

# **XGBoost Machine Learning Algorithm: A Literature Review of Performance, Optimization, Scalability, Interpretability, and Domain Applications**

**Rupali Kumari**  
Government Engineering College Banka  
GEC Banka  
Bihar, India - 813102

**Amit Kumar**  
Government Engineering College Banka  
GEC Banka  
Bihar, India - 813102

**Nilabh Kumar**  
Government Engineering College Banka  
GEC Banka  
Bihar, India - 813102

## **ABSTRACT**

XGBoost is consistently presented as a high-performing tree-boosting algorithm that is both accurate and scalable, but the advantages are context dependent rather than universal. Across the 20 studies synthesized in this review, XGBoost consistently outperforms simpler baselines and several traditional machine-learning approaches in educational prediction, mining, seismic early warning, house-price prediction, and survival regression, whereas tuned alternatives such as CatBoost or specialized optimization pipelines may achieve better results on specific datasets, as in concrete strength estimation. The most obvious pattern is that XGBoost's efficacy is highly dependent on hyperparameter selection: for example, meta-learning-based recommendations improved accuracy in 78.28% of 198 datasets relative to grid search. These results are significant because, although XGBoost is a popular general-purpose prediction engine, its usefulness is influenced by domain-specific calibration, computing efficiency, data imbalance, and interpretability. XGBoost can scale to big datasets, with GPU implementations reaching  $3\times$  to  $6\times$  speedups over 4-core CPUs and  $1.2\times$  over 24-core CPUs, and a separate benchmarking study found an average GPU speedup of  $7.26\times$ . However, interpretability remains a challenge in critical sectors, motivating techniques that emulate XGBoost with transparent decision trees. The evidence supports XGBoost as a strong baseline and frequently a top performer, but not always the best choice, especially where interpretability, extreme class imbalance, or accurately calibrated rival boosters are critical.

## **General Terms**

Machine Learning, Algorithms, Performance Evaluation

## **Keywords**

XGBoost, Extreme gradient boosting, Machine learning, CatBoost, Hyperparameter optimization, Scalable tree boosting, Interpretability

## **1. INTRODUCTION**

Extreme Gradient Boosting, or XGBoost [1], has become one of the most significant machine-learning algorithms in modern predictive modeling due to its excellent accuracy and practical scalability. As a gradient boosting framework, it successively constructs an ensemble of decision trees, with each new tree rectifying the mistakes of the preceding ones. This design has made it especially attractive for structured data, where nonlinear interactions, heterogeneous feature effects, and missingness are common. The reviewed literature portrays XGBoost not merely as a classifier or regressor, but as a flexible modeling backbone that is increasingly adapted through GPU acceleration [2, 3], automated hyperparameter optimization [4, 5, 6], domain-specific loss functions [7], and interpretability post-processing [8]. However, in high-stakes settings such as healthcare and finance, predictive performance alone is insufficient if the rationale behind predictions cannot be understood [8]. In large-scale applications, the numerous configurable parameters can make reliable deployment computationally costly [9, 10]. In imbalanced classification, the advantages of boosted trees are not assured [11]. These issues have led to a number of studies that compare XGBoost to other learners [12, 13, 14], improve its training approach [15], and apply it to specific tasks such as survival regression [7] and seismic early warning [16]. Because the literature includes technical breakthroughs, benchmark comparisons, and application-specific studies, a review is required to identify what is known about XGBoost's capabilities and limitations. A review that integrates these strands can clarify where XGBoost is robustly effective, where it depends on tuning or augmentation, and where competing approaches may be preferable. The

present review therefore examines the evidence on XGBoost as a machine-learning algorithm across predictive tasks, focusing on performance, computational efficiency, interpretability, and practical deployment considerations. It addresses four questions:

- How does XGBoost perform relative to competing learners across classification, regression, and other predictive tasks?
- How strongly do hyperparameter selection and the choice of optimization strategy affect its performance?
- What scalability and efficiency gains are reported, particularly with GPU acceleration?
- How interpretable are its predictions, and how does class imbalance affect its relative standing?

The remainder of the paper is organized as follows. Section 2 describes the search, selection, and synthesis methods. Section 3 reports the results, including study characteristics, quantitative outcomes by theme, and the strength of evidence. Section 4 discusses the findings, Section 5 identifies gaps and future directions, and Section 6 concludes.

## 2. METHODS

### 2.1 Search Strategy

We performed a comprehensive search across more than 220 million academic papers from the Semantic Scholar and OpenAlex databases. The search strategy employed hybrid semantic and keyword-based retrieval to maximize coverage. Search queries included:

- “XGBoost algorithm applications and performance in machine learning”
- “Extreme gradient boosting model benchmarking accuracy feature importance”
- “XGBoost classification regression studies with comparative evaluation”
- “Gradient boosting trees XGBoost variants optimization and interpretability”
- “Systematic review of XGBoost in predictive modeling and data mining”

### 2.2 Study Selection

Initial database searching identified 221 records. After duplicate removal and relevance-based filtering, 114 records were screened against the eligibility criteria. Of these, 94 papers were excluded, resulting in 20 papers included in the final synthesis (Fig. 1). The eligibility criteria were:

- Uses XGBoost:** Does the study explicitly use XGBoost, extreme gradient boosting, or XGBoost-based modeling as a core method?
- Empirical Study:** Does the paper report an empirical study with real data, experiments, or benchmark evaluation rather than only theory or commentary?
- Predictive Task:** Does the study apply XGBoost to a predictive task such as classification, regression, forecasting, ranking, or anomaly detection?
- Performance Reported:** Does the paper report at least one evaluation metric such as accuracy, AUC, F1, RMSE, MAE, precision, recall, or  $R^2$ ?
- Baseline Comparison:** Does the study compare XGBoost against at least one other model, algorithm, or baseline method?

