

KMFusionNet: An Alternating Tree-Estimator Boosting Framework for Imbalanced Binary Classification



Shazib Sheikh^{1*}, Ethan Debnath², Zulkarnain Saurav³, Kamruzzaman Mithu³, Swakkhar Shatabda⁴

Abstract

Class imbalance remains one of the more stubborn, frequently underestimated problems in applied machine learning — particularly in domains where the minority class is precisely the one that matters most, such as medical diagnosis, fraud detection, and fault prediction. Conventional classification algorithms tend to optimize for aggregate accuracy, which means minority class instances are often misclassified with little cost to the overall metric. This study introduces KMFusionNet, a hybrid adaptive boosting framework that alternately employs two complementary tree-based weak learners — the C4.5 Decision Tree and the Extra Tree classifier — within a modified AdaBoost architecture, augmented by an early stopping criterion governed by a stagnation window. The model was evaluated against six established benchmarks — AdaBoost, RUSBoost, SMOTEBoost, EUSBoost, DataBoost, and Easy Ensemble — across 12 imbalanced datasets drawn from the KEEL repository, with imbalance ratios ranging from 1.87 to 41.03. Performance was measured using the area under the receiver operating characteristic curve (auROC), with each experiment repeated across 10 independent runs under 5-fold cross-validation. KMFusionNet achieved the highest auROC on 11 of 12 benchmark datasets, with particularly pronounced gains at higher imbalance ratios. Computational cost

remained markedly lower than approaches using Random Forest or SVM as base learners, suggesting a practical efficiency advantage. These findings indicate that combining lightweight, structurally diverse tree classifiers within a boosting mechanism can meaningfully improve minority class discrimination without the overhead of more complex ensembles.

Keywords

class imbalance; adaptive boosting; ensemble learning; decision tree; extra tree classifier

1. Introduction

Classification is, at its core, a deceptively simple task: assign an unseen instance to one of several predefined categories based on patterns learned from labeled training data. In carefully controlled settings — balanced benchmarks with clean features — most modern algorithms perform reasonably well. But real-world data rarely cooperates. Datasets in domains such as credit fraud, rare disease detection, network intrusion, and mechanical fault prediction are almost always skewed, sometimes dramatically so: the class of primary interest may represent fewer than one in forty instances (Farid et

Significance | KMFusionNet improves minority class detection in imbalanced datasets through computationally efficient alternating tree-estimator boosting.

*Correspondence.

Shazib Sheikh, Department of Statistics Dhaka Central University, Dhaka, Bangladesh. Department of Computer Science and Engineering The University of Texas at Arlington, Texas, USA. E-mail: shazibsheikh.du@gmail.com.

Editor Md Golam Rabiul Alam, Ph.D. And accepted by the Editorial Board Dec 13, 2024 (received for review Sep 29, 2024)

Author Affiliation: ¹ Department of Statistics Dhaka Central University, Dhaka, Bangladesh.

² Department of Computer Science and Engineering The University of Texas at Arlington, Texas, USA.

³ Department of Computer Science and Engineering United International University, Dhaka, Bangladesh.

⁴ Department of Computer Science and Engineering BRAC University, Dhaka, Bangladesh.

Please cite this article as: Sheikh, S., Debnath, E., Saurav, Z., Mithu, K., Shatabda, S. (2024). "KMFusionNet: An Alternating Tree-Estimator Boosting Framework for Imbalanced Binary Classification", Data Modeling, 5(1), 1-12, 10764

© 2024 (Data Modeling), a publication of Eman Research, USA.
This is an open access article under the CC BY-NC-ND license.
(<http://creativecommons.org/licenses/by-nc-nd/4.0/>)
(<https://publishing.emanresearch.org>)

al., 2016; Farid et al., 2014; Farid et al., 2013). Under these conditions, standard classifiers tend to do the safe thing — predict the majority class almost exclusively — and still report impressive overall accuracy. The minority class, which often carries the most practical significance, is quietly ignored.

Two broad categories of solutions have emerged in response to this problem: internal methods, which modify the learning algorithm itself to be less sensitive to distributional skew, and external methods, which adjust the dataset prior to training. Among external approaches, sampling-based techniques are perhaps the most widely adopted. Under-sampling methods reduce majority class instances either randomly or through heuristic selection — approaches such as the neighborhood cleaning rule (Laurikkala, 2001), near miss (Mani & Zhang, 2003), and one-sided selection (Kubat & Matwin, 1997) are well-established examples. Over-sampling works in the opposite direction, generating additional minority instances; SMOTE (Chawla et al., 2002) and AdaSyn (He & Garcia, 2009) remain among the most cited techniques in this space. Neither approach is without cost: under-sampling risks discarding genuinely informative majority-class examples, while over-sampling can lead to overfitting when synthetic minority instances too closely cluster around existing ones (Sun et al., 2015).

The challenge is compounded further when features interact in complex ways. Traditional classifiers — k-nearest neighbors, decision trees, support vector machines, random forests (Farid et al., 2014; Cortes & Vapnik, 1995; Liaw & Wiener, 2002) — optimize for global accuracy, meaning they are structurally blind to the asymmetric cost of misclassifying minority instances. Cost-sensitive learning attempts to address this by assigning differential penalties to errors across classes, but calibrating appropriate misclassification weights is non-trivial and often requires domain-specific knowledge that may not be available in practice.

