Accuracy, Interpretability and Differential Privacy via Explainable Boosting

Harsha Nori
Microsoft
hanori@microsoft.com

Zhiqi Bu
University of Pennsylvania
zbu@sas.upenn.edu

Judy Hanwen Shen
Stanford University
judyshen@stanford.edu

Rich Caruana
Microsoft
rcaruana@microsoft.com

Janardhan Kulkarni
Microsoft
jakul@microsoft.com

Abstract

We show that adding differential privacy to Explainable Boosting Machines (EBMs), an algorithm for training Generalized Additive Models (GAMs), yields state-of-the-art accuracy while protecting privacy. Our experiments on multiple classification and regression datasets show that DP-EBMs suffer surprisingly little accuracy loss, even in the high privacy regime ($\varepsilon < 1$). An additional benefit is that the trained models are perfectly globally and locally interpretable, which is often desirable in many of the same settings as differential privacy.

1 Introduction

Security researchers have consistently shown that machine learning models can leak information about their training data [1, 2]. In some industries, like healthcare, finance, and criminal justice, models are trained and deployed on sensitive information, where this form of leakage might be especially disastrous. To combat these situations, researchers have embraced differential privacy, which establishes a strong mathematical standard for privacy guarantees on algorithms [3, 4]. In many of these high-stakes situations, model interpretability is also critical to provide audits, help models work with domain experts like doctors, and correct unwanted errors before deployment [5, 6]. In this paper, we address both concerns by developing a differentially private algorithm for learning Generalized Additive Models (GAMs) [7]. We show that this method can provide strong privacy guarantees, high accuracy, and perfect interpretability on tabular datasets.

While GAMs are traditionally fit using smooth low-order splines, we focus on the Explainable Boosting Machine (EBM), a modern implementation that learns each shape function using boosted decision trees [8, 9]. EBMs are especially interesting because they often match the accuracy of complex blackbox algorithms (like XGBoost, random forests, and neural networks [10, 11]), while having a simple final structure and optimization procedure [12, 13]. We exploit this simplicity to develop a differentially private learning method for EBMs – DP-EBM – which adds calibrated noise during the training process. Our experiments on classification and regression datasets show that this algorithm retains surprisingly high accuracy even with strong differential privacy guarantees.

Preprint. Under review.

2 Preliminaries

2.1 Explainable Boosting Machine

Explainable Boosting Machines belongs to the family of Generalized Additive Models (GAMs), which are restricted machine learning models that have the form:

$$g(E[y]) = \beta + f_0(x_0) + f_1(x_1) + ...f_k(x_k)$$

where β is an intercept, each f_j is a univariate function that operates on a single input feature x_j, and g is a link function that adapts the model to different settings like classification and regression [7].

While GAMs are more flexible than linear models (where each function f_j is further restricted to be linear), they are significantly less flexible than most machine learning models due to their inability to learn high-order interactions between features (ex: $f_j(x_0, x_1)$). However, this restricted, additive structure has the benefit of making GAMs perfectly interpretable. At prediction time, each feature contributes a single score, which are then summed together and passed through a link function. These scores are exact local feature importance measures, which can be sorted, compared, and reasoned about. In addition, each function f_k can be visualized to provide an exact global description of how the model operates across varying inputs.

Research has shown that fitting GAMs with boosted decision trees leads to more accurate models [8, 5]. EBMs are the latest open-source implementation of boosted tree based GAMs [9]. We build on top of this software and briefly outline the training process below.

The EBM training procedure starts by binning the data in each feature into B large bins, similar to other algorithms like LightGBM [14]. It then proceeds to learn each univariate shape function by using gradient boosted decision trees. However, unlike traditional boosting methods, EBMs use a *cyclic* gradient boosting procedure in which each tree is carefully restricted to train on only one feature at a time in a "round-robin" fashion. During each round of training, EBMs visit each of the k input features in order, and grow a shallow decision tree on the pseudo-residuals. This tree is penalized by an extremely low learning rate to mitigate the impact of the ordering of features and then simply added into the function for that feature. Finally, after growing a decision tree for the k-th feature, the boosting procedure cycles back to the first feature, and repeats for all E epochs.

