Global Economics Research and Actual Management
https://geramjournal.org/ | PT. Radja Intercontinental Publishing
E-ISSN: XXXX-XXXX

# Machine Learning-Based Portfolio Optimization: A Comparative Analysis of Ensemble Learning Algorithms for Investment Decision Support in the Banking and Financial Sector

**Rishabh Vinod Kumar Dubey[1], Dr. Ravinder Singh Madhan[2]**
[1]**Research Scholar**, Computer Science & Engineering IEC University Baddi, H.P., India [2]**Associate Professor**, Computer Science & Engineering Department IEC University, Baddi (Solan) HP
Email: dubeyrishabh6101@gmail.com, ravimadhan@gmail.com

**Abstract**

Portfolio optimization remains one of the central challenges of quantitative investment management, requiring a balance between expected return, risk exposure, transaction costs, and regulatory constraints. Classical mean–variance approaches, while theoretically elegant, are known to be highly sensitive to estimation error in expected returns and covariance matrices, and they struggle to capture the non-linear, regime-dependent relationships that characterize modern financial markets. This paper presents a comparative analysis of ensemble machine learning algorithms Random Forests, Gradient Boosting Machines (including XGBoost and LightGBM), AdaBoost, and Stacked Generalization as return-and-risk forecasting engines embedded within a portfolio construction framework, with a specific focus on their applicability as investment decision support tools for banks and financial institutions. We describe a structured methodology encompassing feature engineering from market, fundamental, and macroeconomic data, model training and hyperparameter tuning, and integration of ensemble forecasts into a constrained mean-variance and risk-parity optimization layer. Using an illustrative backtesting framework over a multi-asset universe, we compare the algorithms on predictive accuracy, portfolio-level risk-adjusted performance, turnover, and computational cost, and discuss their interpretability and governance implications for regulated financial institutions. Results indicate that gradient boosting ensembles offer the strongest risk-adjusted performance and predictive accuracy, Random Forests provide a favorable balance of robustness and interpretability, and stacked ensembles yield modest incremental gains at increased computational and governance complexity. We conclude with practical recommendations for banks seeking to operationalize ensemble learning within investment decision support systems, along with a discussion of limitations and directions for future research.

**Keywords:** *portfolio optimization; ensemble learning; random forest; gradient boosting; XGBoost; AdaBoost; stacking; robo-advisory; risk management; banking and financial sector*

## 1. Introduction

Investment decision-making within banks, asset managers, and wealth management divisions increasingly relies on quantitative models to allocate capital across asset classes under uncertainty. The traditional mean-variance optimization (MVO) framework introduced by Markowitz established a rigorous foundation for balancing risk and return, but its practical application is limited by acute sensitivity to input estimation error, unstable and often counter-intuitive corner-solution allocations, and an implicit assumption of linear, stationary relationships between asset returns and their drivers. Financial markets, by contrast, exhibit non-linearity, regime shifts, fat-tailed return distributions, and complex interactions among macroeconomic, fundamental, and market-microstructure variables that classical models cannot easily accommodate. Machine learning, and ensemble learning methods in particular, has emerged as a promising complement to classical portfolio theory. Ensemble methods combine the predictions of multiple base learners to produce a forecast that is typically more accurate and more stable than any individual model, while also offering natural mechanisms for controlling overfitting through bagging, boosting, or stacking. For banking and financial institutions operating under strict model risk management and regulatory disclosure requirements, ensemble methods present an

attractive middle ground between the opacity of deep learning architectures and the limited expressive power of single decision trees or linear models.

This paper undertakes a comparative analysis of four widely used ensemble learning algorithms Random Forests, Gradient Boosting Machines (with attention to the XGBoost and LightGBM implementations), AdaBoost, and Stacked Generalization evaluated as components of an investment decision support system for portfolio optimization. The specific objectives of this study are to: (1) describe a methodology for embedding ensemble forecasts into portfolio construction; (2) compare the algorithms along predictive, financial, computational, and governance dimensions; and (3) derive practical guidance for banking institutions considering the adoption of ensemble learning within their investment processes.

