised compression: the baseline GBM retains all 561 engineered signals, while LDA-based pipelines are constrained to five discriminant components. On this dataset, reducing to $C - 1$ directions can discard nonlinear interaction structure that trees can exploit in the original space, which helps explain why full-feature GBM leads in cross-validated accuracy even though LDA-based models remain competitive on the held-out test set.

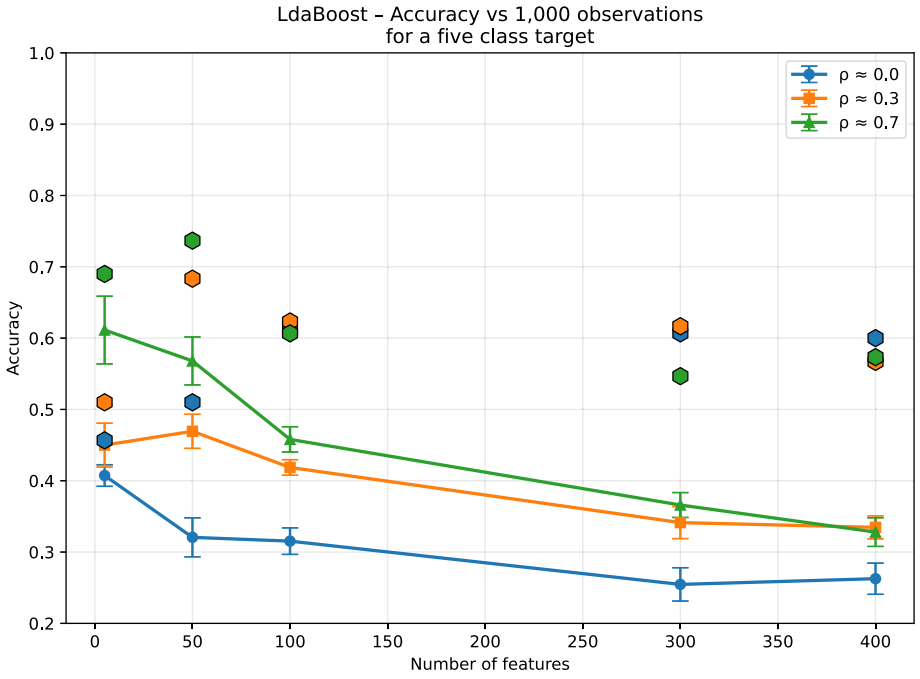
4.4.2 | Other Datasets

Table 1 provides a comprehensive overview of the classification accuracy of the various analyzed approaches across multiple datasets. Overall, LdaBoost or LDA + GBM deliver the best cross-validated performance in four of the five benchmarks, while the standard GBM retains a small advantage on HAR despite using the full set of 561 features.

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License



(a)



(b)

FIGURE 4 | LdaBoost accuracy (multiclass target) by correlation levels (ρ) and smallest N: 3 classes (top), 5 classes (bottom). Points indicate the accuracy of the Bayes classifier.

The Yeast dataset (8 features) showed substantial improvements, with cross-validated accuracy rising from 0.568 (SD = 0.037) for GBM to 0.614 (SD = 0.030) with LDA + GBM, while LdaBoost reached 0.589 (SD = 0.032) and PCA + GBM 0.582 (SD = 0.023). This confirms the benefit of incorporating discriminant components in multiclass problems even when the feature space is relatively low dimensional.

With a small number of features such as in the IRIS dataset, LDA + GBM delivered the highest accuracy (0.980, SD = 0.020). LdaBoost and PCA + GBM (both 0.967, SD = 0.031) and GBM (0.960, SD = 0.044) performed similarly.

For the Sonar dataset (binary target, 60 features), the baseline GBM obtained 0.694 (SD = 0.173), LDA + GBM

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

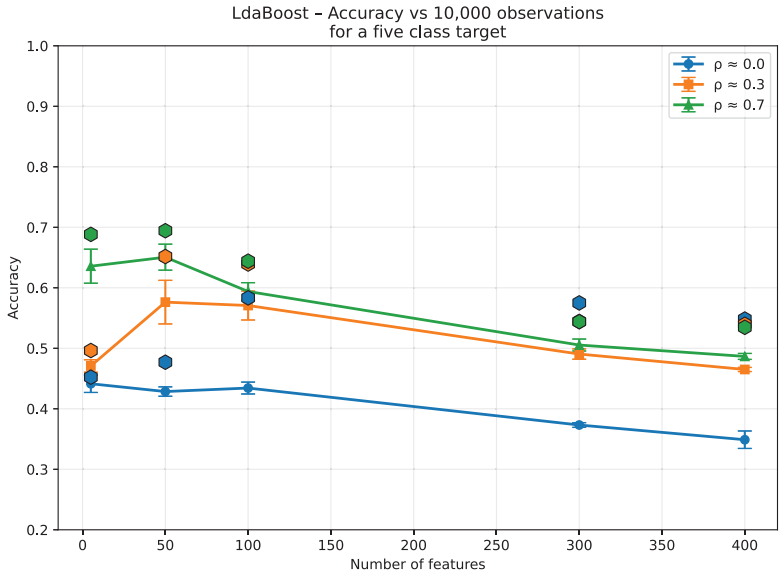


FIGURE 5 | LdaBoost accuracy (multiclass target with five classes) by correlation levels (ρ) and largest N.

