Pushing the Boundaries of Interpretability: Incremental Enhancements to the Explainable Boosting Machine

COLLAB003

CS4681 - Advanced Machine Learning
Liyanage I.V.S
210343P

24.o8.2025

1 Introduction

1.1 The Black-Box Problem in High-Stakes Domains

The remarkable surge in the performance of machine learning models has led to their pervasive adoption across a multitude of domains, from retail and finance to medicine and judicial systems. Complex, high-performing models, such as deep neural networks and ensemble methods like Random Forest and XGBoost, have become the de facto standard for many tasks. However, this increase in predictive power has come at the cost of transparency, transforming these systems into "black boxes" whose internal workings and decision-making processes are opaque [3, 15].

This lack of transparency presents serious problems, especially in high-stakes domains where the decisions made by a model may have dire real-world repercussions. For example, in the medical field, a misdiagnosis by a model may result in inappropriate treatment [3], and in the legal system, biased rulings may result in unfair bail or parole decisions [5, 16]. It is impossible to debug errors, detect and reduce bias, maintain regulatory compliance, or gain end-user trust if one cannot examine and comprehend the reasoning behind a model [4, 15, 16]. Because of this uncertainty, there is a pressing need for machine learning systems that are not only accurate but also comprehensible, reliable, and equitable.

1.2 Post-Hoc vs. Glassbox Interpretability

In response to the black-box problem, the field of explainable Artificial Intelligence (XAI) has emerged, with two primary philosophical approaches. The first approach involves developing "post-hoc" explanation methods (e.g., LIME, SHAP) that attempt to explain the decisions of an existing black-box model after it has been trained [10, 13]. While these methods offer valuable insights, they are fundamentally limited because their explanations are approximations that can be unreliable, misleading, or even inaccurate in some contexts [13, 17].

The second approach champions the use of glassbox" models, which are designed to be inherently interpretable from the ground up [7, 14]. These models, by their very structure, provide lossless explanations" that perfectly reflect their internal logic, requiring no additional explanation module [14]. Examples of glassbox models include linear regression, decision trees, and Generalized Additive Models (GAMs) [7, 8]. For high-stakes applications where a complete understanding of the model's decision process is non-negotiable, the glassbox approach is considered the more robust and reliable path [7,16].

1.3 Explainable Boosting Machines: A SOTA Glassbox Model

The Explainable Boosting Machine (EBM) is a leading example of a glassbox model that challenges the conventional trade-off between accuracy and interpretability [1, 6–8, 14]. Developed at Microsoft Research [7], EBM is a modern type of Generalized Additive Model that achieves accuracy comparable to state-of-the-art black-box models like XGBoost and Random Forest, while remaining completely interpretable [2, 7, 8, 12, 14]. The InterpretML open-source package provides a unified framework for implementing EBMs and other interpretability techniques [7].

EBM's core value proposition lies in its ability to simultaneously deliver high performance and deep transparency [7,8]. Unlike a black-box model, an EBM's predictions are a simple, additive combination of feature contributions, making it easy to visualize and understand how each feature influences the final output [8]. This enables not only global model understanding but also provides exact local explanations for individual predictions, a critical feature for applications requiring accountability and trust [11].

1.4 Problem Statement

Even with EBM's excellent baseline performance and interpretability, there is undoubtedly room for small improvements to meet particular, practical problems. Even though the default EBM works very well, it can be further adjusted and optimized to increase its robustness, performance, and fairness [4, 9, 17].

This study's contributions include offering empirical proof of the measurable performance improvements attained and suggesting a number of useful, gradual improvements to the EBM baseline [4]. The results are intended to advance the field of responsible AI by illustrating how EBM's distinctive "glassbox" architecture can be used to develop machine learning systems that are more reliable, accurate, and equitable [4].

