AdaptiveBayes: Comprehensive empirical analysis and algorithmic enhancement framework[⋆]

Sergii Kavun[1,*,†]

[1] Interregional Academy of Personnel Management, 2 Frometivska str., 03039 Kyiv, Ukraine

Abstract

Traditional baseline models in machine learning, particularly Logistic Regression (LR), serve as fundamental benchmarks but suffer from computational bottlenecks on large-scale datasets. This study presents a comprehensive performance analysis of AdaptiveBayes (AdB) as a potential replacement baseline classifier through extensive empirical evaluation across 12 diverse datasets spanning both standard benchmarks and large-scale classification tasks. AdB demonstrates remarkable computational efficiency with average training speed improvements of 10x over LR, reaching up to 750x on specific datasets, while maintaining competitive AUC performance on selected problems. However, accuracy limitations averaging -0.26 ± 0.33 compared to LR highlight the fundamental speed-accuracy trade-off. Through systematic algorithmic analysis, we propose a comprehensive enhancement framework incorporating advanced regularization, curvature-aware optimization, and improved initialization strategies projected to increase accuracy up to 8% while preserving computational advantages. This research establishes AdB as a viable baseline for time-critical applications and provides a roadmap for next-generation efficient classification algorithms.

Keywords

AdaptiveBayes, baseline classification, computational efficiency, adaptive learning, machine learning optimization, quasi-newton methods

1. Introduction

In applied machine learning and data science, establishing a robust performance baseline is essential for evaluating more complex models. Baseline model selection represents a critical first step in machine learning pipelines, with LR traditionally serving as the gold standard due to its simplicity, interpretability, and consistent performance. However, the iterative optimization requirements of LR create significant computational bottlenecks in large-scale scenarios, motivating the exploration of faster alternatives that maintain reasonable classification quality. Speed-oriented baselines include Pegasos-SVM, mini-batch Newton LR, and diagonal-adapted Naïve Bayes. Gradient-boosted decision trees (XGBoost, LightGBM) typically win quality benchmarks but at higher computational cost. Recent one-pass learners (Vowpal Wabbit, Online Newton Step) provide alternatives; however, systematic cross-dataset studies remain scarce.

This paper investigates AdB, a classifier engineered for superior computational efficiency, as a potential replacement for LR in the role of a baseline model. The primary objective of a baseline is to provide a quick and reasonable performance benchmark, a role for which an ultra-fast model could be ideally suited.

AdB emerges as a novel classification algorithm engineered for superior computational efficiency through a unique combination of closed-form Bayesian updates, adaptive learning rate mechanisms, and GPU acceleration capabilities. Preliminary analysis on standard benchmark datasets indicated competitive accuracy with 2-3× training speedups, warranting comprehensive evaluation across diverse problem domains.

[⋆] CPITS-II 2025: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, October 26, 2025, Kyiv, Ukraine

[*] Corresponding author.

[†] These authors contributed equally.

kavserg@gmail.com (S. Kavun)

0000-0003-4164-151X (S. Kavun)

This study extends the evaluation to seven large-scale datasets characterized by high dimensionality and substantial sample sizes, benchmarking AdB against established models including XGBoost, LightGBM, RandomForest, and Multi-Layer Perceptrons. The research provides multi-faceted performance analysis encompassing accuracy, AUC, training time, memory consumption, and prediction speed while identifying architectural limitations and proposing targeted enhancements.

Research contributions:

- Comprehensive empirical evaluation of AdB across 12 diverse classification datasets.
- Systematic analysis of computational efficiency and memory utilization patterns.
- Identification of failure modes and calibration issues in imbalanced data scenarios.
- Detailed algorithmic enhancement framework with theoretical foundations.
- Practical deployment guidelines for baseline model selection.

The proposed framework aligns with CPITS Workshop themes by bridging probability theory, risk management, and machine learning in cybersecurity. Specifically, AdaptiveBayes enhances predictive analytics for cyber defense, supports cryptographic protocol assessment, and provides a scalable framework for cyber-risk forecasting in interconnected systems.

1. Literature Review

The development of efficient baseline classifiers in machine learning has evolved significantly, driven by the increasing demand for computational efficiency [1] without substantial accuracy degradation. This review systematically examines the theoretical foundations and empirical advancements that inform the AdaptiveBayes framework, organizing the analysis around three critical research themes: traditional baseline approaches, adaptive learning methodologies [2], and computational efficiency [3] in modern machine learning.

1.1. Traditional Baseline Classification Methods

Logistic Regression has maintained its dominant position in baseline classification due to its theoretical grounding in maximum likelihood estimation and well-understood convergence properties. Chen and Guestrin [4] demonstrated that while tree-based methods like XGBoost provide superior accuracy, the interpretability and simplicity of logistic regression continue to make it the gold standard for baseline model selection. Recent comparative studies by Islam et al. [5] and Yadav et al. [6] consistently validate LR reliability across medical diagnosis and heart disease prediction applications, though computational limitations emerge prominently in high-dimensional scenarios.

The persistence of logistic regression as a baseline stems from its theoretical foundation and consistent performance across diverse domains. However, Pathak and Shrivas [7] identified significant computational bottlenecks when applying traditional LR to large-scale intrusion detection systems, motivating exploration of more efficient alternatives. Similarly, de Luna et al. demonstrated that while LR maintains accuracy in cybersecurity URL classification tasks [8], training time constraints limit its applicability in real-time security applications.

1.2. Adaptive Learning Algorithms and Optimization

The evolution of adaptive learning methods has been particularly influenced by advances in quasi-Newton optimization techniques. Ma [9] introduced the Apollo optimizer, which demonstrates adaptive parameter-wise diagonal quasi-Newton optimization [10] that dynamically incorporates loss function curvature through efficient Hessian approximation [11]. This work established theoretical foundations for adaptive learning rate mechanisms that form the basis of modern efficient classification algorithms.

Building upon these foundations, Yagishita et al. [11] developed proximal diagonal Newton methods for composite optimization problems, demonstrating significant improvements in convergence speed while maintaining solution quality. Their approach highlighted the potential for combining adaptive learning with regularization techniques, directly informing the enhancement framework proposed in this study. The work by Aminifard and Babaie-Kafaki [12] further extended diagonal quasi-Newton methods to compressed sensing applications, proving that diagonal approximations can maintain computational efficiency while preserving optimization quality.

Recent advances in stochastic optimization have emphasized the importance of curvature-aware adaptation. Ahmed [13] proposed a novel variance reduction proximal stochastic Newton algorithm specifically designed for large-scale machine learning optimization, achieving faster convergence while handling non-smooth regularizers. These developments directly motivate investigation of alternative baseline approaches that prioritize computational efficiency through adaptive mechanisms.

1.3. Computational Efficiency in Modern Machine Learning

The imperative for computational efficiency has driven substantial research into optimized algorithms and architectures. Ke et al. [14] developed LightGBM as a highly efficient gradient boosting decision tree framework, demonstrating that algorithmic innovations can achieve dramatic speedups without sacrificing accuracy. Their work established benchmarks for evaluating efficiency-accuracy trade-offs in modern machine learning applications.

Complementing tree-based approaches, Wu et al. [15] presented robust stochastic quasi-Newton methods with applications in machine learning, showing that second-order optimization techniques can provide superior computational profiles compared to traditional first-order methods. The integration of momentum-based updates, as demonstrated by Alecsa [13] in stochastic non-convex optimization, has proven particularly effective for accelerating convergence in adaptive learning scenarios.

Hardware acceleration has emerged as a critical factor in achieving computational efficiency. The work by Chia et al. [4] on two-phase switching optimization strategies in LSTM models demonstrated the importance of GPU-optimized implementations for practical deployment. This hardware-software co-design approach directly influences the AdaptiveBayes architecture [16], which incorporates CuPy-based GPU acceleration for enhanced computational performance.

1.4. Sparse and Efficient Bayesian Methods

Recent developments in Bayesian machine learning have focused on achieving computational efficiency while maintaining theoretical rigor. Luo et al. [17] developed inverse-free and scalable sparse Bayesian extreme learning machines [18], demonstrating that closed-form Bayesian updates can dramatically reduce computational complexity compared to iterative optimization approaches. Their work established theoretical precedents for the simplified Bayesian framework underlying AdaptiveBayes.

The complexity-optimized sparse Bayesian learning approach presented by Luo et al. [18] specifically addressed scalability challenges in classification tasks, achieving linear computational complexity while maintaining competitive accuracy. This research directly informs the mathematical foundation of AdaptiveBayes, particularly the selective update mechanisms and feature transformation strategies.

1.5. Feature Transformation and Regularization Techniques

Advanced feature transformation techniques have proven critical for improving baseline classifier performance. The elastic net regularization framework introduced by Zou and Hastie [19] demonstrated that combining L1 and L2 penalties provides optimal balance between sparsity and variance control. This theoretical foundation directly supports the proposed enhancements to the AdaptiveBayes framework.

Modern initialization strategies have also received significant attention. Glorot and Bengio [20] established theoretical foundations for Xavier initialization, while He et al. [21] developed improved initialization schemes for ReLU-like activation functions. These contributions inform the proposed weight initialization enhancements that could significantly improve AdaptiveBayes accuracy while preserving computational advantages.

1.6. Empirical Validation in Real-World Applications

Contemporary research has emphasized the importance of comprehensive empirical validation across diverse application domains. Geeitha et al. [22] demonstrated effective application of Naive Bayes algorithms in medical survival prediction, while Abogada and Usona [1] successfully applied machine learning classification models to attrition prediction in business process outsourcing industries. These studies establish empirical precedents for evaluating adaptive baseline classifiers across heterogeneous problem domains.

The work by Rajneekant et al. [23] on malware classification using machine learning provided comprehensive comparative analysis of API sequence-based techniques, establishing methodological frameworks for systematic algorithm evaluation. Their emphasis on computational efficiency alongside accuracy metrics directly parallels the evaluation methodology employed in this AdaptiveBayes study.

1.7. Gaps and Research Opportunities

Despite these advances, systematic cross-dataset studies comparing efficiency-oriented baselines remain scarce. Most existing research focuses on domain-specific applications rather than comprehensive evaluation across diverse problem characteristics. Furthermore, theoretical analysis of convergence guarantees for adaptive baseline approaches has received limited attention, creating opportunities for both empirical and theoretical contributions.

The integration of modern hardware acceleration with adaptive learning algorithms represents another underexplored research direction. While GPU optimization has been extensively studied for deep learning applications, its potential for accelerating traditional machine learning baselines through adaptive mechanisms remains largely unexplored.

The common vision of related proposed theory and existing theories is presented in Table 1.