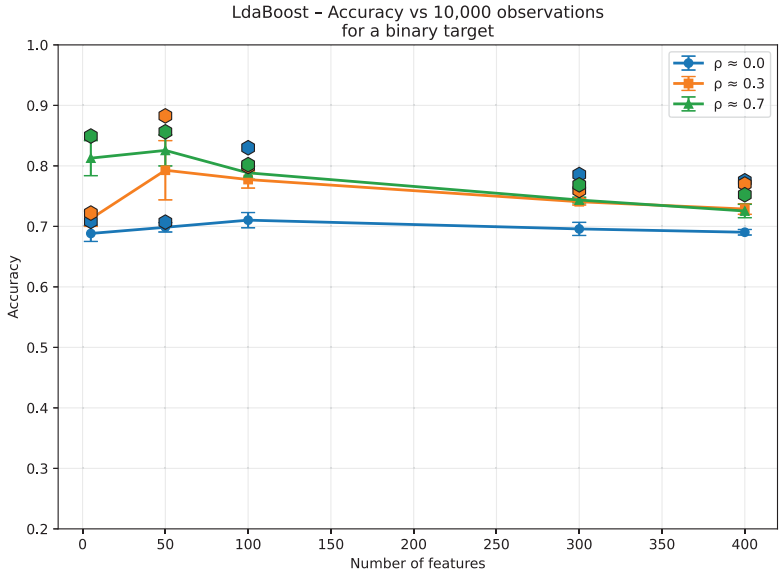


FIGURE 6 | Accuracy over 10,000 observations for a binary target.

0.688 (SD = 0.052), and LdaBoost 0.721 (SD = 0.075), while PCA + GBM achieved the best performance at 0.764 (SD = 0.016). In the Rainfall dataset (binary target, 11 features), performances were comparable, with GBM at 0.861 (SD = 0.025), PCA + GBM at 0.863 (SD = 0.015), LDA + GBM at 0.860 (SD = 0.021), and LdaBoost slightly ahead at 0.864 (SD = 0.025).

Regarding test accuracy, for HAR the LDA + GBM model reached 0.953, LdaBoost 0.951, GBM 0.940, and PCA + GBM 0.905. In the Rainfall dataset, GBM achieved 0.879, PCA + GBM 0.8721, LDA + GBM 0.8607, and LdaBoost 0.8584.

Pairwise comparisons based on the Wilcoxon rank-sum test with Holm correction for multiple testing (Table A3 in the Appendix) reveal no statistically significant differences among the cross-validated accuracy distributions across methods, with the exception of the HAR dataset, where PCA + GBM significantly underperforms relative to the other methods in terms of accuracy.

These findings suggest that the benefits of LdaBoost become increasingly evident as the complexity of the classification task increases, particularly with respect to the number of classes, though not necessarily with feature dimensionality in multiclass settings. In contrast, for binary classification, the added value of LdaBoost tends to grow with higher feature dimensionality. These trends support the overall effectiveness and robustness of the integrated LDA-boosting approach across a range of data scenarios.

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 1 | Cross-validated accuracy (mean with SD in parentheses), test accuracy, and test runtime in seconds (s).

	Dataset	GBM	PCA + GBM	LDA + GBM	LdaBoost
CV accuracy	IRIS	0.960 (0.044)	0.967 (0.031)	**0.980 (0.020)**	0.967 (0.031)
	YEAST	0.568 (0.037)	0.582 (0.023)	**0.614 (0.030)**	0.589 (0.032)
	HAR	**0.995 (0.002)**	0.946 (0.006)	0.982 (0.004)	0.978 (0.006)
	RAINFALL	0.861 (0.025)	0.863 (0.015)	0.860 (0.021)	**0.864 (0.025)**
	SONAR	0.694 (0.173)	**0.764 (0.016)**	0.688 (0.052)	0.721 (0.075)
Test accuracy	HAR	0.940	0.905	**0.953**	0.951
	RAINFALL	**0.879**	0.872	0.8607	0.858
Test runtime (s)	HAR	675.07	652.62	10.25	286.22
	RAINFALL	0.53	0.70	0.27	2.33

Note: Best per row in bold. HAR test values use the official holdout split. RAINFALL test values use train-split fallback validation due to missing external test labels.

TABLE 2 | Runtime and accuracy ablation on HAR and YEAST dataset: CV-accuracy and CV-runtime in seconds (s) with different discriminant components (c). Standard deviations in parenthesis.

		$c = 1$ (HAR)/$c = 1$ (YEAST)		$c = 3$ (HAR)/$c = 5$ (YEAST)	
Scenario	**Model**	**Accuracy**	**Runtime (s)**	**Accuracy**	**Runtime (s)**
HAR	LDA + GBM	0.620 (0.014)	31.910 (4.411)	0.859 (0.012)	33.468 (2.867)
YEAST	LDA + GBM	0.382 (0.020)	0.418 (0.016)	0.554 (0.023)	1.000 (0.006)

4.5 | Accuracy and Runtime-Ablation Evidence

To complement the complexity discussion in Section 3, we conducted an LDA component sensitivity analysis by varying the number of retained discriminant components (c) on the HAR and YEAST datasets, as they present the largest number of classes among the considered datasets. For each dataset, hyperparameters were freely tuned using tenfold cross-validation. We have chosen to fit the LDA + GBM (one-shot transformation) across different values of $c \in \{1, C/2 \text{ (if } C \text{ is even)}, (C + 1)/2 \text{ (if } C \text{ is odd)}, C - 1\}$, where C denotes the number of classes. Performance was evaluated in terms of cross-validated accuracy and runtime.

Table 2 reports CV-accuracies and mean fit time (s) with standard deviations for HAR and YEAST.

Table 2 illustrates that LdaBoost requires a large—ideally maximal—number of LDA components, whereas using fewer components reduces the algorithm's competitiveness. In both categorical target scenarios, the maximum number of components (see Table 1) yields an accuracy improvement of about 60%, compared to a single component.

4.6 | Comparative Performance of Different Pipelines: Simulations

We first begin our simulations with a moderate sample size ($N = 1{,}000$). Figures 1 and 2 illustrate the accuracy trends observed in multiclass and binary classification scenarios, respectively, as the number of features varies at an intermediate mean correlation level ($\rho \approx 0.5$). These figures highlight the comparative performance advantages of LdaBoost over classical GBM and show that LdaBoost can maintain robust predictive accuracy as feature dimensionality increases.