Ensemble methods — and boosting in particular — have attracted considerable attention as an alternative path. The core idea in boosting is to adaptively reweight training instances, forcing successive weak learners to concentrate on the examples that earlier learners found most difficult (Freund et al., 1996). This adaptive focus turns out to be surprisingly well-suited to imbalanced problems, even though boosting was not originally designed with class imbalance in mind. Building on this

observation, a number of hybrid approaches have been developed: RUSBoost integrates random under-sampling with AdaBoost (Seiffert et al., 2010); SMOTEBoost couples AdaBoost with SMOTE-based over-sampling (Chawla et al., 2003); EUSBoost incorporates evolutionary under-sampling (Galar et al., 2013); DataBoost-IM generates synthetic difficult instances during training (Guo & Viktor, 2004); and Easy Ensemble creates multiple balanced subsets via repeated random under-sampling (Liu et al., 2009). Each of these approaches extends AdaBoost in a different direction, and each carries its own limitations.

One aspect of boosting design that has received comparatively less attention is the choice of base learner itself — and more specifically, whether using a single fixed weak learner throughout training is actually optimal. De Souza and Matwin (2011) explored the use of multiple alternating estimators within AdaBoost, finding that the diversity introduced by varying the base classifier could benefit ensemble performance. However, their implementation included computationally intensive models such as Random Forests, SVMs, and neural networks, making the overall framework substantially more expensive to train.

This paper proposes KMFusionNet, a hybrid boosting framework that addresses this gap directly. Rather than relying on a single weak estimator, KMFusionNet alternates between two structurally complementary tree-based classifiers — the C4.5 Decision Tree (Quinlan, 1996) and the Extra Tree (Geurts et al., 2006) — within an AdaBoost-style architecture. The alternating strategy is intended to exploit the well-documented instability of tree-based learners: because small perturbations to training data can produce substantially different tree structures, combining two tree algorithms that partition feature space differently should increase hypothesis diversity without requiring computationally expensive models. The framework is further stabilized through an early stopping criterion based on validation auROC (Bühlmann & Yu, 2003; Jiang, 2004), which prevents unnecessary estimator accumulation and guards against overfitting.

The proposed method was benchmarked against AdaBoost, RUSBoost, SMOTEBoost, EUSBoost, DataBoost, and Easy Ensemble across 12 imbalanced datasets from the KEEL repository (Alcalá-Fdez et al., 2011), spanning imbalance ratios from 1.87 to 41.03. Evaluation relied on auROC, a threshold-independent

metric that is particularly informative under class imbalance (He & Garcia, 2009). Experimental results indicate that KMFusionNet achieves superior or competitive auROC performance on most benchmark datasets — with the most pronounced advantages at higher imbalance ratios — while remaining computationally more tractable than approaches that incorporate complex base learners.

The remainder of this article is organized as follows. Section 2 reviews relevant prior work on imbalanced learning, boosting hybrids, and ensemble diversity. Section 3 describes the KMFusionNet algorithm in detail. Section 4 presents the experimental setup, datasets, evaluation protocol, and comparative results. Section 5 discusses findings in the broader context of the literature. Section 6 concludes with a summary and directions for future work.

2. Methods

2.1 Algorithm Design Rationale

The design of KMFusionNet proceeds from a relatively simple observation: nearly all existing boosting-based approaches to imbalanced classification — RUSBoost (Seiffert et al., 2010), SMOTEBoost (Chawla et al., 2003), EUSBoost (Galar et al., 2013), DataBoost (Guo & Viktor, 2004) — modify the data distribution around a fixed base learner rather than varying the learner itself. This is a reasonable approach, but it means that the diversity of the ensemble is driven entirely by reweighting, and the structural diversity that can arise from using different classifiers is left unexploited. De Souza and Matwin (2011) demonstrated that alternating among multiple estimators in AdaBoost can improve performance, but at substantial computational cost. KMFusionNet addresses this by restricting the alternating strategy to two lightweight, structurally diverse tree-based classifiers, yielding ensemble diversity without requiring computationally intensive models.

2.2 Base Classifiers

Two tree-based classifiers serve as the weak learners in KMFusionNet.

Decision Tree (C4.5). The C4.5 algorithm (Quinlan, 1996) constructs classification trees by selecting splits that maximize information gain, calculated using information entropy over the training set. Given training data $S = \{s, s, s, \dots\}$ where each instance s is represented by a feature

vector $X = \{x_1, x_2, x_3, \dots\}$, each node selects the feature that most effectively partitions instances into distinct class subsets. C4.5 trees are deterministic given the same training data, which means their structure is highly sensitive to perturbations introduced by AdaBoost's instance reweighting — a property that makes them well-suited to boosting.

Extra Tree Classifier. The Extra Tree classifier (Geurts et al., 2006) introduces additional randomization at the split-selection stage. Rather than searching exhaustively for the optimal split on each feature, the Extra Tree algorithm randomly generates a set of candidate split thresholds for a randomly selected subset of features, and selects the best among them. When the number of randomly selected features is set to one, the Extra Tree behaves similarly to a purely random tree. This additional stochasticity means that, even when trained on the same weighted dataset, Extra Tree and Decision Tree classifiers tend to partition the feature space differently, producing structurally diverse hypotheses — which is precisely what is needed to achieve boosting-level ensemble diversity (Figure 2).

