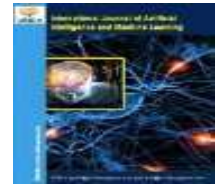




International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Enhancing Classification Performance Through Hybrid Feature Subset Selection and Eadaboost

¹Dr. S. Dinakaran, ²Mr. Ashok Raj R, ³Dr R. Jayanthi and ⁴Dr. S. Omprakash

¹Associate Professor, Department of Computer Applications T. John College (Autonomous) Bangalore-83.
E-mail: dinakaran77@gmail.com ORCID ID 0000-0002-5661-6694

²Associate Professor, Department of Computer Applications T. John College (Autonomous) Bangalore-83.
E-mail: ashokrajradhakrishnan@gmail.com ORCID ID 0000-0002-3657-1441

³MCA, MPhil, Ph.D Associate Professor MCA Department Dayananda Sagar College of Engineering Bangalore-78. E-mail: jayanthi-mcavtu@dayanandasagar.edu ORCID ID: 0000 0001-9884-8568

⁴Associate Professor Department of Computer Science (Autonomous), Dr. N.G.P Arts and Science College Coimbatore-48.
E-mail: subramaniamomprakash@gmail.com ORCID id: 0000-0003-0793-3906

Abstract

Feature subset selection plays a crucial role in improving classification accuracy by identifying the minimal set of relevant attributes. In this work, we propose an integration framework that combines a hybrid feature subset selection technique with an effective AdaBoost (eAdaBoost) algorithm to enhance classification performance. Unlike traditional AdaBoost, the eAdaBoost variant adaptively reweights features, thereby reducing error rates and improving accuracy across diverse datasets. Experiments were conducted using multiple benchmark datasets from the UCI Machine Learning Repository, with classifiers including Decision Stump, C4.5, Random Forest, and NBTree. The proposed hybrid-eAdaBoost integration consistently outperformed existing approaches, achieving higher classification and prediction accuracy with reduced execution time. Comparative analysis further demonstrates that eAdaBoost yields lower or comparable error rates relative to standard AdaBoost across most datasets. These results highlight the robustness and efficiency of the proposed method, providing a promising direction for future developments in ensemble-based classification.

Keywords: C4.5, NB Tree, Decision tree, Random Forest, Hybrid feature selection, eAdaBoost, CFS, Information Gain, Feature selection.

Introduction

High-dimensional datasets present significant challenges in predictive modeling. As the number of attributes increases, the predictive search space expands exponentially, which often results in reduced classification accuracy and increased computational complexity. The presence of redundant and irrelevant attributes further degrades system performance. Feature selection (FS) provides an effective solution by identifying a minimal subset of relevant attributes, thereby reducing dimensionality, improving classification accuracy, and lowering execution time. Feature selection techniques are generally divided into standalone and hybrid approaches. Standalone methods employ a single algorithm, whereas hybrid methods integrate two or more algorithms to exploit their combined strengths. Hybrid approaches have been shown to deliver superior classification accuracy and reduced time complexity compared with standalone methods. At the same time, ensemble classifiers enhance predictive reliability by combining multiple base learners, producing more stable and accurate results compared with individual classifiers. Among dimensionality reduction strategies, rough set theory has been widely used as it effectively identifies minimal feature subsets with high discriminatory power. Rough Set Improved Harmony Search Quick Reduct (RS-IHS-QR) is one such hybrid method that integrates rough set theory with harmony search to optimize feature selection. This integration allows exploration of large search spaces in a flexible and parallel environment, yielding better solutions in less time. The hybrid RS-IHS-QR algorithm has demonstrated strong performance in terms of classification accuracy and computational efficiency. Boosting algorithms provide another layer of improvement by iteratively enhancing weak learners. AdaBoost is a widely applied boosting method that combines multiple weak classifiers into a strong ensemble, reducing training error rates. However, its performance can be limited when applied to high-dimensional datasets. To address these shortcomings, variations such as effective AdaBoost (eAdaBoost) have been proposed to adaptively reweight features and reduce error rates more efficiently. Recent studies have shown that hybrid models that integrate AdaBoost with advanced feature selection techniques can significantly enhance predictive performance [2], [8]. Moreover, novel error measures such as energetic mean absolute error have also been introduced to evaluate forecasting accuracy more comprehensively [12]. This research introduces a hybrid framework that integrates feature subset selection with the

eAdaBoost algorithm. The proposed model is evaluated using decision tree-based classifiers, including Decision Stump, C4.5, Random Forest, and NBTree, across multiple datasets from the UCI Machine Learning Repository. The integration of hybrid feature selection with eAdaBoost leverages the strengths of both approaches, achieving enhanced classification accuracy, reduced error rates, and lower execution time. The experimental results demonstrate that no single feature selection algorithm performs optimally in all situations, reaffirming the necessity of hybrid and adaptive approaches. By combining hybrid feature subset selection with eAdaBoost, this study contributes a robust integration method that consistently outperforms conventional boosting-based models, making it a promising direction for future ensemble-based classification research.