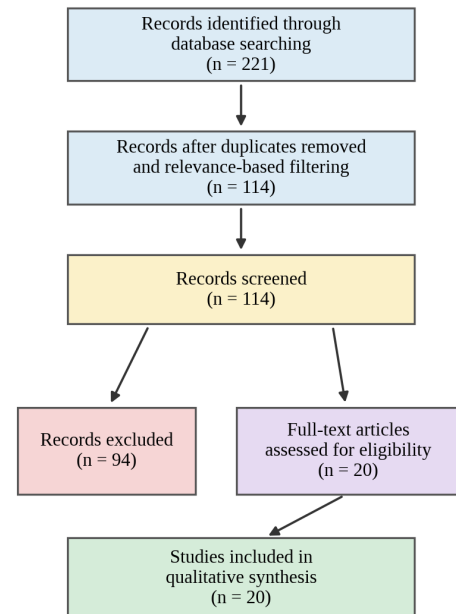


Fig. 1. PRISMA flow diagram for the study of the XGBoost algorithm.

- Application Breadth:** Does the study present a clearly described application domain or dataset rather than a purely methodological note?
- Interpretability or Tuning:** Does the paper discuss feature importance, hyperparameter tuning, optimization, or model interpretability for XGBoost?

All included studies satisfy the stated eligibility criteria.

### 2.3 Data Extraction and Synthesis

Data extraction focused on the following variables:

- Application:** the main application domain or task where XGBoost is used.
- Problem Type:** whether the paper addresses classification, regression, ranking, forecasting, anomaly detection, or another predictive task.
- Data Source:** the dataset(s), feature type(s), or input data modality used with XGBoost.
- Model Setup:** the XGBoost configuration or pipeline details if reported, including preprocessing, tuning, feature selection, or ensemble setup.
- Performance:** the primary evaluation metrics and the reported results, using only exact values stated in the paper.
- Comparison:** the baseline methods or competing models XGBoost was compared against and the nature of the comparison.
- Key Finding:** the paper’s main conclusion about XGBoost’s effectiveness, strengths, limitations, or suitability.

Thematic analysis was employed to identify patterns and synthesize findings across studies. Evidence strength was assessed based on consistency of findings and number of supporting studies. Quantities that appear in the Results (counts, shares, and the figures) are either values reported verbatim in the included studies or simple

tallies and ratios computed from them; where a value is derived, this is stated.

### 3. RESULTS

This section reports the results of the review in three parts: the characteristics of the included studies (Section 3.1), the quantitative findings organized by theme (Section 3.2), and a cross-study evaluation of the strength of the evidence (Section 3.3).

#### 3.1 Characteristics of the Included Studies

Twenty studies met all eligibility criteria (Fig. 1); they are summarized in Table 3.1. The studies were published between 2016 [1] and 2025 [10]. Four (20%) appeared in 2016–2019 and sixteen (80%) in 2020–2025, with 2022 (four studies) and 2024 (five studies) together contributing nine of the twenty (45%) (Fig. 2a). The evidence base is therefore dominated by recent work. Classification was addressed in 14 of the 20 studies (70%) and regression in 11 (55%); several studies address both. Ranking appears only in the GPU-acceleration study [2], and survival regression only in the accelerated-failure-time extension [7] (Fig. 2b). Fourteen studies are methodological, system, or benchmarking papers, while six are domain applications: mining [5, 17], education [18], real estate [6], seismic early warning [16], and concrete strength estimation [19] (Fig. 2c). Optimization is a dominant concern. Ten of the 20 studies (50%) center on tuning or optimization [3, 12, 13, 20, 5, 6, 4, 14, 9, 17], which supports treating XGBoost less as a fixed algorithm than as a configurable modeling framework. The explicit strategies were genetic algorithms (GA) [20, 5], Bayesian optimization [3, 14], particle swarm optimization (PSO) [6], the Runge–Kutta-based RUN optimizer [17], randomized search [14], grid search [13], and meta-learning [4] (Fig. 3a). The most frequent comparators were other gradient-boosting implementations, notably CatBoost (four studies) and LightGBM (three studies), followed by random forests, linear or logistic models, SVM/SVR, and AdaBoost (Fig. 3b).

#### 3.2 Key Findings and Quantitative Results

Table 3.2 lists the quantitative and comparative outcomes extracted from the included studies; the subsections below interpret them theme by theme.

**3.2.1 XGBoost is consistently strong, but rarely universally dominant.** Research consistently highlights XGBoost as a top-tier benchmark for predicting structured data, though its superiority depends on specific circumstances rather than being universal. Although XGBoost frequently surpasses conventional techniques such as random forests, support vector machines, and standard linear models in various applications, comparative evaluations indicate that alternatives such as CatBoost or finely tuned gradient-boosting frameworks can deliver equivalent or superior performance depending on the particular dataset and evaluation criteria [13, 14, 11]. Among the studies reporting an explicit head-to-head outcome (Table 3.2), an XGBoost-based model ranked first in three [18, 17, 6], whereas a competing learner matched or exceeded XGBoost in four [13, 14, 11, 19]. In education, XGBoost improved student-outcome forecasting relative to the standard PFA algorithm and outperformed Random Forest and AdaBoost [18]. In contrast, in estimating concrete strength, XGBoost ranked second with an MAE of about 3 N/mm<sup>2</sup>, trailing CatBoost, which achieved 2.72 N/mm<sup>2</sup> [19] (Fig. 4a); CatBoost's error was therefore roughly 9% lower (derived: 0.28/3). Such results indicate that, while XGBoost

serves as a robust versatile predictor, the optimal algorithm is contingent on the alignment between the model architecture, data characteristics, and optimization strategy. **Confidence:** Strong, because the direction of evidence is consistent across diverse tasks, although the ranking among booster variants varies by context.

**3.2.2 Hyperparameter optimization is central to XGBoost's performance.** A primary finding across various studies is that the predictive effectiveness of XGBoost is strongly influenced by hyperparameter selection. Investigations into genetic algorithms, particle swarm optimization, Bayesian methods, randomized search, and meta-learning have regularly shown performance improvements over default settings [5, 6, 4, 14] (Fig. 3a), and the contrast between tuned and default settings is itself the subject of dedicated studies [12, 9]. Specifically, GA-XGBoost achieved an R of 0.941, an RMSE of 0.369, and an MAE of 0.308 in predicting mining subsidence [5] (Fig. 5b), and a GA-optimized XGBoost was also reported on the UCI Liver Disorders data set [20]. In house-price estimation, PSO-XGBoost proved to be the most accurate among the five algorithms tested [6]. Moreover, meta-learning hyperparameter recommendations enhanced XGBoost's accuracy in 78.28% of the 198 datasets compared to standard grid-search baselines [4] (Fig. 5a), i.e., roughly 155 datasets (derived). The remaining 21.72% (roughly 43 datasets) did not show an improvement, so tuning gains, while common, are not guaranteed on every dataset. The benchmark results also indicate that the optimization strategy materially affects both processing time and predictive standing, with Bayesian optimization and randomized search offering distinct efficiency–performance balances [14, 13, 3]. Verma [9] examined search spaces for `min_child_weight`, `n_estimators`, `max_depth`, and `learning_rate` across six datasets. Consequently, XGBoost is more accurately characterized as a flexible modeling framework that requires careful adjustment to be successful. **Confidence:** Strong, because multiple optimization strategies independently point to the same conclusion.