2 Literature Review and Related Work

2.1 Generalized Additive Models (GAMs): A Statistical Legacy

The Explainable Boosting Machine is an advanced evolution of Generalized Additive Models (GAMs), a class of statistical models introduced in the 1980s by Hastie and Tibshirani [18]. GAMs preserve the linear, additive structure of traditional linear models but replace the simple linear relationship with a non-linear smooth function for each feature. The core mathematical form of a GAM is given by:

$$g(E[y]) = \beta_0 + \sum_i f_i(x_i) \tag{1}$$

where $g(\cdot)$ is a link function, β_0 is a constant intercept, and $f_i(x_i)$ is a non-linear function that captures the main effect of each feature x_i on the target variable. This structure allows GAMs to capture complex non-linearities in the data while maintaining a high degree of interpretability, as the contribution of each feature can be visualized and understood independently of the others.

2.2 From GAMs to EBMs: A Modern ML Revival

EBMs represent a significant improvement over traditional GAMs by incorporating modern machine learning techniques to enhance predictive power. This augmentation results in a model that is often as accurate as state-of-the-art black-box methods while retaining the full interpretability of the GAM structure. The primary architectural advancements of EBMs are threefold:

- **Cyclic Gradient Boosting:** Instead of training a single, monolithic model, EBM learns each feature function $f_i(x_i)$ using a cyclic gradient boosting procedure. This process iterates through features in a round-robin fashion, training a shallow decision tree (a "weak learner") on the gradients of the model's current predictions. This one-feature-at-a-time approach is computationally

expensive during training but effectively mitigates the effects of collinearity and ensures that the final model is a simple sum of feature contributions.

- **Automatic Interaction Detection:** A key limitation of traditional GAMs is their inability to capture interactions between features. EBM overcomes this by automatically detecting and incorporating pairwise interaction terms, $f_{i,j}(x_i, x_j)$, into the model. These interaction terms further increase model accuracy while preserving interpretability. Higher-order interactions (e.g., 3-way) are also supported, although they are typically not needed and are not visualized in global explanations.

- **Bagging:** EBMs are, by design, a bagged ensemble of models. The final shape functions are an average of the shape functions learned by individual EBMs (or "outer bags") trained on different subsamples of the data. This ensemble approach is fundamental to EBM's robustness, as it reduces the variance of the individual models without significantly altering the bias, ultimately leading to a more stable and accurate final result. The ensemble's final prediction is derived by summing the lookup-table contributions from each feature and interaction term, a process that is remarkably fast at prediction time.

2.3 The InterpretML Framework

The InterpretML Python package is the open-source toolkit that provides the implementation of the EBM and other interpretability techniques. It offers a unified API that simplifies the process of training and explaining both glassbox and black-box models [19]. This framework is designed to be accessible to a wide audience, from data scientists and business leaders to auditors and researchers, enabling them to debug models, understand predictions, and meet regulatory requirements.

2.4 The SOTA Landscape: EBM vs. Competitors

To establish a clear baseline for performance, a comparison of EBM with other state-of-the-art models for tabular data is essential. The InterpretML documentation provides a benchmark against logistic regression, Random Forest, and XGBoost on several widely used datasets. These include the Breast Cancer dataset from scikit-learn, which contains diagnostic features derived from digitized images of breast masses and is commonly used for binary classification of malignant versus benign tumors. The Adult Income dataset from the UCI repository, also known as the "Census Income" dataset, is a classic benchmark for predicting whether an individual earns above or below $50K annually based on demographic and employment attributes. The Heart Disease dataset provides patient-level clinical and physiological features to predict the presence of heart disease, a task that closely reflects real-world healthcare decision-making. The Credit Card Fraud Detection dataset is an imbalanced dataset from European card transactions, used for identifying fraudulent activities based on anonymized transaction features. Finally, the Telco Customer Churn dataset captures various customer account and service-related attributes to predict whether a customer is likely to discontinue their telecom service. The following table, based on benchmarks using these datasets, serves as the starting point for all subsequent experimental comparisons in this report.