The simulation results clearly demonstrate that LdaBoost consistently outperformed GBM across varying numbers of features, particularly evident in scenarios with multiclass targets (Figure 1).

As the number of features increased, both LdaBoost and LDA + GBM consistently outperformed the standard GBM model, with particularly notable accuracy gains in the 50–100 feature range. This highlights the effectiveness of LDA-based feature extraction. Moreover, the performance gap between GBM and the two LDA-integrated methods remained relatively stable, emphasizing the advantage of combining discriminant analysis with boosting in high-dimensional classification settings.

A similar pattern emerged in the binary classification setting (Figure 2): although performance differences were less pronounced, LdaBoost continued to outperform GBM, especially at intermediate levels of feature dimensionality (around 50–100 features). However, as dimensionality increased further ($p >$ 270), the accuracy of the two methods tended to converge.

Notably, in both binary and multiclass settings, LDA + GBM closely mirrored the performance trajectory of LdaBoost. In the binary case, LdaBoost consistently outperformed LDA + GBM, particularly at moderate levels of feature dimensionality, suggesting that in more complex feature spaces, LDA-based feature extraction alone can enhance the predictive power of boosting methods.

For binary targets, this behavior is noteworthy because $C = 2$ implies that each LDA fit yields only one discriminant component. Consequently, each boosting stage is trained on a one-dimensional score, and LdaBoost's competitiveness comes from repeatedly updating that score direction across rounds rather than from a high-dimensional transformed representation.

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

This also clarifies why the method is conceptually strongest in multiclass settings, where $C - 1$ provides a richer discriminant subspace.

These results are systematically the same over other mean levels of correlations among features (Figures A2 and A3 in the Appendix). Particularly, the added value of LdaBoost (and LDA + GBM) over GBM increases when the mean correlation increases (Figure A3).

With an increased number of observations, Figure A1 in the Appendix replicates Figure 2 (binary target) for $N = 10{,}000$, showing that the two newly introduced algorithms continue to outperform GBM across all feature sets—unlike in the $N = 1000$ case, where the performance gap disappeared for larger numbers (270–300) of features.

4.7 | LdaBoost Simulations

To deeply assess how LdaBoost works in different scenarios, this subsection provides further insight into how LdaBoost accuracy varies.

Analyzing first the role of mean feature correlation ρ with $N = 1{,}000$ observations, the binary simulations reveal that higher correlations ($\rho = 0.7$) deliver the highest accuracies when the feature space is moderately sized, reaching mean values around 0.80 at $p = 5$ and $p = 50$. At $\rho = 0.3$, accuracies lie between 0.59 and 0.77, whereas under no correlation ($\rho = 0.0$) they remain between 0.55 and 0.69. As dimensionality increases toward $p = 400$, the advantage of highly correlated features diminishes and the mean accuracy falls back to roughly 0.63, consistent with the onset of the Hughes phenomenon.

Figure 3 illustrates how LdaBoost accuracy varies over both the number of features and ρ in a binary classification task, also in comparison with accuracy of the optimal Bayes classifier. The model benefits the most from 50 to 100 features when correlations are moderate or high, while strong correlations become detrimental once p exceeds roughly 200. In contrast, under $\rho \approx 0$ the curves are flatter, with modest gains up to 300 features before a slight decline.

These results suggest that in this finite-sample regime LdaBoost capitalizes on redundant information when dimensionality is controlled, but still faces the usual degradation as N/p shrinks. When features are effectively independent, increases in dimensionality yield only incremental improvements, confirming that the architecture remains sensitive to the balance between sample size and feature count.

Figure 4 illustrates how accuracy varies for multiclass targets with 3 (top) and 5 classes (bottom). The main finding is that, in all scenarios, accuracy is higher for binary targets and decreases as the number of classes increases. Moreover, as the number of covariates increases, the accuracies tend to converge to the same value regardless of the level of correlation, and this effect is more pronounced when the target variable has fewer classes.

This may be a clear finding of the so-called Hughes phenomenon [33], shown here for binary targets. The core idea is that as the number of features increases, the classification accuracy of statistical pattern recognizers does not necessarily improve—and may even decline or stabilize—especially when the number of training samples is limited. In fact, data become sparse in high dimensions, and the classifier has fewer observations per unit volume of the feature space to estimate class boundaries reliably. Hughes showed that there is an optimal number of features for a given sample size and classifier type. Beyond that point, adding more features does not improve and may harm classification accuracy.

This effect is more pronounced when the features are highly correlated or when the sample size is low relative to the number of features. Figures 3 and 4 clearly illustrate both effects, especially the sharp decline in accuracy for high levels of feature correlation. Moreover, regarding the latter aspect, given that the sample size (N) is always fixed at 1000, for a large number of features (p), the ratio N/p declines substantially. This Hughes phenomenon is less marked when the number of classes increases (Figure 4) Moreover, Figures 5 and 6 illustrate simulation results with 10,000 observations for multiclass (5 classes) and binary target, respectively. These pictures highlight that, as expected, large samples mitigate the Hughes effect, especially for the binary target.

The large sample experiment ($N = 10{,}000$) confirms that LdaBoost tracks the optimal Bayes estimator remarkably closely in the multiclass setting (Figure 5). Differences are quite stable, with the smallest gaps observed when the dependence structure strengthens.

An improved pattern emerges for the binary target (Figure 6). LdaBoost holds its Bayes-level accuracy for more than 100 features and even benefits from growing correlation, which slightly narrows the residual gap to the optimal Bayes estimator before the Hughes effect eventually dampens both models beyond that threshold. This behavior suggests that, in the binary case, the discriminant projections within the boosting loop capture nearly all of the Bayes-optimal separation in moderately high-dimensional settings, such as those where researchers typically work with a few dozen informative, correlated covariates.

5 | Discussion

This section discusses the rationale and interpretation of the proposed LdaBoost method, with particular attention to its learning mechanism and its relationship to the gradient boosting framework.