3 Algorithms

Algorithm 1: Differentially Private Explainable Boosting Machine (DP-EBM)

Input: $X \in \mathbb{R}^{n \times K}$: data, $y \in \mathbb{R}^n$: labels, α: learning rate, E: number of epochs, S: number of leaf nodes, R: range of y_i, ε: privacy parameter, H_k: DP Histograms for each feature k

Output: f: learned functions (one per feature)

$f = \{f_1, ..., f_k\} \leftarrow \{0, ..., 0\}$ `// initialize f as k functions`
$\{r_1, ..., r_n\} \leftarrow \{y_1, ..., y_n\}$ `// initialize residuals`
for $epoch(e) \leftarrow 1$ **to** E **do**
  foreach *feature* k **do**
    $X_k \leftarrow X[:, k]$
    $s_1, .., s_S \leftarrow$ `RandomlySplitData(`X_k, S`) // S = 3 yields 3 leaf nodes`
    foreach s_j **do**
      $\mathbf{T} \leftarrow \sum_{i \in s_j} r_i(t)$
      $\hat{\mathbf{T}} := \mathbf{T} + \sigma_e R \cdot \mathcal{N}(0, 1)$ `// Add noise to sum of residuals`
      $\mu \leftarrow \alpha \hat{\mathbf{T}} / \sum_{\text{bin} \subset s_j} |H_k(\text{bin})|$
      for *bin* $\in s_j$ **do**
        `/* Each split` s_j `contains contiguous bins */`
        $f_k[\text{bin}] \leftarrow f_k[\text{bin}] + \mu$
    foreach x_i **do**
      $r_i(t) \leftarrow y_i - \sum_1^k f_k(x_{i,k})$ `// update residuals`
return $f_1, ..., f_K$

Theorem 3.1. Algorithm 1 is (ϵ, δ)-private.

To add differential privacy guarantees to the EBM algorithm, we make several modifications to the training procedure. We first spend a small portion of our privacy budget to generate differentially private bins for each feature of the dataset. After this pre-processing step, we begin boosting. In traditional tree building algorithms, there are two major data-intensive operations: learning the structure of the tree (what feature and feature threshold to install at each node in the tree), and calculating the predicted value of each leaf node [15]. Prior work on differentially private tree learning typically splits budget between choosing which features to split on, where to split them, and learning prediction values for each leaf node [16–18].

EBMs naturally avoid spending any privacy budget on choosing which features to include in a tree. The algorithm is restricted to grow trees on only one feature at a time, and visits each feature on a data-agnostic "round-robin" schedule. Furthermore, by choosing the splitting thresholds at random, we can learn the entire structure of each tree without looking at the training data. Prior work and our empirical evaluations both show that choosing random splits results in little accuracy loss [19]. We therefore spend the entirety of our budget per iteration on learning the values for each leaf node. Each leaf contains a disjoint subset of the data, and the predicted value is simply an average of the residuals for the data belonging to the node. To calculate a private average, we simply add calibrated Gaussian noise to the sum of residuals per leaf node, and divide by the previously calculated differentially private counts from our private binning pre-procedure [4]. The learned tree is added into the feature function f_k, and the cyclic boosting procedure continues. A full sketch of the DP-EBM algorithm is described in Algorithm 1.

4 Experiments

We compare the following algorithms on four classification and three regression datasets:

- DP EBM: Differentially private EBM described in Algorithm 1. We use the following (default) parameters for all experiments: max_bins = 32, learning_rate = 0.01, n_epochs = 300, max_leaves = 3. We allocate 10% of the total budget to binning and 90% to training.
- Generalized Linear Models: Linear and Logistic Regression are widely prevalent methods for interpretable machine learning. For both models, we use the implementation in IBM's differential privacy library [20]. This software implementation follows the algorithms described in [21, 22] for linear regression and in [23] for logistic regression.
- DP Boost: DPBoost is a private gradient boosted decision tree algorithm introduced by [18]. It builds on top of LightGBM, a popular boosting framework [14]. We use the parameters recommended by the DPBoost authors' open source repository for all experiments.