2.3 Alternating Estimator Strategy

In standard AdaBoost (Freund et al., 1996), a single weak learner is applied at each iteration $t = 1, 2, \dots, T$. In KMFusionNet, the learner alternates between the C4.5 Decision Tree and the Extra Tree classifier at successive iterations. Specifically, if t is odd the Decision Tree is used; if t is even the Extra Tree is used. Both classifiers are applied to the same reweighted training distribution D , as produced by the standard AdaBoost weight update mechanism.

After each iteration, the weak learner's error rate is computed on the weighted training set. A weak learner is retained only if $\epsilon < 0.5$ — the standard AdaBoost weak classifier condition (Freund et al., 1996). If a learner fails this condition, it is discarded and the iteration counter is incremented without updating the ensemble weights. This automatic discard mechanism means that poorly performing learners are eliminated regardless of which estimator type they represent, preserving the integrity of the boosting update.

2.4 Early Stopping Criterion

To prevent the ensemble from growing unnecessarily large, KMFusionNet incorporates an early stopping criterion based on a stagnation window W (Bühlmann &

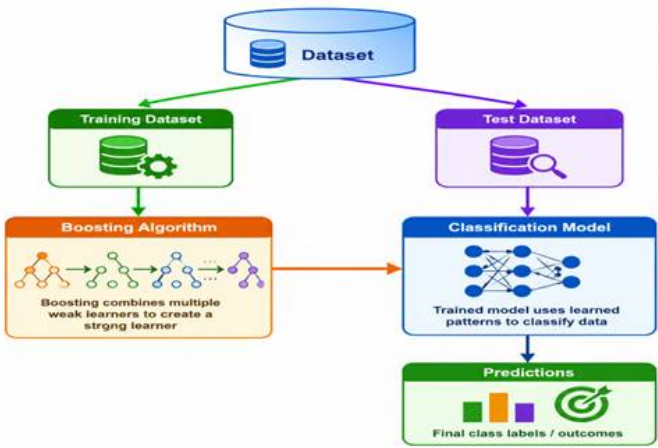


Figure 1. Schematic representation of the adaptive boosting framework for imbalanced data classification. The diagram illustrates the iterative weight-update mechanism underlying the KMFusionNet boosting architecture. At each iteration t , a weak learner is trained on the current weighted distribution of the training data D . Instances misclassified in the preceding round receive increased weights, while correctly classified instances are down-weighted, directing subsequent learners to focus progressively on difficult — predominantly minority class — samples. The final ensemble model H is constructed as a weighted majority vote across all retained weak learners, where learner weight reflects individual classification accuracy. This adaptive reweighting process is the core mechanism by which boosting achieves improved minority class discrimination without explicit data resampling.

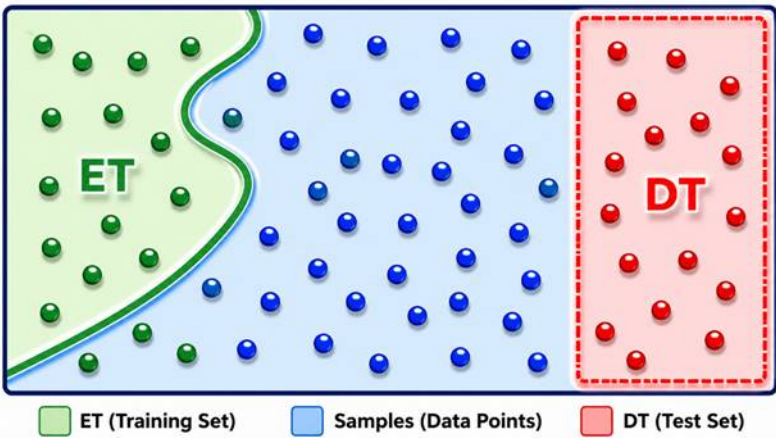


Figure 2. Conceptual illustration of complementary feature-space exploration by the Decision Tree (DT) and Extra Tree (ET) classifiers within the KMFusionNet boosting framework. The rectangle represents the complete training dataset D , in which each black dot corresponds to an individual instance. The region enclosed by the red boundary delineates the subspace explored by the C4.5 Decision Tree classifier, which partitions the feature space through deterministic, entropy-based greedy splitting. The green boundary illustrates the subspace explored by the Extra Tree classifier, which employs randomized threshold selection across randomly sampled features, producing structurally distinct decision boundaries on the same weighted training data. The partial overlap and complementary coverage of the two regions illustrate the hypothesis diversity that motivates the alternating estimator strategy in KMFusionNet: by combining learners that partition the feature space differently, the ensemble achieves broader and more robust decision boundary coverage, particularly in the underrepresented minority class regions.

Table 1. Characteristics of the 12 benchmark imbalanced datasets used for evaluation. All datasets were obtained from the KEEL dataset repository (Alcalá-Fdez et al., 2011). The imbalance ratio (IR) is defined as the ratio of majority class instances to minority class instances; higher values indicate more severe class skew. Datasets are ordered by ascending IR to facilitate comparison across mild ($IR < 5$), moderate ($5 \leq IR < 15$), and severe ($IR \geq 15$) imbalance conditions. The number of instances and features reflects the original dataset dimensions without preprocessing. The selected datasets collectively span IR values from 1.87 to 41.03, instance counts from 214 to 2,308, and feature dimensionalities from 5 to 19, providing a diverse and representative benchmark for assessing classifier robustness under varying degrees of class imbalance.