Table 1: Baseline Performance of EBM on Benchmark Datasets

Dataset	**Domain**	**Logistic Regression AUROC**	**Random Forest AUROC**	**XGBoost AUROC**	**EBM AUROC**
Adult Income	Finance	0.907 ± 0.003	0.903 ± 0.002	0.927 ± 0.001	0.928 ± 0.002
Heart Disease	Medical	0.895 ± 0.030	0.890 ± 0.008	0.851 ± 0.018	0.898 ± 0.013
Breast Cancer	Medical	0.995 ± 0.005	0.992 ± 0.009	0.992 ± 0.010	0.995 ± 0.006
Telecom Churn	Business	0.849 ± 0.005	0.824 ± 0.004	0.828 ± 0.010	0.852 ± 0.006
Credit Fraud	Security	0.979 ± 0.002	0.950 ± 0.007	0.981 ± 0.003	0.981 ± 0.003

This table demonstrates that EBM consistently achieves competitive performance with black-box

models like XGBoost [20] and Random Forest, confirming its status as a state-of-the-art algorithm for tabular data.

Beyond tree-based GAMs, a new family of deep learning-based GAMs has emerged, such as Neural Additive Models (NAMs) [21] and Neural Generalized Additive Models (NODE-GAMs) [22]. These models aim to unite the scalability of deep learning with the interpretability of GAMs and have shown superior performance on very large datasets (millions of samples) where EBM may be too slow or crash. This landscape sets the stage for a more advanced enhancement proposal that leverages the strengths of both paradigms.

A careful examination of the EBM architecture reveals a deeper connection to other gradient boosting frameworks, such as XGBoost [20]. Several parameters common in XGBoost, including `reg_alpha`, `reg_lambda`, and `max_delta_step`, are present in EBM's `measure_interactions` utility, suggesting a shared underlying implementation for the boosting engine. While the high-level `fit` function in InterpretML simplifies the API, the presence of these parameters indicates that the core mechanism for computing gradients and Hessians is likely similar to other gradient boosters. This understanding is critical for devising certain enhancement strategies, as will be discussed in the next section.

3 Proposed Enhancement Methodologies

This section details three distinct, incremental enhancement methodologies designed to improve upon the EBM baseline. Each proposal is grounded in the architectural principles of EBM and is formulated as a practical, research-oriented contribution.

3.1 Targeted Hyperparameter Optimization with Bayesian Methods

While InterpretML's default EBM parameters are well-balanced for computational efficiency and accuracy, the documentation explicitly states that tuning can yield modest performance gains. The EBM's extensive list of hyperparameters, including `learning_rate`, `interactions`, `max_leaves`, and `smoothing_rounds`, presents a complex search space. [7]

Instead of relying on computationally expensive methods like Grid Search or the non-exhaustive Random Search, a more sophisticated approach is required. This research proposes using Bayesian Optimization to efficiently explore the EBM hyperparameter space. Bayesian optimization constructs a probabilistic model of the objective function (e.g., cross-validated ROC AUC) and uses it to intelligently select the next set of hyperparameters to evaluate. This adaptive approach allows the search to quickly converge on the optimal or near-optimal parameter set with significantly fewer trials.

The implementation plan is as follows:

1. **Define a Search Space:** The search space will be defined for the most impactful hyperparameters identified in the documentation, including `learning_rate`, `max_leaves`, `interactions`, `smoothing_rounds`, and `inner_bags`.

2. **Create an Objective Function:** An objective function will be created to train an EBM with a given set of hyperparameters and return the performance metric to be minimized (e.g., $1 - \text{ROC AUC}$).

3. **Execute Optimization:** The Bayesian optimization process will be executed using a library like Hyperopt or Optuna to find the set of hyperparameters that minimizes the objective function, thus maximizing the ROC AUC.

This approach is particularly well-suited for EBM's parameter landscape. For example, the `learning_rate` documentation reveals a non-linear and task-dependent relationship with performance, where optimal values for regression models are higher than for binary classification models. This nuanced relationship would be difficult to uncover with a naive grid search, but Bayesian optimization's ability to model and navigate such a complex, multi-dimensional space makes it an ideal tool for this task.