		Regression | RMSE (Lower is better)		
dataset	ε	DPBoost	DP Linear Regression	DP-EBM
cal-housing	0.5	391316 ± 67258	111437 ± 1284	**79479 ± 2019**
	1.0	204742 ± 8031	109293 ± 1434	**75843 ± 1055**
	2.0	122153 ± 3703	108975 ± 1317	**73081 ± 635**
	4.0	93794 ± 3178	108002 ± 1426	**72603 ± 805**
	8.0	87140 ± 2340	107553 ± 1550	**72395 ± 782**
	Non-Private	**46502 ± 787**	69284 ± 1414	57038 ± 380
elevators	0.5	0.047 ± 0.004	6.102 ± 3.363	**0.006 ± 0.002**
	1.0	0.024 ± 0.002	2.677 ± 0.867	**0.005 ± 0.000**
	2.0	0.013 ± 0.001	1.297 ± 0.397	**0.005 ± 0.000**
	4.0	0.008 ± 0.001	0.661 ± 0.530	**0.004 ± 0.000**
	8.0	0.006 ± 0.000	0.387 ± 0.181	**0.004 ± 0.000**
	Non-Private	**0.002 ± 0.000**	0.003 ± 0.000	**0.002 ± 0.000**
wine-quality	0.5	4.843 ± 0.814	3.886 ± 1.874	**0.938 ± 0.012**
	1.0	2.014 ± 0.140	2.032 ± 0.647	**0.822 ± 0.034**
	2.0	1.324 ± 0.087	1.628 ± 0.418	**0.776 ± 0.012**
	4.0	0.927 ± 0.034	0.891 ± 0.067	**0.742 ± 0.015**
	8.0	0.857 ± 0.014	0.879 ± 0.112	**0.729 ± 0.009**
	Non-Private	**0.619 ± 0.013**	0.753 ± 0.011	0.702 ± 0.014

To evaluate performance, we generate 5 train-test splits and report the average and standard deviation of test performance for each model at varying ε and fixed $\delta = 1e-5$. We use root mean squared error (RMSE) as the metric for regression, and area under the ROC curve for classification.

All datasets, with the exception of the healthcare dataset, are publicly available and the script to reproduce these results will be made public in our repository. We include results from this private, real world medical dataset to highlight how these models might perform in a high stakes settings where both privacy and interpretability are critical.

		Classification | AUROC (Higher is better)		
dataset	ε	DPBoost	DP Logistic Regression	DP-EBM
adult-income	0.5	0.535 ± 0.039	0.474 ± 0.082	$\mathbf{0.874 \pm 0.004}$
	1.0	0.498 ± 0.047	0.441 ± 0.085	$\mathbf{0.883 \pm 0.005}$
	2.0	0.600 ± 0.037	0.526 ± 0.117	$\mathbf{0.888 \pm 0.005}$
	4.0	0.726 ± 0.012	0.537 ± 0.087	$\mathbf{0.890 \pm 0.005}$
	8.0	0.791 ± 0.008	0.521 ± 0.058	$\mathbf{0.893 \pm 0.004}$
	Non-Private	$\mathbf{0.929 \pm 0.004}$	0.633 ± 0.096	0.928 ± 0.004
credit-fraud	0.5	0.398 ± 0.122	0.579 ± 0.033	$\mathbf{0.966 \pm 0.007}$
	1.0	0.422 ± 0.144	0.505 ± 0.100	$\mathbf{0.966 \pm 0.006}$
	2.0	0.431 ± 0.096	0.568 ± 0.115	$\mathbf{0.968 \pm 0.008}$
	4.0	0.530 ± 0.137	0.496 ± 0.104	$\mathbf{0.969 \pm 0.008}$
	8.0	0.573 ± 0.184	0.622 ± 0.070	$\mathbf{0.971 \pm 0.008}$
	Non-Private	0.577 ± 0.203	0.907 ± 0.017	$\mathbf{0.979 \pm 0.004}$
healthcare	0.5	0.506 ± 0.047	0.455 ± 0.040	$\mathbf{0.787 \pm 0.023}$
	1.0	0.518 ± 0.049	0.431 ± 0.040	$\mathbf{0.811 \pm 0.010}$
	2.0	0.507 ± 0.047	0.518 ± 0.038	$\mathbf{0.825 \pm 0.011}$
	4.0	0.538 ± 0.054	0.518 ± 0.041	$\mathbf{0.835 \pm 0.009}$
	8.0	0.665 ± 0.026	0.529 ± 0.027	$\mathbf{0.838 \pm 0.010}$
	Non-Private	0.836 ± 0.011	0.741 ± 0.018	$\mathbf{0.846 \pm 0.010}$
telco-churn	0.5	0.496 ± 0.060	0.635 ± 0.122	$\mathbf{0.826 \pm 0.017}$
	1.0	0.605 ± 0.034	0.384 ± 0.169	$\mathbf{0.828 \pm 0.018}$
	2.0	0.750 ± 0.024	0.523 ± 0.136	$\mathbf{0.829 \pm 0.016}$
	4.0	0.762 ± 0.006	0.697 ± 0.072	$\mathbf{0.830 \pm 0.017}$
	8.0	0.786 ± 0.003	0.739 ± 0.059	$\mathbf{0.832 \pm 0.016}$
	Non-Private	0.831 ± 0.010	0.838 ± 0.012	$\mathbf{0.845 \pm 0.013}$