Firstly, the proposed approach builds upon the general principles of gradient boosting. In classical GBM for categorical targets, subsequent learners are not trained directly on the original class labels, but on pseudo-residuals, that is, the negative gradients of the loss function with respect to the current model [32]. These quantities are continuous and differ substantially from the original labels, yet they provide the supervision signal that drives the iterative refinement of the model.

In our method, rather than applying LDA directly to these continuous residuals, we use the gradient information to construct pseudo-labels. Specifically, after computing the residuals, each observation is reassigned to the class corresponding to the largest positive gradient component. This step, which discretizes the

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

functional gradient and is not part of the classical GBM iterative procedure, should be regarded as a heuristic procedure inspired by loss-function minimization through gradient descent. The gradient information provides the motivation for the pseudo-label construction, but the specific discretization rule is not theoretically implied by functional gradient descent. LDA is then applied to these dynamically relabeled data, thereby preserving its natural formulation as a method for supervised dimensionality reduction with categorical labels. In other words, the pseudo-label identifies the class toward which the model "should move" in order to correct its current error.

This procedure introduces a form of dynamic supervision, in which the labels are no longer fixed but evolve across iterations, reflecting the current state of the model. As a consequence, LDA is not used to separate the original classes, but rather to identify, at each iteration, the directions in the feature space that best discriminate among the required corrections. In this sense, the resulting discriminant projection is locally adapted to the model's errors and allows the learning process to focus on the most challenging regions of the problem.

From this perspective, the relabeling step can be viewed in connection with cost-sensitive and reweighted classification frameworks, where the learning problem is modified to emphasize difficult or misclassified observations [34, 35], as well as with hard assignment strategies derived from soft signals, as in expectation–maximization (EM) algorithms [36]. More specifically, the residual vector for each observation can be interpreted as a class-wise cost or utility signal, analogous to the class-dependent weights used in cost-sensitive learning. Our relabeling step corresponds to a hard assignment that selects, for each observation, the class associated with the largest reduction in loss, thereby converting a continuous supervision signal into a categorical one. This is closely related to hard assignment variants of the EM algorithm, where latent soft responsibilities are replaced by their maximum a posteriori values. Under this view, LDA is applied to a sequence of dynamically updated partitions of the data, each reflecting the current gradient of the loss, and its optimization criterion remains well-defined at every iteration.

To this end, LDA plays a dynamic role within this framework: at each iteration, it identifies a discriminant subspace that emphasizes the directions where the model is currently making the largest errors, as indicated by the gradient signal.

One limitation of this pseudo-labeling strategy is that it may favor classes that are strongly underestimated or only weakly plausible, as large residuals can dominate the update even when the corresponding class has low prior probability. To mitigate this issue while still leveraging gradient information, possible alternative rules (to be assessed in future studies) that combines (or weights) the gradients $g_{i,c}^{(m)}$ with the associated class probabilities $p_{i,c}^{(m)}$ in a unified scoring function can be assessed. Then, discriminant scores such as $p_{i,c}^{(m)} + \lambda g_{i,c}^{(m)}$ can be used, where $\lambda \in [0, 1]$ controls the influence of the gradient term. This formulation would allow the method to account not only for the model's current belief but also for the direction and magnitude of correction suggested by the gradient. In this way, the approach integrates ideas from gradient boosting and pseudo-labeling, promoting classes that are both probable and systematically underpredicted, while filtering out low-confidence assignments. Future studies may exploit such alternative labeling.

Second, in the binary classification setting, Fisher Discriminant Analysis yields a single discriminant component, and it may therefore appear somewhat surprising that LdaBoost performs competitively at all given the resulting one-dimensional representation. However, this component is obtained through a supervised procedure explicitly designed to maximize class separation, and therefore represents a highly informative projection of the original feature space. Rather than constituting a generic dimensionality reduction, it can be interpreted as a targeted feature extraction step that concentrates the relevant signal for the prediction task into a single direction.

Even when operating on a one-dimensional representation, Gradient Boosting Machines with decision trees retain considerable modeling flexibility. Decision trees can introduce multiple splits along the projected axis, enabling the approximation of complex and nonlinear relationships through combinations of threshold-based rules. As a result, the restriction to a single feature does not necessarily imply a substantial loss in the expressive power of the overall model.

From a statistical perspective, the binary case can also be viewed as a setting in which the method trades representational richness for noise reduction and stability. By projecting the data onto the most discriminative direction, the approach may reduce variance and improve generalization, particularly in high-dimensional or noisy scenarios. This perspective helps explain why competitive empirical performance can still be observed despite the apparent constraint. At the same time, the advantages of the proposed approach become more pronounced in the multiclass setting, where multiple discriminant components provide a richer representation for the subsequent boosting stage.

Third, despite the strong predictive accuracy of LDA-based methods, an important finding is that GBM (using cross-validated metrics) outperforms the other approaches on the largest and most feature-rich HAR dataset. A possible explanation is that, in this setting, the supervised dimensionality reduction induced by LDA may compress the original information too aggressively. In particular, LDA produces at most $C - 1$ discriminant components, thereby representing the data through a small number of global linear directions; this can be limiting when the discriminative signal is distributed across many variables or depends on nonlinear relationships among features [37]. By contrast, GBM can naturally capture threshold effects, feature interactions, and complex nonlinear structures through ensembles of decision trees, making it particularly effective in high-dimensional settings with heterogeneous features [32]. This suggests that, for rich datasets such as HAR, preserving the original feature space may offer a substantial advantage.

Moreover, the literature highlights several well-known limitations of LDA/FDA: performance may degrade when the Gaussian class-conditional assumption and the homogeneity of covariance matrices are violated [37]; when classes exhibit multimodal structure that cannot be captured by a single discriminant direction [38]; and in high-dimensional regimes, where noise accumulation can adversely affect linear discriminants [39]. These factors

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

provide a plausible explanation for the observed behavior on the HAR dataset. Notice, however, that LdaBoost outperforms GBM on the test set for the HAR dataset.