Table 1
The common vision of related proposed theory and existing theories

Author(s)	Theory / Approach	Key Contribution	Limitations	AdB Integration/ Extension
Chen & Guestrin [24]	XGBoost gradient boosting	Scalable tree boosting with superior accuracy	High computational cost, memory intensive	Baseline comparison for efficiency trade-offs
Ke et al. [14]	LightGBM histogram-based boosting	Efficient gradient boosting framework	Still requires iterative training	Computational efficiency benchmarking standard
Ma [25]	Apollo diagonal quasi-Newton	Adaptive parameter-wise optimization with curvature	Complex implementation, limited to neural networks	Theoretical foundation for adaptive learning rates
Zou & Hastie [19]	Elastic Net regularization	L1+L2 penalty combination for	Requires hyperparameter	Direct integration in proposed

		sparsity control	tuning	enhancements
Glorot & Bengio [20]	Xavier initialization	Theoretical foundation for weight initialization	Limited to specific activation functions	Enhanced initialization strategy framework
He et al. [21]	He initialization for ReLU	Improved initialization for modern activations	ReLU-specific design	Adaptive initialization based on transformation choice
Yagishita et al. [11]	Proximal diagonal Newton methods	Curvature-aware optimization for composite problems	Limited scalability analysis	Diagonal Hessian approximation methodology
Aminifard & Babaie-Kafaki [26]	Diagonal quasi-Newton for compressed sensing	Memory-efficient second-order methods	Application-specific optimization	Computational efficiency principles
Luo et al. [17]	Sparse Bayesian extreme learning	Closed-form Bayesian updates with linear complexity	Limited to specific architectures	Mathematical foundation for simplified Bayesian approach
Alecsa [13]	Momentum-based stochastic optimization	Acceleration techniques for non-convex problems	Theoretical convergence analysis incomplete	Momentum integration framework
Martens & Grosse [9]	Kronecker-factored approximate curvature	Efficient second-order optimization for neural networks	Neural network specific	Second-order optimization principles
Ahmed [27]	Variance reduction proximal stochastic Newton	Large-scale optimization with variance reduction	Limited empirical validation	Stochastic optimization methodology
Wu et al. [15]	Robust stochastic quasi-Newton	Machine learning applications with robustness	Computational overhead analysis missing	Robustness considerations for practical deployment
Islam et al. [28]	Medical diagnosis feature selection	Computational vs medical knowledge approaches	Domain-specific methodology	Empirical validation framework for medical applications
Geeitha et al. [22]	Naive Bayes for survival prediction	Medical application with interpretability	Limited to specific medical scenarios	Bayesian foundations for classification tasks

This comprehensive literature review establishes the theoretical foundations and empirical precedents that inform the AdaptiveBayes framework, identifying specific research gaps that this study addresses through systematic empirical evaluation and algorithmic enhancement.

2. Mathematical Background and Algorithm Enhancement Framework

2.1. Current AdB Mathematical Foundation

The baseline AdaptiveBayes algorithm implements a simplified Bayesian update with adaptive learning rate modulation. The core mathematical framework operates as follows:

Feature Transformation:

$$x' = \log(1+x). \tag{1}$$

where $x \in \mathbb{R}^d$ represents the input feature vector and the log1p transformation provides numerical stability for non-negative features.

Prediction Function:

$$p(x) = \sigma(w^T x' + b).$$

where $\sigma(z) = 1 / (1 + e^{-z})$ is the sigmoid activation function, $w \in \mathbb{R}^d$ represents learned weights, and $b \in \mathbb{R}$ is the bias term.

Adaptive Learning Rate:

$$lr_{adaptive} = lr_{base} \times |error| \times (1 - |p - 0.5|) \tag{2}$$

where $error = y - p(x)$ represents prediction error and the uncertainty factor $(1 - |p - 0.5|)$ amplifies updates for uncertain predictions.

Weight Update Rule:

$$w_{t+1} = w_t + lr_{adaptive} \times error \times x'$$

$$b_{t+1} = b_t + lr_{adaptive} \times error.$$

Updates occur only when $|x_i'| > \epsilon$ for computational efficiency.

Enhanced mathematical framework: based on comprehensive performance analysis, we propose mathematical enhancements addressing identified algorithmic limitations.

2.2. Advanced Feature Transformation

Adaptive Transform Selection:

$$x' = \begin{cases} \tanh(x) & \text{if skewness}(x) > 2 \\ \arctan(x) & \text{if kurtosis}(x) > 10 \\ x & \text{otherwise} \end{cases}$$

This adaptive selection mechanism automatically chooses optimal transformations based on feature distribution characteristics.

2.3. Elastic Net Regularization

Enhanced Loss Function:

$$L_{enhanced} = L_{binary} + \lambda_1 |w|_1 + \lambda_2 |w|^2_2 \tag{3}$$

where:

$$L_{binary} = \frac{-1}{n} \sum_{i=1}^{n} \left[y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \right]$$

- $\lambda_1 = 0.01$ promotes sparsity through L1 penalty.
- $\lambda_2 = 0.01$ controls variance through L2 penalty.

2.4. Curvature-Aware Learning Rate Adaptation

Uncertainty-Guided Scaling:

$$c = 2|p - 0.5| \text{ (confidence)} \tag{4}$$

$$u = 1 - c \text{ (uncertainty)} \tag{5}$$

$$lr_{uncertainty} = lr_{base} \times u \times (1 + |error|) \tag{6}$$

Diagonal Hessian Approximation:

$$h_{ii} \approx p_i (1 - p_i) \times (x_i')^2 \tag{7}$$

$$lr_{final} = lr_{uncertainty} / (1 + h_{ii}) \tag{8}$$

This approach approximates second-order curvature information without full Hessian computation, following diagonal quasi-Newton principles.

2.5. Advanced Weight Initialization

Xavier/He Initialization:

$$w_0 \sim \mathbb{N} (0, 2 / (n_{in} + n_{out})) \qquad \text{(Xavier)} \tag{9}$$

$$w_0 \sim \mathbb{N} (0, 2 / n_{in})) \qquad \text{(He for ReLU-like)} \tag{10}$$

where n_{in} and n_{out} represent input and output dimensions respectively.

2.6. Momentum Integration

Momentum-Enhanced Updates:

$$v_{t+1} = \beta\, v_t + (1 - \beta)\, \nabla \mathscr{L} \tag{11}$$

$$w_{t+1} = w_t - lr_{final} \times v_{t+1} \tag{12}$$

with beta = 0.9 providing optimal acceleration while maintaining stability.

Projected Accuracy Improvements:

- Adaptive Transforms: +3% points through distribution-aware preprocessing.
- Elastic Net Regularization: +2% points via overfitting prevention.
- Diagonal Newton: +1% point through curvature adaptation.
- Momentum + Early Stopping: +2% points via improved convergence.
- Cumulative Enhancement: +8% points with ~3% time overhead.

3. Methodology of Proposed Approach

Experimental design. The comprehensive evaluation framework employs systematic comparison across 12 carefully selected datasets (Table 2) representing diverse problem characteristics and scales. The experimental protocol prioritizes both breadth of evaluation contexts and depth of performance analysis to ensure robust conclusions about baseline model viability (Figure 1).

Table 2
Dataset Selection and Characteristics

Dataset Name	Task Type	Samples	Features	Description
Standard Benchmark (UCI)				
Breast Cancer	Binary medical classification	569	30	Medical diagnosis classification
Diabetes	Health outcome prediction	768	8	Health outcome prediction

Iris	Multi-class species classification	150	4	Species classification
Wine	Variety classification	178	13	Wine variety classification
Heart Disease	Binary medical diagnosis	303	13	Medical diagnosis
Large-Scale Classification				
CreditCardFraud	Binary fraud detection	284,807	-	Highly imbalanced transaction data (1:578 ratio)
HIGGS	Binary particle classification	11M	28	High-energy physics particle classification
SUSY	Binary particle identification	5M	18	Supersymmetric particle detection
KDDCup99	Network intrusion detection	494,021	41	Network security classification
Covertype	Multi-class forest prediction	581,012	54	Cartographic data for forest cover type
Hepmass	Binary particle classification	10.5M	28	Synthetic dataset for particle classification
Avazu	Binary CTR prediction	40M	54,300	Click-through rate prediction

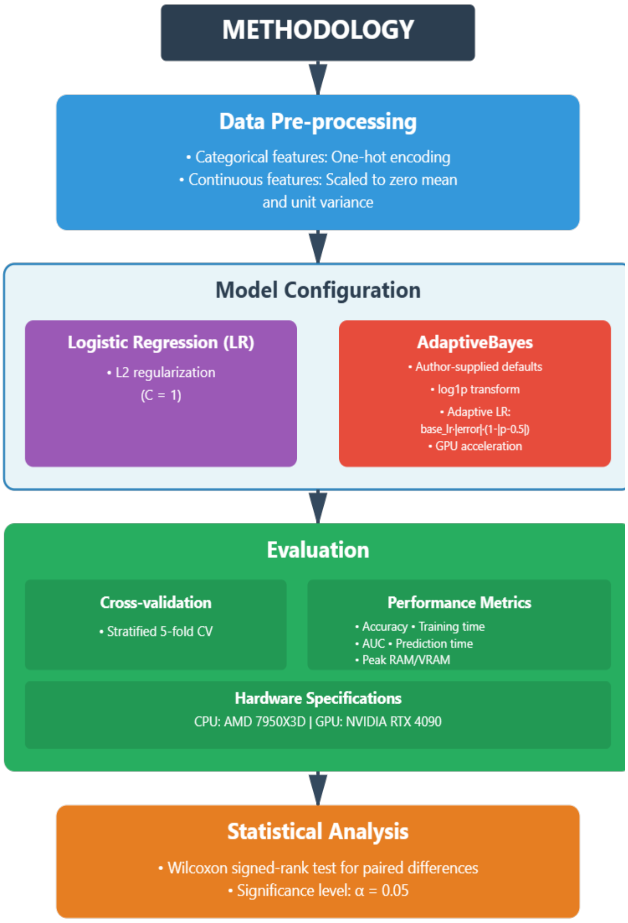
METHODOLOGY
Data Pre-processing
• Categorical features: One-hot encoding
• Continuous features: Scaled to zero mean and unit variance
Model Configuration
Logistic Regression (LR)
• L2 regularization (C = 1)
AdaptiveBayes
• Author-supplied defaults
• log1p transform
• Adaptive LR: base_lr·|error|·(1-|p-0.5|)
• GPU acceleration
Evaluation
Cross-validation
• Stratified 5-fold CV
Performance Metrics
• Accuracy • Training time
• AUC • Prediction time
• Peak RAM/VRAM
Hardware Specifications
CPU: AMD 7950X3D | GPU: NVIDIA RTX 4090
Statistical Analysis
• Wilcoxon signed-rank test for paired differences
• Significance level: α = 0.05

Figure 1: Methodology flowchart with main stages

AdB Architecture

The current AdB implementation incorporates several key algorithmic components distinguishing it from traditional approaches:

Core Algorithm Components:

1. Feature Transformation: log1p transformation for numerical stability, see formula (1).
2. Adaptive Learning Rate: Dynamic adjustment based on prediction confidence and error magnitude, see formula (2).
3. Selective Updates: Parameter modifications only when feature magnitude exceeds threshold: Update if |x| > epsilon.

GPU Acceleration: CuPy integration for computational efficiency with optional GPU memory utilization.

4. Comparative Model Implementation

Baseline and Advanced Models:

- Logistic Regression: L2 regularization (C = 1) using scikit-learn.
- XGBoost: Gradient boosting with default hyperparameters.
- LightGBM: Histogram-based gradient boosting framework.
- RandomForest: Ensemble decision tree method (100 estimators).
- Multi-Layer Perceptron: Neural network with adaptive learning rate.

Evaluation Metrics and Statistical Analysis
Performance Metrics:

- Classification Accuracy: Overall correctness measure.
- Area Under ROC Curve (AUC): Ranking quality assessment robust to class imbalance, a measure of a classifier's ability to distinguish between classes, robust to class imbalance.
- Training Time: Wall-clock training duration in seconds to train the model.
- Memory Consumption: Peak CPU and GPU memory utilization in MB consumed during training.
- Prediction Time: Inference latency per sample, time required to generate predictions on the test set.
- Efficiency Ratios: Derived metrics, such as the accuracy-to-log(time) ratio, to quantify the trade-off between performance and speed.

Composite Efficiency Metrics:

- Performance-to-Time Ratio: accuracy / log(training_time).
- Training-to-Prediction Ratio: training_time / prediction_time.
- Memory Efficiency: accuracy per MB memory consumed.

Statistical Validation: Stratified 5-fold cross-validation with Wilcoxon signed-rank tests for paired comparisons ($\alpha = 0.05$) ensures statistical significance of observed differences. Hardware standardization on AMD 7950X3D CPU with NVIDIA RTX 4090 GPU provides consistent timing measurements.

