Predictive Uncertainty in Gradient-Boosted Regression Trees : A Muon Energy Reconstruction Case Study

Nikolaos Karafyllis
School of Electrical and Computer Engineering
National Technical University of Athens
Athens, Greece
nickarafyllis@gmail.com

Abstract

Accurate energy reconstruction of muon events detected by the ANNIE experiment is of paramount importance for subsequent analytical endeavors. While, on an ML approach, Gradient Boosted Regression Trees (GRBTs) have been previously employed to estimate energy values, no effort has been made to quantify the uncertainty associated with these point-predictions. In this work, we explore different methods to quantify and evaluate uncertainty for our task, mainly using the CatBoost model. Empirically, we find that IBUG+CBU method achieves the best probabilistic and point-prediction performance.
Source code of this work is be available at https://github.com/nickarafyllis/ReconstructionUncertainty.

1 Introduction

Boosted decision Trees (BDTs) have been extensively used in the realm of particle physics since the pioneering approaches that utilized them for particle identification [23] [25] The use of BDTs gained significant momentum with the remarkable success of XGBoost [5] in the Higgs Boson Machine Learning Challenge [16] on Kaggle. Since then, multiple implementations of BDTs have emerged, *e.g.* LightGBM [15] , Catboost [22], NGBoost [6] that together with XGBoost achieve state-of-the-art results on tabular (structured) data.

Uncertainty remains an inherent challenge in the utilization of BDTs and other machine learning models. While these models excel in prediction and classification tasks, they often operate within an environment characterized by ambiguity and imprecision. Understanding, quantifying, and mitigating uncertainty is crucial for deploying these models reliably and for interpreting their outcomes accurately.

In the context of muon energy reconstruction in the ANNIE experiment, addressing these uncertainties becomes paramount to ensure precise and dependable analysis of particle interactions within the detector.

In this report, we make an introduction to concept of predictive uncertainty and to uncertainty quantification and evaluation methods, using Boosted Decision Trees (BDTs). We outline the application of a BDT with uncertainty estimation algorithm to the muon energy reconstruction task. Furthermore, we present a comparison between the different uncertainty quantification methods within our task, where we find that IBUG+CBU method achieve the best probabilistic and point-prediction performance. Finally, we discuss the significance of uncertainty estimation on practical applications within the HEP field.

2 Background

2.1 Gradient Boosted Regression Trees

Gradient Boosting [7] is a powerful machine learning algorithm of combining many "weak" learners iteratively to create a "strong" learner. Each subsequent learner added attempts to fix the errors of its predecessor, thus minimizing the overall prediction error, as seen in Fig. 1.

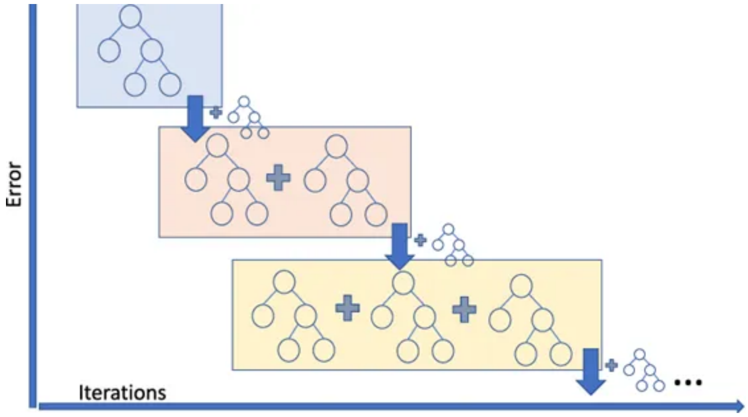


Figure 1. Schematical representation of gradient boosting regression in regards to algorithm iterations. Source: [3]

"The 'gradient' term in Gradient boosting refers to the gradient of the loss function with respect to the predictions made by the current ensemble of models at each step, guiding the addition of new models to minimize the loss."

Gradient Boosting vs Gradient Descent "Gradient descent descends the gradient by introducing changes to parameters, whereas gradient boosting descends the gradient by introducing new models.", *a stack exchange user*.