**3.2.3 Efficiency and scalability are defining strengths, especially with GPU support.** Another consistent finding concerns computational efficiency on large datasets. The foundational research highlighted techniques such as sparsity-aware algorithms and weighted quantile sketching, which enable XGBoost to process billions of records with minimal computational overhead [1]. GPU acceleration further improved these capabilities: performance increases of 3 to 6 times were observed compared with 4-core i7 CPUs and 1.2 times over 24-core Xeon processors [2], while another benchmarking study reported an average GPU speedup of 7.26 times [3] (Fig. 6). The improvements were evident on massive datasets such as the Higgs collection, which contains 10 million entries [2]. Efficiency has also been pursued algorithmically, for example through a five-stage modular XGBoost compared against the standard implementation [10]. Nevertheless, these efficiency gains come with caveats, such as memory constraints on extremely large data and the fact that faster training does not always guarantee superior model generalization [3]. Consequently, while XGBoost offers significant scalability, its real-world performance remains bounded by hardware capacity and data volume. **Confidence:** Strong, given the convergence between foundational and benchmark-oriented studies.

**3.2.4 XGBoost is useful for nonlinear regression and forecasting across applied domains.** Regression-focused studies show that XGBoost is particularly effective when the target relationship is nonlinear and tabular features interact in complex ways. This is visible in house-price prediction [6], concrete compressive-strength

Table 1. Summary of surveyed papers. Abbreviations: AFT, accelerated failure time; ANN, artificial neural network; CV, cross-validation; GBDT, gradient-boosted decision trees; PFA, performance factors analysis; PPV, peak particle velocity; UCI, UCI Machine Learning Repository.

| Ref. | Study                    | Key focus                                                  | Problem type                          | Data source / model setup                                                        | Comparative context                                   |
|------|--------------------------|------------------------------------------------------------|---------------------------------------|----------------------------------------------------------------------------------|-------------------------------------------------------|
| [1]  | Chen & Guestrin, 2016    | Scalable tree boosting system                              | Regression / classification           | Not specified; sparsity-aware algorithm, weighted quantile sketch                | Existing systems                                      |
| [2]  | Mitchell & Frank, 2017   | GPU acceleration                                           | Classification / regression / ranking | Higgs dataset (large-scale); CUDA-based GPU tree construction                    | CPUs                                                  |
| [3]  | Anghel et al., 2018      | Benchmarking and optimization                              | Binary / multi-class classification   | Higgs, Epsilon, Microsoft, Yahoo; Bayesian optimization, GPU acceleration        | LightGBM, CatBoost                                    |
| [12] | Bentéjac et al., 2019    | Comparative analysis of XGBoost                            | Classification / regression           | Not detailed; tuned vs default settings                                          | Random forests, gradient boosting                     |
| [13] | Bentéjac et al., 2020    | Comparative analysis of gradient boosting algorithms       | Classification                        | 28 UCI datasets; grid search, stratified 10-fold CV                              | LightGBM, CatBoost, random forests                    |
| [20] | Chen et al., 2020        | Genetic-algorithm optimization                             | Classification                        | UCI Liver Disorders data set; GA-optimized parameters                            | Existing approaches                                   |
| [8]  | Sagi & Rokach, 2021      | Interpretability via tree approximation                    | Classification                        | Not specified; transforming forests into interpretable trees                     | Decision forests, GBDT                                |
| [18] | Asselman et al., 2021    | Student performance prediction                             | Classification                        | Three datasets; ensemble of RF, AdaBoost, XGBoost                                | Original PFA, RF, AdaBoost                            |
| [11] | Lai et al., 2021         | Imbalanced classification                                  | Classification                        | Simulated and real imbalanced datasets; setup not specified                      | AdaBoost, logistic regression                         |
| [7]  | Barnwal et al., 2022     | Survival regression                                        | Survival regression                   | Real and simulated datasets; AFT loss in XGBoost, GPU support                    | Survival-modeling baselines                           |
| [15] | Dong et al., 2022        | Regression boosting with shallow neural nets               | Regression                            | Several datasets; NNBoost framework                                              | XGBoost, GBDT                                         |
| [6]  | Sheng & Yu, 2022         | House price prediction                                     | Regression                            | Ames, Iowa housing data; PSO-optimized XGBoost                                   | Four other ML algorithms                              |
| [5]  | Gu et al., 2022          | Mining subsidence prediction                               | Regression                            | Domestic mining subsidence datasets; GA-XGBoost                                  | XGBoost, random deep forest, gradient boost           |
| [14] | Florek & Zagdański, 2023 | Classification benchmarking                                | Binary / multi-class classification   | Public real-world datasets; randomized search, Bayesian optimization             | GBM, LightGBM, CatBoost                               |
| [4]  | Marinho et al., 2024     | Hyperparameter recommendation                              | Regression / classification           | 198 datasets; meta-learning for hyperparameters                                  | Standard grid search                                  |
| [16] | Li et al., 2024          | Earthquake intensity threshold classification              | Classification                        | 4,353 acceleration records; 5 P-wave-derived features; XGBoost-ITD               | SVM and traditional methods                           |
| [17] | Sun et al., 2024         | Peak particle velocity (PPV) prediction in open-pit mining | Regression                            | 188 blasting operation records; RUN-XGBoost                                      | Sadovsky formula, PSO-XGBoost, Ridge, LASSO, SVM, SVR |
| [9]  | Verma, 2024              | Hyperparameter search spaces                               | Regression / classification           | Six datasets; min_child_weight, n_estimators, max_depth, learning_rate           | Not explicitly compared                               |
| [19] | Agaji & Igah, 2024       | Concrete strength estimation                               | Regression                            | Not specified; linear regression, decision tree, XGBoost, CatBoost, ANN compared | CatBoost and other models                             |
| [10] | Li et al., 2025          | Multi-stage XGBoost efficiency                             | Classification / regression           | Multiple datasets; five-stage modular XGBoost                                    | Standard XGBoost                                      |