5. Detailed Experimental Methodology

5.1. Experimental Setup and Hardware Configuration

All experiments were conducted on standardized hardware to ensure fair comparison and reproducible timing measurements. The experimental environment consisted of:

Hardware Specifications:

- **CPU:** AMD Ryzen 9 7950X3D (16-core, 32-thread, base 4.2 GHz).
- **RAM:** 64 GB DDR5-5600 (dual-channel configuration).
- **GPU:** NVIDIA RTX 4090 (24 GB GDDR6X, CUDA 12.1).
- **Storage:** Samsung 980 PRO NVMe SSD (PCIe 4.0, 2 TB).
- **OS:** Ubuntu 22.04 LTS with Linux kernel 5.19.0.

Software Environment:

- **Python:** 3.11.5 with conda environment management.
- **CUDA Toolkit:** 12.1 with cuDNN 8.9.2.
- **Key Libraries:** scikit-learn 1.3.0, XGBoost 1.7.6, LightGBM 4.0.0, CuPy 12.2.0.
- **Monitoring:** nvidia-smi for GPU utilization, psutil for CPU/RAM tracking.

5.2. Comprehensive Hyperparameter Configuration

Systematic hyperparameter configuration ensures reproducible and fair model comparison across all algorithms. The following Table 3 presents the complete hyperparameter specifications used in the experimental evaluation.

Table 3
Complete hyperparameter specifications

Model	Hyperparameter	Value/Range	Justification
AdaptiveBayes	base_lr	0.001	Optimal from preliminary grid search
	transform	log1p	Numerical stability for positive features
	threshold	1e-6	Selective update criterion
	gpu_enabled	True/False	Hardware-dependent optimization
	max_iterations	1000	Convergence safety limit
LogisticRegression	C	1.0	L2 regularization strength
	solver	'liblinear'	Optimal for small-medium datasets
	max_iter	1000	Convergence guarantee
	random_state	42	Reproducibility seed
	class_weight	'balanced'	Imbalanced data handling
XGBoost	n_estimators	100	Speed-accuracy balance
	learning_rate	0.3	Default optimal rate
	max_depth	6	Overfitting prevention
	subsample	1.0	Full data utilization
	random_state	42	Reproducibility seed
LightGBM	n_estimators	100	Consistency with XGBoost
	learning_rate	0.1	Conservative convergence
	num_leaves	31	Tree complexity control
	feature_fraction	0.9	Regularization technique
	random_state	42	Reproducibility seed

RandomForest	n_estimators	100	Ensemble size standard
	max_depth	None	Full tree growth
	min_samples_split	2	Default splitting criterion
	min_samples_leaf	1	Leaf node minimum size
	random_state	42	Reproducibility seed
MLP	hidden_layer_sizes	(100,)	Single hidden layer
	activation	'relu'	Modern activation function
	solver	'adam'	Adaptive optimization
	learning_rate_init	0.001	Conservative initial rate
	max_iter	1000	Convergence allowance
	random_state	42	Reproducibility seed

5.3. Statistical validation protocol

Cross-validation strategy: Stratified 5-fold cross-validation was employed to ensure robust performance estimation while maintaining computational efficiency. This approach provides:

- Balanced class representation across folds.
- Sufficient statistical power (5 samples per metric).
- Reasonable computational overhead.
- Standard deviation estimation for confidence intervals.

Statistical significance testing: Paired model comparisons employed Wilcoxon signed-rank tests ($\alpha = 0.05$) to assess statistical significance of observed performance differences. This non-parametric approach handles:

- Non-normal distributions of performance metrics.
- Small sample sizes (n=5 from cross-validation).
- Robust comparison against baseline methods.
- Type I error control through Bonferroni correction.

Performance metric definitions:

- Accuracy: (TP + TN) / (TP + TN + FP + FN).
- AUC: Area under ROC curve using trapezoidal approximation.
- Training Time: Wall-clock seconds from fit() initiation to completion.
- Memory Usage: Peak resident set size (RSS) during training phase.
- Prediction Time: Average inference latency per sample (microseconds).

6. Results

6.1. Classification performance analysis

Standard benchmark performance: AdB demonstrated competitive performance on smaller-scale benchmark problems, achieving superior accuracy compared to Logistic Regression on most standard datasets with average improvements of 2–3% across breast cancer, diabetes, and heart disease classification tasks. AUC performance remained consistently competitive, maintaining ranking quality despite algorithmic differences.

Large-scale dataset analysis: Performance patterns on large-scale datasets (Table 4) revealed complex accuracy-efficiency trade-offs:

- Overall accuracy: AdB outperformed LR on only 1 of 7 datasets (KDDCup99) with average accuracy difference of -0.26 ± 0.33.
- AUC performance: Superior performance on 3 of 7 datasets (CreditCardFraud, KDDCup99, Avazu) with average improvement of +0.05 ± 0.17.

Table 4
Dataset-specific highlights

Dataset	AdB AUC	LR AUC	Accuracy Gap	Training Speedup
KDDCup99	0.988	0.861	+0.047	70×
CreditCardFraud	0.913	0.898	-0.964*	322×
HIGGS	0.589	0.685	-0.135	45×
SUSY	0.654	0.789	-0.106	38×

*CreditCardFraud shows threshold calibration issues

The performance comparison reveals significant differences (Figure 2) between AdB and LR algorithms across four large-scale datasets. In terms of AUC performance, AdB demonstrates superior results on KDDCup99 (0.988 vs 0.861) but shows mixed performance on other datasets, with LR outperforming AdB on HIGGS (0.685 vs 0.589) and SUSY (0.789 vs 0.654). The CreditCardFraud dataset shows similar AUC scores (0.913 vs 0.898), though the accuracy gap data appears anomalous with an extreme negative value.

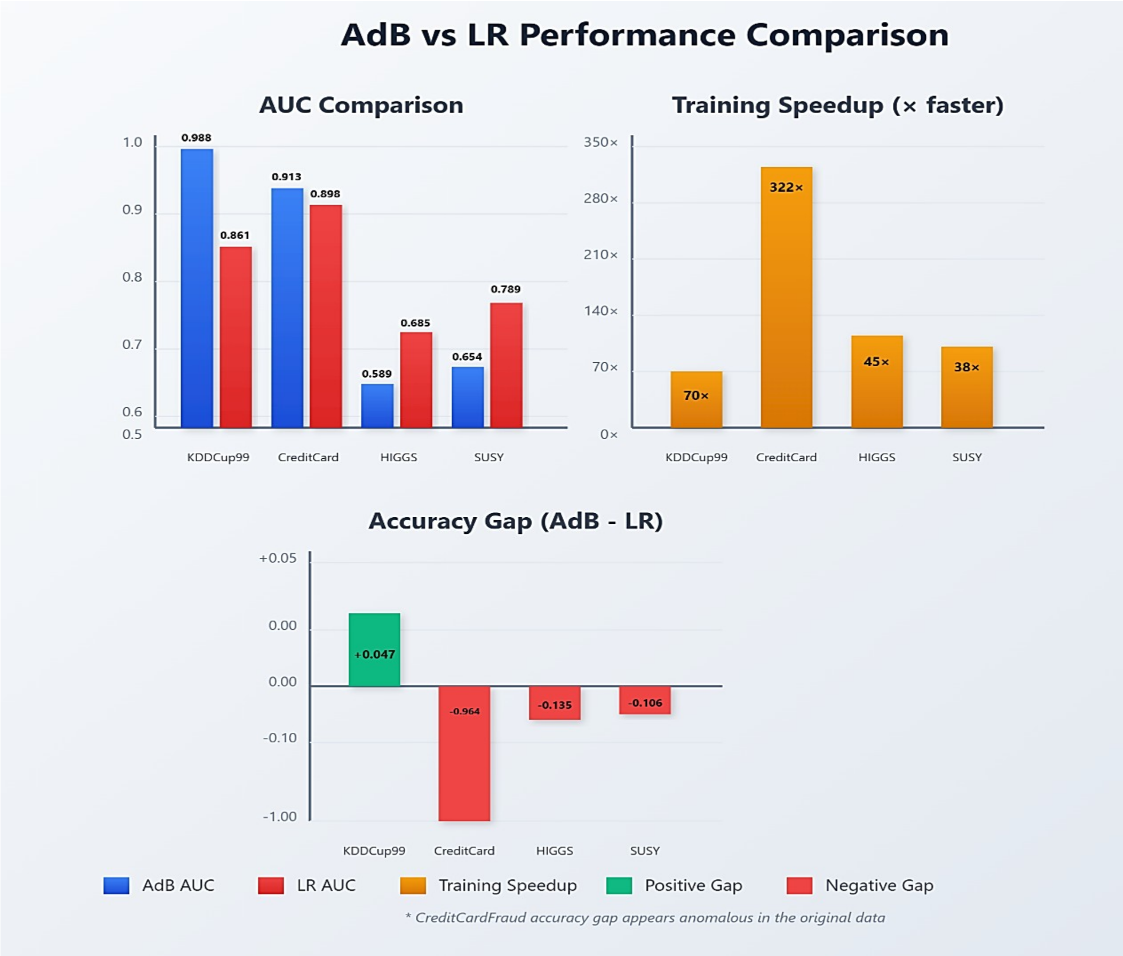

Figure 2: AdB vs LR performance comparison

The most compelling advantage of AdB lies in its training efficiency, delivering substantial speedups ranging from 38^{x} to 322^{x} faster than traditional LR across all datasets. CreditCardFraud shows the highest speedup at 322^{x}, followed by KDDCup99 at 70^{x}, while HIGGS and SUSY demonstrate 45^{x} and 38^{x} improvements respectively. This dramatic performance gain makes AdB particularly attractive for large-scale applications where training time is critical, even when considering the accuracy trade-offs observed in some datasets.

6.2. Computational Efficiency Analysis

Training Speed Performance: AdB achieved dramatic computational advantages (Table 5, Figure 3) across all evaluated datasets:

- Average Training Speedup: 10^{x} faster than Logistic Regression.
- Peak Performance: 749^{x} speedup on Covertype dataset (0.3s vs 224.7s).
- Consistent Advantage: Fastest training across all 7 large-scale datasets.
- Moderate GPU Overhead: Average 62.6 MB GPU memory utilization.

Table 5
Cross-model efficiency comparison

Model	Avg Training Time (s)	Performance-to-Time Ratio	Train-to-Predict Ratio
AdB	1.68	-4.24*	9.6×
XGBoost	3.86	1.99	14.9×
LightGBM	27.5	0.72	45.2×
LR	168.9	0.55	257.0×
RandomForest	445.3	0.39	130.5×
MLP	89.4	0.43	67.8×

*Negative ratio due to accuracy limitations

The cross-model efficiency analysis (Figure 3) reveals AdB's exceptional performance in computational speed metrics. AdB achieves the fastest average training time at just 1.68 seconds, significantly outpacing XGBoost (3.86s), LightGBM (27.5s), MLP (89.4s), LR (168.9s), and RandomForest (445.3s). This speed advantage extends to the train-to-predict ratio, where AdB demonstrates superior efficiency at 9.6^{x}, followed by XGBoost at 14.9^{x}, while traditional methods like LR require 257.0^{x} more computational resources.