Gradient Boosting Regression Trees (the regression version of BDTs) use regression trees as weak learners. Each tree has a few layers and tries to estimate(or fix) the residuals(usually MSE :mean squared error) of the previous tree. The final model makes a prediction by summing the values of the leaves traverses of all subtrees.

An example of the tree ensemble structure of BDTs is shown in Fig. 2 :

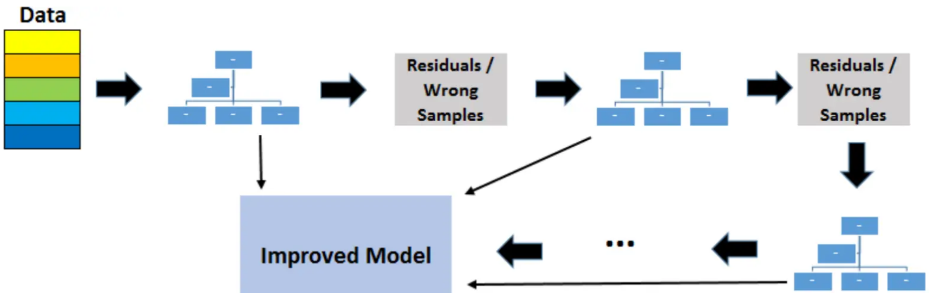


Figure 2. Gradient boosted decision tree: sequentially connected weak learners
Image source: `https://towardsdatascience.com/gradient-boosted-decision-trees-explained-9259bd8205af`

Several libraries of Gradient Boosted Regression Trees exist, including AdaBoost, Scikit-learn BDT, XGBoost, LightGBM, NGBoost and CatBoost. In the original attempt to find which one suits better in the problem discussed in §3, we found that CatBoost is superior, as shown in Table 1.

For this reason, we mainly focus on uncertainty using the CatBoost model.

Table 1. BDT model point-prediction comparison

BDT model	RMSError
Scikit BDT	33.16
XGBoost	**32.84**
LightGBM	34.27
NGBoost	33.01
CatBoost	**31.50**

2.2 Sources of Predictive Uncertainty

> Successful decisions under uncertainty depend on our minimizing our ignorance, accepting inherent randomness and knowing the difference between the two.
>
> *Craig Fox*
> *UCLA Business School*

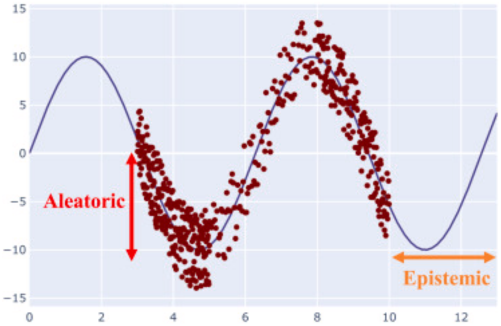


Figure 3. A schematic view of the main differences between aleatoric and epistemic uncertainties, when having only one independent variable. Source: [1]

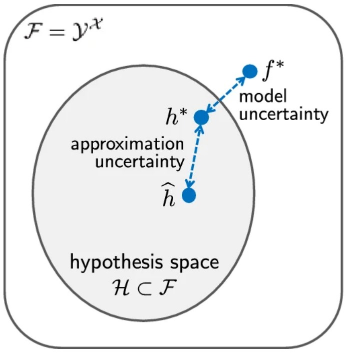
While many sources of uncertainty may exist, they are generally categorized as either aleatoric or epistemic.

- Aleatoric(aka statistical) uncertainty refers to the inherent randomness and complexity of the data(due to noise, unseen features *etc*.), that is considered irreducible. An example is the unpredictability in the outcomes of rolling dice.
- Epistemic(aka systematic) uncertainty refers to the "lack of knowledge about the behaviour of the system, that is conceptually resolvable" [13]. In a supervised-learning context, epistemic uncertainty can be distinguished in model and approximation uncertainty as shown in Fig. 4. Model uncertainty comes from being unsure about one's model choice due to a limited number of training samples. Approximation uncertainty, on the other hand, stems from the model's ability to approximate the training samples accurately, often due to its generalization ability.