estimation [19], mining subsidence prediction [5], and peak particle velocity (PPV) prediction [17]. The PPV study, based on 188 blasting operation records, reported that RUN-XGBoost outperformed the empirical Sadovsky formula, traditional XGBoost, PSO-XGBoost, Ridge, LASSO, SVM, and SVR, demonstrating

that optimization-enhanced XGBoost can capture multivariate blast dynamics more effectively than linear or single-learner alternatives [17]. Similarly, GA-XGBoost produced strong subsidence prediction values [5], and PSO-XGBoost was the best house-price predictor among the methods tested [6]. The regression evidence also ex-

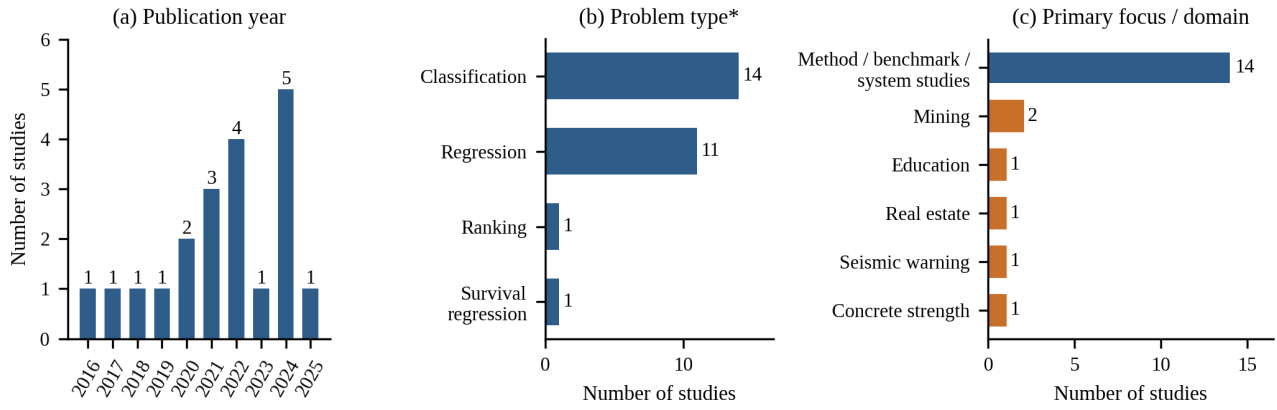


Fig. 2. Characteristics of the 20 included studies: (a) publication year; (b) problem type (\*studies may address more than one problem type, so counts sum to more than 20); (c) primary focus or application domain.

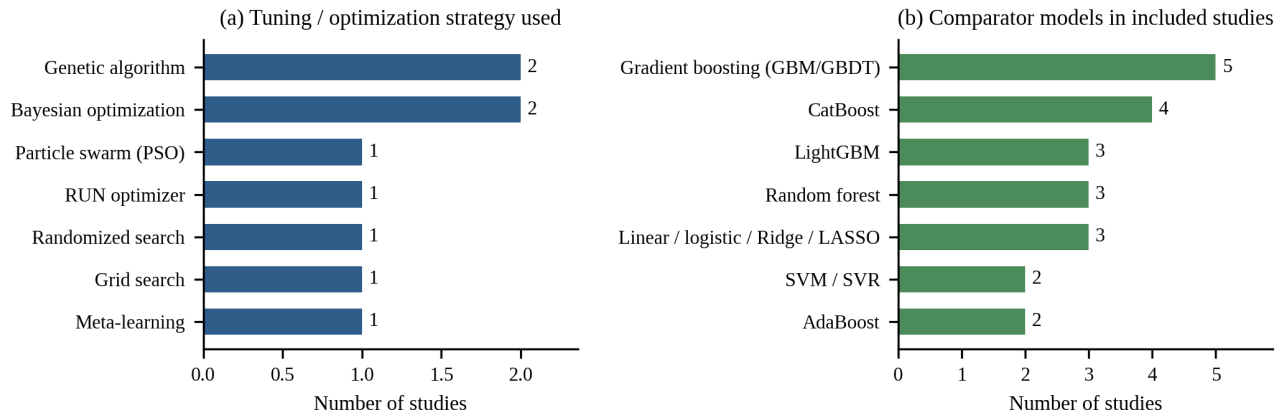


Fig. 3. (a) Explicit tuning or optimization strategies used in the included studies; (b) comparator models that appear in the included studies (a study may use several comparators).

tends beyond conventional squared-error settings: an accelerated-failure-time loss with GPU support enables survival regression in XGBoost [7], and a neural-network boosting variant (NNBoost) has been proposed and compared against XGBoost and GBDT [15]. This cluster of findings suggests that XGBoost is especially valuable for engineering and economic forecasting where nonlinear interactions dominate and exact functional forms are unknown. **Confidence:** Moderate to strong, because the regression evidence is consistent but concentrated in specific applied domains.

**3.2.5 XGBoost in classification: reliable performance, yet sensitive to class imbalance.** Research on classification presents a more nuanced view. XGBoost demonstrated robust performance in predicting student outcomes [18], identifying seismic intensity thresholds [16], and on standard classification benchmarks [14]. In seismic early warning, using five P-wave-derived features from 4,353 acceleration records, the model reached 93% accuracy for low-intensity events and 88% for high-intensity events [16] (Fig. 4b). However, its effectiveness is less consistent on imbalanced datasets; logistic regression was shown to exceed XGBoost on highly skewed simulated data, with performance rankings vary-

ing with the minority-class ratio and data source [11]. Furthermore, comparative studies indicate that optimized gradient boosting or CatBoost can marginally exceed XGBoost's results in certain real-world scenarios [13, 14]. These findings suggest that, while XGBoost is a formidable tool for classification, its edge diminishes in the presence of extreme class imbalance or when compared with well-calibrated competitors. **Confidence:** Moderate, because the classification evidence is strong overall but heterogeneous across data balance and benchmark conditions.

