Proceedings of CONF-MLA 2026 Symposium: Explainable Computing, Modeling & Data Science in Complex Systems
DOI: 10.54254/2755-2721/2026.GF34864

Machine Learning for Estimation: Comparing Tree Ensembles and Deep Learning on Tabular Data

Yaoping Wang

School of Arts and Sciences, Rutgers University–New Brunswick, New Brunswick, USA
13380284790@163.com

Abstract. Parametric models widely used in estimation often violate assumptions in practical research. As flexible alternatives, machine learning methods, especially tree ensembles and deep neural networks, impose fewer prior assumptions on functional forms for parameter estimation. This paper systematically examines eight estimation methods—ordinary least squares, ridge regression, lasso, random forest, XGBoost, LightGBM, multilayer perceptron (MLP), and deep neural networks—across four simulation regimes (linear, semiparametric, nonlinear, and high-dimensional sparse) and two real-world datasets: the Home Credit Default Risk dataset (307,511 samples) and the PIMA Indians Diabetes dataset (768 samples). This study evaluate model bias, mean squared error (MSE) and computational overhead to determine the applicable scenarios for each method category. Experimental results demonstrate that tree-based methods perform steadily across various scenarios. Although deep neural networks incur higher computational overhead, they achieve the minimal MSE when facing strong nonlinearity with moderate or large sample sizes. These findings provide actionable guidance for selecting estimation methods based on data characteristics, bridging theoretical advances in machine learning and practical estimation. The results verify that no single estimator outperforms all others across all data scenarios. The optimal selection relies on the joint effects of data nonlinearity, dimensionality and sample size, which highlights the necessity of diagnosis-oriented method selection in empirical studies.

Keywords: statistical inference, machine learning, parameter estimation, tree ensembles, deep learning

1. Introduction

Arameter estimation is a fundamental task in both natural and social sciences. Classical methods, notably ordinary least squares (OLS) and generalized linear models, offer consistency, normality, and exact inference under correct specification [1]. Nevertheless, these favorable properties will no longer hold once model assumptions are violated. Empirical data often contain nonlinearities, high-dimensional spaces, and complex interactions that parametric models cannot capture without extensive manual specification [2].

Machine learning (ML) approaches, including tree ensemble models (e.g., random forests (RF) and gradient boosting (GB)) and deep neural networks (DNN), can learn flexible functional forms

DOI: 10.54254/2755-2721/2026.GF34864

directly from raw data [3-5]. Nevertheless, their applicability to formal statistical inference still remains an open research problem [6, 7]. Existing comparisons focus on narrow method subsets or data regimes; few studies benchmark classical and ML estimators across controlled nonlinearity, dimensionality, and sample-size conditions within a unified framework [1, 2]. As a result, applied researchers lack clear guidelines for model selection. Three core research questions are addressed: whether ML estimators achieve comparable reliability to classical methods, what types of data scenarios can benefit from the flexibility of ML models, and how much computational overhead these methods introduce.

This work addresses these questions through a controlled simulation framework spanning four data-generating processes and two real-world datasets, with a practical discussion of computation-performance trade-offs. The findings help researchers judge whether the increased complexity of ML models is worthwhile for improving estimation accuracy.

2. Preliminaries and literature review

2.1. Classical estimation framework

The study first introduces the standard linear regression model as follows:

$$y = X\beta + \varepsilon \tag{1}$$

where $y \in \mathbb{R}^n$ is the response vector, $X \in \mathbb{R}^{n\times p}$ is the design matrix, $\beta \in \mathbb{R}^p$ is the coefficient vector, and $\varepsilon \sim N(0, \sigma^2 I)$ represents independent and identically distributed (i.i.d.) Gaussian noise. According to the Gauss-Markov theorem, the OLS estimator is derived by solving the following objective function:

$$\beta_{OLS} = (X^\top X)^{-1} X^\top y\# \tag{2}$$

which is the best linear unbiased estimator [8]. The lasso extends this framework by adding l_1 regularization for simultaneous estimation and variable selection in high-dimensional settings [9]. The Lasso estimator solves:

$$\min_\beta \|y - X\beta\|_2^2 + \lambda\|\beta\|_1 \tag{3}$$

where λ governs the penalization intensity. These estimators enjoy $\sqrt{n}$ -consistency and asymptotic normality under correct specification, but degrade under nonlinearity or heavy-tailed errors. Ridge employs l_2 penalization to control variance at the cost of bias without variable selection. Lasso enables automatic feature selection via l_1 penalization yet can be unstable under high multicollinearity. These limitations motivate the exploration of flexible ML methods.

2.2. Machine learning for estimation

Random forests (RF) construct an ensemble of B decision trees trained on bootstrap samples and average their predictions [3]. The random forest predictor is given by:

$$f(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \tag{4}$$

DOI: 10.54254/2755-2721/2026.GF34864

where T_b denotes the b-th tree. Gradient Boosting (GB) constructs additive ensemble models in a stage-wise manner [4]:

$$f_m(x) = f_{m-1}(x) + \nu \cdot h_m(x) \tag{5}$$

where h_m is a weak learner fitted to the negative gradient of the loss at step m and ν is the learning rate. XGBoost (eXtreme Gradient Boosting) and LightGBM (Light Gradient Boosting Machine) are optimized implementations with level-wise and leaf-wise tree growth, respectively [10, 11]. Random forests reduce variance through averaging, whereas boosting iteratively reduces bias. Deep neural networks (DNN) approximate the regression function through L compositional layers:

$$f(x) = W_L \sigma(W_{L-1} \sigma(\dots W_1 x + b_1 \dots) + b_{L-1}) + b_L \tag{6}$$

where σ is a nonlinear activation function (typically ReLU). DNNs have achieved state-of-the-art performance on image and text processing tasks. However, their high optimization variance and heavy computational overhead restrict their application to medium-scale tabular datasets.

2.3. Related work

Numerous studies have investigated the statistical properties of ML-based estimators. Scornet et al. established L^2 consistency of random forests under the additive regression model [12], Mentch and Hooker derived a central limit theorem for random forest predictions under subsampling [14], and Biau and Scornet provided a comprehensive survey of consistency, convergence rates, and variable importance [19].

For statistical inference, Wager and Athey developed honest random forests that yield asymptotically normal estimates of conditional average treatment effects [6], and Lei et al. constructed quantile regression forests with asymptotically valid prediction intervals [18]. In causal inference, Athey and Imbens extended random forests to generalized causal forests for heterogeneous treatment effect estimation [16], Künzel et al. proposed an ensemble-of-metalearners framework that adaptively selects the optimal combination of base learners [17], and Chernozhukov et al. introduced double/debiased machine learning for using flexible ML methods in nuisance function estimation while preserving $\sqrt{n}$-consistency for the target parameter [7].

For deep learning theories, Farrell et al. proved the asymptotic normality and semiparametric efficiency of DNN estimators under standard regularity conditions, laying a solid foundation for neural network-based statistical inference [13]. Regarding uncertainty quantification, Angelopoulos and Bates proposed conformal prediction, a distribution-free framework with few assumptions on data distribution [15].

The above research indicates that ML models are mainly applied as first-stage estimators, and statistical inference is implemented via second-stage approaches that are robust to the approximation errors from the first stage. Different from prior literature that primarily derives asymptotic properties theoretically, This work provides a systematic empirical comparison across controlled simulation regimes spanning linearity, mild misspecification, strong nonlinearity, and high-dimensional sparsity, thereby complementing theoretical results with practical guidance for method selection.