Reducible vs irreducible The distinction of uncertainty into aleatoric and epistemic uncertainty is a rather convenient way to be aware of which uncertainties can be (easily) reduced and which cannot. Epistemic uncertainty can be reduced by observing more data in the same setting of the problem. While considered irreducible, aleatoric uncertainty can be also reduced by adding new features as shown in the example of Fig. 5. In general, "embedding data in higher dimensional space will reduce aleatoric and increase epistemic uncertainty, because fitting a model will become more difficult and require more data", as discussed in [14].

A more concrete demonstration on the distinction of uncertainties can be seen in the classification example of Fig.6

Overall, distinction of uncertainty is context-depended due to its ambiguous nature.


$\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$
f^*
model uncertainty
h^*
approximation uncertainty
$\widehat{h}$
hypothesis space
$\mathcal{H} \subset \mathcal{F}$

	point prediction	probability
ground truth	$f^*(\boldsymbol{x})$	$p(\cdot \mid \boldsymbol{x})$
best possible	$h^*(\boldsymbol{x})$	$p(\cdot \mid \boldsymbol{x}, h^*)$
induced predictor	$\hat{h}(\boldsymbol{x})$	$p(\cdot \mid \boldsymbol{x}, \hat{h})$

Figure 4. Different types of uncertainties related to different types of discrepancies and approximation errors: f^* is the best overall predictor, h^* is the best predictor within the hypothesis space(training data), and $\widehat{h}$ the predictor produced by the learning algorithm. Source: [14]

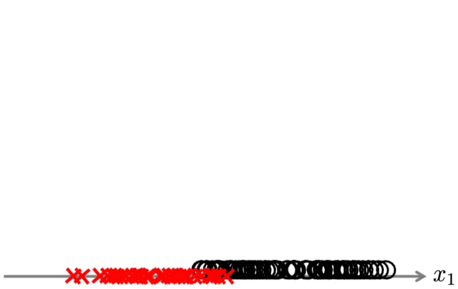


Figure 5. Left: The two classes are overlapping, which causes (aleatoric) uncertainty in a certain region of the instance space. Right: By adding a second feature, and hence embedding the data in a higher-dimensional space, the two classes become separable, and the uncertainty can be resolved. Source: [14]

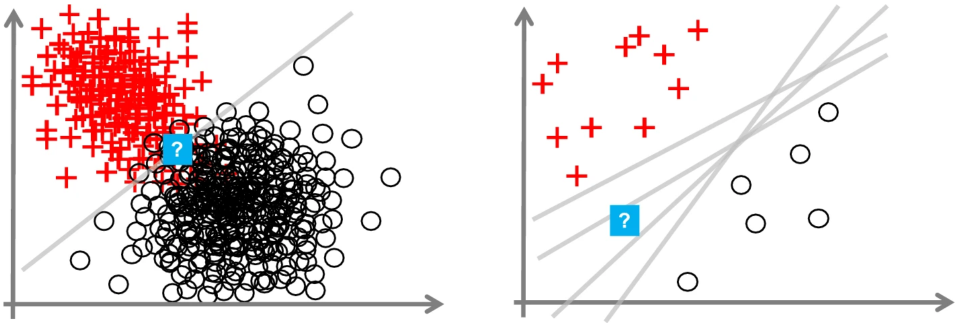


Figure 6. Left: Even with precise knowledge about the optimal hypothesis, the prediction at the query point (indicated by a question mark) is aleatorically uncertain, because the two classes are overlapping in that region. Right: A case of epistemic uncertainty due to a lack of knowledge about the right hypothesis, which is in turn caused by a lack of data. Source: [14]

2.3 Uncertainty Quantification in BDTs

Several methods for capturing uncertainty on machine learning models exist [8]. However, here we focus on BDT specific methods (not black-box methods) that seem to exploit better the structure of BDTs.