**3.2.6 The interpretability challenge: making sense of the black box.** XGBoost is often described as a "black box": it produces high-quality predictions, but understanding its internal logic remains difficult. This lack of transparency is particularly problematic in sensitive fields such as healthcare and finance, where knowing the reasoning behind a decision is critical [8]. The literature suggests that XGBoost is not inherently transparent; rather, researchers achieve transparency through outside tools, such as approximating its complex forests with simpler decision trees [8] or using SHAP to explain specific outcomes in studies such as earthquake detection [16] and blast-vibration prediction [17]. Inter-

Table 2. Quantitative and comparative outcomes reported in the included studies.

| Theme          | Outcome                                          | Reported result                                                                                                           | Study      | Context                                  |
|----------------|--------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|------------|------------------------------------------|
| Scalability    | GPU speedup over 4-core i7 CPU                   | $3 \times$ to $6 \times$                                                                                                  | [2]        | Higgs dataset (10 million entries)       |
| Scalability    | GPU speedup over 24-core Xeon CPU                | $1.2 \times$                                                                                                              | [2]        | Higgs dataset                            |
| Scalability    | Average GPU speedup in benchmarking              | $7.26 \times$                                                                                                             | [3]        | Higgs, Epsilon, Microsoft, Yahoo         |
| Tuning         | Datasets on which meta-learning beat grid search | 78.28% ( $\approx 155$ of 198 datasets; derived)                                                                          | [4]        | 198 regression / classification datasets |
| Tuning         | GA-XGBoost, mining subsidence                    | $R = 0.941$ ; $RMSE = 0.369$ ; $MAE = 0.308$                                                                              | [5]        | Domestic mining subsidence data          |
| Tuning         | PSO-XGBoost, house price                         | Most accurate of five algorithms tested                                                                                   | [6]        | Ames, Iowa housing data                  |
| Regression     | RUN-XGBoost, PPV prediction                      | Outperformed Sadosky formula, XGBoost, PSO-XGBoost, Ridge, LASSO, SVM, SVR                                                | [17]       | 188 blasting operation records           |
| Classification | Seismic intensity, low-intensity events          | 93% accuracy                                                                                                              | [16]       | 4,353 records; 5 P-wave features         |
| Classification | Seismic intensity, high-intensity events         | 88% accuracy                                                                                                              | [16]       | 4,353 records; 5 P-wave features         |
| Classification | Student performance prediction                   | Improved on PFA; exceeded RF and AdaBoost                                                                                 | [18]       | Three datasets                           |
| Comparative    | Concrete strength, MAE                           | XGBoost $\approx 3$ N/mm <sup>2</sup> vs CatBoost 2.72 N/mm <sup>2</sup> (CatBoost best)                                  | [19]       | Five algorithms compared                 |
| Imbalance      | Logistic regression vs XGBoost, skewed data      | Logistic regression exceeded XGBoost on highly skewed simulated data; rankings varied with minority ratio and data source | [11]       | Simulated and real datasets              |
| Benchmark      | CatBoost / LightGBM vs XGBoost                   | CatBoost sometimes slightly better in accuracy and AUC; LightGBM usually faster but not consistently more accurate        | [13], [14] | 28 UCI datasets; public datasets         |

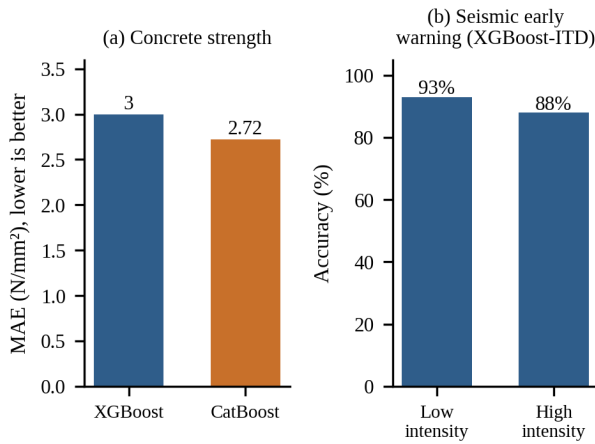


Fig. 4. Selected comparative and classification results: (a) mean absolute error in concrete-strength estimation, XGBoost vs CatBoost; (b) accuracy of XGBoost-ITD in seismic early warning.

pretability is therefore not part of the algorithm itself but is something that must be added. In sectors where clear explanations are non-negotiable, XGBoost either needs these extra layers of support or may be replaced by simpler, more transparent models. **Confidence:** Moderate, as the findings on interpretability are consistent across the literature but vary in their approach.

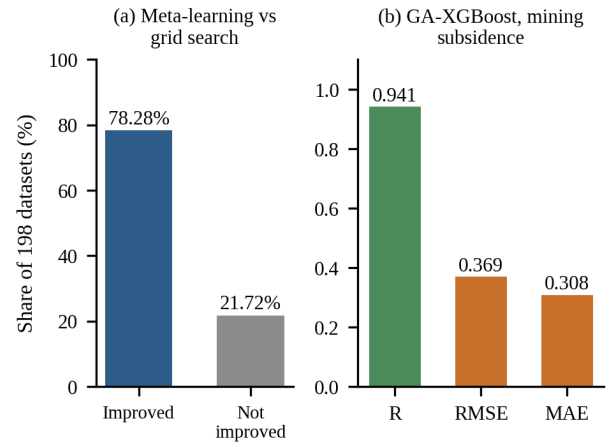


Fig. 5. Tuning results: (a) share of 198 datasets on which meta-learning-based hyperparameter recommendation did or did not improve XGBoost accuracy relative to grid search; (b) metrics reported for GA-XGBoost in mining subsidence prediction.