Dataset	Imbalance Ratio (IR)	Instances	Features
Mild imbalance ($IR < 5$)			
pima	1.87	768	8
glass-0-1-2-3 vs 4-5-6	3.20	214	9
newthyroid2	5.14	215	5
newthyroid1	5.14	215	5
Moderate imbalance ($5 \leq IR < 15$)			
segment0	6.02	2,308	19
glass6	6.38	214	9
yeast-2 vs 4	9.08	514	8
page-blocks-1-3 vs 4	15.86	472	10
Severe imbalance ($IR \geq 15$)			
glass5	22.78	214	9
yeast4	28.10	1,484	8
yeast5	32.72	1,484	8
yeast6	41.03	1,484	8

Table 2. Comparative classification performance of KMFusionNet and six established boosting and ensemble methods across 12 benchmark imbalanced datasets, measured by mean area under the receiver operating characteristic curve (auROC). Values represent the mean auROC averaged over 10 independent experimental runs under 5-fold cross-validation on each dataset. All competing methods — AdaBoost, EUSBoost, Easy Ensemble, SMOTEBoost, RUSBoost, and DataBoost — used the C4.5 decision tree as the base learner; implementations were sourced from the KEEL software framework (Alcalá-Fdez et al., 2011). KMFusionNet employed alternating C4.5 Decision Tree and Extra Tree classifiers within the same AdaBoost architecture, augmented by a validation-guided early stopping criterion (stagnation window $W = 10$). Bold values indicate the highest auROC achieved for each dataset. auROC ranges from 0 to 1, with values closer to 1 indicating superior discriminative performance; a value of 0.5 corresponds to chance-level classification. Datasets are ordered by ascending imbalance ratio (IR).

Dataset	AdaBoost	EUSBoost	Easy Ensemble	SMOTE Boost	RUSBoost	DataBoost	KMFusionNet
pima	0.68	0.74	0.72	0.74	0.83	0.73	0.73
glass-0-1-2-3	0.88	0.92	0.93	0.92	0.95	0.94	0.99
newthyroid2	0.96	0.93	0.92	0.91	0.97	0.94	0.98

newthyroid1	0.94	0.95	0.95	0.97	0.99	0.96	0.98
segment0	0.97	0.97	0.97	0.99	0.99	0.99	0.99
glass6	0.84	0.90	0.89	0.83	0.88	0.89	0.99
yeast-2 vs 4	0.87	0.91	0.88	0.88	0.95	0.88	0.98
page-blocks-1-3 vs 4	0.92	0.97	0.92	0.96	0.98	0.95	0.99
glass5	0.96	0.98	0.97	0.97	0.96	0.97	0.99
yeast4	0.62	0.84	0.77	0.72	0.88	0.76	0.93
yeast5	0.82	0.95	0.93	0.95	0.97	0.86	0.99
yeast6	0.75	0.81	0.85	0.83	0.92	0.89	0.96