2.3.1 Catboost with Uncertainty

CatBoost [19] is a high performance gradient boosting decision trees library, with uncertainty estimation support.

To estimate **data uncertainty**, CatBoost employs a specific loss function called RMSEwithUncertainty. This loss function is designed to minimize two key aspects simultaneously:

- Point-Error (RMSE - Root Mean Square Error): This component focuses on reducing the discrepancy between the actual target values and the predicted values generated by the model. RMSE measures the average distance between the predicted values and the observed values, giving higher weight to larger deviations.
- Proper Scoring Rule (NLL - Negative Log-Likelihood §2.4.3): This element aims at optimizing a scoring rule that evaluates the predicted probability distribution's accuracy against the true distribution of the data. Negative log-likelihood is a common proper scoring rule used to measure the quality of probabilistic predictions.

By minimizing both RMSE and NLL simultaneously within the RMSEwithUncertainty loss function, CatBoost attempts to generate predictions that not only provide a point prediction (the sum of predictions from the subtrees) but also offer an understanding of the uncertainty associated with those predictions in the form of an expected variance within the predicted distribution (the mean of variances predicted from the subtrees).

The output of CatBoost, therefore, includes not just a single point prediction but also information about the uncertainty or variability associated with that prediction.

So, given the vector of variances predicted from each subtree $s = (s_0, ..., s_N - 1)$, we calculate data uncertainty as $\bar{s}$.

To capture **epistemic** (knowledge) uncertainty, an ensemble of catboost models is employed. Thus, epistemic uncertainty is calculated as the variance of the point predictions from each model. When using the ensemble, data uncertainty is modeled as the mean data uncertainty across all models.

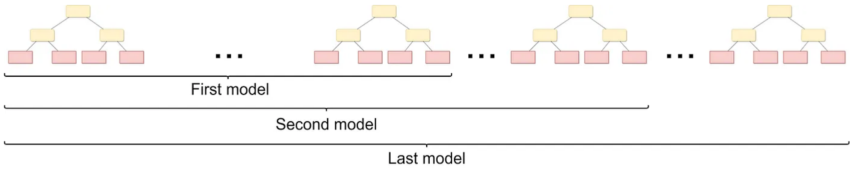


Figure 7. Virtual ensemble of catboost models: a single model is actually used

In practice, training an ensemble of several CatBoost models can be too expensive, so we use a virtual ensemble as in Fig. 7.

To calculate the total uncertainty we use the rule of total variation:

Total Uncertainty = Knowledge Uncertainty + Expected Data Uncertainty

or

$$\underbrace{V_p(y|x,D)[y]}_{\text{Total Uncertainty}} \approx \underbrace{\frac{1}{M}\sum_{m=1}^{M}\left[\left(\sum_{m=1}^{M}\frac{\mu_m}{M}\right) - \mu_m\right]}_{\substack{\text{Knowledge Uncertainty}\\ \text{(Variance of point-predictions}\\ \text{of each model)}}} + \underbrace{\frac{1}{M}\sum_{m=1}^{M}\sigma_m^2}_{\substack{\text{Expected Data Uncertainty}\\ \text{(Mean data uncertainty}\\ \text{of each model)}}} .$$

2.3.2 IBUG: Instance Based Uncertainty Quantification

IBUG [4] is a method to extend any Gradient Boosted Regression Tree (GBRT) point predictor to produce probabilistic predictions. IBUG uses the point-prediction of the underlying GBRT as predicted mean and it finds the k-nearest training instances to estimate the uncertainty of the target prediction. In this context, affinity is measured as the number of times both instances appear in the same leaf throughout the ensemble. It uses a validation dataset to tune the number of k and calibrate the prediction variances.

IBUG is a nearest neighbors approach and thus seems well-suited to estimating aleatoric uncertainty since it can quantify the range of outcomes to be expected given the observed features. However, by tuning k and variances on held-out data(validation set), it optimizes prediction uncertainty encompassing both aleatoric and epistemic uncertainty.