### 3.3 Summary of Key Findings and Strength of Evidence

Table 3.3 consolidates the six themes with their effect direction, confidence rating, and supporting studies, and Fig. 7 visualizes the confidence ratings. Evidence is strongest for scalability, hyperpa-

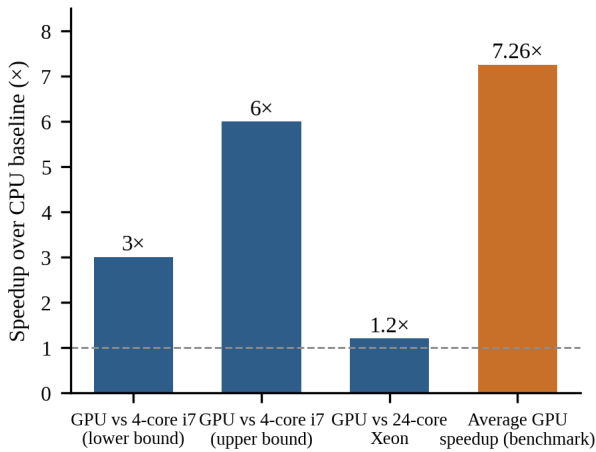


Fig. 6. Reported GPU speedups for XGBoost. The 3 $\times$  and 6 $\times$  bars are the lower and upper ends of the range reported against a 4-core i7 CPU; the dashed line marks parity with the CPU baseline.

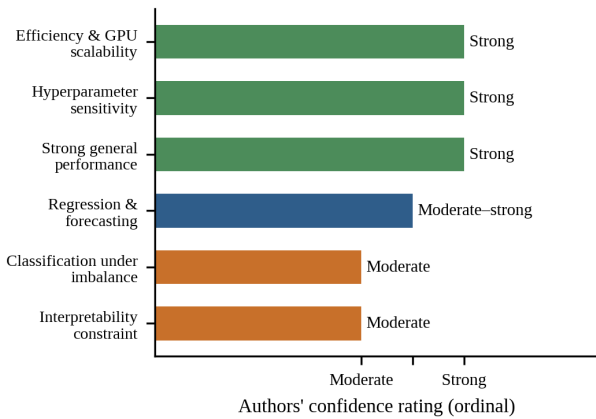


Fig. 7. Confidence ratings assigned to the six evidence themes (ordinal encoding of the ratings in Table 3.3: Moderate = 2, Moderate to strong = 2.5, Strong = 3).

parameter sensitivity, and general predictive performance (three supporting studies each, rated Strong), moderate-to-strong for regression and forecasting utility, and moderate for classification under imbalance and for interpretability. The confidence ratings are the review's qualitative judgments, based on consistency of findings and number of supporting studies; no pooled effect estimates were computed.

## 4. DISCUSSION

### 4.1 Primary Findings and Their Interpretation

The synthesis shows that XGBoost's core value lies in its combination of strong predictive capacity and engineering flexibility. Its performance is not simply the result of boosting trees, but of a design that supports sparse data handling, scalable training, and configurable regularization [1]. The consistency of gains across heterogeneous applied domains suggests that XGBoost is especially well suited to tabular data where interactions are nonlin-

ear but still learnable from finite features. The strongest pattern is that performance improves substantially when optimization is treated as part of the modeling pipeline rather than an afterthought. Meta-learning, genetic algorithms, particle swarm optimization, and Bayesian search all indicate that XGBoost's effective hypothesis space is highly sensitive to parameterization [4, 5, 14]. A second interpretive insight is that XGBoost appears most robust when the task structure matches boosted trees' strengths: heterogeneous predictors, moderate-to-large sample sizes, and nonlinear response surfaces. This likely explains why it performs well in mining, educational prediction, real-estate, and engineering regression, where interactions among features matter and parametric linearity is unrealistic [18, 17, 6]. In contrast, severe class imbalance weakens its relative advantage because the model can still prioritize the majority class unless the loss structure and tuning are adapted [11]. The literature therefore suggests not a universal superiority claim, but a conditional one: XGBoost is highly effective when properly tuned and when the data structure aligns with boosted-tree assumptions. The evidence base is strongest for performance and scalability, moderately strong for regression and classification utility, and weaker for interpretability (Fig. 7). That hierarchy matters because it shows where confidence should be placed: the computational and predictive benefits are well supported, while transparency remains an external constraint rather than an intrinsic property of the algorithm.

### 4.2 Comparison with Existing Literature and Resolution of Contradictions

The comparisons across studies indicate that XGBoost sits in a competitive middle ground rather than occupying a permanent first place. In benchmark analyses, CatBoost sometimes achieved slightly better generalization accuracy and AUC, while LightGBM was usually faster but not consistently more accurate [13, 14]. This is not a contradiction so much as a reflection of differing inductive biases. CatBoost's gradient handling may reduce prediction shift in categorical-heavy or benchmark-like settings, whereas LightGBM's efficiency-oriented design favors speed. XGBoost remains highly competitive because it balances both accuracy and runtime, but the reviewed evidence shows that no single gradient-boosting framework dominates every dataset. The most substantive contradiction concerns imbalanced classification. One study found logistic regression superior to XGBoost under severe imbalance, whereas other classification studies reported strong XGBoost performance in more balanced or less pathological settings [11, 14, 16]. The likely explanation is that imbalance changes what "good performance" means: accuracy can be misleading, and even F1 or the geometric mean (G-mean) may depend on the minority ratio and sample size. XGBoost's flexible tree structure may not automatically compensate for rare-class sparsity, especially when defaults are used, while a simpler linear model can sometimes provide more stable decision boundaries in extreme imbalance. Another plausible explanation is dataset scale and simulation design, because the imbalanced-data study explicitly distinguished simulated and real datasets and observed different rankings across them [11]. This suggests that reported superiority may be contingent on class prevalence, sample size, and the representativeness of the benchmark. Methodologically, the field has evolved toward better optimization and hardware acceleration. Newer studies incorporate meta-learning, Bayesian tuning, and GPU support, which likely reduce the gap between XGBoost and its competitors by allowing more careful calibration [4, 3]. That progression implies that earlier default-parameter comparisons may understate XGBoost's po-

Table 3. Evidence of XGBoost performance by theme.