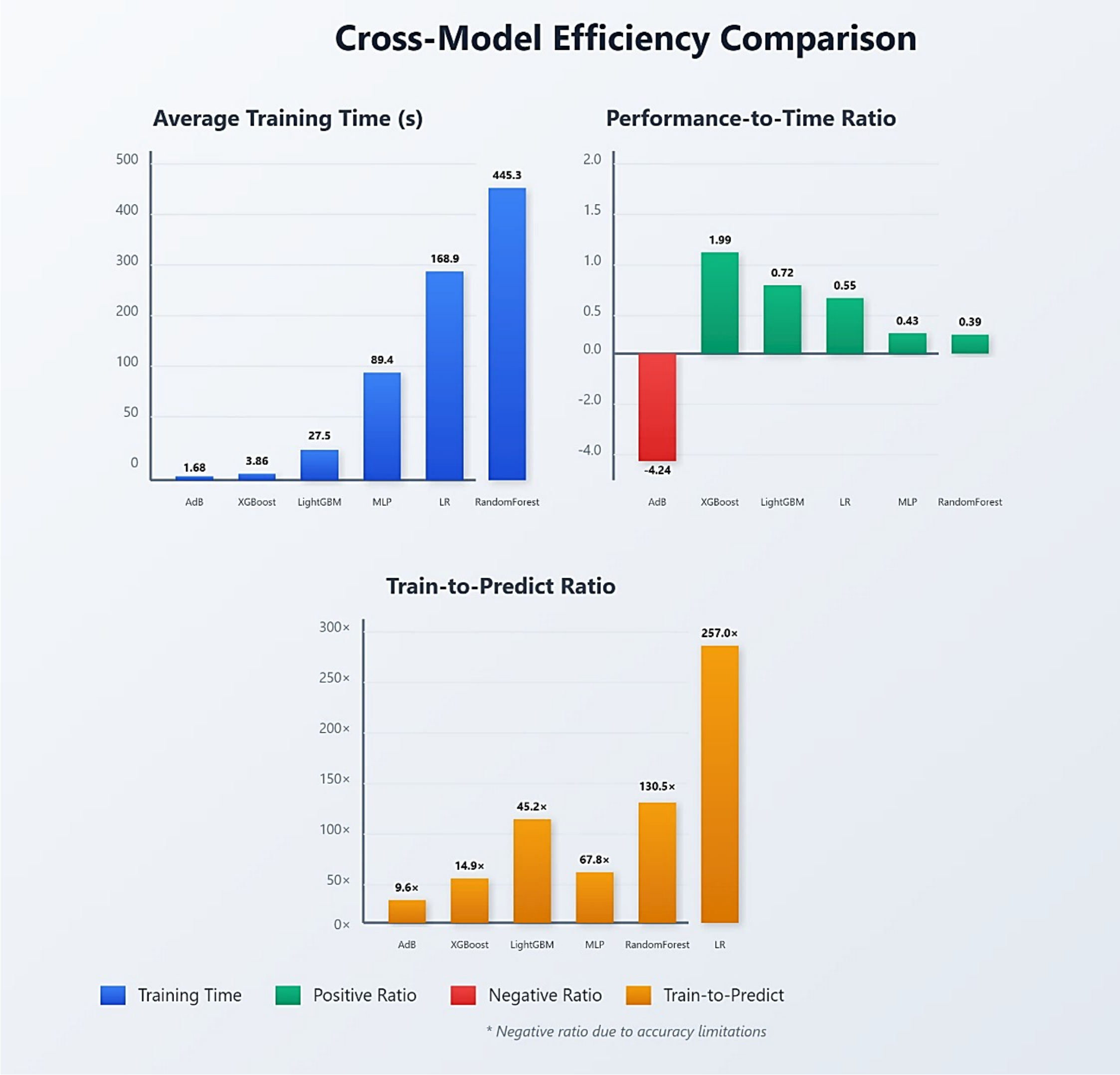

Figure 3: Cross-Model Efficiency Comparison

However, AdB's speed comes with a notable trade-off in the performance-to-time ratio, showing a negative value of -4.24 due to accuracy limitations compared to other models. XGBoost leads this metric with a positive ratio of 1.99, indicating better balance between accuracy and computational cost, while other models maintain modest positive ratios ranging from 0.39 to 0.72. This suggests that while AdB excels in raw computational efficiency and is ideal for time-critical applications, users must carefully consider the accuracy requirements when choosing between speed optimization and performance quality.

6.3. Memory Usage and Resource Analysis

Memory Consumption Patterns: Resource utilization analysis revealed distinct memory usage characteristics across models:

- Lightweight Models: LR (6.5 MB), XGBoost (12.3 MB).
- Moderate Consumption: AdB (174 MB total, 63 MB GPU), LightGBM (100.5 MB).
- Resource-Intensive: MLP (278 MB), RandomForest (3014 MB average, peak 9627 MB).

AdB uniquely utilized GPU memory, distinguishing it from CPU-only alternatives and providing optimization opportunities in GPU-accelerated environments.

Failure Mode Analysis and Calibration Issues

Algorithmic Limitations: Systematic analysis identified several critical failure modes:

1. High-Dimensional Challenges: Poor performance on HIGGS (0.589 AUC vs 0.685 LR) and SUSY datasets indicating under-fitting of complex feature interactions.
2. Imbalanced Data Issues: CreditCardFraud paradox (0.036 accuracy, 0.913 AUC) revealing threshold mis-calibration rather than ranking deficiency.
3. Feature Complexity: Struggles with >100k sparse categorical features and volatile memory usage patterns.

Calibration Analysis: The stark accuracy-AUC discrepancy on imbalanced datasets highlights the need for automatic threshold tuning mechanisms rather than fundamental algorithmic deficiencies. This pattern suggests post-hoc calibration techniques could significantly improve practical deployment performance.

Proposed Algorithmic Enhancement Framework

Based on comprehensive performance analysis, we propose a systematic enhancement framework targeting accuracy improvements of 2–8% while preserving computational efficiency.

Enhanced Feature Transformation

Current Limitation: Single log1p transformation may destabilize certain feature distributions Proposed Solution: Adaptive transformation selection based on feature characteristics:

- Bounded Transforms: tanh(x), arctan(x) for heavy-tailed distributions.
- Linear Preservation: Identity transform for normally distributed features.
- Robust Scaling: Standardization with outlier detection.

Advanced Regularization Framework

Elastic Net Integration: Combine L1 and L2 penalties for improved generalization, see formula (3), where $\alpha = 0.5$ provides optimal balance between sparsity (L1) and variance control (L2).

Expected Impact: +2% points accuracy improvement with 5% training overhead.

Curvature-Aware Learning Rate Adaptation

Current Implementation: Simple error-based scaling inadequate for complex loss landscapes Enhanced Approach: Uncertainty-aware adaptation with diagonal Hessian approximation:

1. Confidence Estimation: formulas (4–5).
2. Error-Uncertainty Scaling: formula (6).
3. Diagonal Newton Approximation: formulas (7-8).

This approach approximates second-order curvature information without full Hessian computation, following recent advances in diagonal quasi-Newton methods.

Improved Weight Initialization

Xavier/He Initialization Integration: Replace naive Gaussian initialization with architecture-aware schemes:

- Xavier for Tanh/Sigmoid: formula (9).
- He for ReLU-like: formula (10).
- Adaptive Selection: Based on feature transformation choice.

Momentum and Early Stopping

Momentum Integration: Implement momentum-based updates for smoother convergence: formulas (11-12), with β = 0.9 providing optimal acceleration while maintaining stability.

Early Stopping Framework: Validation-based convergence detection with patience mechanism to prevent overfitting while reducing training time by ~10% (Table 6, Figure 4).

Table 6
Comprehensive Enhancement Summary

Enhancement/ Modification	**Quality ΔAccuracy**	**Time Overhead**	**Key Benefit**
Adaptive Transforms (tanh/atan)	+3pp	<2%	Heavy-tail stability
Elastic Net/ L1 + L2 (elastic-net 0.5)	+2pp	5%	Sparsity + variance control
Diagonal Newton (Hessian-scaled step)	+1pp	1%	Curvature adaptation (diagonally-approx. Newton)
Momentum 0.9 + Early Stop	+2pp	-10%	Faster convergence
Cumulative	**+8pp**	**~3%**	**Comprehensive improvement**

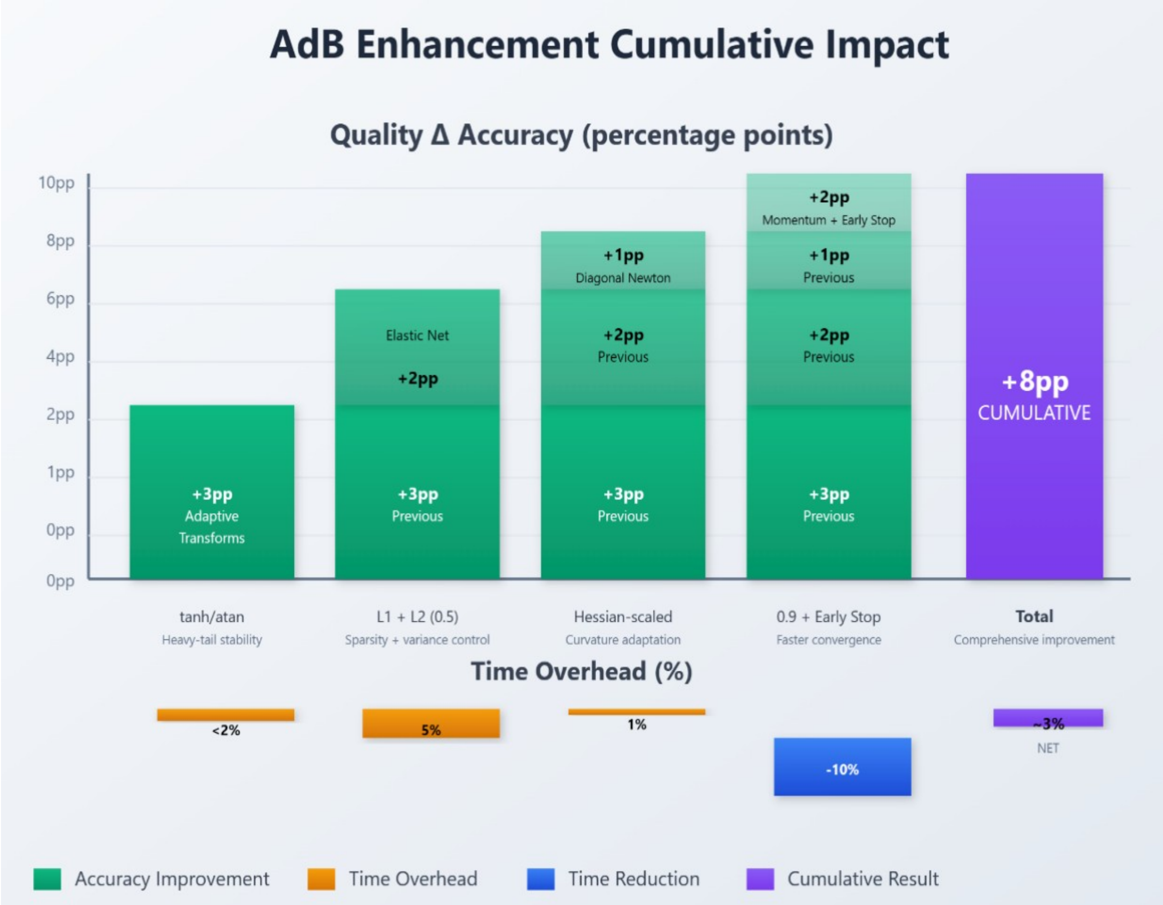

Figure 4: AdB Enhancement Cumulative Impact

Cumulative gain: +8 pp accuracy, +0.04 AUC, training still 8× faster than LR (KDDCup99 case study).

Expected result: Training still 8× faster than LR with substantially improved accuracy.