The paper has also limitations.

Firstly, the simulation study is based on a multivariate normal data-generating process with class-specific means, a setting in which Linear Discriminant Analysis is known to be asymptotically optimal. As a consequence, the proposed LdaBoost method is evaluated in a scenario that is particularly favorable to its underlying structure, and the observed performance may therefore be partly expected. While this choice allows for a clear and interpretable comparison with the Bayes optimal classifier in a controlled framework, it does not fully reflect the complexity of real-world data, which often exhibit non-Gaussian distributions, nonlinear class boundaries, and heterogeneous feature types. Future work will extend the empirical analysis to more general settings, including non-Gaussian and nonlinear scenarios, in order to better assess the robustness and generalizability of the proposed approach.

Second, since LDA is traditionally defined for fixed class labels, the use of LdaBoost—where samples are relabeled based on residuals and LDA is iteratively refitted—may appear as a heuristic attempt to extend LDA to a setting that is more naturally suited to continuous targets. In the Discussion section, we provide a methodological motivation for this procedure, grounded in the gradient information arising from the loss-minimization framework of gradient descent, while explicitly recognizing that the pseudo-labeling construction is a heuristic algorithmic adaptation and is not theoretically implied by GBM. We also discuss related machine-learning approaches based on pseudo-labels and possible improvements for future studies.

6 | Conclusion

To our knowledge, no previous study has investigated the use of LDA in conjunction with gradient boosting. Specifically, there is no established practice or empirical evidence suggesting that LDA enhances performance or is inherently compatible with gradient boosting models. Beyond analyzing the effects of PCA and LDA as feature extraction techniques on the accuracy of gradient boosting machines, we propose an integrated iterative algorithm—LdaBoost—that applies LDA-based feature transformation at each boosting iteration.

Empirically, we compare the classification performance of standard GBM with PCA, GBM with LDA preprocessing, and the proposed LdaBoost algorithm across five benchmark datasets. In most cases, LdaBoost or GBM with LDA preprocessing improved on standard GBM in cross-validated and test accuracy, the HAR dataset being the unique case demonstrating that baseline GBM can still be competitive when all features are retained.

Overall, the results show that LdaBoost yields increased performance gains as classification tasks become more complex in terms of the number of target classes. However, this pattern is less pronounced when the number of features is increased in multiclass settings. In contrast, in binary classification tasks, LdaBoost exhibits a clear advantage as the dimensionality of the features increases. These findings highlight the adaptability and robustness of the LDA-boosting framework across a variety of data structures.

Two main conclusions emerge from the empirical results.

First, the strong performance of the LDA-based approaches in the binary classification setting, despite relying on a single discriminant component, further confirms the effectiveness of the GBM framework itself. Predictive accuracy appears to depend less on the dimensional richness of the covariates and more on the sequential boosting mechanism, where multiple low-complexity trees iteratively correct classification errors and progressively concentrate the relevant predictive signal.

Second, standard GBM outperforms the LDA-based variants only in the most complex and feature-rich dataset. This suggests that, in high-dimensional settings characterized by large p and more complex class structures, the LDA compression may become excessively restrictive, removing useful multidimensional information and nonlinear interactions that standard GBM is instead able to exploit more effectively across successive boosting iterations.

Concerning LdaBoost, simulation studies indicate improved accuracy in binary classification under moderate feature complexity and high feature correlation, and also clearly reveal the presence of the Hughes phenomenon (curse of dimensionality).

Together, these insights underscore LdaBoost's robust adaptability to different data conditions, demonstrating its practical utility in diverse classification scenarios, particularly in settings characterized by high dimensionality and substantial feature correlation. In particular, the comparison with the optimal Bayes estimator (especially for the $N = 10{,}000$ simulations and binary target) shows that LdaBoost stays virtually coincident with the Bayes-optimal curve for more than 100 correlated covariates, demonstrating that the method preserves near-optimal discrimination in the most relevant feature ranges.

Author Contributions

Both authors contributed to the conceptualization of the study. P.G.L. wrote the manuscript, interpreted the results, and critically revised the manuscript, while G.E.D. developed the code and conducted the simulations. Both authors approved the final version of the manuscript.

Acknowledgments

Generative AI was employed solely for rephrasing some sentences and paragraphs of this manuscript, which have always been scrutinized by a human. Specifically, Gemini 3.1 Pro in the agentic mode was used for this purpose. The design of the study and the algorithm code were not done by the AI.

Funding

The authors have nothing to report.

Conflicts of Interest

The authors declare no conflicts of interest.

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Data Availability Statement

The datasets are freely available upon request from the first author g.donninelli@campus.unimib.it.

References

1. L. Van Der Maaten, E. O. Postma, H. Jaap Van Den Herik, et al., "Dimensionality Reduction: A Comparative Review," *Journal of Machine Learning Research* 10, no. 66–71 (2009): 13.

2. N. Bacanin, M. Zivkovic, T. Bezdan, D. Cvetnic, and L. Gajic, "Dimensionality Reduction Using Hybrid Brainstorm Optimization Algorithm," in *Proceedings of International Conference on Data Science and Applications: ICDSA 2021*, vol. 2 (Spring, 2021), 679–692.

3. X. Huang, W. Lei, and Y. Ye, "A Review on Dimensionality Reduction Techniques," *International Journal of Pattern Recognition and Artificial Intelligence* 33, no. 10 (2019): 1950017.

4. Z. Hu, K. Shukla, G. E. Karniadakis, and K. Kawaguchi, "Tackling the Curse of Dimensionality With Physics-Informed Neural Networks," *Neural Networks* 176 (2024): 106369.

5. M. García-Torres, R. Ruiz, and F. Divina, "Evolutionary Feature Selection on High Dimensional Data Using a Search Space Reduction Approach," *Engineering Applications of Artificial Intelligence* 117 (2023): 105556.