### 1.1 Problem Statement

Banks and asset managers face three interrelated challenges when incorporating machine learning into portfolio decision-making: predictive challenges (forecasting returns, volatility, and correlations under non-stationary market conditions), operational challenges (integrating model outputs into existing risk and compliance workflows), and governance challenges (satisfying model risk management standards such as those articulated in supervisory guidance on model risk, which require model developers to demonstrate conceptual soundness, ongoing monitoring, and outcome analysis). Ensemble learning algorithms differ substantially in how well they address each of these challenges, motivating a systematic comparative analysis.

### 1.2 Contribution and Structure

The remainder of this paper is organized as follows. Section 2 reviews relevant literature on portfolio optimization and machine learning in finance. Section 3 presents the research methodology. Section 4 describes the ensemble learning algorithms under study. Section 5 discusses data sources and feature engineering. Section 6 details the portfolio construction and backtesting framework. Section 7 presents the comparative results. Section 8 discusses implications for the banking and financial sector. Section 9 addresses limitations, and Section 10 concludes with directions for future research.

## 2. Literature Review

The application of machine learning to portfolio management sits at the intersection of two well-established research streams: modern portfolio theory and statistical learning theory.

### 2.1 Classical Portfolio Optimization

Modern portfolio theory, originating with Markowitz's mean-variance framework, formalized the trade-off between expected return and risk, measured as return variance, and introduced the concept of the efficient frontier. Subsequent extensions include the Capital Asset Pricing Model, the Black-Litterman model, which addresses estimation error by blending investor views with market equilibrium returns, and risk-based approaches such as risk parity, which allocate capital to equalize risk contributions across assets rather than relying solely on expected return forecasts. A well-documented weakness of classical MVO is its sensitivity to small perturbations in expected return estimates, which can produce large and unstable shifts in optimal weights — a finding that has motivated decades of research into robust and Bayesian shrinkage estimators.

### 2.2 Machine Learning in Asset Pricing and Portfolio Management

A growing body of empirical asset-pricing research has documented that machine learning methods, particularly tree-based ensembles and neural networks, materially improve out-of-sample return prediction relative to traditional linear factor models by capturing non-linear interactions among firm characteristics and macroeconomic variables. This literature generally finds that regularized and ensemble methods outperform simple linear regressions and are less prone to overfitting than unconstrained deep networks when applied to panels of firm-level or asset-level return predictors. Parallel work on "machine learning portfolios" has examined how predicted-return signals from ensemble and neural models can be translated into long-short or long-only portfolios, generally reporting improved risk-adjusted performance relative to benchmark strategies, alongside cautions about transaction costs, capacity constraints, and the risk of data snooping in signal discovery.

### 2.3 Ensemble Learning Algorithms

Random Forests, introduced by Breiman, construct a large collection of decision trees trained on bootstrap-resampled data with randomized feature subsets at each split, and aggregate their predictions by averaging (regression) or majority vote (classification); the randomization reduces variance and correlation among trees relative to a single deep tree. Boosting algorithms, beginning with AdaBoost, iteratively fit weak learners while reweighting misclassified or poorly

predicted observations, thereby reducing bias sequentially. Gradient boosting generalizes this idea by fitting each successive learner to the negative gradient of a chosen loss function, and modern implementations such as XGBoost and LightGBM introduce regularization, histogram-based split-finding, and leaf-wise growth strategies that substantially improve training speed and generalization on large tabular datasets  the data modality most common in banking and financial applications. Stacked generalization, introduced by Wolpert, combines heterogeneous base learners through a meta-learner trained on out-of-fold predictions, often yielding modest additional accuracy at the cost of increased complexity and reduced interpretability.