2.4 Uncertainty Evaluation Metrics

based on Uncertainty Toolbox Tutorial

While for single-point predictions accuracy metrics like RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) are commonly used, the evaluation of uncertainty quantification requires other evaluation methods.

There is still no golden metric in Uncertainty Quantification so we should inspect various metrics simultaneously.

2.4.1 Average Calibration Metrics

"Calibration refers to the statistical consistency between the distributional forecasts(i.e., the predictive distribution) and the observations and is a joint property of the predictions and the events that materialize." [11]

Validity / Coverage

When making a prediction, one can form an α-prediction interval that aims to capture observed values $\alpha\%$ of the time. We can iterate over values of α and see the proportion of the test data that actually fall within the prediction interval. The calibration(or reliability) diagram then shows the predicted proportion of the test data we expect to lie inside the interval on the x-axis and the observed proportion of the test data inside the interval on the y-axis.

A perfectly calibrated model should plot the identity function. Any deviation from a perfect diagonal represents miscalibration. We can use the area between the produced curve and the f(x) = x line to gauge how miscalibrated our model is. Using this miscalibration area we can calculate **MACE** (mean absolute calibration error) and **RMSCE** (root mean square calibration error). The reliability diagram of a slightly overconfident model can be seen in Fig. 8.

Expected Normalized Calibration Error - ENCE Expected Normalized Calibration Error (ENCE) [18] is analogous to the expected calibration error (ECE) [21] used in classification. To evaluate how calibrated the predicting model is, we compare per bin j two quantities : the root of the mean variance (RMV) of each distribution and the empirical root mean square error (RMSE):

$$ENCE = \frac{1}{N}\sum_{j=1}^{N}\frac{|RMV(j) - RMSE(j)|}{RMV(j),} \tag{1}$$

This score averages the calibration error in each bin, normalized by the bin's mean predicted variance, since for a larger variance, we expect naturally larger errors.

2.4.2 Sharpness / Interval Length

"Sharpness refers to the concentration of the predictive distributions and is a property of these distributions only." [10]

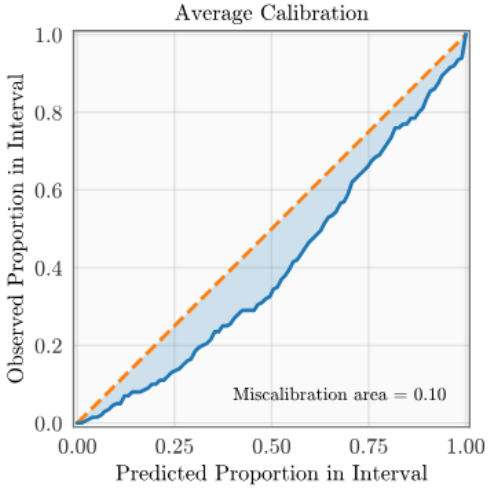


Figure 8. Reliability diagram of a slightly overconfident model: predicted proportion of data within intervals is more than the observed proportion.

Sharpness is a measure of how concentrated the predictive distribution is. It is evaluated *solely* based on the predictive distribution, and neither the datapoint nor the ground truth distribution are considered when measuring sharpness. An example can be seen in Fig. 9.

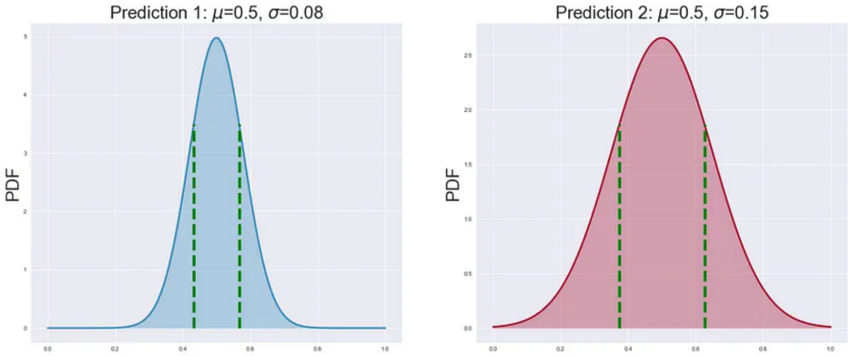


Figure 9. Sharpness example: Distribution on the left is sharper than the right one, therefore more confident in its predictions.