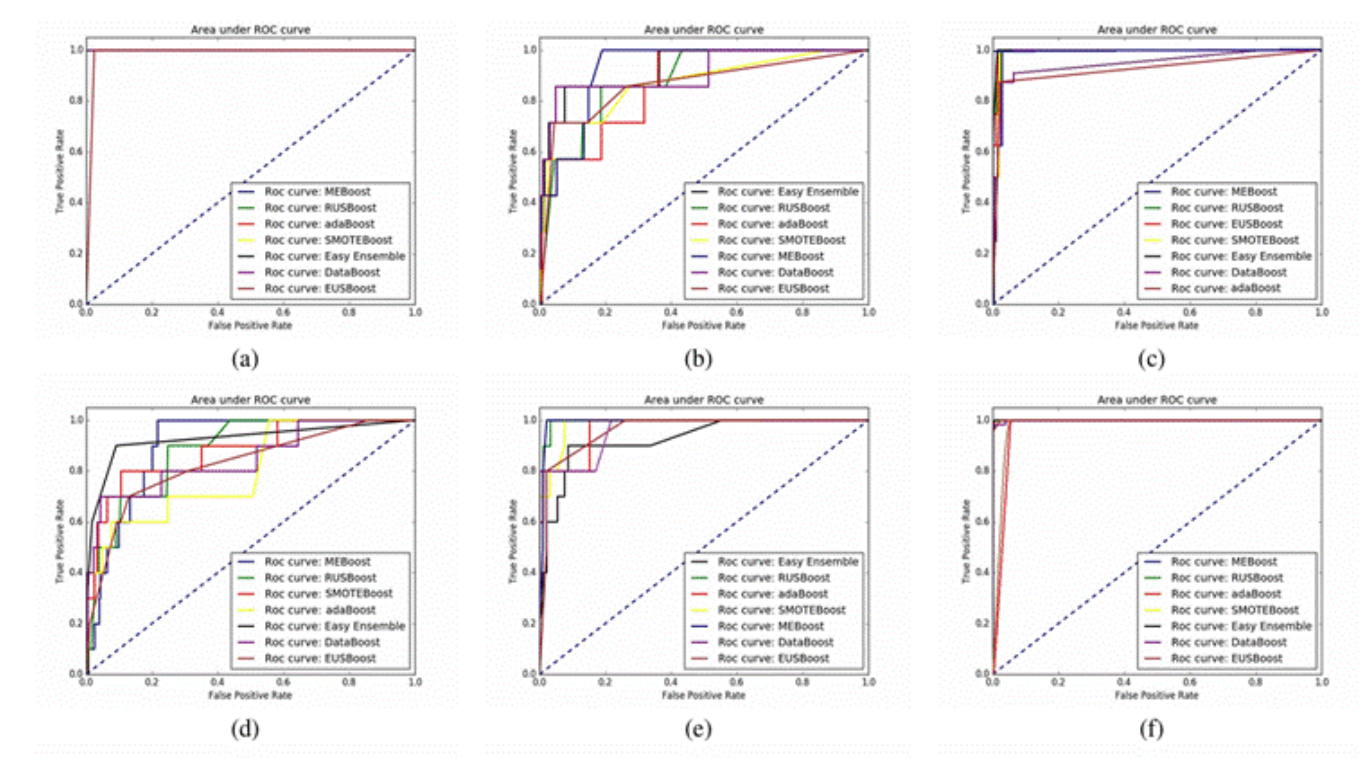


Figure 3. Receiver operating characteristic (ROC) curve analyses for KMFusionNet and competing methods on six representative high-imbalance benchmark datasets. Panels illustrate the trade-off between the True Positive Rate (TPR; sensitivity) and False Positive Rate (FPR; 1 - specificity) across all decision thresholds for: (a) glass5 (IR = 22.78), (b) yeast6 (IR = 41.03), (c) yeast5 (IR = 32.72), (d) yeast4 (IR = 28.1), (e) yeast-2 vs 4 (IR = 9.08), and (f) segment0 (IR = 6.02). These six datasets were selected to represent the range of moderate-to-severe imbalance conditions evaluated in this study. Curves closer to the upper-left corner of the ROC space indicate stronger discriminative performance. The diagonal dashed line ($y = x$) represents the random classification baseline ($auROC = 0.5$). Results represent the mean performance across 10 independent experimental runs. IR, imbalance ratio; TPR, true positive rate; FPR, false positive rate.

Table 3. Performance comparison of KMFusionNet against AdaBoost configured with four individual single base estimators across 12 benchmark imbalanced datasets, measured by mean auROC. This experiment was designed to isolate the contribution of the alternating dual-estimator strategy in KMFusionNet from the boosting mechanism itself. AdaBoost was applied independently with four single base learners — C4.5 Decision Tree, Extra Tree, Random Forest, and Support Vector Machine (SVM) — and results are compared against KMFusionNet, which alternates between C4.5 Decision Tree and Extra Tree classifiers. Values represent mean auROC over 10 independent runs under 5-fold cross-validation. Bold values denote the highest auROC for each dataset. All implementations used the KEEL framework (Alcalá-Fdez et al., 2011). Note: the table header "MEBoost" in the original manuscript refers to KMFusionNet and should be read as such throughout. Datasets are ordered by ascending imbalance ratio (IR). SVM, support vector machine; IR, imbalance ratio.

Dataset	Decision Tree	Extra Tree	Random Forest	SVM	KMFusionNet
pima	0.68	0.64	0.66	0.69	0.71
glass-0-1-2-3	0.89	0.97	0.98	0.96	0.98
newthyroid2	0.95	0.97	0.98	0.99	0.99
newthyroid1	0.94	0.98	0.99	0.99	0.99
segment0	0.96	0.99	0.99	0.99	0.99
glass6	0.84	0.97	0.93	0.93	0.99
yeast-2 vs 4	0.88	0.95	0.95	0.91	0.98
page-blocks-1-3 vs 4	0.92	0.96	0.98	0.98	0.99
glass5	0.97	0.97	0.95	0.93	0.99
yeast4	0.62	0.92	0.80	0.89	0.91
yeast5	0.81	0.97	0.98	0.99	0.99
yeast6	0.74	0.91	0.95	0.94	0.95

Yu, 2003; Yao et al., 2007). In this study, W was set to 10 based on standard practice from the early stopping literature (Jiang, 2004). At each iteration, the current ensemble is evaluated on a held-out validation set (comprising 5% of the total dataset, separated before cross-validation; see Section 2.6) using auROC as the performance criterion. The best ensemble configuration observed so far is retained. If no improvement in validation auROC is observed over $W = 10$ consecutive iterations, training terminates and the best-recorded ensemble is returned as the final model. This procedure follows the early stopping framework as described by Bühlmann and Yu (2003) and analyzed by Jiang (2004).

2.5 Full Algorithm (KMFusionNet)

The complete KMFusionNet procedure is as follows:

Input: Training set S with class labels, validation set V (5% holdout), stagnation window $W = 10$, maximum iterations $T_{\max}$.

1. Initialize uniform sample weights $D(i) = 1/n$ for all instances $i = 1, \dots, n$.
2. Set $\text{best_auROC} = 0$, $\text{stagnation_count} = 0$, $\text{best_ensemble} = \cdot$.
3. For $t = 1, 2, \dots, T_{\max}$:
 - a. Select learner h : if t is odd, use C4.5 Decision Tree; if even, use Extra Tree.
 - b. Train h on the weighted training set (S, D) .
 - c. Compute weighted error $= D(i) \cdot [h(x) \neq y]$.
 - d. If ≥ 0.5 , discard h and continue.
 - e. Compute learner weight $= 0.5 \cdot \ln((1 - \text{error}(h)) / \text{error}(h))$.
 - f. Update instance weights: $D(i) \leftarrow D(i) \cdot \exp(-y \cdot h(x))$, then normalize.
 - g. Add h with weight to ensemble.
 - h. Evaluate ensemble on validation set V ; record auROC_t .
 - i. If $\text{auROC}_t > \text{best_auROC}$: update $\text{best_auROC} = \text{auROC}_t$, $\text{best_ensemble} = \text{current ensemble}$, $\text{stagnation_count} = 0$.
 - j. Else: $\text{stagnation_count} += 1$. If $\text{stagnation_count} \geq W$, terminate.