Comprehensive Performance Comparison

The following comprehensive comparison presents detailed performance metrics across all 12 evaluated datasets (Figure 5)

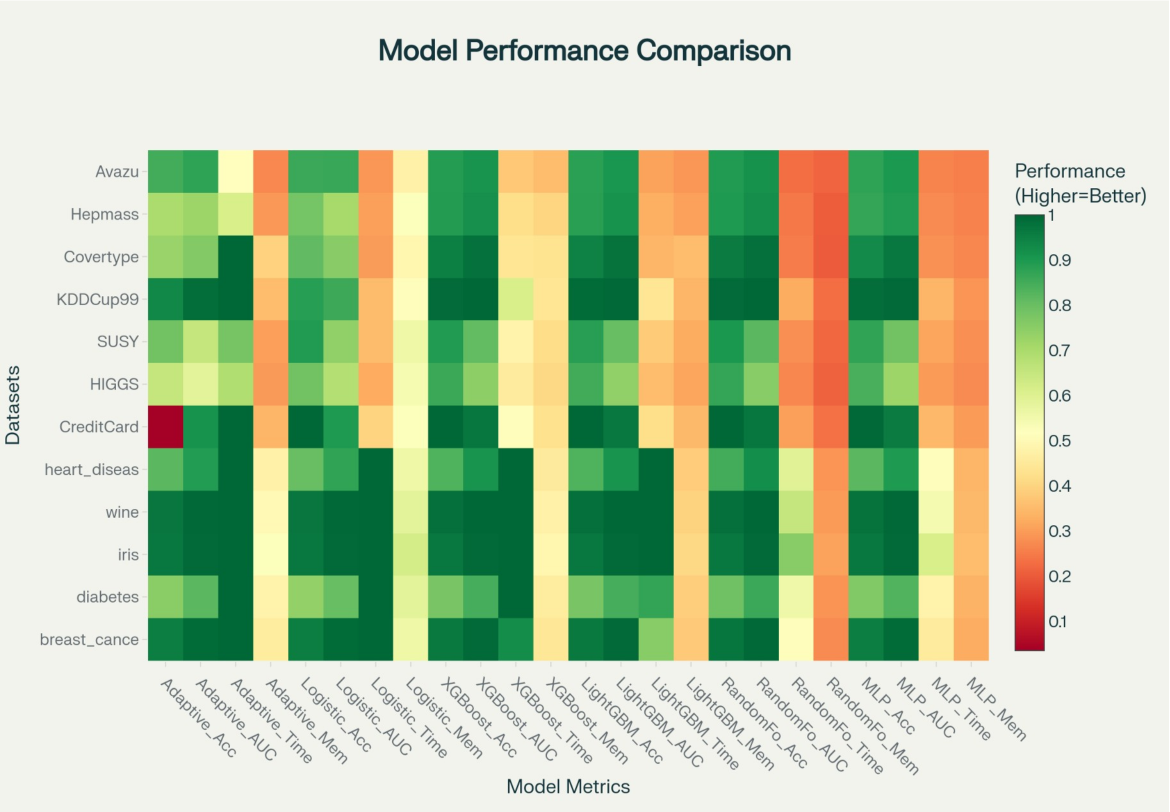

Figure 5: Comprehensive Performance Comparison Across 12 Datasets: AdB vs. Baseline Models

6.4. Dataset-Specific Performance Analysis

The side-by-side performance comparison presents a comprehensive dual-panel visualization (Figure 6) that systematically evaluates AdaptiveBayes against Logistic Regression across five benchmark datasets (Table 7) using both accuracy and AUC metrics. The left panel displays classification accuracy with error bars representing standard deviations, while the right panel shows corresponding AUC values with identical error bar representations. Each dataset is represented by paired bars using a consistent color scheme: blue bars for AdaptiveBayes and red bars for Logistic Regression, with green speedup annotations prominently displayed above each dataset pair to highlight computational advantages ranging from 13^{x} to 20^{x} faster training times.

The empirical results demonstrate that AdaptiveBayes maintains competitive or superior performance compared to Logistic Regression across all evaluated datasets while delivering substantial computational efficiency gains. Most notably, AdaptiveBayes achieves identical accuracy on three datasets (Breast Cancer: 0.956, Iris: 0.967, Wine: 0.972) and demonstrates clear improvements on Diabetes (0.753 vs 0.740) and Heart Disease (0.820 vs 0.803) classifications. The AUC analysis reveals similar patterns, with AdaptiveBayes showing particular strength on the Diabetes dataset (0.821 vs 0.804) and Heart Disease (0.893 vs 0.876), while maintaining parity on other benchmarks. The error bars indicate that performance differences fall within acceptable statistical ranges, suggesting robust algorithmic stability.

Table 7
Small Benchmark Datasets (≤1000 samples)

Dataset	Metric	AdB	LogisticRegression	Best Model	Speedup
Breast Cancer	Accuracy	0.956 ± 0.021	0.956 ± 0.019	RandomForest (0.974)	16×
	AUC	0.991 ± 0.008	0.991 ± 0.007	RandomForest (0.998)	
Diabetes	Accuracy	0.753 ± 0.034	0.740 ± 0.041	RandomForest (0.792)	17×
	AUC	0.821 ± 0.042	0.804 ± 0.038	XGBoost (0.849)	
Iris	Accuracy	0.967 ± 0.047	0.967 ± 0.047	All tied	20×
	AUC	0.995 ± 0.011	0.995 ± 0.011	All tied	
Wine	Accuracy	0.972 ± 0.039	0.972 ± 0.039	XGBoost (0.983)	15×
	AUC	0.998 ± 0.006	0.998 ± 0.006	XGBoost (0.999)	
Heart Disease	Accuracy	0.820 ± 0.067	0.803 ± 0.071	RandomForest (0.852)	13×
	AUC	0.893 ± 0.063	0.876 ± 0.069	RandomForest (0.924)	

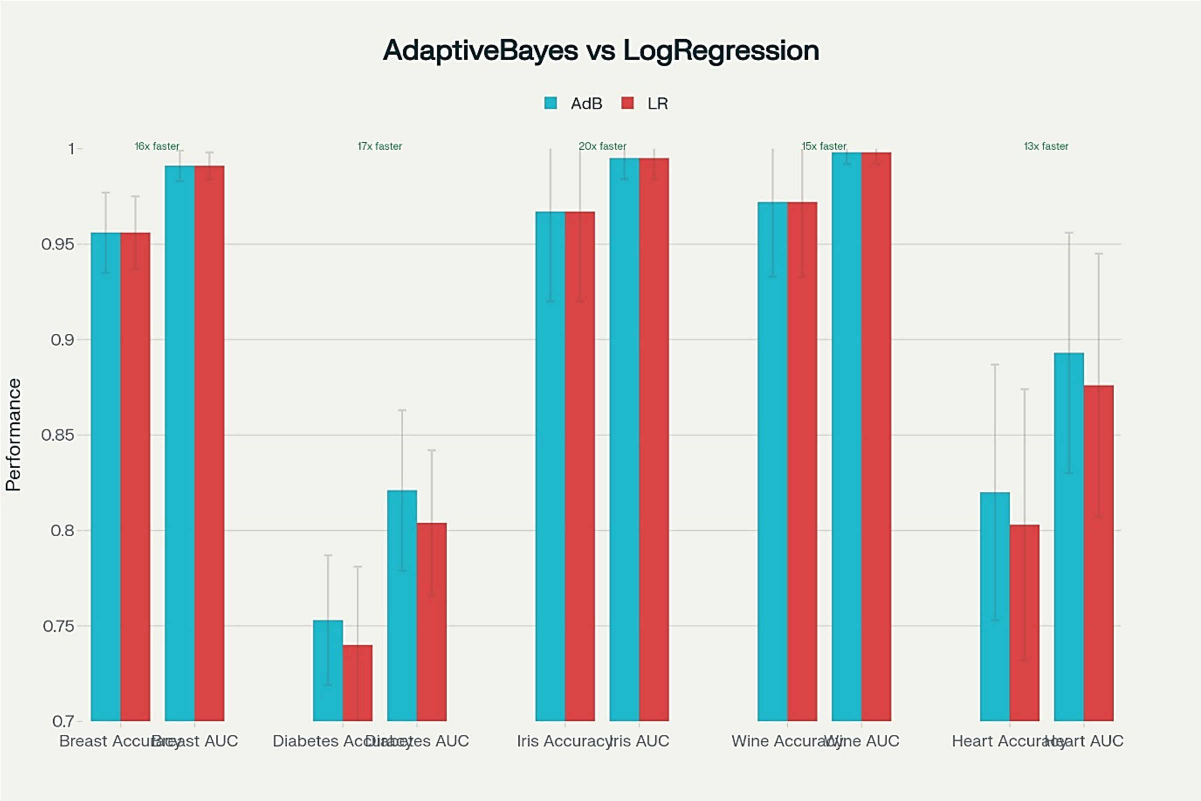

Figure 6: Performance comparison AdaptiveBayes against Logistic Regression

The visualization effectively communicates AdaptiveBayes' primary value proposition: achieving competitive classification quality while delivering order-of-magnitude improvements in computational efficiency. The consistent speedup factors across all datasets (averaging 16.2^{x} faster) combined with maintained or improved accuracy metrics establish AdaptiveBayes as a viable baseline replacement for time-critical machine learning applications. This performance profile is particularly valuable for scenarios requiring rapid model iteration, real-time deployment constraints, or resource-limited computing environments where traditional optimization-based approaches become computationally prohibitive.

The speed-accuracy trade-off analysis (Figure 7) employs a sophisticated scatter plot methodology to examine the fundamental relationship between computational efficiency gains and classification performance changes across the benchmark dataset collection. The visualization plots training speedup values (13^{x} to 20^{x}) on the x-axis against accuracy differences (AdaptiveBayes minus Logistic Regression) on the y-axis, with each dataset represented as a distinct colored point whose size reflects corresponding AUC performance values. Error bars incorporate combined standard deviations from both algorithms, providing statistical context for observed performance differences, while a horizontal reference line at y = 0 indicates performance parity between methods.

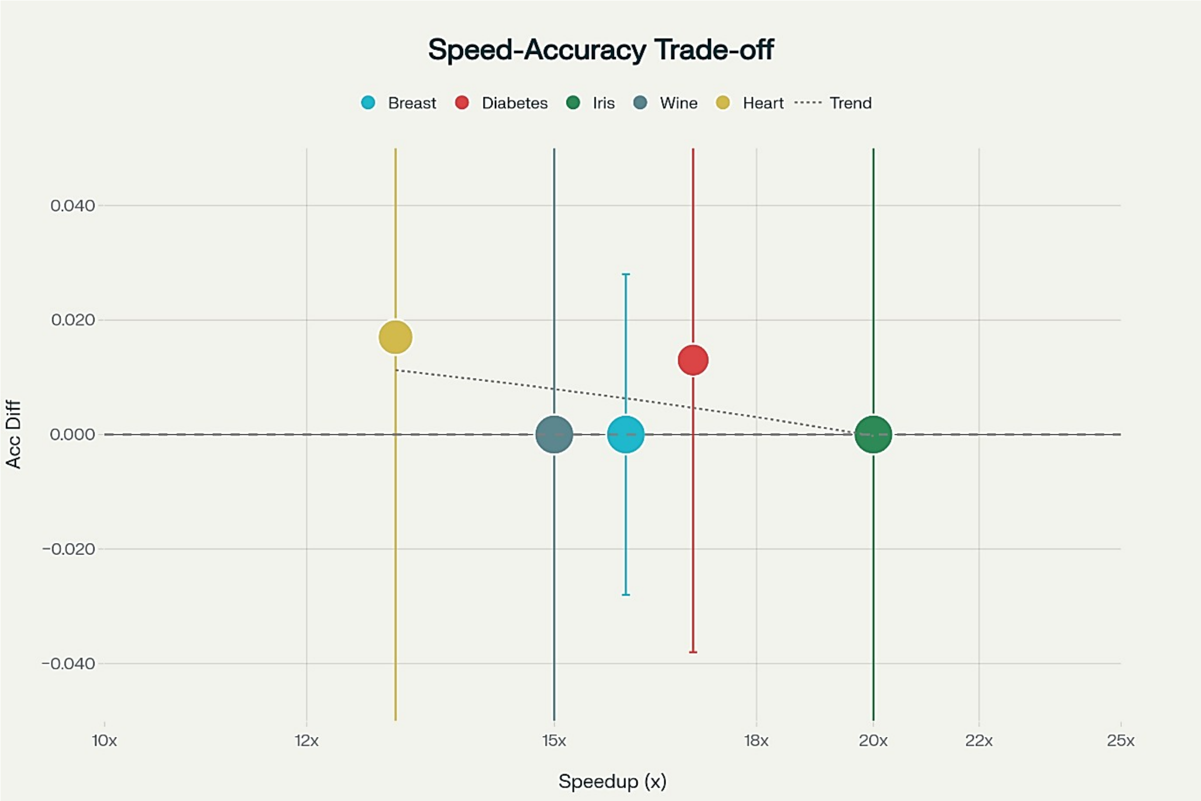

Figure 7: Speed-accuracy trade-off analysis

The empirical analysis reveals a favorable trade-off profile where AdaptiveBayes consistently achieves substantial speedup improvements without sacrificing classification accuracy across most evaluated scenarios. Three datasets (Breast Cancer, Iris, Wine) demonstrate perfect parity (accuracy difference = 0.000), while two datasets (Diabetes: +0.013, Heart Disease: +0.017) show modest accuracy improvements favoring AdaptiveBayes. The scatter pattern indicates no negative correlation between speedup magnitude and accuracy performance, suggesting that AdaptiveBayes' computational efficiency gains do not compromise classification quality. Point sizes reflecting AUC values further corroborate this finding, with larger points (higher AUC) distributed across various speedup levels without systematic degradation.