DOI: 10.54254/2755-2721/2026.GF34864

3. Experimental design and methodology

3.1. Simulation framework

This study designs four simulation scenarios with increasing complexity, ranging from ideal conditions for classical estimators to challenging settings where ML methods are supposed to achieve superior performance. Each scenario follows the general regression model:

$$y = f(X) + \varepsilon \tag{7}$$

where X_j~$N(0,1)$ for $j = 1, \ldots, p$, ε~$N(0, \sigma^2)$ with $\sigma = 1$, and f varies by scenario. All simulations use $n = 500$ observations repeated 100 times with independent draws to obtain stable Monte Carlo estimates. The linear scenario adopts $f(X) = X\beta$ with $p = 10$ and all coefficients non-zero, serving as the baseline where classical methods are optimal.

The semiparametric scenario introduces a mild sinusoidal perturbation: $f(X) = X\beta + 0.3\sin(X_1)$ with $p = 10$, representing mild misspecification where the true functional form deviates locally from linearity.

The nonlinear scenario incorporates trigonometric, logarithmic, and product interactions: $f(X) = \sin X_1 + \log(1 + |X_2|) + X_3 X_4$, with $p = 10$, producing strong nonlinearities that linear models cannot approximate without explicit basis expansion.

The high-dimensional sparse scenario uses $f(X) = X\beta$ with $p = 100$ but only $s = 5$ non-zero coefficients, representing settings where the signal is sparse relative to the feature space.

3.2. Methods and implementation

Eight estimation methods are compared across all scenarios. OLS, ridge regression ($\lambda = 1$), and lasso (λ selected by 5-fold cross-validation) represent classical linear estimators. Random forest uses 200 trees with maximum depth 10 and default minimum samples per leaf, providing a balance between bias and variance through ensemble averaging. XGBoost and LightGBM are configured with 200 trees, maximum depth 6, and learning rate 0.1; LightGBM additionally employs gradient-based one-side sampling to accelerate training without compromising accuracy. MLP employs one hidden layer of 50 units with ReLU activation, trained for 500 iterations using the Adam optimizer. DNN uses three hidden layers of 128, 64, and 32 units respectively, with ReLU activation and dropout 0.2, trained for 200 epochs using Adam (lr=0.001, batch size 64). A tuned DNN configuration (wider 256-128-64 architecture, batch normalization, increased dropout, learning rate scheduling, and extended training) was also evaluated. A supplementary grid search (max_depth 3/6/9, learning_rate 0.05/0.1/0.2, n_estimators 100/200) over XGBoost and LightGBM on the nonlinear scenario found default parameters near-optimal, consistent with reported robustness [10, 11]. All models are evaluated on 30 percent held-out test sets, with tree hyperparameters selected based on recommended defaults from the literature.

3.3. Evaluation metrics

Three primary metrics are used for evaluation. The test MSE is calculated on hold-out test sets, and the average results over 100 repeated experiments are reported with standard errors in parentheses.. Estimation bias is computed as:

DOI: 10.54254/2755-2721/2026.GF34864

$$\text{Bias}(\beta) = \|\mathbb{E}[\beta] - \beta\|_2 \tag{8}$$

This metric is applied to linear and semiparametric scenarios where the true coefficient vector is available. For nonlinear scenarios, estimation bias cannot be separated from variance without a predefined parametric target. Thus, we take test MSE as the major metric to evaluate model accuracy.

4. Results and analysis

4.1. Simulation study findings

In the linear scenario, OLS achieves the lowest MSE (1.02), followed closely by ridge (1.02) and lasso (1.01). Tree-based methods exhibit higher MSE (RF: 1.92, LightGBM: 1.67) due to variance inflation under correct specification. The DNN (1.21) falls between, approximately 19 percent above OLS. Estimation bias is negligible.

Under mild semiparametric misspecification with a sinusoidal perturbation, OLS and ridge maintain the lowest MSE (1.02), while tree-based methods (RF: 1.95, XGBoost: 1.85, LightGBM: 1.67) show modestly higher error. The DNN (1.22) performs competitively. The small perturbation magnitude relative to the noise level ($\sigma = 1$) limits the advantage of flexible methods in this regime.