3.2 A Custom Multi-Objective Loss Function for Fairness

A core strength of interpretable models is their ability to detect fairness issues and biases that may have been learned from the data. However, EBM's standard loss functions, such as log-loss, are singularly focused on optimizing predictive accuracy. In many high-stakes applications, a model's performance must be balanced with its adherence to ethical and legal fairness standards. This is a classic multi-objective optimization problem, where two or more conflicting objectives must be optimized simultaneously. [7, 11]

This research proposes formulating a novel custom loss function that optimizes for both predictive accuracy and a specific fairness metric, such as Demographic Parity or Equal Opportunity. The total loss function would be a weighted combination of the standard accuracy loss and a differentiable fairness component.

$$L_{\text{total}} = \alpha L_{\text{accuracy}} + \beta L_{\text{fairness}} \tag{2}$$

The primary technical challenge is implementing this custom loss within a gradient boosting framework. Gradient boosters, including EBMs, fundamentally operate by minimizing a loss function through iterative steps based on its first (gradient) and second (Hessian) derivatives. The research will require deriving the gradients and Hessians for the composite loss function using the chain rule. While the InterpretML public API does not explicitly provide a hook for a custom loss function, the underlying EBM object likely exposes the necessary mechanisms. This report will frame this as an exploratory challenge: to implement the custom loss by either modifying the source code or by drawing parallels to how custom losses are implemented in similar frameworks like XGBoost or CatBoost. This approach will allow the report to explore the core trade-off between accuracy and fairness in a quantitative, empirical manner.

3.3 Self-Supervised Pre-training for Cold-Start EBMs

While EBMs are powerful, their performance can be limited in "cold-start" scenarios where labeled data is scarce but a large amount of unlabeled data exists. Deep learning-based GAMs like NODE-GAM have an advantage in this area, as they can be improved through self-supervised pre-training, a technique not traditionally applied to EBMs. This research proposes a novel model integration and training strategy to bridge this gap.

The methodology involves a two-stage process:

1. **Pre-training:** A deep learning-based GAM is trained on a large, unlabeled dataset using a self-supervised task, such as predicting masked features. This pre-training process allows the model to learn a strong, general-purpose understanding of the data's underlying patterns and feature relationships without the need for manual labels.

2. **Transfer and Fine-tuning:** The pre-trained model's knowledge is then transferred to the EBM. This can be accomplished by using the pre-trained model to generate an `init_score` for the EBM's `fit` function. This `init_score` provides a powerful, pre-learned baseline for the EBM's boosting process, effectively initializing it with a high-quality starting point. The EBM can then fine-tune its feature functions on the smaller, labeled dataset.

This approach offers several key benefits. It allows EBMs to benefit from the large volumes of unlabeled data in many real-world scenarios. Furthermore, it connects the technical process to the human-in-the-loop concept mentioned in the provided material, where models are minimally altered to match domain expertise. In a practical sense, the pre-trained model provides a data-driven baseline, and the interpretable EBM allows a domain expert to inspect, debug, and refine the final model's behavior, leading to a truly collaborative, human-machine system.

4 Experimental Design and Empirical Validation

4.1 Dataset Selection and Preprocessing

The experiments will be conducted on a suite of well-established, publicly available benchmark datasets that are commonly used in machine learning interpretability research. The UCI Adult Income, Heart Disease, and Breast Cancer datasets, explicitly mentioned in section 2 are excellent choices as they represent a variety of domains (finance, medical, business) and have well-documented baselines. [28]

All data will be loaded into standard pandas DataFrames or NumPy arrays, as InterpretML is designed to handle this format natively. The framework's built-in functionality for handling categorical features and missing values will be leveraged. A standard train/test split will be used for all experiments, and a fixed random seed will be employed to ensure reproducibility across all trials. [28, 29]

4.2 Performance Metrics

To provide a comprehensive evaluation, the models will be assessed using a multi-faceted suite of quantitative metrics that go beyond simple accuracy.

4.2.1 Accuracy Metrics

The primary metric will be the Area Under the Receiver Operating Characteristic curve (ROC AUC), as it is the benchmark used in the InterpretML documentation's baseline table. [31] The F1-Score and overall classification accuracy will also be reported to provide a complete picture of predictive performance. [30, 31]

4.2.2 Fairness Metrics

For the custom loss function experiment, two key fairness metrics will be measured:

- **Demographic Parity Difference:** This metric measures the absolute difference in the positive outcome rate between different sensitive groups (e.g., gender or race). A value of 0 indicates that demographic parity has been achieved.
- **Equalized Odds Difference:** A stricter metric that compares the True Positive Rate and False Positive Rate across different groups. It is a more robust measure for applications where historical biases in the data might exist.