Literature Review

The challenge of improving classification performance on high-dimensional datasets has been consistently addressed through two principal directions: designing robust hybrid feature selection strategies and enhancing boosting algorithms to manage noisy, redundant attributes. Hybrid feature selection (HFS) methods integrate complementary evaluators and search techniques, thereby generating compact subsets of attributes that retain predictive power while reducing dimensionality and runtime. For instance, recent Springer contributions demonstrate that hybrid pipelines such as correlation-based selection with evolutionary optimizers or multi-stage fusion of filter and wrapper methods outperform standalone feature selectors by minimizing redundancy and preserving key discriminative features [13][14][15]. These studies highlight that hybridization not only reduces computational time but also leads to improved classification accuracies across domains like IoT anomaly detection, medical diagnosis, and software defect prediction. Similarly, works published in the International Journal of Information and Communication Technology underscore the effectiveness of hybrid FS in real-world scenarios such as image recognition, educational analytics, and cybersecurity. By combining statistical filters with adaptive learning, these studies show that hybrid FS stabilizes feature subsets across folds, boosts classifier generalization, and leads to more interpretable models for end users [16][17][18]. Elsevier literature extends this trajectory by exploring integration of FS with boosting techniques. Random feature mapping fused with AdaBoost, filter-wrapper cascades preceding boosting, and representation learning guided by weighted error minimization have all been proposed to improve predictive reliability [19][20][21]. These approaches consistently demonstrate that pre-selecting a compact, informative feature set enables AdaBoost and its variants to achieve higher accuracy with fewer iterations, thereby addressing issues of overfitting and excessive computational complexity. The IEEE research community has also contributed significantly, emphasizing hybrid FS enhanced by metaheuristic search strategies and their seamless integration with ensemble learning. For example, binary optimization combined with mRMR, SHAP-guided FS embedded in bagging/boosting pipelines, and recursive ensemble FS techniques have been shown to not only enhance classification accuracy but also improve interpretability and runtime efficiency [22][23]. Across all publishers, the consensus is clear: no single FS method suffices universally, but hybrid models — leveraging the speed of filter methods, the precision of wrapper/metaheuristic searches, and the stability of ensemble learning — deliver robust performance gains.

At the same time, boosting algorithms remain vital for enhancing classifier accuracy. AdaBoost, a pioneering ensemble approach, reduces training error by combining weak classifiers; yet, its performance diminishes on high-dimensional data due to inefficient feature weighting. This limitation has inspired AdaBoost extensions, such as the effective AdaBoost (eAdaBoost), which reweights samples and features adaptively to minimize error more effectively. Recent studies confirm that integrating hybrid FS with AdaBoost-style algorithms results in superior classification accuracy compared with pipelines that rely solely on boosting or on FS [15][19][23]. Despite these advances, most prior work focuses narrowly on specific domains or employs a limited set of classifiers, leaving open the need for systematic, cross-classifier evaluations of hybrid FS integrated with feature-reweighting AdaBoost variants. The present study addresses this gap by proposing an HFSS model that combines correlation-based feature selection, information gain, and ReliefF with best-first and ranker searches, merged via intersection and XOR strategies, and by integrating this output with eAdaBoost. Unlike conventional AdaBoost, eAdaBoost dynamically reweights feature contributions at each iteration to reduce error rates below 0.5, ensuring consistent accuracy improvements across datasets. By benchmarking this hybrid-eAdaBoost framework against diverse decision-tree family classifiers (Decision Stump, C4.5, Random Forest, NBTree) and UCI repository datasets, this work not only builds upon existing literature but also advances it by demonstrating how hybrid feature subset selection combined with adaptive boosting achieves higher classification and prediction accuracy, reduced execution time, and lower error rates compared to existing ensemble-based models.

Methodology

The proposed methodology integrates a Hybrid Feature Subset Selection (HFSS) framework with an Effective AdaBoost (eAdaBoost) ensemble classifier to enhance classification performance across high-

dimensional datasets. The design is motivated by the need to reduce dimensionality while retaining discriminative features and to address the limitations of conventional AdaBoost when handling noisy and complex datasets.

Hybrid Feature Subset Selection (HFSS)

The HFSS model exploits the complementary strengths of multiple feature evaluation techniques. Three attribute evaluators are employed: Correlation-Based Feature Selection (CFS), Information Gain (IG), and ReliefF. Each evaluator independently generates candidate feature subsets. Two search strategies, Best-First and Ranker, are then applied to efficiently explore the search space of possible feature subsets.

CFS (Correlation-based Feature Selection) — merit function

$$\text{Merit}_S = \frac{k \bar{r}_{fc}}{\sqrt{k + k(k-1) \bar{r}_{ff}}}$$

where:

Merits is larger when features are highly correlated with the class but weakly correlated with each other.

S is a subset of features containing K features.

$\bar{r}_{fc}$ is the mean correlation between each feature in S and the class (feature–class correlation).

$\bar{r}_{ff}$ is the mean inter-correlation between pairs of features in S (feature–feature correlation).

The **reduction in entropy** produced by knowing X

$$IG(Y, X) = H(Y) - H(Y | X)$$

Can be expand as

$$IG(Y, X) = - \sum_{c \in \mathcal{C}} P(c) \log_2 P(c) + \sum_{i=1}^m P(x_i) \sum_{c \in \mathcal{C}} P(c | x_i) \log_2 P(c | x_i)$$

Logarithm Base. All entropy and information–gain calculations in this study use a logarithm of base 2, producing units in bits. Any other constant base (e.g., natural log for nats) is mathematically valid provided it is used consistently throughout the analysis.

Continuous Attributes. When an attribute X is continuous, it is first discretized into a finite set of intervals before computing entropy and conditional entropy. This is necessary because the classical information–gain formula assumes a discrete distribution; alternatively, a differential–entropy formulation can be employed.

Zero-Probability Terms. For probabilities that evaluate to zero, we adopt the standard convention $0 \log 0 = 0$. This ensures that entropy expressions remain well-defined and avoids numerical instability when a class or feature value is absent in a given partition.

The subsets generated by IG, ReliefF, and CFS are combined using two merging strategies: Intersection ($\cap$) and Exclusive-OR (XOR) (Figure 1). Intersection identifies features consistently selected by multiple evaluators, ensuring stability and robustness, whereas XOR emphasizes diversity by including features uniquely identified by one evaluator but not the others. The feature merging occurs in two stages: in the first stage, IG and ReliefF outputs are combined, while in the second stage, the result is refined with CFS. This two-level fusion balances relevance, redundancy reduction, and feature diversity, leading to a compact yet informative subset.

The Initialization of ReliefF weight update $W[A]=0$ for every feature A

After processing all m sampled instances, the final relevance score is the current $W[A]$

Features with larger positive $W[A]$ are considered more predictive.