| Theme                                      | Key finding                                                                                                                                                                                     | Population / applicability                                                          | Effect direction                        | Confidence         | Supporting studies                                           |
|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-----------------------------------------|--------------------|--------------------------------------------------------------|
| Classification performance under imbalance | Logistic regression outperformed XGBoost on highly imbalanced simulated data, while performance was similar in less extreme settings [11].                                                      | Imbalanced classification; simulated and real datasets                              | Mixed / negative under severe imbalance | Moderate           | Lai et al. [11]; Florek & Zagdański [14]; Li et al. [16]     |
| Regression and forecasting utility         | RUN-XGBoost outperformed the Sadosky formula and several ML baselines in PPV prediction; PSO-XGBoost achieved the highest house-price prediction accuracy [17], [6].                            | Engineering, mining, and real-estate regression                                     | Positive                                | Moderate to strong | Sheng & Yu [6]; Sun et al. [17]; Gu et al. [5]               |
| Scalability and GPU acceleration           | GPU XGBoost produced $3\times$ to $6\times$ speedups over a 4-core i7 CPU and $1.2\times$ over 24-core Xeon CPUs [2]; average GPU speedup $7.26\times$ [3].                                     | Large-scale, computation-intensive structured-data settings                         | Positive                                | Strong             | Mitchell & Frank [2]; Anghel et al. [3]; Chen & Guestrin [1] |
| Hyperparameter sensitivity                 | Meta-learning improved XGBoost in 78.28% of datasets [4]; GA-XGBoost reached $R = 0.941$ , $RMSE = 0.369$ , $MAE = 0.308$ [5].                                                                  | General ML datasets and domain-specific regression / classification tasks           | Positive                                | Strong             | Marinho et al. [4]; Gu et al. [5]; Verma [9]                 |
| Strong general predictive performance      | XGBoost often outperforms classical baselines, but CatBoost or tuned boosting can be superior; $MAE \approx 3$ vs $2.72$ N/mm <sup>2</sup> for XGBoost vs CatBoost in concrete prediction [19]. | Structured-data tasks across applied domains; some findings from benchmark datasets | Mixed-positive                          | Strong             | Chen & Guestrin [1]; Bentéjac et al. [13]; Agaji & Igah [19] |
| Interpretability constraints               | XGBoost remains a black-box model, prompting surrogate-tree approximation for transparent decision-making [8].                                                                                  | High-stakes decision contexts such as healthcare and finance                        | Negative for interpretability           | Moderate           | Sagi & Rokach [8]; Li et al. [16]                            |

tential, but it also reinforces that its practical success depends on a substantial amount of expert or automated tuning.

### 4.3 Practical Implications

For practitioners, the central implication is that XGBoost should be treated as a highly capable starting point for structured prediction, not as a plug-and-play guarantee of optimal performance. In engineering and forecasting settings such as mining subsidence, blast-vibration, house-price prediction, and concrete strength estimation, XGBoost and its optimized variants are particularly attractive because they capture nonlinearities without requiring deep-model complexity [5, 17, 6, 19]. For these applications, systematic tuning is not optional; it is part of the method's value proposition. In clinical or high-stakes settings, the interpretability limitation becomes more consequential. XGBoost can be deployed where predictive accuracy is essential, but decisions affecting patients or financial clients may require surrogate trees or explainability tools to make predictions actionable [8]. For severely imbalanced classification problems, users should not assume XGBoost will outperform simpler methods, particularly when minority events are rare and evaluation emphasizes recall-oriented metrics [11]. In such cases, model choice should be driven by calibration to class prevalence, not by algorithm popularity alone. From a systems perspective, the algorithm's GPU compatibility and scalability make it suitable for large datasets and time-sensitive pipelines [2, 3]. That said, hardware acceleration does not remove the need for careful tuning or validate model generalization on its own. The literature supports broad use of XGBoost in computationally demanding settings, but the decision to adopt it should remain conditional on interpretability needs, class balance, and tuning capacity.

### 4.4 Strengths and Limitations

A major strength of this review is the integration of methodological, computational, and application-oriented evidence, allowing XGBoost to be evaluated as both an algorithm and a deployed modeling strategy. The literature spans foundational system design, GPU acceleration, benchmark comparisons, and domain-specific applications, which provides a broad basis for synthesis. Another strength is the emphasis on tuned versus default configurations, since this distinction is essential for understanding real-world performance. The included studies also have limitations. Many are benchmark comparisons rather than prospective applied evaluations, and several abstracts do not fully report dataset characteristics, preprocessing, or exact metric distributions. Some application studies rely on relatively narrow domains such as mining, earthquake prediction, or educational analytics, which limits transferability. In addition, a number of papers report only the best-performing metric or qualitative superiority rather than complete performance distributions, which constrains direct comparison. This review has limitations as well. The synthesis is based on reported abstracts and extracted data, so some methodological details are not available. No formal risk-of-bias assessment was performed, and the literature may overrepresent studies with positive results because XGBoost is frequently benchmarked in contexts where high performance is expected. Finally, because exact pooled estimates were not provided, the review cannot make quantitative summary claims beyond the reported metrics.

## 5. GAPS AND FUTURE DIRECTIONS

The literature points to several specific gaps. First, interpretability remains underdeveloped relative to performance optimization:

existing work often adds post hoc explanation or surrogate trees, but there is little direct evidence on how these tools affect decision quality in high-stakes settings [8, 16]. Second, the evidence on severe class imbalance is limited and inconsistent, suggesting a need for targeted studies that systematically vary minority prevalence, sample size, and evaluation metrics to clarify when XGBoost is genuinely competitive [11]. Third, many studies report only headline metrics without complete uncertainty estimates or calibration diagnostics, making it difficult to assess robustness. Future studies should also test whether the gains from meta-learning, genetic algorithms, and particle swarm optimization generalize across domains rather than only within individual datasets [4, 5, 6]. More work is needed on exact model behavior under resource constraints, especially memory-limited large-scale data. For the population represented in this literature, namely structured tabular predictive tasks, future research would benefit from harmonized benchmarks that jointly report accuracy, runtime, calibration, and interpretability so that XGBoost can be compared fairly with LightGBM, CatBoost, and simpler baselines.