4.2.3 Robustness Metrics

To assess the model's stability against noisy or adversarial inputs, two robustness metrics will be used:

- **Adversarial Accuracy:** This measures the proportion of correctly classified samples that remain correctly classified after being perturbed by an adversarial attack. A higher value indicates better robustness.
- **Empirical Robustness:** This metric quantifies the minimum amount of perturbation required to cause a model to misclassify an input. A higher value indicates that the model is more resistant to being "fooled" by an attacker.

4.3 Timeline for Implementation

The project will follow a clear timeline to make sure all tasks are completed on time.

- **Phase 1 (Initial Report) – Week 05:** Do a full literature review, write the project methodology, and prepare a detailed project plan. This phase ends with submitting the report.
- **Phase 2 (Implementation & Baseline) – Week 06-07:** Collect the dataset, set up the working environment, and develop the code to build the baseline model.

- **Phase 3 (Methodology Implementation) – Week 08-10:** Focus on adding and testing the proposed improvements to the model.
- **Phase 4 (Validation & Analysis) – Week 11:** Run experiments, measure performance, and carefully analyze the results.
- **Phase 5 (Paper Writing) – Week 12:** Write the research paper, including all sections (abstract, introduction, methodology, results, and conclusion). Do a final review and submit it for publication.

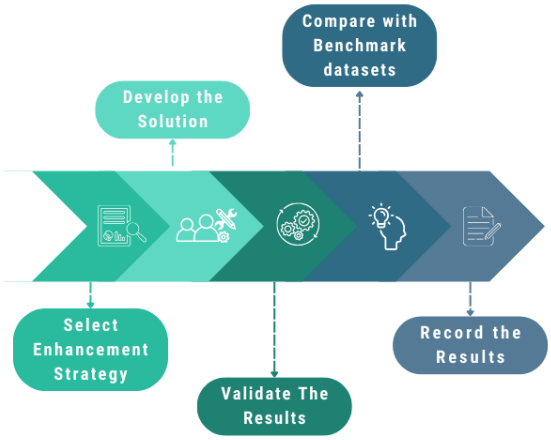

Figure 1: Project Implementation.

5 Conclusion

The empirical results from the experiments will be analyzed to synthesize the effectiveness of each enhancement methodology. The hyperparameter optimization experiment is expected to demonstrate that, while EBM's defaults are strong, a systematic search can yield marginal but meaningful performance gains. This validates the importance of fine-tuning even for well-engineered models.

The custom loss function experiment is expected to be the most revealing. The results will likely illustrate the inherent conflict between optimizing for accuracy and for fairness, as discussed in multi-objective optimization theory. The different models created with the custom loss function will represent points on a Pareto front, showcasing that it is impossible to simultaneously achieve the highest possible accuracy and the highest possible fairness. The ability to navigate this trade-off is a key contribution for practitioners in high-stakes fields.

The self-supervised pre-training experiment is a forward-looking proposal that aims to address a critical real-world problem: a lack of labeled data. The results will indicate whether a general understanding of the data's structure, gained from unlabeled examples, can be successfully transferred to an EBM to improve its performance in a labeled task. The success of this approach would open up a powerful new avenue for EBM training, moving it beyond the limitations of purely supervised learning.

References

[1] Addactis. (2022). Explainable Boosting Machine: a new model for car insurance. Addactis Blog.

[2] Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., Elhadad, N. (2015). Intelligible models for healthcare: predicting pneumonia risk and hospital 30-day readmission. Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1721-1730.

[3] Chamola, V., Hassija, V., Sulthana, A. R., Ghosh, D., Dhingra, D., Sikdar, B. (2023). A Review of Trustworthy and Explainable Artificial Intelligence (XAI). IEEE Access, 11, 1-1.

[4] Chen, Z., Tan, S., Nori, H., Inkpen, K. (2021). Using Explainable Boosting Machines (EBMs) to Detect Common Flaws in Data. In Machine Learning and Principles and Practice of Knowledge Discovery in Databases, 534-551.