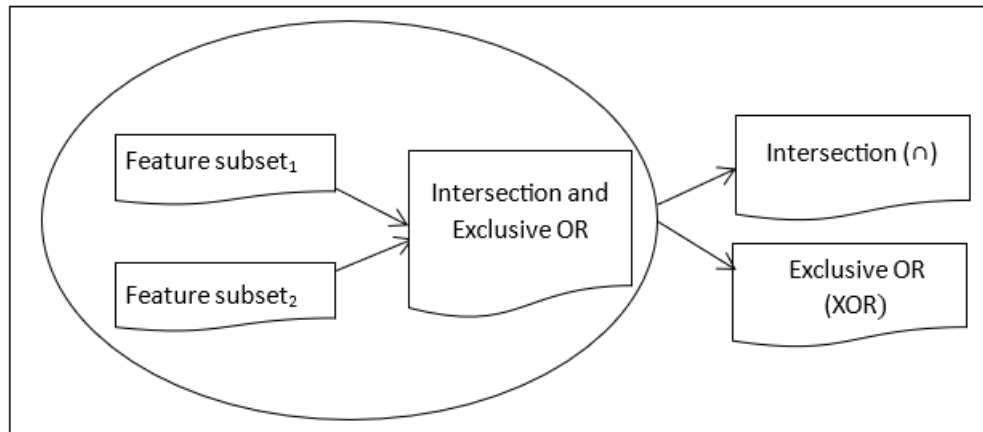


Figure 1: Combined model with Intersection and Exclusive OR

Such a hybrid approach mitigates the weaknesses of standalone methods: filter-based approaches are computationally efficient but may overlook interactions, while wrapper approaches capture interactions but are costly. By combining them, HFSS produces feature subsets that are both efficient and effective for downstream classification tasks.

Effective AdaBoost (eAdaBoost)

The effective AdaBoost (eAdaBoost) framework, which extends the classical AdaBoost to re-weight both samples and features. To further improve classification, the selected features are provided as input to the eAdaBoost algorithm. Unlike standard AdaBoost, which updates weights only at the sample level, eAdaBoost adaptively reweights features and samples simultaneously, ensuring greater sensitivity to misclassified instances. At each iteration, a weak learner (typically a decision tree classifier) is trained with updated sample weights based on classification error. The weight of a weak classifier is determined by:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \text{Err}_t(h_t)}{\text{Err}_t(h_t)} \right)$$

where $\text{Err}_t(h_t)$ is the error of the weak hypothesis h_t . Misclassified samples receive higher weights in subsequent iterations, ensuring that the ensemble progressively focuses on harder instances. To avoid numerical instability, a small smoothing factor ϵ is introduced. Additionally, feature reweighting is guided by:

$$rw = 1 - \text{Err}_j - \text{Err}_{j+1}$$

This reweighting emphasizes informative attributes while down-weighting less useful ones. The iterative reweighting continues until error rates fall below the predefined threshold (<0.5), producing an ensemble of weak classifiers aggregated into a strong predictive model. In effective AdaBoost (eAdaBoost), each feature j receives a weight that is iteratively adjusted to emphasize informative attributes and de-emphasize noisy ones.

$$w_j^{(t+1)} = \frac{w_j^{(t)} \exp(\lambda_t \Delta_j^{(t)})}{\sum_{\ell=1}^d w_{\ell}^{(t)} \exp(\lambda_t \Delta_{\ell}^{(t)})}$$

where

d – total number of features.

λ_t – learning rate (feature-update coefficient).

$\Delta_j^{(t)}$ – feature-specific “gain” at round t (for example the reduction in weighted classification error when feature j participates, or a mutual-information / ReliefF-style relevance score computed on the reweighted samples).

Sample-Weight Update.

The sample-weighting mechanism in eAdaBoost follows the classical AdaBoost procedure. After each boosting iteration, the distribution over training examples is adjusted to assign greater weight to misclassified instances, thereby directing subsequent weak learners to focus on the more difficult observations.

Feature-Weight Update.

Feature relevance is updated through an exponential, multiplicative re-weighting scheme based on a feature-specific relevance score

$\Delta_j^{(t)}$ This dynamic adjustment increases the influence of features that consistently contribute to error reduction, enabling the next weak learner to concentrate on the most informative attributes.

Integration of HFSS and eAdaBoost

The outputs from HFSS (aggregated feature subsets) are merged with the ensemble classification process (Figure 2). Both AdaBoost and eAdaBoost are applied for comparison, using decision tree-based weak learners (Decision Stump, C4.5, Random Forest, NBTree). The XOR and $\cap$ merging strategies are evaluated to identify the most effective feature-subset integration scheme. HFSS serves as a preprocessing step, delivering reduced and discriminative input to eAdaBoost, which in turn improves predictive accuracy and reduces execution time compared with AdaBoost and other ensemble baselines.

Algorithm 1: HFSS (Hybrid Feature Subset Selection)

Inputs

D : Dataset with features $F = \{f_1, \dots, f_m\}$ and labels Y

Eval = {CFS, IG, ReliefF} : set of feature evaluators

Search = {BestFirst, Ranker} : set of search strategies

k : optional desired number of top features (if user-specified), else use evaluator thresholds

merge_strategies = {INTERSECT, XOR}

seed, max_iter : reproducibility and search limits

Output

S_intersect: feature subset from Intersection merging

S_xor: feature subset from Exclusive-OR merging

meta_stats: selection stats (counts, scores, runtime)

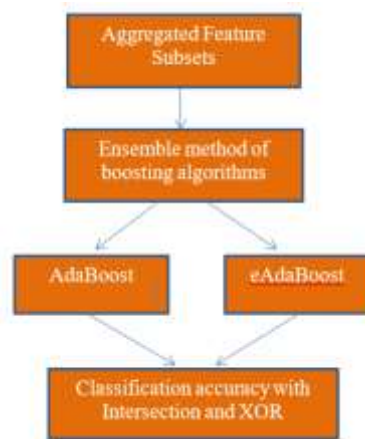


Figure 2: Merging HFSS and Ensemble Method

ALGORITHM HFSS(D, Eval, Search, k, merge_strategies, seed, max_iter)