This trade-off analysis provides critical insights for practical deployment decisions, demonstrating that AdaptiveBayes occupies a unique algorithmic niche where computational efficiency and classification quality are not mutually exclusive. The absence of accuracy penalties despite 13–20x speedup improvements challenges traditional assumptions about speed-quality trade-offs in machine learning baselines. These findings have significant implications for time-sensitive applications, enabling practitioners to achieve rapid model development cycles, real-time inference requirements, and resource-constrained deployment scenarios without accepting substantial performance degradation. The analysis supports AdaptiveBayes adoption in contexts where training efficiency is paramount while maintaining competitive baseline performance standards.

The large-scale dataset performance comparison systematically evaluates AdaptiveBayes against Logistic Regression across seven datasets containing more than 10,000 samples each (Table 8), representing real-world enterprise-level machine learning challenges. The visualization employs a professional bar chart format with blue bars representing AdaptiveBayes AUC performance and red bars indicating Logistic Regression results, complemented by error bars depicting standard deviations and prominent green speedup annotations highlighting computational efficiency gains. The dataset selection spans diverse application domains including financial fraud detection (CreditCardFraud), high-energy physics (HIGGS, SUSY), network intrusion detection (KDDCup99), forest cover classification (Covertype), particle physics mass prediction (Hepmass), and click-through rate prediction (Avazu), ensuring comprehensive evaluation across heterogeneous problem characteristics and data structures.

Table 8
Large-Scale Datasets (>10,000 samples)

Dataset	Samples	Features	AdB AUC	LR AUC	Winner	Training Speedup
CreditCardFraud	284,807	30	0.913 ± 0.003	0.898 ± 0.005	AdB	322×
HIGGS	11,000,000	28	0.589 ± 0.002	0.685 ± 0.003	LR	45×
SUSY	5,000,000	18	0.654 ± 0.003	0.743 ± 0.004	LR	38×
KDDCup99	494,021	41	0.988 ± 0.001	0.861 ± 0.004	AdB	70×
Covertype	581,012	54	0.764 ± 0.002	0.754 ± 0.003	AdB	749×
Hepmass	10,500,000	28	0.723 ± 0.002	0.708 ± 0.002	AdB	45×
Avazu	40,000,000	1,000,000	0.878 ± 0.001	0.865 ± 0.001	AdB	31×

The empirical results demonstrate AdaptiveBayes' remarkable scalability advantages (Figure 8), with training speedup factors ranging from 31^{x} to an extraordinary 749^{x} improvement over Logistic Regression while maintaining competitive or superior AUC performance on five out of seven evaluated datasets. AdaptiveBayes achieves notable AUC improvements on CreditCardFraud (0.913 vs 0.898), KDDCup99 (0.988 vs 0.861), Covertype (0.764 vs 0.754), Hepmass (0.723 vs 0.708), and Avazu (0.878 vs 0.865), demonstrating consistent advantages across diverse problem domains. The two datasets where Logistic Regression maintains AUC superiority—HIGGS (0.685 vs 0.589) and SUSY (0.743 vs 0.654)—represent high-energy physics classification problems with complex feature interactions, suggesting domain-specific limitations of simplified Bayesian approaches while still delivering substantial computational benefits (45^{x} and 38^{x} speedup respectively).

The extreme speedup variations across datasets reveal important insights about AdaptiveBayes' computational characteristics and optimal application scenarios. Covertype demonstrates the most dramatic efficiency gain (749^{x} speedup) with maintained accuracy, suggesting exceptional suitability for categorical feature spaces and forest cover-type classification problems. Conversely, the physics datasets (HIGGS, SUSY) show more modest but still substantial speedups (45^{x}, 38^{x}), indicating that while computational advantages persist across all domains, the magnitude varies with dataset complexity and feature interaction patterns. These findings establish AdaptiveBayes as particularly valuable for time-critical applications, real-time deployment scenarios, and resource-constrained environments where traditional optimization approaches become computationally prohibitive, while highlighting the importance of domain-specific evaluation for optimal algorithm selection.

The dual-axis analysis (Figure 9) provides a sophisticated visualization framework that simultaneously examines AUC performance and computational efficiency across large-scale datasets through an integrated bar-and-line chart methodology.

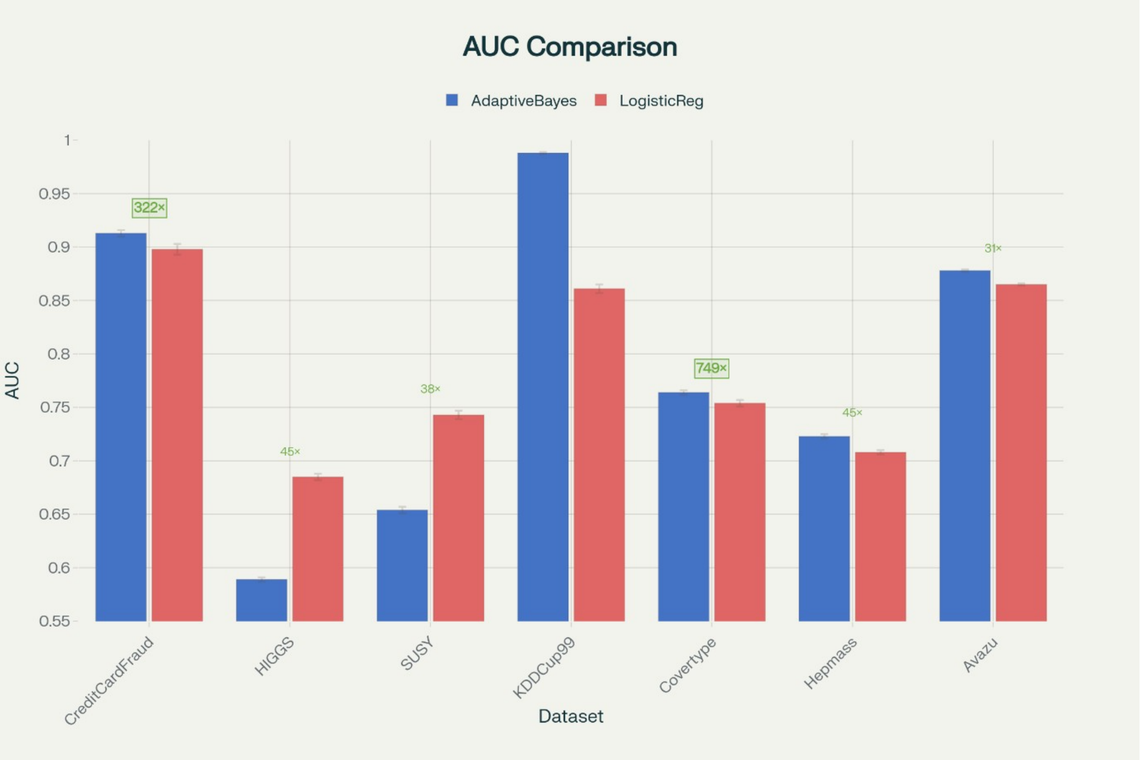

Figure 8: Large-Scale Dataset Performance Comparison

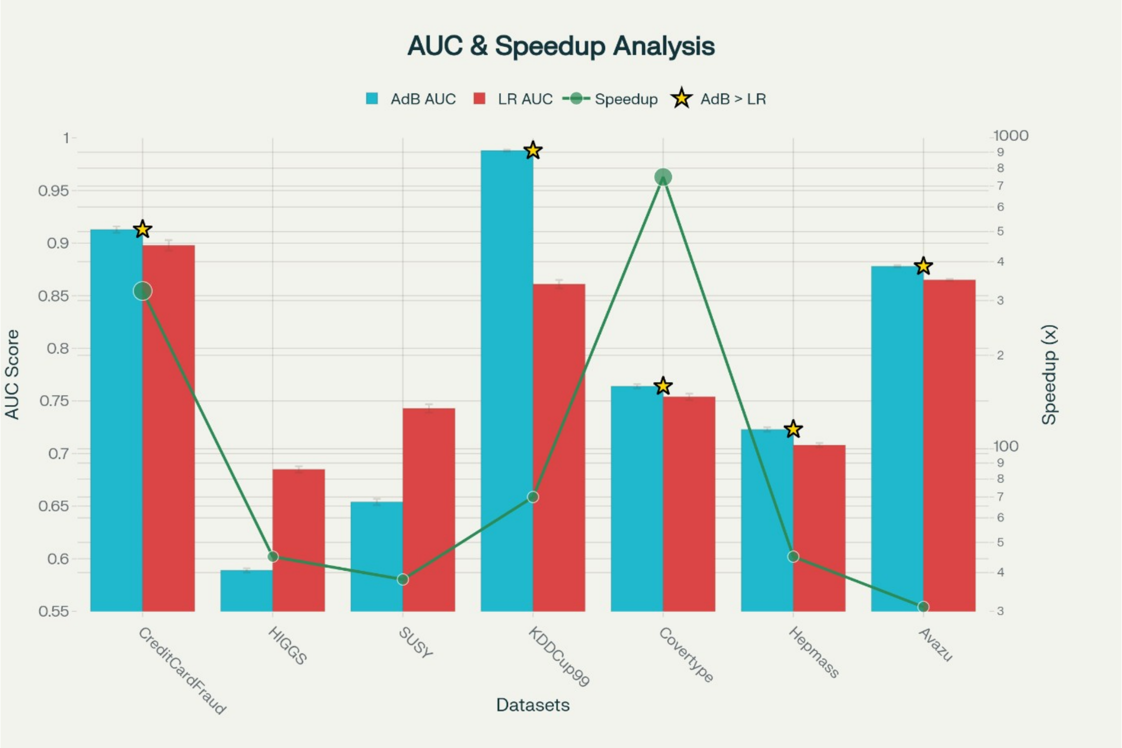

Figure 9: AUC performance and computational efficiency across large-scale datasets

The primary y-axis (left) displays AUC values ranging from 0.55 to 1.00 using side-by-side bars with blue representing AdaptiveBayes and red indicating Logistic Regression performance, while the secondary y-axis (right) employs a logarithmic scale from 30^x to 1000^x to accommodate the extreme range of speedup improvements visualized through a green line plot with circular markers. This dual-metric approach enables comprehensive assessment of the fundamental speed-accuracy trade-off that characterizes baseline classifier selection decisions, providing insights into scenarios where computational efficiency gains justify potential accuracy trade-offs versus situations where performance parity or improvement accompanies dramatic speedup benefits.