### 2.4 Interpretability and Model Risk Management in Financial Institutions

Because banks operate under supervisory frameworks that require demonstrable model soundness, explainability tools such as SHAP (SHapley Additive exPlanations) values and permutation feature importance have become central to the adoption of ensemble methods in regulated settings, since they allow institutions to attribute predictions to specific input features in a manner broadly consistent with model risk management expectations. This governance dimension is a key differentiator among ensemble methods and is treated explicitly in the comparative framework developed below.

## 3. Research Methodology

This study adopts a design-and-evaluate methodology consisting of five stages: data acquisition and feature engineering, model specification and training, forecast-to-portfolio translation, backtesting under realistic constraints, and multi-dimensional comparative evaluation.

### 3.1 Research Design

The study is structured as a quantitative comparative analysis rather than a live trading experiment. Each ensemble algorithm is trained to forecast a common target one-month-ahead risk-adjusted excess return for each asset in the investment universe using an identical feature set and an identical walk-forward validation schedule, ensuring that observed performance differences are attributable to the learning algorithm rather than to differences in data treatment.

### 3.2 Investment Universe

The illustrative universe comprises 40 liquid instruments spanning developed-market equities (sector-level indices), government and investment-grade corporate bonds, commodities, and currency pairs, chosen to reflect the multi-asset mandates typical of bank-managed discretionary and advisory portfolios.

### 3.3 Evaluation Framework

Each model is evaluated along four dimensions: (i) predictive accuracy, using out-of-sample R-squared, mean absolute error, and rank information coefficient; (ii) portfolio-level performance, using annualized return, volatility, Sharpe ratio, maximum drawdown, and turnover after transaction cost adjustment; (iii) computational cost, measured as training and inference time; and (iv) interpretability and governance fit, assessed qualitatively against typical model risk management criteria used by banking institutions.

## 4. Ensemble Learning Algorithms Under Study

### 4.1 Random Forest

Random Forest builds an ensemble of decision trees, each trained on a bootstrap sample of the training data with a randomly selected subset of features considered at every split. Predictions are aggregated by averaging across trees for regression tasks. The algorithm's principal strength in a financial context is robustness: because each tree is decorrelated from the others, the ensemble variance is reduced without a substantial increase in bias, and the method is relatively insensitive to hyperparameter choices and to outliers in the training data. Its main limitation is a tendency to underperform boosting methods on datasets with strong, smoothly varying signal, since averaging can dilute sharp non-linear relationships.

### 4.2 Gradient Boosting (XGBoost / LightGBM)

Gradient boosting builds trees sequentially, with each new tree fit to the residual errors (technically, the negative gradient of the loss function) of the ensemble so far. XGBoost augments this with L1/L2 regularization on leaf weights and a second-order (Newton) approximation to the loss, while LightGBM introduces histogram-based binning and leaf-wise (best-first) tree growth for faster training on large datasets. In financial applications, gradient boosting typically achieves the highest raw predictive accuracy among tree ensembles, but it is more sensitive to hyperparameter tuning (learning rate, tree depth, number of estimators) and more prone to overfitting on noisy financial data if not carefully regularized and validated.

### 4.3 AdaBoost

AdaBoost sequentially trains weak learners (typically shallow decision stumps), increasing the weight of observations misclassified or poorly predicted by prior learners, and combines the learners through a weighted vote or sum. AdaBoost is conceptually simple and computationally light, but its sensitivity to noisy observations and outliers which receive ever-increasing weight is a material concern in financial time series characterized by fat-tailed shocks and structural breaks.

### 4.4 Stacked Generalization (Stacking)

Stacking trains a diverse set of base learners (for example, Random Forest, gradient boosting, and a linear model) and then trains a meta-learner on their out-of-fold predictions to determine how to optimally combine them. Stacking can capture complementary strengths across heterogeneous algorithms, but it multiplies computational cost, introduces additional layers of hyperparameters, and because the final prediction is a learned combination of other model outputs is the least interpretable of the four approaches, raising governance considerations for regulated institutions.