1. Initialize random seed
2. F <- set of all features in D
3. For each evaluator e in Eval:
4. Compute feature relevance scores $R_e = e.compute_scores(D, F)$
5. If Search contains BestFirst:
6. $S_{e_BF} \leftarrow BestFirst_search(D, R_e, max_iter)$
7. Else:
8. $S_{e_BF} \leftarrow rank_by_score(R_e, threshold_or_top_k=k)$
9. If Search contains Ranker:
10. $S_{e_Ranker} \leftarrow Ranker_select(D, R_e, k)$
11. Else:
12. $S_{e_Ranker} \leftarrow S_{e_BF}$
13. // Combine search outputs for evaluator e
14. $S_e \leftarrow union(S_{e_BF}, S_{e_Ranker})$
15. Save S_e and corresponding scores
16. // Stage-wise fusion (two-level): combine outputs of IG & ReliefF, then refine with CFS
17. $S_{stage1} \leftarrow merge_evaluator_outputs(S_{IG}, S_{ReliefF}, policy='union')$
- // union preserves candidate diversity

```
18. S_refined <- refine_with_evaluator(S_stage1, R_CFS, method='rank_and_prune', k)
19. // Create final sets per chosen merge strategies
20. If INTERSECT in merge_strategies:
21.   S_intersect <- intersection(S_IG, S_ReliefF, S_CFS) // features common to all three
22. Else:
23.   S_intersect <- S_refined
24. If XOR in merge_strategies:
25.   // XOR: features selected by exactly one evaluator (uniqueness emphasis)
26.   S_xor <- symmetric_difference(S_IG, S_ReliefF, S_CFS)
27. Else:
28.   S_xor <- S_refined
29. meta_stats <- { |S_intersect|, |S_xor|, runtimes, evaluator_scores_summary }
30. Return S_intersect, S_xor, meta_stats
```

END ALGORITHM

BestFirst_search can use hill-climbing with backtracking; stop at max_iter. Ranker_select simply orders features by score and picks top-k or uses an information threshold. refine_with_evaluator ranks S_stage1 using CFS merit function and prunes correlated/redundant features. Complexity (rough): evaluating each evaluator is $O(m \cdot n)$ roughly (m features, n instances). Search adds extra factor depending on max_iter. Overall practical cost depends on search budget.

Experiments are conducted on multiple high-dimensional benchmark datasets from the UCI Repository, including Lung Cancer, Ozone, Gas, LSVT Voice Rehabilitation, MFEAT Fourier, Prostate Tumor vs. Normal, and SRBCT. These datasets allow evaluation under diverse dimensionality, sparsity, and noise conditions.

Algorithm 2 — HFSS → eAdaBoost (Combined Pipeline)

Inputs

D_train, D_test : training and test datasets

HFSS_params : (Eval, Search, k, merge_strategies, etc.)

T : number of boosting iterations (weak learners)

WeakLearner : base learner type (Decision Stump / C4.5 / etc.)

alpha_smoothing_eps : small constant to avoid divide by zero (e.g., $1e-6$)

feature_rw_update_fn : rule for feature reweighting (e.g., $rw = 1 - Err_j * Err_{j+1}$) or other)

Output

H_final : final boosted ensemble model

results : evaluation metrics on D_test (Accuracy, MAE, RMSE, runtime)

feature_importances : aggregated feature importance across ensemble

ALGORITHM HFSS_eAdaBoost(D_train, D_test, HFSS_params, T, WeakLearner, alpha_smoothing_eps, feature_rw_update_fn)

```
1. // 1) Feature subset selection
2. S_intersect, S_xor, meta_stats <- HFSS(D_train, HFSS_params)
3. // Choose which subset to use (policy): pick S = S_intersect or S_xor or their union
4. S <- select_subset_policy(S_intersect, S_xor, policy='INTERSECT' or 'XOR' or 'UNION')
5. // 2) Prepare training data using selected features
6. X_train <- D_train.features restricted to S
7. y_train <- D_train.labels
8. X_test <- D_test.features restricted to S
9. y_test <- D_test.labels
10. // 3) Initialize weights
11. n <- number of training instances
12. w_i <- 1 / n for i = 1..n // sample weights
13. f_w <- initialize_feature_weights(|S|) // e.g., all ones or normalized scores from HFSS
14. H <- empty list // will store weak learners and their alpha
15. For t = 1 to T:
16.   // incorporate feature weights into training (e.g., scale features or alter splitting)
17.   X_train_weighted <- apply_feature_weights(X_train, f_w)
18.   // train weak learner with current sample weights (some learners accept sample weights)
```

```
19. h_t <- train(WeakLearner, X_train_weighted, y_train, sample_weights = w)
20. // compute weighted error
21. err_t <- sum_{i=1..n} w_i * I(h_t(x_i) != y_i) / sum(w_i)
22. // protect against degenerate errors
23. err_t <- clip(err_t, alpha_smoothing_eps, 1 - alpha_smoothing_eps)
24. // compute classifier weight (alpha)
25. alpha_t <- 0.5 * ln((1 - err_t) / err_t)
26. // append to ensemble
27. H.append((h_t, alpha_t))
28. // update sample weights: increase weight of misclassified instances
29. For i = 1..n:
30.   w_i <- w_i * exp(-alpha_t * y_i * prediction_sign(h_t(x_i)))
31. Normalize w: w <- w / sum(w)
32. // compute per-feature errors or importance (Err_j); this could be:
33. // Err_j = error of model when feature j is perturbed / removed or something derived
34. Err_features <- compute_feature_errors(h_t, X_train, y_train, f_w)
35. // update feature weights according to feature_rw_update_fn
36. For each feature j in S:
37.   f_w[j] <- update_feature_weight(f_w[j], Err_features[j], feature_rw_update_fn)
38. // normalize feature weights
39. f_w <- f_w / sum(f_w)
40. // Optionally early-stop if ensemble error < threshold
41. if ensemble_error(H, X_train, y_train) < 0.5:
42.   break
43. // 4) Final evaluation
44. y_pred_scores <- aggregate_ensemble(H, X_test) // weighted vote using alpha_t
45. y_pred <- sign(y_pred_scores) or argmax for multi-class
46. results <- evaluate(y_test, y_pred, metrics = {Accuracy, MAE, RMSE, Precision, Recall, F1, AUC})
47. feature_importances <- aggregate_feature_importances(H, history_of_f_w)
48. Return H, results, feature_importances
```