6. K. Fukunaga, *Introduction to Statistical Pattern Recognition* (Elsevier, 2013).

7. G. James, D. Witten, T. Hastie, et al., *An Introduction to Statistical Learning*, vol. 112 (Springer, 2013).

8. T. Zhou, D. Tao, and W. Xindong, "Manifold Elastic Net: A Unified Framework for Sparse Dimension Reduction," *Data Mining and Knowledge Discovery* 22 (2011): 340–371.

9. S. T. Roweis and L. K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science* 290, no. 5500 (2000): 2323–2326.

10. J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science* 290, no. 5500 (2000): 2319–2323.

11. M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Computation* 15, no. 6 (2003): 1373–1396.

12. Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Roux, and M. Ouimet, "Out-Of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering," *Advances in Neural Information Processing Systems* 16 (2003): 2197–2219.

13. S.-A. Sadegh-Zadeh, N. Sadeghzadeh, O. Soleimani, S. S. Ghidary, S. Movahedi, and S.-Y. Mousavi, "Comparative Analysis of Dimensionality Reduction Techniques for EEG-Based Emotional State Classification," *American Journal of Neurodegenerative Disease* 13, no. 4 (2024): 23–33.

14. R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction," *Journal of Applied Science and Technology Trends* 1, no. 1 (2020): 56–70.

15. T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, 2016), 785–794.

16. M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do We Need Hundreds of Classifiers to Solve Real World Classification Problems?," *Journal of Machine Learning Research* 15, no. 1 (2014): 3133–3181.

17. J. J. Gabriel and L. J. Anbarasi, "Optimizing Coronary Artery Disease Diagnosis: A Heuristic Approach Using Robust Data Preprocessing and Automated Hyperparameter Tuning of Extreme Gradient Boosting," *IEEE Access* 11 (2023): 112988–113007.

18. A. Demiriz, K. P. Bennett, and J. Shawe-Taylor, "Linear Programming Boosting via Column Generation," *Machine Learning* 46, no. 1–3 (2002): 225–254.

19. J. K. Bradley and R. E. Schapire, "Filterboost: Regression and Classification on Large Datasets," *Advances in Neural Information Processing Systems* 20 (2008): 185–192.

20. K. Crammer, J. Keshet, and Y. Singer, "Kernel Design Using Boosting," in *Advances in Neural Information Processing Systems* 15, ed. S. Becker, S. Thrun, and K. Obermayer (MIT Press, 2003), 553–60.

21. B. Rakhshan and G. Rabusseau, "Tensorized Random Projections," in *International Conference on Artificial Intelligence and Statistics* (PMLR, 2020), 3306–3316.

22. M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and Robust Automated Machine Learning," in *Advances in Neural Information Processing Systems*, vol. 28, ed. C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Curran Associates, 2015), https://proceedings.neurips.cc/paper_files/paper/2015/file/11d0e6287202fced83f79975ec59a3a6-Paper.pdf.

23. J. Ma and Y. Yuan, "Dimension Reduction of Image Deep Feature Using PCA," *Journal of Visual Communication and Image Representation* 63 (2019): 102578.

24. A. Subasi and M. I. Gursoy, "EEG Signal Classification Using PCA, ICA, LDA and Support Vector Machines," *Expert Systems With Applications* 37, no. 12 (2010): 8659–8666.

25. M. Leo, C. Distante, M. Bernabei, and K. Persaud, "An Efficient Approach for Preprocessing Data From a Large-Scale Chemical Sensor Array," *Sensors* 14, no. 9 (2014): 17786–17806.

26. M. Luštrek and B. Kaluža, "Fall Detection and Activity Recognition With Machine Learning," *Informatica* 33, no. 2 (2009): 197–204.

27. M. Cann, *Feature Reduction Using PCA, LDA, and PCA + LDA to Improve Decision Tree C4.5 Classification of Human Activity Recognition Dataset* (2020), Web document. Available at: https://mattcann1.github.io/mattcann1/docs/Feature%20Reduction%20using%20PCA,%20LDA,%20AND%20PCA+LDA%20to%20Improve%20Decision%20Tree%20C4.5%20Classification%20of%20Human.pdf.

28. M. Pechenizkiy, A. Tsymbal, and S. Puuronen, "On Combining Principal Components With Fisher's Linear Discriminants for Supervised Learning," *Foundations of Computing and Decision Sciences* 31, no. 1 (2006): 59–74.

29. J. Yang and J.-y. Yang, "Why Can LDA Be Performed in PCA Transformed Space?," *Pattern Recognition* 36, no. 2 (2003): 563–566.

30. J. Li, B. Zhao, and H. Zhang, "Face Recognition Based on PCA and LDA Combination Feature Extraction," in *2009 First International Conference on Information Science and Engineering* (IEEE, 2009), 1240–1243.

31. Z. Liu, J. Wang, G. Liu, and Z. Lin, "Discriminative Low-Rank Preserving Projection for Dimensionality Reduction," *Applied Soft Computing* 85 (2019): 105768, https://doi.org/10.1016/j.asoc.2019.105768.

32. J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics* 29, no. 5 (2001): 1189–1232.

33. G. Hughes, "On the Mean Accuracy of Statistical Pattern Recognizers," *IEEE Transactions on Information Theory* 14, no. 1 (1968): 55–63.

34. C. Elkan, "The Foundations of Cost-Sensitive Learning," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)* (Lawrence Erlbaum Associate, 2001), 973–978.

35. B. Zadrozny, J. Langford, and N. Abe, "Cost-Sensitive Learning by Cost-Proportionate Example Weighting," in *Proceedings of the Third IEEE International Conference on Data Mining (ICDM)* (IEEE, 2003), 435–442.

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

36. A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood From Incomplete Data via the Em Algorithm," *Journal of the Royal Statistical Society: Series B* 39, no. 1 (1977): 1–38.

37. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning* (Springer, 2009).

38. M. Sugiyama, "Dimensionality Reduction of Multimodal Labeled Data by Local Fisher Discriminant Analysis," *Journal of Machine Learning Research* 8 (2007): 1027–1061.

39. J. Fan and Y. Fan, "High Dimensional Classification Using Features Annealed Independence Rules," *Annals of Statistics* 36, no. 6 (2008): 2605–2637.

Appendix A

As previously mentioned, we adopt an alternative tuning strategy. To evaluate how the performance of different algorithms depends solely on the selected features, we fix the hyperparameters of GBM (max_depth, n_estimators, learning_rate, and min_samples_leaf) at their optimal values and apply them unchanged to PCA + GBM, LDA + GBM, and LdaBoost.

Table A1 presents the results. The findings are largely consistent with those obtained under the free- tuning strategy (Table 1). In particular, GBM outperforms the other methods only on the HAR dataset, while LDA-based features and LdaBoost achieve superior performance in the remaining cases. This pattern also holds for the SONAR dataset, where PCA + GBM performs better than the other methods, in line with the free-tuning results. Overall, the ranking of methods in terms of test accuracy remains unchanged across the two tuning strategies.

In the HAR dataset both LDA + GBM and LdaBoost match the benchmark (Support Vector Machine (0.977) and Adaboost (0.977) found by Luštrek and Kaluža) while using drastically fewer features, underscoring the benefit of integrating discriminant components within boosting frameworks. Macro-F1 results mirror the accuracy rankings: GBM (F1 = 0.995, SD = 0.002), PCA + GBM (F1 = 0.949, SD = 0.009), LDA + GBM (F1 = 0.983, SD = 0.004), and LdaBoost (F1 = 0.980, SD = 0.006).

From a computational perspective (Table A2), the LDA-based approaches remain very competitive while keeping strong accuracy. One-shot LDA + GBM is generally the fastest option (and in HAR it is dramatically faster than PCA + GBM), while still delivering high predictive performance. At the same time, LdaBoost deserves a positive emphasis: despite its iterative LDA updates, it keeps a favorable speed-accuracy trade-off, remains substantially faster than PCA + GBM on massive dataset such as HAR, and preserves consistently strong predictive performance across datasets (Table A3).

TABLE A1 | Cross-validated accuracy (mean with SD in parentheses) and test accuracy where available.

	Dataset	GBM	PCA + GBM	LDA + GBM	LdaBoost
CV	IRIS	0.960 (0.044)	0.933 (0.060)	**0.973 (0.053)**	0.967 (0.054)
	YEAST	0.568 (0.037)	0.504 (0.023)	0.569 (0.035)	**0.589 (0.032)**
	HAR	**0.995 (0.002)**	0.946 (0.009)	0.982 (0.004)	0.978 (0.006)
	RAINFALL	0.861 (0.025)	0.860 (0.019)	0.856 (0.024)	**0.864 (0.025)**
	SONAR	0.694 (0.173)	0.645 (0.132)	0.625 (0.150)	**0.721 (0.075)**

Note: Best per row in bold.

TABLE A2 | Cross-validated and test runtime in seconds (s).

	Dataset	GBM	PCA + GBM	LDA + GBM	LdaBoost
CV runtime (s)	IRIS	0.12	0.15	**0.04**	2.18
	YEAST	**2.38**	3.65	3.12	16.42
	HAR	429.84	407.42	**9.18**	204.65
	RAINFALL	**0.22**	1.48	1.07	2.56
	SONAR	0.14	0.19	**0.05**	2.24
Test runtime (s)	HAR	675.07	652.62	**10.25**	286.22
	RAINFALL	0.53	0.70	**0.27**	2.33

Note: Best per row in bold.

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE A3 | Holm-corrected *p*-values for pairwise comparisons across methods in each dataset.

Dataset	Comparison	Holm-corrected *p*-value
HAR	GBM_vs_PCA + GBM	**0.012**
HAR	GBM_vs_LDA + GBM	0.888
HAR	GBM_vs_LdaBoost	0.750
HAR	PCA + GBM_vs_LDA + GBM	**0.012**
HAR	PCA + GBM_vs_LdaBoost	**0.012**
HAR	LDA + GBM_vs_LdaBoost	0.412
RAINFALL	GBM_vs_PCA + GBM	0.613
RAINFALL	GBM_vs_LDA + GBM	0.387
RAINFALL	GBM_vs_LdaBoost	0.930
RAINFALL	PCA + GBM_vs_LDA + GBM	1.000
RAINFALL	PCA + GBM_vs_LdaBoost	1.000
RAINFALL	LDA + GBM_vs_LdaBoost	0.930
IRIS	GBM_vs_PCA + GBM	0.786
IRIS	GBM_vs_LDA + GBM	0.786
IRIS	GBM_vs_LdaBoost	0.952
IRIS	PCA + GBM_vs_LDA + GBM	0.573
IRIS	PCA + GBM_vs_LdaBoost	0.952
IRIS	LDA + GBM_vs_LdaBoost	0.952
SONAR	GBM_vs_PCA + GBM	0.136
SONAR	GBM_vs_LDA + GBM	0.090
SONAR	GBM_vs_LdaBoost	0.070
SONAR	PCA + GBM_vs_LDA + GBM	0.932
SONAR	PCA + GBM_vs_LdaBoost	0.932
SONAR	LDA + GBM_vs_LdaBoost	0.932
YEAST	GBM_vs_PCA + GBM	1.000
YEAST	GBM_vs_LDA + GBM	1.000
YEAST	GBM_vs_LdaBoost	0.625
YEAST	PCA + GBM_vs_LDA + GBM	1.000
YEAST	PCA + GBM_vs_LdaBoost	0.625
YEAST	LDA + GBM_vs_LdaBoost	0.375

Note: Significant (at the 0.05 significance level) contrasts in Bold.