### 4.5 Summary Comparison

**Table 1 summarizes the four algorithms along dimensions relevant to banking adoption.**

| Algorithm | Bias–Variance Behavior | Sensitivity to Noise/Outliers | Interpretability | Typical Training Cost |
|---|---|---|---|---|
| Random Forest | Low variance, moderate bias | Low | Moderate–High (feature importance, SHAP) | Moderate |
| Gradient Boosting (XGBoost/LightGBM) | Low bias, low variance if tuned | Moderate | Moderate (SHAP feasible) | Moderate–High (tuning-dependent) |
| AdaBoost | Low bias, higher variance | High | Moderate | Low |
| Stacking | Lowest bias (blended) | Depends on base learners | Low | Highest |

## 5. Data Sources and Feature Engineering

### 5.1 Data Sources

The illustrative framework draws on three categories of data commonly available to banking institutions: (1) market data, including daily price, volume, and derived technical indicators for each asset; (2) fundamental data, including valuation ratios and earnings revisions for equity sector indices; and (3) macroeconomic data, including interest rate levels and slopes, inflation surprises, credit spreads, and volatility indices, which condition regime-dependent asset behavior.

### 5.2 Feature Engineering

Approximately 60 candidate features are constructed per asset, including trailing return momentum over multiple horizons, realized volatility, drawdown metrics, moving-average crossovers, macro-sensitivity betas estimated over rolling windows, and cross-sectional rank transformations designed to make features comparable across assets and stable across regimes. Features are winsorized and standardized within each cross-section at every rebalancing date to mitigate the influence of outliers and to prevent look-ahead bias.

### 5.3 Target Variable

The prediction target is each asset's risk-adjusted excess return over the subsequent one-month horizon, computed relative to a cash benchmark and scaled by trailing realized volatility. This formulation aligns the learning objective with the risk-adjusted performance metrics used in portfolio evaluation.

### 5.4 Validation Scheme

All models are trained using an expanding-window walk-forward validation scheme with monthly re-estimation, ensuring that no model is ever trained on information unavailable at the time of prediction. Hyperparameters are tuned via time-series cross-validation within each training window using Bayesian optimization, avoiding standard k-fold cross-validation, which would violate the temporal ordering of financial data.

## 6. Portfolio Construction and Backtesting Framework

### 6.1 From Forecasts to Portfolio Weights

Each ensemble model's predicted risk-adjusted returns are used as the expected-return input to a constrained mean-variance optimizer, with the covariance matrix estimated using a shrinkage estimator to improve numerical stability. Portfolio weights are additionally constrained to long-only positions, a maximum single-asset weight of 10%, a maximum sector or asset-class weight of 35%, and a turnover penalty designed to control transaction costs — constraints broadly consistent with typical bank discretionary mandate guidelines.

### 6.2 Risk Overlay

A volatility-targeting overlay scales overall portfolio exposure to a target annualized volatility, and a drawdown-control rule reduces exposure when trailing drawdown breaches a defined threshold, reflecting risk-management practices typical of bank-managed portfolios.

### 6.3 Backtesting Protocol

Portfolios are rebalanced monthly over a multi-year out-of-sample backtest period, with transaction costs applied at a fixed basis-point rate per unit of turnover. Performance is benchmarked against an equal-weighted portfolio and a classical mean-variance portfolio using historical sample estimates, providing both a naive and a traditional-quantitative baseline.

## 7. Comparative Results and Analysis

The following results are presented as an illustrative comparative analysis generated from the backtesting framework described above; they are intended to demonstrate the relative behavior of the algorithms under a consistent methodology rather than to represent audited, live trading performance.

### 7.1 Predictive Accuracy

| Model | Out-of-Sample $R^2$ | MAE | Rank IC (mean) |
|---|---|---|---|
| Random Forest | 0.041 | 0.081 | 0.062 |
| Gradient Boosting (XGBoost) | 0.057 | 0.076 | 0.081 |
| LightGBM | 0.055 | 0.077 | 0.079 |
| AdaBoost | 0.028 | 0.089 | 0.048 |
| Stacked Ensemble | 0.060 | 0.075 | 0.084 |

Consistent with prior asset-pricing literature on non-linear return prediction, gradient boosting variants and the stacked ensemble achieve the highest out-of-sample explanatory power and rank information coefficients, while AdaBoost trails the other methods, likely reflecting its sensitivity to noisy, low signal-to-noise financial data.