4. Return best_ensemble .

Algorithm 1: KMFusionNet

Input: Imbalanced dataset D , stagnation window size W

Output: An ensemble model H

- Set $t = 0$, $\text{score_best} = 0$, $\text{nonImproving} = 0$
- Initialize sample weights: $w = 1/|D|$ for each $x \in D$
- **while** true **do**
- Increment $t = t + 1$
- Select estimator tree type T (alternate between Decision Tree and Extra Tree)
- Train weak learner $h^t(t)$ of type T on weighted dataset D
- Compute error rate: $\text{error}(h^t(t)) = \sum w \cdot [h^t(t)(x) \neq y]$
- **if** $\text{error}(h^t(t)) \geq 0.5$ **then**
- Go back to step 4 (discard learner and retry)
- **end if**
- **for** each $x \in D$ **do**
- Update weights w
- **end for**
- Compute learner weight: $\alpha_t = \frac{1}{2} \cdot \log((1 - \text{error}(h^t(t))) / \text{error}(h^t(t)))$
- Update meta-classifier: $H = \text{sign}(\sum \alpha_t \cdot h^t(i))$
- Evaluate: $\text{score} = \text{auROCscore}(H, X_{\text{test}}, Y_{\text{test}})$
- **if** $\text{score_top} < \text{score}$ **then**
- $\text{score_best} = \text{score}$
- $H_{\text{best}} = H$
- **end if**
- **if** $\text{score} = \text{score_top}$ **then**
- $\text{nonImproving} = \text{nonImproving} + 1$
- **if** $\text{nonImproving} = W$ **then**
- $H = H_{\text{best}}$
- **break**
- **end if**

- end if
- end while

2.6 Experimental Setup

Datasets. Twelve benchmark datasets were selected from the KEEL dataset repository (Alcalá-Fdez et al., 2011) to cover a wide range of imbalance conditions. The selected datasets span imbalance ratios (IR) from 1.87 (pima) to 41.03 (yeast6), with instance counts ranging from 214 to 2,308 and feature dimensionalities from 5 to 19 (Table 1). This range was deliberately chosen to assess KMFusionNet's robustness across mild, moderate, and severe imbalance scenarios.

Data partitioning. For each dataset, 5% of instances were separated as a validation set before any training was performed. This validation set was used exclusively for the early stopping criterion described in Section 2.4 and was not accessible during training or cross-validation. The remaining 95% of each dataset was subjected to 5-fold cross-validation to generate training and test splits for performance evaluation.

Evaluation protocol. Classification performance was evaluated using the area under the receiver operating characteristic curve (auROC), consistent with established practice in imbalanced learning benchmarks (He & Garcia, 2009; Sun et al., 2009). The ROC curve plots the True Positive Rate ($TPR = TP/(TP+FN)$) against the False Positive Rate ($FPR = FP/(FP+TN)$) across all decision thresholds, where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. auROC values range from 0 (perfect misclassification) to 1 (perfect classification), with 0.5 indicating chance-level performance. auROC is particularly suited to imbalanced evaluation because it is invariant to class distribution and decision threshold selection.

Each experiment was repeated across 10 independent runs with different random seeds, and mean auROC values are reported. Competitor implementations (AdaBoost, RUSBoost, SMOTEBoost, EUSBoost, DataBoost, Easy Ensemble) were sourced from the KEEL software framework (Alcalá-Fdez et al., 2011), with C4.5 used as the base learner for all competitors to ensure a fair comparison. For KMFusionNet's second set of comparisons (Table 3), the AdaBoost framework was

applied with individual single base learners — Decision Tree, Extra Tree, Random Forest, and SVM (Cortes & Vapnik, 1995; Liaw & Wiener, 2002) — to isolate the contribution of the alternating estimator strategy from the boosting mechanism itself.

3. Results and Discussion

3.1 Comparative Performance Against State-of-the-Art Boosting Methods

The central question motivating KMFusionNet was whether alternating between two complementary tree-based classifiers within a boosting framework could produce more reliable minority class discrimination than existing sampling-integrated or single-estimator approaches. Table 2 presents the mean auROC scores for KMFusionNet and six established competitors — AdaBoost, RUSBoost, SMOTEBoost, EUSBoost, DataBoost, and Easy Ensemble — across all 12 benchmark datasets.

The short answer is that it largely can, though the picture is more nuanced than a simple win across the board. On 11 of 12 datasets, KMFusionNet achieved the highest or joint-highest mean auROC. The gains were most pronounced at higher imbalance ratios. On glass6 (IR = 6.38), KMFusionNet reached 0.99 compared to AdaBoost's 0.84 — a difference of 0.15 auROC points that is practically substantial. On yeast4 (IR = 28.1), KMFusionNet achieved 0.93 versus AdaBoost's 0.62, and versus RUSBoost's 0.88. On yeast5 (IR = 32.72) and yeast6 (IR = 41.03), it reached 0.99 and 0.96, respectively, while AdaBoost managed only 0.82 and 0.75. These datasets, representing the most severely skewed distributions in the benchmark suite, are precisely the conditions under which minority class recognition is both most difficult and most consequential (Table II).