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

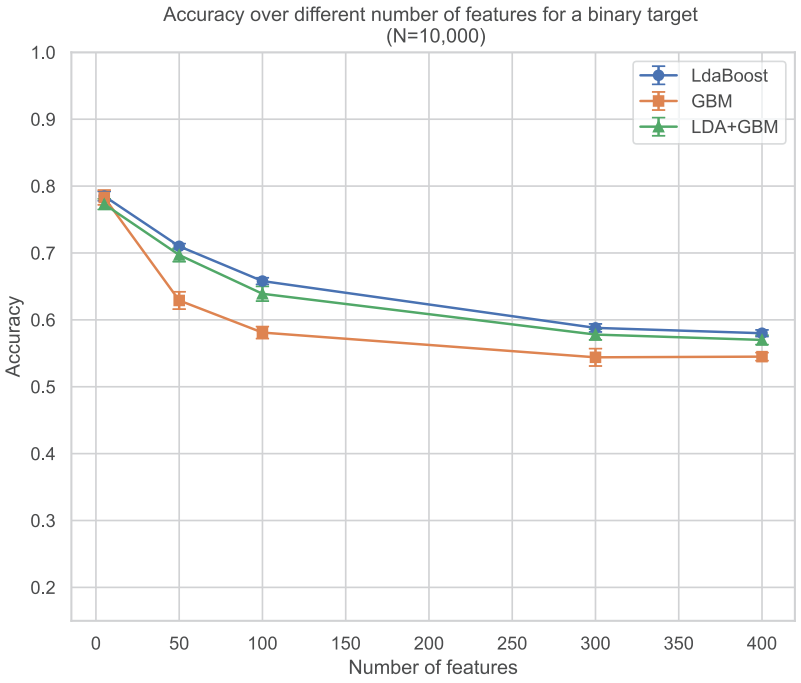


FIGURE A1 | Comparison of the different pipelines (binary target) in the medium-correlation scenario and largest N ($\rho \approx 0.5$, $N = 10{,}000$).

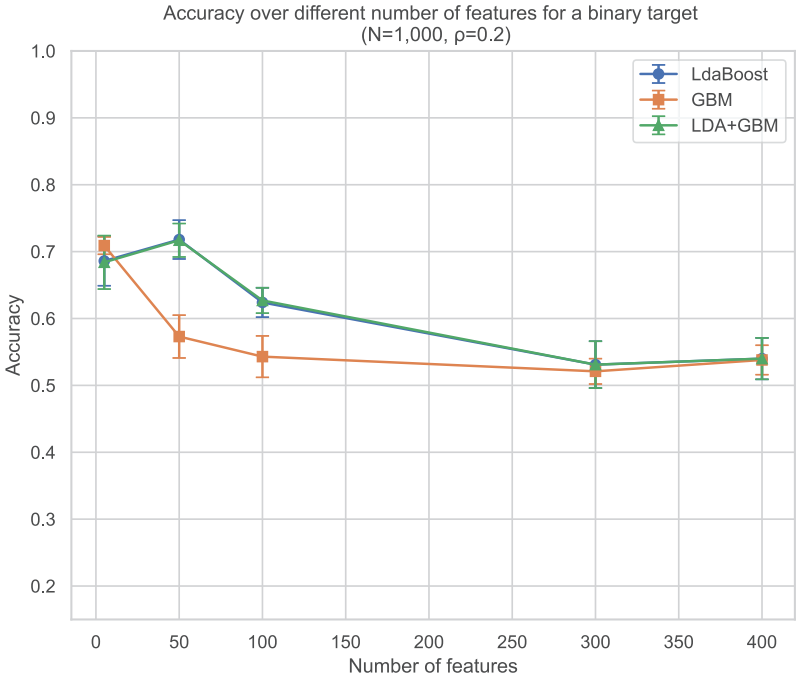


FIGURE A2 | Comparison of the different pipelines (binary target) in a low-correlation scenario and smallest N.

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

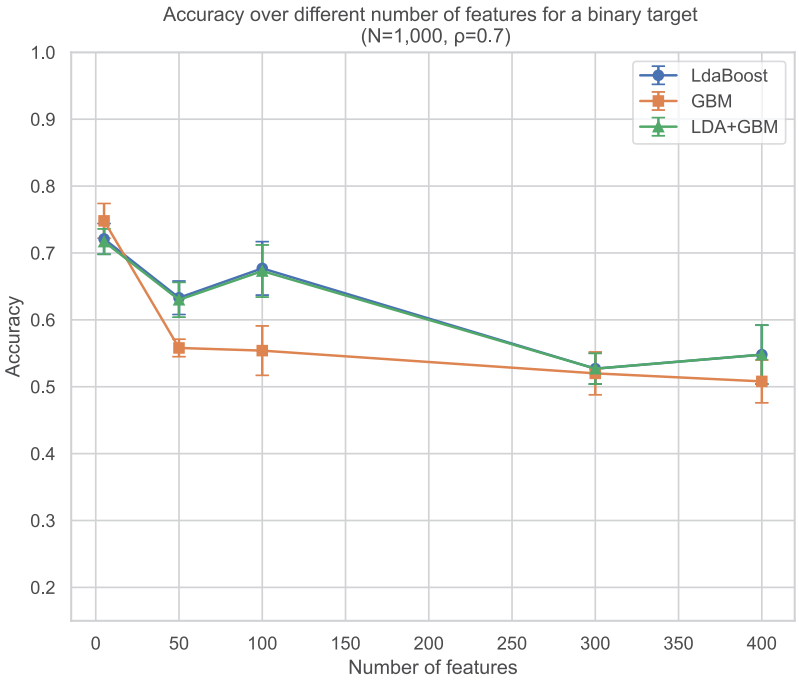


FIGURE A3 | Comparison of the different pipelines (binary target) in a high-correlation scenario and smallest N.

19321872a, 2026, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/sam.70114 by CochraneItalia, Wiley Online Library on [17/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License