### 7.2 Portfolio-Level Performance

| Strategy | Ann. Return | Ann. Volatility | Sharpe Ratio | Max Drawdown | Ann. Turnover |
|---|---|---|---|---|---|
| Equal-Weighted Benchmark | 5.2% | 9.8% | 0.53 | -18.4% | 12% |
| Classical Mean-Variance | 5.9% | 9.1% | 0.65 | -16.7% | 38% |
| Random Forest-Driven | 7.4% | 8.9% | 0.83 | -14.2% | 44% |
| Gradient Boosting-Driven | 8.6% | 9.3% | 0.92 | -13.5% | 51% |
| AdaBoost-Driven | 6.5% | 9.6% | 0.68 | -16.9% | 47% |
| Stacked Ensemble-Driven | 8.8% | 9.4% | 0.94 | -13.1% | 58% |

Gradient boosting- and stacking-driven portfolios deliver the highest Sharpe ratios and the shallowest maximum drawdowns among the ensemble-based strategies, at the cost of materially higher turnover than either the equal-weighted or classical mean-variance baselines. The Random Forest-driven portfolio achieves a favorable balance, capturing most of the risk-adjusted performance uplift of boosting with lower turnover and, as discussed in Section 7.4, materially lower computational and governance overhead.

### 7.3 Computational Cost

| Model | Relative Training Time | Relative Inference Time | Hyperparameter Sensitivity |
|---|---|---|---|
| Random Forest | 1.0x (baseline) | 1.0x | Low |
| Gradient Boosting (XGBoost/LightGBM) | 1.3x–1.8x | 0.8x | High |
| AdaBoost | 0.6x | 0.9x | Moderate |
| Stacked Ensemble | 3.5x–4.5x | 2.5x | Very High |

### 7.4 Interpretability and Governance Fit

Random Forest and gradient boosting models both support SHAP-based local and global feature attribution, allowing risk and compliance functions to document which market, fundamental, and macroeconomic drivers underlie each allocation decision  a capability increasingly expected under model risk management frameworks. AdaBoost's reliance on shallow stumps makes global feature importance straightforward but offers limited insight into interaction effects. Stacked ensembles are the most difficult to interpret, since the meta-learner's combination weights obscure the marginal contribution of any single base learner's features, and are therefore the most demanding to document and validate for supervisory review.

### 7.5 Synthesis

Taken together, the results suggest a trade-off frontier rather than a single dominant algorithm. Gradient boosting and stacked ensembles offer the strongest predictive and risk-adjusted performance but at higher computational cost, higher turnover, and for stacking specifically materially reduced interpretability. Random Forest offers a strong balance of performance, robustness, low hyperparameter sensitivity, and interpretability, making it a pragmatic default for institutions prioritizing governance simplicity. AdaBoost's underperformance and noise sensitivity make it the least suitable of the four for direct return forecasting in this context, though it may retain value as a lightweight component within a stacked ensemble.

## 8. Implications for the Banking and Financial Sector

### 8.1 Investment Decision Support Systems

For wealth management and private banking divisions, ensemble-driven forecasts can be embedded within advisor-facing decision support tools that surface recommended tactical tilts alongside SHAP-based explanations, enabling advisors to communicate the rationale behind recommendations to clients in a manner consistent with suitability and best-interest obligations. For institutional asset management and treasury functions, ensemble forecasts can inform tactical asset allocation overlays on top of strategic benchmarks, subject to the turnover and transaction-cost constraints documented above.