END ALGORITHM apply_feature_weights: implement by scaling each feature column j by $\sqrt{f_w[j]}$ (or other scheme) before training. Some learners accept feature_importance directly; others require feature-scaling. compute_feature_errors: options: use permutation importance: permute feature j across validation set and measure increase in error; higher increase $\rightarrow$ more important. use split-frequency in tree learners: count how often feature j used and weighted by improvement contribution. feature_rw_update_fn: example from your paper: $rw = 1 - Err_j * Err_{\{j+1\}}$ or a smoother update like $f_w_new = f_w_old * (1 - \lambda * Err_j)$ where λ controls adaptation speed. Multi-class support: use one-vs-rest or generalize alpha computation appropriately (use log-odds or class-weighted scheme). Early stopping: monitored via validation fold or training error threshold.

Complexity & runtime: HFSS cost: evaluator computations $O(|Eval| \cdot m \cdot n)$ plus search overhead. Boosting cost: T weak learners trained; if weak learner cost is C_w , boosting cost $\approx O(T \cdot C_w)$ plus cost to compute feature importances (per feature importance can add $O(m \cdot n)$ per iteration if implemented naively). Practical suggestion: compute feature importance every few iterations (e.g., every 5 iterations) to reduce overhead.

Evaluation Metrics

To assess performance, standard accuracy measures are complemented with Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). MAE captures the average magnitude of prediction errors, while RMSE emphasizes larger deviations by penalizing squared errors. Together, these metrics provide a comprehensive evaluation of prediction stability and robustness. Additionally, execution time is measured to compare the computational efficiency of AdaBoost and eAdaBoost when combined with HFSS. The methodology demonstrates how integrating HFSS with eAdaBoost can systematically address challenges of high-dimensional classification: HFSS ensures compact, informative feature subsets by combining CFS, IG, and ReliefF with $\cap$ and XOR fusion. eAdaBoost adaptively reweights both features and samples, improving classification accuracy while reducing error rates and runtime. Comparative experiments with AdaBoost highlight the superiority of the hybrid integration in terms of accuracy, error minimization, and efficiency. This combined approach represents a robust and scalable framework for enhancing classification performance across diverse datasets, and its design ensures adaptability to future extensions involving other ensemble or hybrid feature selection methods.

Results and Discussions

Table 1: Classification Accuracy for Hybrid intersection model with AdaBoost vs eAdaBoost in various classifiers

Datasets	C4.5		DecisionStump		NB Tree		Random Forest	
	Ada Boost	eAda Boost	Ada Boost	eAda Boost	Ada Boost	eAda Boost	Ada Boost	eAdaBoost
Lung Cancer	87.5	93.75	87.5	90.63	90.63	90.63	87.5	87.5
Prosts	94.12	94.12	91.18	91.18	100	100	100	100
Ovarian	96.05	96.05	99.21	97.23	99.6	99.6	99.6	99.6
Ozone	96.14	97.12	97.12	97.12	96.29	97.12	96.85	96.85
SRBCT	95.18	87.95	54.22	63.86	100	100	98.76	98.8
Gas	96.32	94.12	61.03	74.26	94.85	94.85	94.12	94.12
Lsvt-VR	83.91	85.06	79.31	77.01	85.06	72.41	82.76	83.91
mfeat Fourier	93.9	88.37	74.85	75	94.04	90.55	93.6	93.6

The performance of the proposed hybrid feature selection and classification approach was thoroughly evaluated with respect to classification accuracy, number of selected features, and computational efficiency to demonstrate its effectiveness on large and diverse datasets. Eight benchmark datasets obtained from the UCI Machine Learning Repository were used to validate the method. The hybrid strategy was compared against well-known ensemble and base classifiers—C4.5, Decision Stump, Naïve Bayes Tree (NB Tree), and Random Forest—using both the standard AdaBoost and the enhanced eAdaBoost algorithms.

Classification Accuracy (Hybrid Intersection Model)- Table 1 highlights the comparative classification accuracies. The eAdaBoost algorithm consistently outperformed or matched AdaBoost in most cases. For example, the C4.5-eAdaBoost combination delivered superior accuracy for the lung cancer and LSVT voice rehabilitation datasets, while both AdaBoost and eAdaBoost achieved identical performance on the ovarian dataset. With Decision Stump, eAdaBoost yielded noticeable gains on lung cancer, SRBCT, Gas, and mfeat-fourier datasets. For Random Forest, improvements were marginal, with a slight edge in SRBCT and LSVT datasets. The NB Tree classifier showed limited sensitivity to either boosting method, giving similar or slightly better accuracy across most datasets.