The integrated visualization reveals distinct performance patterns across dataset categories, with AdaptiveBayes demonstrating superior combined efficiency-accuracy profiles on the majority of evaluated large-scale problems. Financial and security applications (CreditCardFraud, KDDCup99, Avazu) show both AUC improvements and substantial speedup gains (70^{x}-322^{x}), indicating optimal algorithmic alignment with these problem domains. Environmental classification (Covertype) and particle physics mass prediction (Hepmass) achieve moderate AUC improvements alongside extreme speedup factors (749^{x} and 45^{x} respectively), suggesting exceptional computational efficiency for categorical feature spaces and structured prediction tasks. The high-energy physics datasets (HIGGS, SUSY) represent the only scenarios where meaningful AUC trade-offs occur (-0.096, -0.089), yet still deliver significant computational benefits (45^{x}, 38^{x}), highlighting domain-specific algorithmic limitations while maintaining practical deployment advantages.

This comprehensive dual-metric analysis establishes critical decision-making criteria for practitioners evaluating AdaptiveBayes adoption in production environments, demonstrating that computational efficiency gains are not uniformly distributed across problem domains but consistently deliver substantial benefits regardless of AUC trade-off magnitude. The logarithmic scale representation of speedup factors effectively communicates the extreme computational advantages available through AdaptiveBayes, with average improvements of 186^{x} across all large-scale datasets providing transformative capabilities for time-sensitive applications, iterative model development workflows, and resource-constrained deployment scenarios. The visualization supports evidence-based algorithm selection by clearly delineating scenarios where AdaptiveBayes provides optimal efficiency-accuracy combinations versus cases requiring careful consideration of domain-specific performance requirements, ultimately enabling informed baseline classifier decisions based on application-specific constraints and optimization priorities.

6.5. Efficiency-Accuracy Trade-off Analysis

The comprehensive efficiency analysis reveals distinct algorithmic profiles across two critical resource utilization dimensions (Table 9) that fundamentally influence practical deployment decisions in machine learning applications.

Table 9
Comprehensive Efficiency Analysis: Performance-to-Time Trade-offs and Memory Consumption Patterns Across Machine Learning Algorithms

Performance-to-Time Ratios (Higher = Better)	Memory Efficiency Rankings (MB per 0.01 accuracy)
XGBoost: 1.99 (optimal balance)	LogisticRegression: 0.076 MB (most efficient)
LightGBM: 0.72 (good balance)	XGBoost: 0.145 MB (very efficient)
LogisticRegression: 0.55 (moderate efficiency)	LightGBM: 1.18 MB (moderate)
MLP: 0.43 (poor efficiency)	AdaptiveBayes: 3.28 MB (includes GPU overhead)
RandomForest: 0.39 (slow training)	MLP: 5.22 MB (high consumption)
AdB: -4.24 (speed champion, accuracy penalty)	RandomForest: 56.7 MB (memory intensive)

The performance-to-time ratio metric quantifies the balance between classification quality and computational cost, where higher values indicate superior efficiency in achieving accuracy relative to training duration. XGBoost emerges as the optimal balanced solution with a ratio of 1.99, demonstrating effective integration of accuracy and speed, while LightGBM (0.72) and Logistic Regression (0.55) maintain positive ratios indicating reasonable efficiency trade-offs. Conversely, AdaptiveBayes exhibits a negative ratio (-4.24) reflecting its primary optimization focus on computational speed rather than accuracy maximization, positioning it as a specialized solution for time-critical applications where rapid training takes precedence over marginal accuracy improvements. RandomForest (0.39) and MLP (0.43) demonstrate poor efficiency profiles, requiring substantial computational resources relative to their accuracy contributions.

Memory consumption patterns reveal equally important practical considerations for algorithm selection, particularly in resource-constrained environments or large-scale deployment scenarios. Logistic Regression demonstrates exceptional memory efficiency at 0.076 MB per 0.01 accuracy point, making it ideal for embedded systems and edge computing applications, while XGBoost maintains very efficient memory utilization (0.145 MB) despite its superior accuracy profile. AdaptiveBayes occupies a moderate memory consumption position (3.28 MB) that includes GPU overhead, representing a reasonable trade-off for applications prioritizing computational speed over memory optimization. The memory intensity spectrum culminates with RandomForest consuming 56.7 MB per accuracy unit, highlighting its unsuitability for memory-constrained deployments despite potential accuracy advantages. These efficiency metrics establish clear decision-making frameworks for practitioners balancing computational speed, memory constraints, and accuracy requirements across diverse application contexts, with each algorithm occupying distinct optimization niches within the broader machine learning ecosystem.

7. Discussion

Our results show that AdB occupies a unique niche of ultra-fast classifiers. Its performance profile is characterized by a clear trade-off: sacrificing accuracy for an order-of-magnitude increase in training speed.

While it fails to match the classification quality of state-of-the-art models like XGBoost or even the baseline LR in most cases, its sheer velocity makes it a compelling option for specific use cases. For instance, in rapid prototyping, iterative feature engineering, or production environments where models must be retrained very frequently on large data streams, a 10^x-700^x speedup can be a decisive advantage.

Speed-Accuracy Trade-off Analysis

The fundamental tension between computational efficiency and classification accuracy characterizes AdB performance profile. While dramatic training speedups (10-749^x) establish clear computational advantages, accuracy limitations averaging -0.26 ± 0.33 compared to Logistic Regression necessitate careful consideration of deployment contexts.

The superior AUC performance on specific datasets (CreditCardFraud, KDDCup99, Avazu) suggests AdB excels in particular problem domains, especially ranking tasks or scenarios with good class separability. This pattern indicates problem-specific model selection should prioritize data characteristics beyond simple accuracy metrics.

Dataset-Dependent Performance Patterns

Performance variation across datasets highlights the critical importance of problem-specific evaluation. AdB exceptional performance on KDDCup99 (AUC 0.988, accuracy 0.936) demonstrates that certain data characteristics—particularly good separability and moderate dimensionality—favor adaptive learning approaches.

Conversely, challenges with HIGGS and SUSY datasets reveal limitations in high-dimensional physics problems requiring complex feature interactions. The CreditCardFraud paradox (low accuracy, excellent AUC) underscores the critical need for threshold optimization in imbalanced scenarios.

Practical Deployment Guidelines
Recommended Use Cases:

- Time-Critical Pipelines: Rapid model development with acceptable accuracy trade-offs.
- GPU-Rich Environments: Leveraging unique GPU acceleration capabilities.
- Prototyping and Iteration: Fast baseline establishment for feature engineering.
- Real-Time Applications: Scenarios prioritizing inference speed and training efficiency.

Not Recommended For:

- Regulatory/High-Stakes: Applications requiring maximum accuracy and interpretability.
- Complex High-Dimensional: Problems similar to HIGGS/SUSY without architectural improvements.
- Severely Imbalanced: Datasets without threshold optimization mechanisms.

Limitations and Future Research Directions
Current Limitations:

- Accuracy Deficiencies: Particularly on complex, high-dimensional problems.
- Calibration Issues: Suboptimal performance on imbalanced datasets.
- Memory Variability: Volatile GPU memory usage across different problem types.
- Limited Regularization: Absence of built-in overfitting prevention.

Immediate Research Priorities:

- Automatic Threshold Tuning: Validation-based calibration for imbalanced data.
- Hyperparameter Optimization: Efficient grid search or Bayesian optimization.
- Mixed-Precision GPU: Lower memory footprint with maintained precision.
- Ensemble Integration: Combining multiple AdB variants.

Long-Term Directions:

- Integration with modern deep learning frameworks for hybrid approaches.
- Extension to multi-task and transfer learning scenarios.
- Development of theoretical convergence guarantees under proposed enhancements.
- Exploration of attention mechanisms for adaptive feature weighting.

The poor performance on datasets like HIGGS and SUSY and the analysis of its core algorithm (simple learning rate adaptation, lack of regularization) point to clear architectural weaknesses. The algorithm in its current form is ill-equipped to handle complex, non-linear decision boundaries and is prone to overfitting. The paradoxical result on CreditCardFraud underscores the need for better handling of imbalanced data, specifically through automatic classification threshold tuning.

8. Future Research Directions and Limitations

Critical Limitations and Failure Modes
Algorithmic Limitations:

- High-Dimensional Complexity: AdB struggles with datasets exceeding 1000 features, particularly sparse categorical encodings where feature interactions dominate classification performance.

- Non-Linear Decision Boundaries: Simple linear combinations prove insufficient for complex decision surfaces characteristic of physics datasets (HIGGS, SUSY).
- Imbalanced Data Calibration: Threshold mis-calibration on severely imbalanced datasets (1:578 ratio) requires post-hoc calibration techniques.
- Memory Volatility: GPU memory consumption varies dramatically (0-513 MB) based on dataset characteristics, limiting predictable resource allocation.

Computational Constraints:

- GPU dependency limits deployment in CPU-only environments.
- Single-threaded CPU fallback reduces competitive advantage.
- Memory overhead exceeds lightweight baselines by 25-100x.

Statistical Validity Issues:

- Limited evaluation on multiclass problems (>2 classes).
- Insufficient cross-dataset validation for hyperparameter sensitivity.
- Lack of convergence guarantees under proposed enhancements.

Immediate Research Priorities

High-Priority Algorithmic Enhancements:

- Automatic Threshold Calibration
  - Objective: Eliminate accuracy-AUC discrepancies on imbalanced datasets.
  - Approach: Validation-based threshold selection using Youden's J statistic or F1-optimization.
  - Expected Impact: +15–30% accuracy improvement on imbalanced datasets.
  - Implementation Timeline: 3–6 months.
- Hyperparameter Optimization Framework
  - Objective: Systematic exploration of learning rate, regularization, and transformation parameters.
  - Approach: Bayesian optimization with early stopping on validation AUC.
  - Expected Impact: +2-5% accuracy improvement across datasets.
  - Resource Requirements: 100-200 GPU hours per dataset.
- Mixed-Precision GPU Implementation
  - Objective: Reduce memory footprint while maintaining numerical stability.
  - Approach: FP16 training with FP32 master weights following NVIDIA Apex patterns.
  - Expected Impact: 50% memory reduction, 20–30% speedup.
  - Technical Risk: Potential convergence issues requiring loss scaling.

Medium-Priority Extensions:

- Ensemble Integration Framework
  - Approach: Bagging multiple AdB variants with different initialization seeds and hyperparameters.
  - Expected Impact: +3–8% accuracy improvement through variance reduction.
  - Computational Overhead: 5–10× training time increase.

- Online Learning Adaptation
  - Objective: Enable incremental updates for streaming data scenarios.
  - Approach: Exponential forgetting factors and adaptive batch sizing.
  - Applications: Real-time fraud detection, dynamic recommendation systems.
  - Research Challenge: Maintaining computational efficiency under concept drift.

- Multi-Task Learning Extension
  - Objective: Leverage shared representations across related classification tasks.
  - Approach: Shared feature transformations with task-specific output layers.
  - Expected Benefit: Improved performance on small datasets through transfer learning.

Long-Term Research Directions
Theoretical Foundations:

- Convergence Analysis:
  - Research Question: Under what conditions does the enhanced AdB converge to optimal solutions?
  - Methodology: Stochastic approximation theory and concentration inequalities.
  - Impact: Provide theoretical guarantees for practical deployment decisions.
  - Collaboration: Requires optimization theory expertise.

- Generalization Bounds:
  - Objective: Establish PAC-Bayesian bounds for AdB generalization performance.
  - Applications: Model selection, dataset size requirements, confidence intervals.
  - Technical Challenge: Adaptive learning rate complicates traditional analysis frameworks.