The nonlinear scenario produces the largest differentials. LightGBM achieves the lowest MSE among tree-based methods (1.70), followed by XGBoost (1.91) and random forest (1.96). The DNN (MSE = 1.46) outperforms all tree ensembles, and paired t-tests confirm statistical significance ($p < 0.001$). This advantage reflects the nature of the DGP: the sine, logarithmic, and product-interaction components form a smooth global function that ReLU-activated DNNs can approximate with relatively few parameters through their compositional structure, whereas tree ensembles must partition the input space into many piecewise constant regions to achieve comparable accuracy. A tuned DNN configuration did not yield improvement (MSE = 1.50; $p = 0.011$), confirming near-optimality at this sample size. All linear methods exhibit substantially higher MSE (2.26).

A sample size sensitivity analysis shows that at $n = 200$, all methods exhibit similar mean MSE (2.0–2.5), with the tuned DNN (1.99) marginally leading. At $n = 500$, DNN (1.46) separates from tree ensembles (LightGBM: 1.70), and at $n = 2000$, DNN achieves MSE of 1.20 compared to LightGBM at 1.39—a 14 percent advantage, indicating that the DNN function approximation advantage is realized primarily at larger sample sizes.

In the high-dimensional sparse setting, lasso achieves the best MSE (1.28), correctly selecting the five non-zero coefficients. Tree-based methods exhibit elevated MSE (LightGBM: 1.81, RF: 2.32, XGBoost: 2.30), and the DNN (MSE = 2.71) struggles without sparsity-inducing regularization.

In summary, classical linear methods are preferred under linear or near-linear processes, the DNN excels under strong nonlinearity at moderate-to-large sample sizes, and lasso is optimal when the model is sparse.

4.2. Real-data validation

Two real-world binary classification datasets are examined. The Home Credit Default Risk dataset (307,511 loan applications, 104 numerical features after removing variables with over 50 percent missingness and imputing remaining missing values with the median) is evaluated via Brier score. Random forest achieves the lowest score (0.0699), followed closely by XGBoost (0.0699) and

DOI: 10.54254/2755-2721/2026.GF34864

logistic regression (0.0729), a 4 percent improvement over the linear baseline. DNN performance (0.0702) is comparable but requires substantially more computation: 27 seconds versus 0.3 for XGBoost and 0.7 for LightGBM. The PIMA dataset (768 observations, 8 numerical features, no missing values, binary classification) represents a complementary small-sample setting.

On the PIMA Diabetes dataset (768 observations), random forest achieves the lowest Brier score (0.159), followed by logistic regression (0.162) and DNN (0.164). The high variance of flexible methods in small samples offsets any potential gains from capturing nonlinear structure.

A supplementary DNN tuning experiment confirmed that the tuned DNN (Brier = 0.164 on PIMA, 0.070 on Home Credit) is statistically indistinguishable from the original configuration. Across both applications, the performance gap depends primarily on the degree of nonlinearity and sample size, rather than on the superiority of any single model class.

4.3. Computational trade-offs

All runtime tests are performed on a standard computing device equipped with an Intel Core i7 CPU and 32 GB RAM, without GPU acceleration. In the n = 500 simulation, tree ensembles train in 0.1–0.2 seconds while DNN training requires 0.6–0.7 seconds. This gap widens substantially at scale: on the Home Credit dataset (50K observations), DNN training takes 27 seconds versus 0.3 seconds for XGBoost and 0.7 seconds for LightGBM. Tree ensembles thus offer the most favorable accuracy-to-cost ratio for tabular data across sample sizes.