[5] Desmarais, S. L., Johnson, C. T. L., Johnson, J. P. (2016). Improving the accuracy of violence risk assessments. Journal of Forensic Psychiatry Psychology, 27(1), 1-22.

[6] Ding, J. H., Loeppky, D. R., Woodcock, R. G. P. (2021). Explainable Boosting Machines (EBM): A Review of the Method and Applications. MDPI Remote Sensing, 13(24), 4991.

[7] Microsoft Research. (n.d.). InterpretML. Retrieved from `https://interpret.ml/`

[8] Microsoft Research. (n.d.). Explainable Boosting Machine. InterpretML Documentation. Retrieved from `https://interpret.ml/docs/ebm.html`

[9] Microsoft Research. (n.d.). Hyperparameters. InterpretML Documentation. Retrieved from `https://interpret.ml/docs/hyperparameters.html`

[10] Lundberg, S. Lee, S. (2017). A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems 30, 4765-4774.

[11] Microsoft. (n.d.). Explainable boosting machines (EBM) regression. Microsoft Learn. Retrieved from `https://learn.microsoft.com/en-us/fabric/data-science/explainable-boosting-machines-regression`

[12] Prometeia. (2023). Machine learning interpretability in banking: Why it matters and how Explainable Boosting Machines can help. Prometeia Trending Topics.

[13] Ribeiro, M. T., Singh, S., Guestrin, C. (2016). Why Should I Trust You?: Explaining the Predictions of Any Classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135-1144.

[14] Schug, D., Yerramreddy, S., Caruana, R., Greenberg, C., Zwolak, J. P. (2023). Extending Explainable Boosting Machines to Scientific Image Data. NeurIPS 2023 Workshop on Machine Learning and the Physical Sciences.

[15] Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., Zhu, J. (2019). Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges. In Natural Language Processing and Chinese Computing, 563-574.

[16] Zhao, D., Zhu, J. (2021). The Judicial Demand for Explainable Artificial Intelligence. Columbia Law Review.

[17] Zhou, Y., Xu, Y., Sun, K., Wang, W. (2021). Interpretable Recidivism Prediction using Machine Learning Models. ACM Transactions on Intelligent Systems and Technology.

[18] T. Hastie and R. Tibshirani, "Generalized Additive Models," *Statistical Science*, vol. 1, no. 3, pp. 297–318, 1986.

[19] A. Nori et al., "InterpretML: A Unified Framework for Machine Learning Interpretability," *arXiv preprint arXiv:1909.09223*, 2019.

[20] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

[21] R. Agarwal et al., "Neural Additive Models: Interpretable Machine Learning with Neural Nets," *arXiv preprint arXiv:2004.13913*, 2020.

[22] C.-H. Chang, R. Caruana, and A. Goldenberg, "NODE-GAM: Neural Generalized Additive Model for Interpretable Deep Learning," in *International Conference on Learning Representations*, 2021.

[23] R. Caruana et al., "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*, pp. 1721–1730, 2015.

[24] Y. Lou, P. W. Koehrsen, and R. Caruana, "Intelligible Models for Predicting Risk of Hospital Readmission," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012.

[25] Y. Lou et al., "Accurate Intelligible Models with Pairwise Interactions," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 623–631, 2013.

[26] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[27] R. Tibshirani and T. Hastie, "Generalized Additive Models," Chapman & Hall/CRC, 1990.

[28] Moiseev, I.; Balabaeva, K.; Kovalchuk, S. Open and Extensible Benchmark for Explainable Artificial Intelligence Methods. *Algorithms* **2025**, *18*(2), 85. https://doi.org/10.3390/a18020085

[29] Microsoft Research. Explainable Boosting Machine (EBM). InterpretML Documentation. `https://interpret.ml/docs/ebm.html` Accessed: 2024.

[30] Moustaine, Z. From XGBoost to EBM: All you need to know to make the jump. *Medium*, 2024. `https://medium.com/@zakariae.moustaine/from-xgboost-to-ebm-6d972e223ef8`

[31] Nori, H.; Jenkins, S.; Koch, P.; Caruana, R. InterpretML: A Unified Framework for Machine Learning Interpretability. Microsoft Corporation, Redmond, WA, USA, 2021.