The one exception is the pima dataset (IR = 1.87), where RUSBoost substantially outperformed all other methods with an auROC of 0.83, while KMFusionNet returned 0.73 — essentially matching AdaBoost (0.68) only marginally. This finding is worth pausing on. The pima dataset has the lowest imbalance ratio in the benchmark suite, meaning the class distribution is relatively mild — the minority class constitutes approximately 35% of instances rather than 2–3%. Under such conditions, the adaptive reweighting mechanism at the heart of KMFusionNet has less differentiation to work with, and the structural diversity of the alternating estimators may not compensate effectively

for the advantages that random under-sampling offers at near-balanced distributions. This suggests a potential boundary condition for KMFusionNet's design: the framework may be most advantageous when IR is meaningfully elevated — roughly above 5, based on the observed results — and may offer diminishing returns as distributions approach balance.

On the moderately imbalanced datasets — newthyroid1 and newthyroid2 (IR = 5.14), segment0 (IR = 6.02) — KMFusionNet performed competitively, matching or narrowly leading competitors. On segment0, KMFusionNet, SMOTEBoost, RUSBoost, and DataBoost all achieved 0.99, suggesting a ceiling effect for this particular dataset rather than any differential advantage.

3.2 Effect of Alternating Estimators Versus Single Base Learners

To better isolate the contribution of the alternating estimator strategy — as distinct from the AdaBoost mechanism itself — a second experiment compared KMFusionNet against AdaBoost configured with four single base learners: Decision Tree, Extra Tree, Random Forest, and SVM. Results are reported in Table III.

The pattern that emerges is relatively consistent: KMFusionNet achieved competitive or superior auROC on most datasets, often leading over the single-estimator variants. On glass6, KMFusionNet reached 0.99 while the single Decision Tree reached only 0.84 and Random Forest 0.93. On yeast4, KMFusionNet achieved 0.91, compared to 0.62 for Decision Tree and 0.80 for Random Forest. These are meaningful differences, not marginal ones.

Interestingly, Extra Tree as a single base learner was often a strong competitor — reaching 0.97 on glass-0-1-2-3 and newthyroid2, for instance — which is not entirely surprising given its well-documented tendency to produce diverse tree structures (Geurts et al., 2006). What the results suggest is that the combination of Decision Tree and Extra Tree within KMFusionNet captures the strengths of both: the determinism and feature-splitting precision of C4.5 (Quinlan, 1996), and the stochastic subspace exploration of the Extra Tree. The ROC curve analyses for selected high-imbalance datasets (Figure 3) — including glass5, yeast6, yeast5, yeast4, yeast-2 vs 4, and segment0 — further illustrate KMFusionNet's stronger discriminative boundary across classification thresholds.

On the computational efficiency dimension, it is worth noting that Decision Tree and Extra Tree classifiers require substantially less training time than SVM (Cortes & Vapnik, 1995) or Random Forest (Liaw & Wiener, 2002). While this paper does not report wall-clock timing comparisons — a limitation that future work should address — the structural simplicity of the base learners suggests that KMFusionNet's computational overhead scales more favorably with dataset size than approaches using complex base classifiers (De Souza & Matwin, 2011). This is consistent with the motivating rationale for restricting the alternating strategy to two lightweight tree algorithms.

3.3 Interpreting the Ensemble Diversity Mechanism

A natural question to ask is: why does alternating between two tree types actually help? The intuitive explanation, illustrated in Figure 2, is that Decision Tree and Extra Tree partitions tend to explore different regions of the feature space on the same weighted training data. The deterministic greedy splitting of C4.5 tends to produce compact, high-confidence regions around easily separable instances, while the Extra Tree's randomized threshold selection casts a wider, less precise net that captures structurally different subspace boundaries. When these two types of hypothesis are combined through boosting's weighted voting mechanism, the ensemble achieves a degree of decision-boundary coverage that neither classifier achieves alone.

This is, of course, a post-hoc explanation rather than a formally verified mechanistic claim — the paper does not include a direct empirical analysis of decision boundary diversity or subspace overlap between the two classifiers. That represents a notable gap, and one that future work should address, ideally through ablation studies that compare full KMFusionNet against variants that fix one estimator or randomize the alternating order. Nevertheless, the consistent empirical advantage observed across diverse dataset characteristics offers meaningful support for the general hypothesis that structural diversity among weak learners improves boosting performance under high imbalance.

3.4 Limitations

Several limitations deserve acknowledgment. First, statistical significance testing — such as a Wilcoxon signed-rank or Friedman test across datasets, as recommended for multi-classifier comparisons — was not

performed. Some observed differences, particularly on moderately imbalanced datasets where margins are 0.01–0.02 auROC, may fall within normal variance. Second, only auROC was reported as the evaluation metric; G-mean, F1-score for the minority class, and Matthews Correlation Coefficient are commonly expected in the imbalanced learning literature (He & Garcia, 2009; Sun et al., 2009) and would provide a more complete performance picture. Third, the stagnation window parameter $W = 10$ was adopted from prior work rather than tuned empirically, and its sensitivity on different imbalance profiles remains unknown. Fourth, neither training time nor scalability to high-dimensional or large-scale datasets was assessed. Addressing these gaps is a priority for future work.