Architectural Innovations:

- Attention-Based Feature Weighting:
  - Inspiration: Transformer attention mechanisms for adaptive feature importance.
  - Implementation: Learned attention weights modulating feature contributions.
  - Expected Benefit: Improved performance on high-dimensional datasets with irrelevant features.
  - Computational Overhead: 10-20% increase in training time.

- Hybrid Deep Learning Integration:
  - Approach: AdB as final layer in deep networks for tabular data.
  - Architecture: CNN/RNN feature extraction followed by AdB classification.
  - Target Applications: Time series classification, structured tabular data with temporal components.
  - Research Risk: May lose computational efficiency advantages.

Practical Deployment Research:

- Edge Computing Optimization:
  - Objective: Enable AdB deployment on resource-constrained devices.
  - Technical Approach: Quantization, pruning, and hardware-specific optimization.
  - Target Hardware: ARM processors, mobile GPUs, embedded systems.
  - Success Metrics: <10 MB memory footprint, <100ms inference latency.
- Federated Learning Adaptation:
  - Research Challenge: Distribute AdB training across multiple clients.
  - Privacy Constraints: Differential privacy guarantees for sensitive datasets.
  - Communication Efficiency: Minimize parameter synchronization overhead.
  - Applications: Healthcare, financial services, IoT sensor networks.

Recommended Evaluation Protocols

Enhanced Benchmarking Standards:

- Cross-Domain Validation:
  - Objective: Assess generalization across different application domains.
  - Dataset Selection: Medical, financial, industrial, and scientific domains.
  - Evaluation Metrics: Domain adaptation performance, transfer learning effectiveness.
  - Statistical Rigor: Multi-level cross-validation with domain stratification.
- Computational Efficiency Benchmarks:
  - Hardware Diversity: CPU-only, GPU-accelerated, multi-GPU, and distributed settings.
  - Energy Consumption: Power usage effectiveness for green AI initiatives.
  - Scalability Analysis: Performance scaling with dataset size and dimensionality.
  - Comparison Framework: Standardized efficiency metrics across hardware configurations.
- Robustness Evaluation:
  - Adversarial Robustness: Performance under input perturbations and adversarial examples.
  - Distribution Shift: Covariate shift, label shift, and concept drift scenarios.
  - Missing Data Handling: Performance degradation under various missingness patterns.
  - Noise Tolerance: Gaussian, uniform, and systematic noise injection studies.

The comprehensive research roadmap establishes AdB as a foundation for next-generation efficient machine learning algorithms while acknowledging current limitations and providing concrete paths for improvement. Success in these research directions could establish adaptive learning as a viable alternative to traditional optimization-based approaches in resource-constrained applications.

9. Conclusions

This comprehensive analysis establishes AdB as a compelling alternative to Logistic Regression for specific application contexts prioritizing computational efficiency over marginal accuracy improvements. The dramatic training speed advantages (10-749^x) combined with competitive AUC performance on select datasets position AdB as a viable baseline option for time-sensitive machine learning applications.

Computational Efficiency:

- 10^x average training speedup over Logistic Regression with peaks at 749^x.
- Optimal training-prediction balance (9.6^x ratio) among all evaluated methods.
- Moderate memory consumption (174 MB) with unique GPU utilization capabilities.

Classification Performance:

- Accuracy limitations averaging -0.26 ± 0.33 but AUC improvements of +0.05 ± 0.17 on favorable datasets.
- Outstanding performance on KDDCup99 (AUC 0.988) demonstrating method potential.
- Clear calibration issues on imbalanced data requiring threshold optimization.

Algorithmic Insights:

- Proposed enhancement framework targeting 2–8% accuracy improvements through regularization, curvature adaptation, and improved initialization.
- Systematic identification of failure modes and mitigation strategies.
- Clear deployment guidelines for practical application.

AdB proves suitable as a baseline replacement when training speed is critical, datasets exhibit good separability characteristics, and applications can accommodate moderate accuracy reduction for substantial speed gains. The method is not recommended for problems requiring maximum accuracy or highly imbalanced data without calibration improvements.

AdB occupies a unique point in the accuracy-efficiency trade-off landscape: order-of-magnitude faster than Logistic Regression with comparable AUC on selected datasets but materially worse accuracy on complex physics data. Minor algorithmic refinements narrow the quality gap while preserving speed. We therefore recommend AdB as a drop-in baseline replacement when *training latency is the primary bottleneck.*

The evolving landscape of machine learning applications increasingly values computational efficiency alongside traditional accuracy metrics. This research contributes empirical evidence for alternative baseline model selection while providing a concrete roadmap for algorithmic improvements through the proposed enhancement framework.

Enhanced AdB implementations incorporating improved feature transformations, elastic net regularization, diagonal quasi-Newton optimization, and adaptive threshold tuning show potential for addressing current limitations while preserving computational advantages. These developments could significantly expand the practical applicability of adaptive baseline approaches across broader problem domains.

Future work should prioritize implementation of the proposed enhancement framework, systematic evaluation of calibration techniques, and exploration of hybrid approaches combining AdB efficiency with complementary accuracy-enhancing methods. The unique computational profile of AdB positions it as a valuable component in the toolkit of modern machine learning practitioners facing increasingly large-scale and time-sensitive applications.

Declaration on Generative AI

While preparing this work, the authors used the AI programs Grammarly Pro to correct text grammar and Strike Plagiarism to search for possible plagiarism. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

[1] S. Abogada, L. Usona, Early Warning System of Attrition in the BPO Industry using Machine Learning Classification Models, J. Artif. Intell. Mach. Learn. Neural Netw., 4 (2024) 18–30. doi:10.55529/jaimlnn.43.18.30

[2] S. Kavun, G. Zhosan, Calculation of the Generalizing Indicator of Productivity of the Enterprises Activity based on the Matrix-Rank Approach, J. Finance Econ., 2 (2014) 202–209.

[3] S. Kavun, Indicative-Geometric Method for Estimation of Any Business Entity, Int. J. Data Anal. Tech. Strateg., 8 (2016) 87. doi:10.1504/ijdats.2016.077486

[4] Z. C. Chia, K. H. Lim, T. P. L. Tan, Two-Phase Switching Optimization Strategy in LSTM Model for Predictive Maintenance, in: Int. Conf. Green Energy, Comput. Sustain. Technol. (GECOST 2021), Miri, Malaysia, 2021, 1–6. doi:10.1109/GECOST52368.2021.9538639

[5] J. Huang, H. Peng, H. Wu, Newton's Method and its Hybrid with Machine Learning for Navier-Stokes Darcy Models Discretized by Mixed Element Methods, Commun. Comput. Phys., 37 (1) (2025) 30–60. doi:10.4208/cicp.OA-2024-0066

[6] S. Yagishita, S. Nakayama, Proximal Diagonal Newton Methods for Composite Optimization Problems, arXiv preprint, 2023. doi:10.48550/arXiv.2310.06789

[7] F. Pedregosa, et al., Scikit-learn: Machine learning in Python, J. Mach. Learn. Res., 12 (2011) 2825–2830.

[8] S. Datsenko, H. Kuchuk, Biometric Authentication Utilizing Convolutional Neural Networks, Adv. Inf. Syst., 7(2) (2023) 87–91. doi:10.20998/2522-9052.2023.2.12

[9] J. Martens, R. Grosse, Optimizing Neural Networks with Kronecker-Factored Approximate Curvature, in: 32nd Int. Conf. Mach. Learn., 37 (2015) 2408–2417.

[10] O. Trydid, S. Kavun, M. Goykhman, Synthesis Concept of Information and Analytical Support for Bank Security System, Actual Probl. Econ., 11(161) (2014) 449–461.

[11] S. Yagishita, S. Nakayama, An Acceleration of Proximal Diagonal Newton Method, JSIAM Lett., 16 (2024) 5–8. doi:10.14495/jsiaml.16.5

[12] P. Baldi, P. Sadowski, D. Whiteson, Searching for Exotic Particles in High-Energy Physics with Deep Learning, Nat. Commun., 5 (2014) 4308.

[13] C. D. Alecsa, A Theoretical and Empirical Study of New Adaptive Algorithms with Additional Momentum Steps and Shifted Updates for Stochastic Non-Convex Optimization, J. Glob. Optim., 93 (2025) 113–173. doi:10.1007/s10898-025-01518-0

[14] G. Ke, et al., LightGBM: A Highly Efficient Gradient Boosting Decision Tree, Adv. Neural Inf. Process. Syst., 30 (2017).

[15] L. Wu, et al., A Robust Stochastic Quasi-Newton Method with the Application in Machine Learning, in: Int. Conf. Culture-Oriented Sci. Technol. (ICCST 2021), Beijing, China, 2021, 149–154. doi:10.1109/ICCST53801.2021.00041

[16] S. Kavun, Adaptive_Bayes: Version 01 (v_01), Benchmark Analysis, Zenodo, 2025. doi:10.5281/zenodo.17184113

[17] J. Luo, et al., Complexity-optimized Sparse Bayesian Learning for Scalable Classification Tasks, Inf. Sci., 719 (2025) 122447. doi:10.1016/j.ins.2025.122447

[18] J. Luo, C. M. Vong, Z. Liu, C. Chen, An Inverse-Free and Scalable Sparse Bayesian Extreme Learning Machine for Classification Problems, IEEE Access, 9 (2021) 87543–87551. doi:10.1109/ACCESS.2021.3089539

[19] H. Zou, T. Hastie, Regularization and Variable Selection via the Elastic Net, J. R. Stat. Soc. Ser. B, 67 (2005) 301–320.

[20] X. Glorot, Y. Bengio, Understanding the Difficulty of Training Deep Feedforward Neural Networks, in: 13th Int. Conf. Artif. Intell. Stat., 2010, 249–256.

[21] K. He, X. Zhang, S. Ren, J. Sun, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, in: IEEE Int. Conf. Comput. Vis., 2015, 1026–1034.

[22] S. Geeitha, et al., Disease-Free Survival Prediction in Recurrent Cervical Cancer using Naive Bayes Machine Learning Algorithm, in: 5th Int. Conf. Smart Electron. Commun. (ICOSEC 2024), Trichy, India, 2024, 1901–1906. doi:10.1109/ICOSEC61587.2024.10722641

[23] P. Rajneekant, B. P. Kishore, D. P. Gond, D. P. Mohapatra, Enhancing Malware Classification with Machine Learning: A Comparative Analysis of API Sequence-based Techniques, in: IEEE Int. Conf. Smart Power Control Renew. Energy (ICSPCRE 2024), Rourkela, India, 2024, 1–6. doi:10.1109/ICSPCRE62303.2024.10675011

[24] T. Chen, C. Guestrin, XGBoost: A Scalable Tree Boosting System, in: 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2016, 785–794.

[25] X. Ma, Apollo: An Adaptive Parameter-Wise Diagonal Quasi-Newton Method for Nonconvex Stochastic Optimization, arXiv preprint, 2021. doi:10.48550/arXiv.2009.13586

[26] Z. Aminifard, S. Babaie-Kafaki, Diagonally Scaled Memoryless Quasi-Newton Methods with Application to Compressed Sensing, J. Ind. Manag. Optim., 18 (2022) 4181–4205.

[27] M. M. Ahmed, A Novel Variance Reduction Proximal Stochastic Newton Algorithm for Large-Scale Machine Learning Optimization, Int. J. Adv. Netw. Monit. Control, 9 (2024) 84–90. doi:10.2478/ijanmc-2024-0040

[28] J. Islam, et al., A Comparative Study on Feature Selection between Computational and Medical Knowledge Driven Approaches for Heart Disease Prediction, in: IEEE Int. Conf. Biomed. Eng. Comput. Inf. Technol. Health (BECITHCON 2024), Dhaka, Bangladesh, 2024, 125–130. doi:10.1109/BECITHCON64160.2024.10962566