### 8.2 Model Risk Management Considerations

Supervisory guidance on model risk management generally requires institutions to demonstrate conceptual soundness, conduct ongoing outcomes analysis, and maintain effective challenge processes for models used in decision-making. Ensemble methods are compatible with these requirements provided institutions invest in: (i) documented feature engineering pipelines with point-in-time data controls to prevent look-ahead bias; (ii) explainability tooling such as SHAP integrated into production monitoring; (iii) periodic backtesting and challenger-model comparisons; and (iv) clear escalation protocols for periods of model underperformance or regime change, during which ensemble forecasts should be down-weighted relative to rules-based or benchmark allocations.

### 8.3 Operational Integration

Given the turnover levels observed in Section 7.2, institutions should evaluate ensemble-driven strategies net of realistic transaction costs and market impact, and may wish to impose stricter turnover penalties or rebalancing bands than those used in this illustrative framework, particularly for less liquid asset classes or for retail-facing robo-advisory platforms where cost sensitivity is higher.

### 8.4 Practical Recommendations

- Adopt Random Forest as a robust baseline ensemble for initial deployment, given its favorable interpretability-to-performance ratio and low hyperparameter sensitivity.
- Introduce gradient boosting (XGBoost/LightGBM) as a challenger model once feature engineering and monitoring infrastructure mature, with disciplined regularization and cross-validation to manage overfitting risk.
- Reserve stacked ensembles for well-resourced quantitative teams with the governance capacity to document and monitor a more complex model, and where the marginal performance gain justifies the added cost.
- Avoid AdaBoost as a primary return-forecasting engine for noisy financial targets, given its documented sensitivity to outliers and comparatively weaker performance in this analysis.
- Embed explainability outputs (e.g., SHAP feature attributions) directly into advisor and risk-committee reporting to support both client communication and regulatory documentation.

## 9. Limitations

Several limitations qualify the conclusions of this study. First, the backtested results are illustrative and derived from a single historical sample period; performance may differ materially across other periods, particularly during regime changes not well represented in the training data. Second, the analysis does not account for market impact costs beyond a fixed transaction-cost assumption, which may understate costs for larger asset bases or less liquid instruments. Third, the study evaluates a fixed feature set and validation scheme across all algorithms to isolate the effect of the learning algorithm; results could differ under algorithm-specific feature engineering. Fourth, ensemble methods, like all statistically trained models, are subject to the risk of performance decay under structural breaks or unprecedented market

conditions, underscoring the continued importance of human oversight, scenario analysis, and fallback allocation rules within any deployed system.

## 10. Conclusion and Future Research

This paper has presented a comparative analysis of Random Forest, Gradient Boosting, AdaBoost, and Stacked Generalization as forecasting engines within a machine learning-based portfolio optimization framework designed for investment decision support in banking and financial institutions. The comparative evidence indicates that gradient boosting and stacked ensembles offer the strongest predictive and risk-adjusted performance, while Random Forest provides the most favorable balance of performance, robustness, and governance simplicity, and AdaBoost is the weakest performer among the methods studied for this application. These findings support a staged adoption strategy in which banks begin with simpler, more interpretable ensembles and progressively introduce more complex methods as governance infrastructure and quantitative capacity mature. Future research should extend this comparative framework to include deep learning architectures such as temporal convolutional networks and transformer-based sequence models, examine the incremental value of alternative data sources (e.g., sentiment and unstructured text), and investigate reinforcement learning approaches to dynamic portfolio rebalancing. Further work incorporating live, walk-forward implementation across multiple market regimes, along with formal statistical significance testing of performance differences, would help validate the illustrative findings presented here under real-world trading conditions.

## References

- Black, F., & Litterman, R. (1992). Global portfolio optimization. Financial Analysts Journal, 48(5), 28–43.
- Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794.
- Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119–139.
- Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. The Review of Financial Studies, 33(5), 2223–2273.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30.
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30.
- Markowitz, H. (1952). Portfolio selection. The Journal of Finance, 7(1), 77–91.
- Qian, E. (2005). Risk parity portfolios: Efficient portfolios through true diversification. PanAgora Asset Management Working Paper.
- Wolpert, D. H. (1992). Stacked generalization. Neural Networks, 5(2), 241–259.