## 6. CONCLUSION

The evidence supports XGBoost as a highly effective and widely applicable machine-learning algorithm for structured prediction, especially when the goal is to model nonlinear relationships in tabular data. Its strongest advantages are predictive flexibility, scalability, and compatibility with GPU acceleration, with reported speedups of  $3\times$  to  $6\times$  over a 4-core i7 CPU and  $1.2\times$  over 24-core Xeon CPUs [2], and an average GPU speedup of  $7.26\times$  in benchmarking work [3]. Its performance can be especially strong in tuned settings, as shown by GA-XGBoost with  $R = 0.941$ ,  $RMSE = 0.369$ , and  $MAE = 0.308$  in mining subsidence prediction [5], and by meta-learning that improved accuracy in 78.28% of datasets [4]. At the same time, the literature does not support viewing XGBoost as universally best. CatBoost and tuned gradient boosting can surpass it in some benchmark settings, and logistic regression may be preferable under severe class imbalance [13, 11, 14]. The evidence therefore favors a conditional conclusion: XGBoost is a strong default choice, but its success depends on dataset structure, tuning strategy, and the practical need for interpretability. The most important unresolved question is how to integrate high predictive accuracy with reliable transparency in high-stakes applications without sacrificing the algorithm's core performance advantages. Addressing that challenge would materially strengthen the case for broader, safer deployment of XGBoost across scientific, industrial, and policy-relevant settings.

## 7. REFERENCES

- [1] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. arXiv [Internet]. 2016. Available from: <https://doi.org/10.48550/arXiv.1603.02754>
- [2] Mitchell R, Frank E. Accelerating the XGBoost algorithm using GPU computing. PeerJ Computer Science [Internet]. 2017 Jul 24;3:e127. Available from: <https://doi.org/10.7717/peerj-cs.127>
- [3] Anghel A, Papandreou N, Parnell TA, Palma AD, Pozidis H. Benchmarking and Optimization of Gradient Boosting Decision Tree Algorithms. arXiv [Internet]. 2018. Available from: <https://doi.org/10.48550/arXiv.1809.04559>
- [4] Marinho TL, Nascimento DC, Pimentel BA. Optimization on selecting XGBoost hyperparameters using meta-learning. Expert Systems [Internet]. 2024;41. Available from: <https://doi.org/10.1111/exsy.13611>
- [5] Gu Z, Cao M, Wang C, Yu N, Qing H. Research on mining maximum subsidence prediction based on genetic algorithm combined with XGBoost model. Sustainability [Internet]. 2022 Aug 22;14(16):10421. Available from: <https://doi.org/10.3390/su141610421>
- [6] Sheng C, Yu H. An optimized prediction algorithm based on XGBoost. In: 2022 International Conference on Networking and Network Applications (NaNA) [Internet]. IEEE; 2022. p. 1–6. Available from: <https://doi.org/10.1109/nana56854.2022.00082>
- [7] Barnwal A, Cho H, Hocking T. Survival regression with accelerated failure time model in XGBoost. Journal of Computational and Graphical Statistics [Internet]. 2022 May 24;31(4):1292–1302. Available from: <https://doi.org/10.1080/10618600.2022.2067548>
- [8] Sagi O, Rokach L. Approximating XGBoost with an interpretable decision tree. Information Sciences [Internet]. 2021 Sep;572:522–542. Available from: <https://doi.org/10.1016/j.ins.2021.05.055>
- [9] Verma V. Exploring Key XGBoost Hyperparameters: A Study on Optimal Search Spaces and Practical Recommendations for Regression and Classification. International Journal of AI Research Education Scientific Methods [Internet]. 2024;12:3259–3266. Available from: <https://doi.org/10.56025/ijaresm.2024.1210243259>
- [10] Li H, Cao Q, Kuo CJ. Multi-Stage XGBoost for Efficient Tree-Based Gradient Boosting. 2025;368–373. Available from: <https://doi.org/10.1109/icipw68931.2025.11386108>
- [11] Lai SBS, Binti Md Shahri NHN, Mohamad MB, Rahman HABA, Rambli AB. Comparing the performance of AdaBoost, XGBoost, and logistic regression for imbalanced data. Mathematics and Statistics [Internet]. 2021 May;9(3):379–385. Available from: <https://doi.org/10.13189/ms.2021.090320>
- [12] Bentéjac C, Csörgő A, Martínez-Muñoz G. A Comparative Analysis of XGBoost. arXiv [Internet]. 2019. Available from: <https://doi.org/10.48550/arXiv.1911.01914>
- [13] Bentéjac C, Csörgő A, Martínez-Muñoz G. A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review [Internet]. 2020;54:1937–1967. Available from: <https://doi.org/10.1007/s10462-020-09896-5>
- [14] Florek P, Zagdański A. Benchmarking state-of-the-art gradient boosting algorithms for classification. arXiv [Internet]. 2023. Available from: <https://doi.org/10.48550/arXiv.2305.17094>
- [15] Dong J, Chen Y, Yao B, Zhang X, Zeng N. A neural network boosting regression model based on XGBoost. Applied Soft Computing [Internet]. 2022;125:109067. Available from: <https://doi.org/10.1016/j.asoc.2022.109067>
- [16] Li S, Chen X, Lu J, Ma Q, Xie Z, Tao D, et al. XGBoost. Earth Science–Journal of China University of Geosciences [Internet]. 2024;49:379. Available from: <https://doi.org/10.3799/dqkx.2023.159>
- [17] Sun M, Yang J, Yang C, Wang W, Wang X, Li H. Research on prediction of PPV in open-pit mine using RUN-XGBoost model. Heliyon [Internet]. 2024 Apr;10(7):e28246.

Available from: <https://doi.org/10.1016/j.heliyon.2024.e28246>

- [18] Asselman A, Khaldi M, Aammou S. Enhancing the prediction of student performance based on the machine learning XGBoost algorithm. *Interactive Learning Environments* [Internet]. 2021 May 19;31(6):3360–3379. Available from: <https://doi.org/10.1080/10494820.2021.1928235>
- [19] Agaji AA, Igah GC. Comparative analysis of linear regression, decision tree, XGBoost, CatBoost, and artificial neural network machine learning algorithms for [Internet]. Springer Science and Business Media LLC; 2024 Nov. Available from: <https://doi.org/10.21203/rs.3.rs-5442566/v1>
- [20] Chen J, Zhao F, Sun Y, Yin Y. Improved XGBoost model based on genetic algorithm. *International Journal of Computer Applications in Technology* [Internet]. 2020;62(3):240. Available from: <https://doi.org/10.1504/IJCAT.2020.106571>