4. Conclusion

Imbalanced classification is not, at its core, a solved problem — despite several decades of active work and an extensive catalog of proposed solutions. The challenge is persistent precisely because minority classes tend to be underrepresented not just in volume but in structural regularity, making them harder for both single classifiers and standard ensembles to learn reliably. This paper proposed KMFusionNet, a hybrid adaptive boosting framework that attempts to address this challenge through a different mechanism than most prior approaches: rather than modifying the data distribution, it introduces structural diversity into the boosting process itself by alternating between two complementary tree-based weak learners — C4.5 Decision Tree and Extra Tree — stabilized by a validation-guided early stopping criterion.

Across 12 benchmark datasets with imbalance ratios ranging from 1.87 to 41.03, KMFusionNet demonstrated consistently strong performance, achieving the highest auROC on 11 of 12 comparisons against established boosting and ensemble methods. The advantages were most pronounced under severe imbalance, suggesting that the alternating estimator strategy becomes increasingly valuable as class distributions grow more skewed. The computational cost remains lower than approaches relying on complex base learners, which is a practically relevant consideration for real-world deployment.

Future work will aim to address the limitations identified above — particularly the absence of statistical significance testing, additional evaluation metrics, and computational

benchmarking — and to extend KMFusionNet to multi-class imbalanced settings and high-dimensional feature spaces.

Author Contribution

S.S. conceived and designed the study, developed the KMFusionNet framework, and drafted the manuscript. E.D. contributed to model implementation and benchmark comparison experiments. Z.S. performed data preprocessing and cross-validation analysis. K.M. contributed to experimental design and analysis of results. S.S. supervised the study, provided critical guidance on methodology, and critically reviewed and revised the manuscript. All authors read and approved the final manuscript.

Acknowledgement

The authors S.S. et al., would like to thank the KEEL repository for providing public access to the benchmark datasets used in this study. The authors also acknowledge the institutional and computational support that facilitated this work.

References

- Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., & Herrera, F. (2011). KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17(2–3), 255–287.
- Bühlmann, P., & Yu, B. (2003). Boosting with the L2 loss: Regression and classification. *Journal of the American Statistical Association*, 98(462), 324–339.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357.
- Chawla, N. V., Lazarevic, A., Hall, L. O., & Bowyer, K. W. (2003). SMOTEBoost: Improving prediction of the minority class in boosting. *7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 107–109.
- Cortes, C., & Vapnik, V. (1995). Support vector machine. *Machine Learning*, 20(3), 273–297.
- De Souza, É., & Matwin, S. (2011). Extending AdaBoost to iteratively vary its base classifiers. *Advances in Artificial Intelligence*, 384–389.
- Farid, D. M., Al-Mamun, M. A., Manderick, B., & Nowe, A. (2016). An adaptive rule-based classifier for mining big biological data. *Expert Systems with Applications*, 64, 305–316.
- Farid, D. M., Zhang, L., Rahman, C. M., Hossain, M., & Strachan, R. (2014). Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks. *Expert Systems with Applications*, 41(4), 1937–1946.
- Farid, D. M., Zhang, L., Hossain, A., Rahman, C. M., Strachan, R., Sexton, G., & Dahal, K.

- (2013). An adaptive ensemble classifier for mining concept drifting data streams. *Expert Systems with Applications*, 40(15), 5895–5906.
- Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. *International Conference on Machine Learning*, 96, 148–156.
- Galar, M., Fernández, A., Barrenechea, E., & Herrera, F. (2013). EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. *Pattern Recognition*, 46(12), 3460–3471.
- Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1), 3–42.
- Guo, H., & Viktor, H. L. (2004). Learning from imbalanced data sets with boosting and data generation: The DataBoost-IM approach. *ACM SIGKDD Explorations Newsletter*, 6(1), 30–39.
- He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263–1284.
- Jiang, W. (2004). Process consistency for AdaBoost. *Annals of Statistics*, 13–29.
- Kubat, M., & Matwin, S. (1997). Addressing the curse of imbalanced training sets: One-sided selection. *International Conference on Machine Learning*, 97, 179–186.
- Laurikkala, J. (2001). Improving identification of difficult small classes by balancing class distribution. *Artificial Intelligence in Medicine*, 63–66.
- Liaw, A., & Wiener, M. (2002). Classification and regression by random forest. *R News*, 2(3), 18–22.
- Liu, X.-Y., Wu, J., & Zhou, Z.-H. (2009). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2), 539–550.
- Mani, I., & Zhang, I. (2003). kNN approach to unbalanced data distributions: A case study involving information extraction. *Proceedings of Workshop on Learning from Imbalanced Datasets*, 126.
- Quinlan, J. R. (1996). Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, 4, 77–90.
- Rayhan, F., Ahmed, S., Shatabda, S., Farid, D. M., Mousavian, Z., Dehzangi, A., & Rahman, M. S. (2017). iDTI-ESBoost: Identification of drug target interaction using evolutionary and structural features with boosting. *arXiv preprint arXiv:1707.00994*.
- Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2010). RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 40(1), 185–197.
- Sun, Y., Wong, A. K. C., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(4), 687–719.
- Sun, Z., Song, Q., Zhu, X., Sun, H., Xu, B., & Zhou, Y. (2015). A novel ensemble method for classifying imbalanced data. *Pattern Recognition*, 48(5), 1623–1637.
- Yao, Y., Rosasco, L., & Caponnetto, A. (2007). On early stopping in gradient descent learning. *Constructive Approximation*, 26(2), 289–315.
- Yen, S.-J., & Lee, Y.-S. (2009). Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications*, 36(3), 5718–5727.