5 Discussion

Our empirical evaluation shows that DP-EBMs perform surprisingly well on tabular datasets, even in the high privacy regime ($\varepsilon < 1$). In the non-private setting, EBMs often perform competitively with other boosting algorithms, like LightGBM [12, 9], but DP-EBMs have a significant advantage in the differentially private setting. We believe this might be explained by the significant privacy budget savings when learning each tree – unlike other algorithms, we spend no budget on learning the tree structure, and focus on learning the best leaf node values. In addition, by growing shallow trees on single features at a time, each leaf tends to contain large portions of the dataset which further minimizes the impact of noise. Conversely, DP-EBMs also benchmark well against differentially private linear methods, which are popular private and interpretable machine learning algorithms.[1] The added flexibility of being able to learn non-linear functions on each feature generally results in a performance boost in the non-private setting, which appears to translate to the private setting. In addition, the iterative nature of gradient boosting might give DP-EBMs the ability to recover from bad applications of noise early in the training process. In short, it appears that DP-EBMs can maintain all the privacy and interpretability of a differentially private linear model with significantly higher accuracy. In the future, we hope to expand upon the simplified DP-EBM algorithm and re-introduce features like bagging, interaction detection, and early stopping detailed in [24, 9, 5].

[1] We observed that both DPBoost and DP Logistic Regression occasionally reported abnormally low AUROCs (< 0.4) in some experimental runs. The AUROC of models predicting at random is typically around 0.5. By inverting predictions, these implementations would paradoxically perform better as ε approaches 0.

References

[1] Nicholas Carlini, Chang Liu, Jernej Kos, Úlfar Erlingsson, and Dawn Song. The secret sharer: Measuring unintended neural network memorization & extracting secrets. 2018.

[2] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706. IEEE, 2019.

[3] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[4] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[5] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1721–1730, 2015.

[6] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[7] Trevor J Hastie and Robert J Tibshirani. *Generalized additive models*, volume 43. CRC press, 1990.

[8] Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158, 2012.

[9] Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*, 2019.

[10] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[11] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[12] Chun-Hao Chang, Sarah Tan, Ben Lengerich, Anna Goldenberg, and Rich Caruana. How interpretable and trustworthy are gams? *arXiv preprint arXiv:2006.06466*, 2020.

[13] Caroline Wang, Bin Han, Bhrij Patel, Feroze Mohideen, and Cynthia Rudin. In pursuit of interpretable, fair and accurate machine learning for criminal recidivism prediction. *arXiv preprint arXiv:2005.04176*, 2020.

[14] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, pages 3146–3154, 2017.

[15] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[16] Sam Fletcher and Md Zahidul Islam. A differentially private decision forest. *AusDM*, 15: 99–108, 2015.

[17] Kai Wen Wang, Travis Dick, and Maria-Florina Balcan. Scalable and provably accurate algorithms for differentially private distributed decision tree learning.

[18] Qinbin Li, Zhaomin Wu, Zeyi Wen, and Bingsheng He. Privacy-preserving gradient boosting decision trees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 784–791, 2020.

[19] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[20] Naoise Holohan. Diffprivlib: The ibm differential privacy library. `https://github.com/IBM/differential-privacy-library`, 2019.

[21] Or Sheffet. Private approximations of the 2nd-moment matrix using existing techniques in linear regression. *arXiv preprint arXiv:1507.00056*, 2015.

[22] Hafiz Imtiaz and Anand D Sarwate. Symmetric matrix perturbation for differentially-private principal component analysis. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2339–2343. IEEE, 2016.

[23] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

[24] Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–631, 2013.