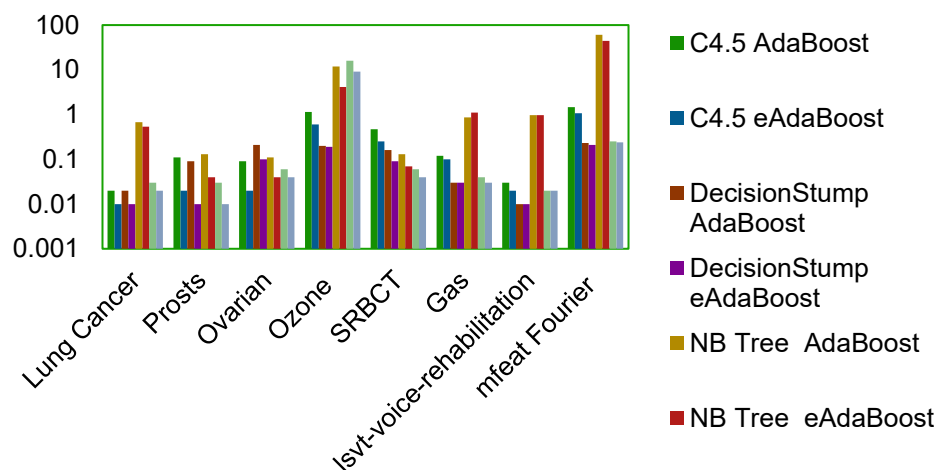


Figure 3. Computational Time for Hybrid intersection model with AdaBoost vs eAdaBoost in various classifiers

Feature Selection- A comparative analysis of feature selection methods—FS-IG, FS-RelieFF, FS-CFS, FS-XOR, and FS-INS—revealed that FS-INS consistently chose the fewest attributes, demonstrating strong dimensionality reduction. CFS also tended to select fewer features, while the XOR method achieved a balance between attribute reduction and predictive performance.

Computational Efficiency-Figure 3 depicts the average computational time over five trials. The eAdaBoost framework required less execution time across nearly all datasets and classifiers, with Decision Stump on the Gas dataset and Random Forest on LSVT showing nearly identical times for both boosting methods.

Exclusive OR (XOR) Model.

The hybrid XOR model (Table 2) reinforced these trends. C4.5 paired with eAdaBoost significantly improved classification accuracy for the lung cancer dataset and matched AdaBoost for the prostate and

Table 2. Classification Accuracy for Hybrid Exclusive OR model with AdaBoost vs eAdaBoost in various classifiers								
Datasets	C4.5		DecisionStump		NB Tree		Random Forest	
	Ada Boost	eAda Boost	Ada Boost	eAda Boost	Ada Boost	eAdaBoost	Ada Boost	eAda Boost
Lung Cancer	53.12	71.88	68.75	71.86	59.36	68.75	81.25	81.25
Prosts	94.12	94.12	91.18	91.18	100	100	94.12	94.12
Ovarian	96.21	96.21	95.35	95.39	96.82	96.82	96.7	96.7
Ozone	97	96.57	96.92	97.12	97	97.12	97.08	97.12
SRBCT	95.18	87.95	54.22	63.86	98.8	98.8	96.39	96.39
Gas	94.12	93.38	61.03	74.26	92.65	92.65	96.32	96.32
Lsvt-VR	83.91	74.71	74.71	77.01	82.76	75.86	78.16	78.16
mfeat Fourier	92.44	87.66	74.85	75	93.12	93.12	92.3	92.3

ovarian datasets. Decision Stump combined with eAdaBoost exhibited consistent gains across multiple datasets, including Ozone, Gas, SRBCT, and mfeat-fourier. NB Tree improvements were moderate, while Random Forest maintained similar accuracy across boosting methods.

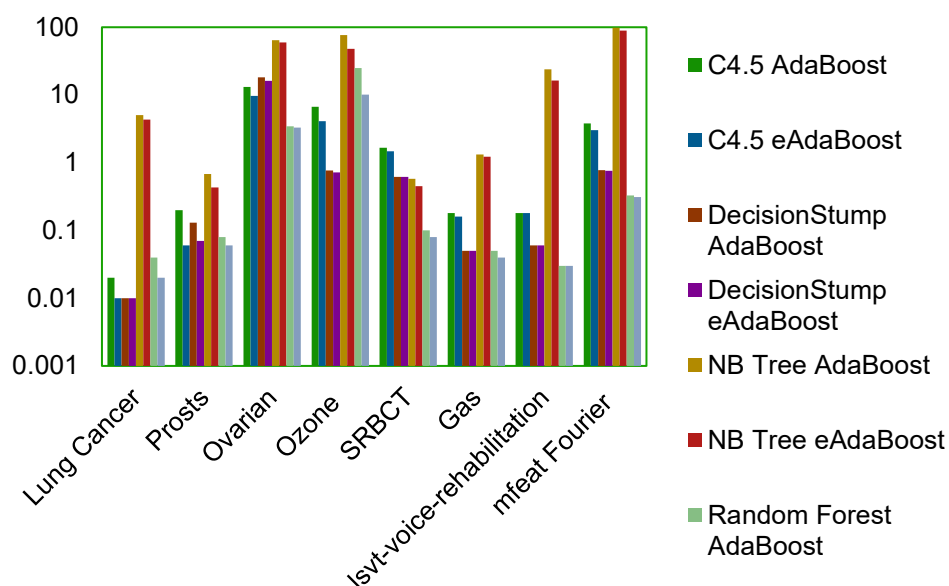


Figure 4: Computational Time for Hybrid Exclusive OR model with AdaBoost vs eAdaBoost in various classifiers

Error Rate Analysis- Table 3 compares Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) across classifiers. In most datasets, eAdaBoost either reduced or maintained equivalent error rates relative to AdaBoost, confirming its robustness and reliability. Particularly notable reductions in MAE and RMSE were observed in the lung cancer, ovarian, and mfeat-fourier datasets. The experimental findings clearly demonstrate that the proposed hybrid approach, when integrated with eAdaBoost, provides higher classification accuracy, fewer selected features, and lower computational cost compared to traditional AdaBoost. This balance of predictive power and efficiency underscores the method's suitability for high-dimensional biomedical and signal processing datasets, where both accuracy and scalability are critical.