5. Conclusion

This paper conducts a systematic comparative analysis between classical estimation methods and machine learning approaches for parameter estimation, based on four simulation scenarios and two real-world applications. The principal findings are as follows. Under correct linear specification, OLS and ridge regression remain optimal both statistically and computationally. Under mild semiparametric misspecification, all methods perform similarly and OLS remains competitive. Under strong nonlinearity, the DNN achieves the lowest MSE, with the advantage growing from marginal at n = 200 to a clear 14 percent lead over tree ensembles at n = 2000 under the examined SNR. In high-dimensional sparse settings, lasso achieves the optimal bias-variance trade-off through automatic variable selection. These results indicate that method selection should be guided by data characteristics rather than by the presumed superiority of any single model class. In practice, tree ensembles offer a robust default for tabular data with unknown structure, while DNNs may be preferred when strong nonlinearity is suspected and sufficient observations are available.

Several limitations suggest directions for future work. First, the scope is restricted to regression with Gaussian errors; extension to classification tasks, survival analysis and heavy-tailed distribution scenarios; Second, the nonlinear DGP (trigonometric, logarithmic, and product interactions) represents one class of nonlinear structure; performance may differ under other forms such as high-order polynomials, threshold effects, or periodic functions. Third, bootstrap uncertainty for ML estimators is known to be anti-conservative [14]; future work should explore conformal prediction and variational inference. Fourth, the threshold at which DNNs consistently dominate tree ensembles depends on the regression function, signal-to-noise ratio, and feature dimensionality, and warrants further investigation. Addressing these limitations would strengthen the evidence base for principled method selection. Simulation code is available upon request.

DOI: 10.54254/2755-2721/2026.GF34864

References

[1] Lehmann, E.L., Romano, J.P. (2006) Testing statistical hypotheses. 3rd ed. Springer, New York.
[2] Hastie, T., Tibshirani, R., Friedman, J. (2009) The elements of statistical learning: data mining, Inference, and Prediction. 2nd ed. Springer, New York.
[3] Breiman, L. (2001) Random forests. Machine Learning, 45: 5–32.
[4] Friedman, J.H. (2001) Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29: 1189–1232.
[5] Goodfellow, I., Bengio, Y., Courville, A. (2016) Deep Learning. MIT Press, Cambridge, MA.
[6] Wager, S., Athey, S. (2018) Estimation and inference of heterogeneous treatment effects using random forests. Journal of the American Statistical Association, 113: 1228–1242.
[7] Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., Robins, J. (2018) Double/debiased machine learning for treatment and structural parameters. The Econometrics Journal, 21: C1–C68.
[8] Hayashi, F. (2000) Econometrics. Princeton University Press, Princeton, NJ.
[9] Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B, 58: 267–288.
[10] Chen, T., Guestrin, C. (2016) XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794.
[11] 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. In: Advances in Neural Information Processing Systems 30 (NeurIPS), 3146–3154.
[12] Scornet, E., Biau, G., Vert, J.-P. (2015) Consistency of random forests. Annals of Statistics, 43: 1716–1741.
[13] Farrell, M.H., Liang, T., Misra, S. (2021) Deep neural networks for estimation and inference. Econometrica, 89: 181–213.
[14] Mentch, L., Hooker, G. (2016) Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. Journal of Machine Learning Research, 17: 1–41.
[15] Angelopoulos, A.N., Bates, S. (2023) A gentle introduction to conformal prediction and distribution-free uncertainty quantification. Foundations and Trends in Machine Learning, 16: 495–607.
[16] Athey, S., Imbens, G. (2016) Recursive partitioning for heterogeneous causal effects. Proceedings of the National Academy of Sciences, 113: 7353–7360.
[17] Künzel, S.R., Sekhon, J.S., Bickel, P.J., Yu, B. (2019) Metalearners for estimating heterogeneous treatment effects using machine learning. Proceedings of the National Academy of Sciences, 116: 4156–4165.
[18] Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R.J., Wasserman, L. (2018) Distribution-free predictive inference for regression. Journal of the American Statistical Association, 113: 1094–1111.
[19] Biau, G., Scornet, E. (2016) A random forest guided tour. TEST, 25: 197–227.