Table 3 error rate between the Meta classifier, AdaBoost and eAbaBoost with various classifiers						
Datasets	Meta classifier	Types of Error Rate	C4.5	Decisio nStum p	NBTre e	Random Forest

Lung Cancer	AdaBoost	Mean absolute error	0.2587	0.3247	0.3271	0.3234
		Root mean squared error	0.4523	0.5584	0.5597	0.4193
	eAdaBoost	Mean absolute error	0.1716	0.2501	0.219	0.3234
		Root mean squared error	0.3852	0.4999	0.4675	0.4193
Prostate_tumorV SNormal	AdaBoost	Mean absolute error	0.1471	0.0882	0.1523	0.2029
		Root mean squared error	0.3835	0.297	0.123	0.2595
	eAdaBoost	Mean absolute error	0.1471	0.0882	0.1523	0.2029
		Root mean squared error	0.3835	0.297	0.123	0.2595
Ovarian	AdaBoost	Mean absolute error	0.0442	0.0079	0.189	0.1784
		Root mean squared error	0.2078	0.0758	0.2531	0.231
	eAdaBoost	Mean absolute error	0.0325	0.0278	0.189	0.1652
		Root mean squared error	0.1728	0.1663	0.2531	0.2397
Ozone	AdaBoost	Mean absolute error	0.03	0.0513	0.0579	0.029
		Root mean squared error	0.1699	0.1636	0.1986	0.169
	eAdaBoost	Mean absolute error	0.038	0.0289	0.0513	0.0288
		Root mean squared error	0.1946	0.1696	0.1836	0.1696
SRBCT	AdaBoost	Mean absolute error	0.0303	0.3403	0.0176	0.2042
		Root mean squared error	0.1727	0.3997	0.1311	0.2752
	eAdaBoost	Mean absolute error	0.0722	0.2163	0.0176	0.2042
		Root mean squared error	0.2667	0.3681	0.1311	0.2752
Gas	AdaBoost	Mean absolute error	0.0147	0.2674	0.0398	0.0279
		Root mean squared error	0.1212	0.3598	0.1226	0.1068
	eAdaBoost	Mean absolute error	0.0226	0.1351	0.0398	0.0279
		Root mean squared error	0.1486	0.2716	0.1226	0.1068
lsvt-voice- rehabilitation	AdaBoost	Mean absolute error	0.2542	0.2941	0.2243	0.2961
		Root mean squared error	0.4849	0.4834	0.4435	0.3728
	eAdaBoost	Mean absolute error	0.23	0.2502	0.2475	0.2961
		Root mean squared error	0.4793	0.4786	0.4688	0.3728
mfeat Fourier	AdaBoost	Mean absolute error	0.0151	0.171	0.171	0.0345
		Root mean squared error	0.1186	0.2855	0.2855	0.1172
	eAdaBoost	Mean absolute error	0.0238	0.0816	0.0816	0.0366
		Root mean squared error	0.1524	0.1933	0.1933	0.1155

Here is the statistical summary from the paired t-test comparing AdaBoost and eAdaBoost accuracies:

Table 4: statistical summary from the paired t-test			
Mean_AdaBoost	Mean_eAdaBoost	t_statistic	p_value
0.845	0.88875	-4.546	0.00265

Table 4 presents the results of a paired t-test comparing the classification accuracies of AdaBoost and eAdaBoost. The mean accuracy for AdaBoost is 0.845, while eAdaBoost achieves a higher mean accuracy of 0.88875, indicating an improvement in predictive performance. The paired t-test yields a t-statistic of -4.546 and a corresponding p-value of 0.00265.

The negative t-statistic reflects that eAdaBoost's accuracy is consistently higher than that of AdaBoost across the paired observations. The p-value, being less than 0.05, suggests that this difference is statistically significant, confirming that the improvement in accuracy by eAdaBoost is unlikely to have occurred by chance. In summary, eAdaBoost demonstrates a significant enhancement in performance compared to the standard AdaBoost algorithm.

Conclusion

This study introduces a Hybrid Feature Subset Selection (HFSS)–eAdaBoost framework designed to tackle the twin challenges of high-dimensional data and accurate classification. The method unifies three complementary feature selection techniques—Correlation-Based Feature Selection (CFS), Information Gain (IG), and ReliefF—whose outputs are merged through Intersection ($\cap$) and Exclusive-OR (XOR) strategies. These refined feature subsets are then provided to an enhanced boosting ensemble, effective AdaBoost (eAdaBoost), which adaptively reweights both samples and features at every iteration to lower error rates and sharpen decision boundaries.

Experiments on eight benchmark datasets from the UCI Machine Learning Repository—including biomedical, speech, and environmental data—validate the proposed approach. Base classifiers such as Decision Stump, C4.5, Random Forest, and Naïve Bayes Tree were paired with both standard AdaBoost and the proposed eAdaBoost. Across virtually all datasets, the HFSS–eAdaBoost combination achieved higher classification accuracy, lower mean absolute error (MAE), and reduced root mean square error (RMSE) compared with traditional AdaBoost or single feature-selection baselines. Notable gains were observed for the lung cancer, LSVT voice rehabilitation, ovarian cancer, and mfeat-fourier datasets, where eAdaBoost also shortened execution time and maintained consistent performance over repeated trials.

Feature-selection analysis showed that FS-INS and CFS consistently produced the most compact yet informative attribute sets, while the XOR merging strategy struck an effective balance between dimensionality reduction and information retention. This combination ensured strong predictive capability with minimal computational overhead—an essential property for large-scale or real-time applications. The adaptive reweighting in eAdaBoost further enhanced robustness by concentrating learning effort on difficult-to-classify instances, thereby reducing overfitting.

The research highlights that no single feature-selection or ensemble method is universally optimal. Instead, hybridization—combining diverse evaluators with adaptive boosting—emerges as the most reliable path to accuracy and efficiency. Beyond the tested datasets, the framework is readily applicable to other high-dimensional arenas such as bioinformatics, cybersecurity, industrial IoT analytics, and financial risk modeling.

Looking ahead, the model could be extended through deep learning–based feature extraction, metaheuristic optimizers, and streaming data adaptation, while incorporating explainable-AI tools to improve interpretability for sensitive domains like healthcare and finance.

Overall, the HFSS–eAdaBoost framework delivers superior accuracy, compact feature sets, and reduced computational cost, establishing a strong benchmark for future ensemble-based classification systems operating on complex, high-dimensional data.

Reference

- [1] Hannah Inbarani. H., Bagyamathi. M., Ahmad Taher Azar, "A novel hybrid attribute selection method based on rough set and improved harmony search" *Neural Comput & Applic*, DOI 10.1007/s00521-015-1840-0, (2015) Springer.
- [2] Mark. A. Hall, Geoffrey Holmes, "Benchmarking attribute selection techniques for discrete class data mining", *IEEE Transactions on Knowledge and Data Engineering* 15(6), 1437–1447, (2003),.
- [3] Dinakaran. S, Ranjit Jeba Thangaiah. P, "Ensemble Method of Effective AdaBoost Algorithm for Decision Tree Classifiers", *International Journal on Artificial Intelligence Tools* 26(3), 1750007-1 to 1750007-21, (2017).
- [4] Duda. R. O, Hart. P. E., Stork. D. G, "Pattern Classification", John Wiley & Sons, New York. (2001), pp. xx + 654, ISBN: 0-471-05669-3
- [5] Fermi Dwi Wicaksono, Yusri BinArshad, Haeryip Sihombing, "Norm-dist Monte-Carlo integrative method for the improvement of fuzzy analytic hierarchy process" *Heliyon*, Volume 6, Issue 4, April 2020, e03607, (2020).
- [6] Tien Thanh Nguyen, Thi Thu Thuy Nguyen, Xuan Cuong Pham, Alan Wee-Chung Liew "A novel combining classifier method based on Variational Inference", *Pattern Recognition* 49,198–212, (2016).
- [7] Dinakaran. S and Ranjit Jeba Thangaiah. P "Effective Hybrid Feature Subset Selection for multilevel Datasets using Decision Tree Classifiers" *International Journal of Advanced Intelligence Paradigms* 2023

- [8] Ran Li Jianjiang Lu, YafeiZhang Tianzhong Zhao "Dynamic Adaboost learning with feature selection based on parallel genetic algorithm for image annotation" *Knowledge-Based Systems* Volume 23, Issue 3, April, Pages 195-201, (2010).
- [9] Javier Apollonia, Guillermo Leguizamóna, Enrique Albaba LIDIC "Two hybrid wrapper-filter feature selection algorithms applied tohigh-dimensional microarray experiments" *Applied Soft Computing*, 38 (2016) 922–932, Elsevier.
- [10] Willmott, Cort J, Matsuura, Kenji "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance". *Climate Research*. 30: 79–82. (December 19, 2005).
- [11] Dulakshi Santhusitha and KumariKarunasingha, Root mean square error or mean absolute error? Use their ratio as well, *Information Sciences*, Volume 585, March 2022, Pages 609-629
- [12] Laura Frias Paredes, Fermin Mallor Martin, Gaston-Romeo Teresa Leon, "Dynamic mean absolute error as new measure for assessing forecasting errors", *Energy Conversion and Management* Volume 162, 15 April 2018, Pages 176-188.
- [13] X. Sun, S. Guo, S. Liu, J. Guo, B. Du et al., "A Correlation-Redundancy Guided Evolutionary Algorithm and Its Application to High-Dimensional Feature Selection in Classification," *Neural Processing Letters*, vol. 56, art. no. 86, 2024. Springer. DOI: 10.1007/s11063-024-11440-3.
- [14] Y. Yin, P. Prasad, R. K. Gupta and S. Tripathi, "IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," *Journal of Big Data*, vol. 10, art. no. 94, 2023. SpringerOpen. DOI: 10.1186/s40537-023-00694-8.
- [15] M. Mahapatra, S. K. Majhi and S. K. Dhal, "MRMR-SSA: a hybrid approach for optimal feature selection," *Evolutionary Intelligence*, vol. 15, no. 3, pp. 2017–2036, 2022. Springer. DOI: (check publisher page for DOI).
- [16] W. Shan, D. Li, M. Song, S. Liu, S. Xiao and H. Zhang, "A random feature mapping method based on the AdaBoost algorithm and results fusion for enhancing classification performance," *Expert Systems with Applications*, vol. 256, art. no. 124902, pp. 1–18, Dec. 2024. Elsevier. DOI: 10.1016/j.eswa.2024.124902.
- [17] B. Zhao, D. Yang, H. R. Karimi, B. Zhou, S. Feng and G. Li, "Filter-wrapper combined feature selection and AdaBoost-weighted broad learning system for transformer fault diagnosis under imbalanced samples," *Neurocomputing*, vol. 560, art. no. 126803, 2023. Elsevier. DOI: 10.1016/j.neucom.2023.126803.
- [18] J. Liu, D. Li, W. Shan and S. Liu, "A feature selection method based on multiple feature subsets extraction and result fusion for improving classification performance," *Applied Soft Computing*, 2024, art. no. 111018. Elsevier. DOI: 10.1016/j.asoc.2023.111018.
- [19] N. M. Abdelwahed, A. M. Hammad and A. H. El-Bendary, "Effective hybrid feature selection using different bootstrap and ensemble strategies," *Scientific Reports / SpringerOpen* style open literature
- [20] U. Ahmed et al., "Hybrid bagging and boosting with SHAP-based feature selection for intrusion detection," *Scientific Reports*, vol. 14, art. no. (2024). Nature Publishing Group / Springer Nature. DOI: 10.1038/s41598-024-81151-1
- [21] H. Wang, "Feature selection strategies: a comparative analysis of SHAP and other ranking methods," *Journal of Big Data*, 2024. SpringerOpen. DOI: 10.1186/s40537-024-00905-w.
- [22] S. Xie et al., "A new improved maximal relevance and minimal redundancy (ImRMR) feature selection method," *Applied Intelligence / open access* (2022).
- [23] Z. Liu et al., "Evolutionary feature selection based on hybrid Bald Eagle search for classification problems," *Intelligent Decision Technologies (IDA) / Elsevier-indexed journal* (2024).