Sharpness can be counted by calculating an average interval length(*e.g.* standard deviation) along all of the predictions.

Calibration vs. Sharpness. By itself, calibration is not enough to guarantee a useful forecast. Sharpness is desired because ideally, the predictive distribution should be tight around the observed data.

2.4.3 Proper Scoring Rules

One class of metrics that considers both calibration and sharpness simultaneously is proper scoring rules.[10]

A proper scoring rule is any function that assigns a score to a predictive probability distribution, where the maximum score of the function is attained when the predictive distribution exactly matches the ground truth distribution (i.e. the distribution of the data).

Negative Log-Likelihood (NLL) / Maximum Likelihood Estimation (MlE)

"The likelihood function describes the joint probability of the observed data as a function of the parameters of the chosen statistical model" (parameters here: mean and variance of predictive distribution)

NLL of the normal distribution is defined as the negative log of the probability density function of the normal distribution:

$$L_i(y_i; \mu_i, \sigma_i^2) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(y_i - \mu_i)^2}{2\sigma_i^2}\right) \tag{2}$$

$$NLL_i(y_i; \mu_i, \sigma_i^2) = -log(L_i) \tag{3}$$

The NLL is being evaluated at each observation using the parameters of the (normal) distribution. To compare NLL between algorithms we average NLLs across all observations. The lower the (average) NLL is the better the fit.

Minimizing the NLL loss is equivalent to performing Maximum Likelihood Estimation (MLE)

Continuous Ranked Probability Score - CRPS "The Continuous Ranked Probability Score, known as CRPS, is a score to measure how a proposed distribution approximates the data, without knowledge about the true distributions of the data."

$$CRPS(P, x_a) = \int_{-\infty}^{\infty} \|P(x) - H(x - x_a)\|_2 dx, \tag{4}$$

where: x_a is the true value, x the predicted value, $P(x)$ the predicted cumulative distribution of x and $H(x)$ is the Heaviside step function $H(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$

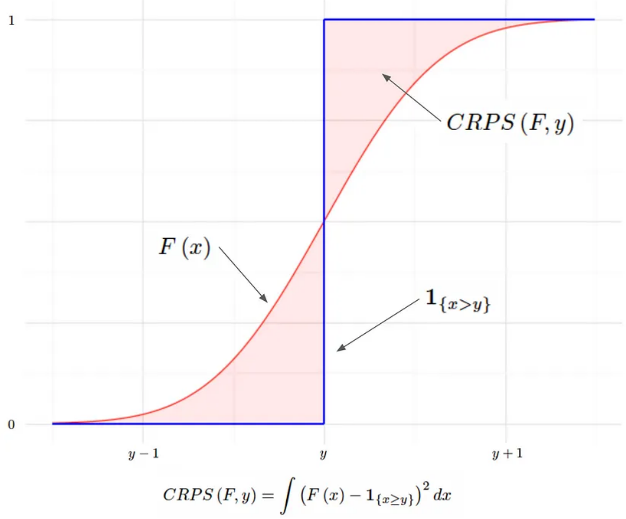


Figure 10. Visualization of the CRPS. The predicted distribution is marked in red, and the ground truth's degenerate distribution is marked in blue. The CRPS is the (squared) area trapped between the two CDFs. Image source

A good explanation of the CRPS formula can be found in the appendix.

2.5 Uncertainty recalibration

Uncertainty recalibration, in the context of machine learning and predictive modeling, refers to the process of adjusting or refining the estimated uncertainties associated with a model's predictions.

Many modern Neural networks are over-confident in their predictions [12], Graph neural networks are under-confident [24]

Given the trained uncalibrated model and a calibration set, we can train an auxiliary model to adjust the predicted distributions of the uncalibrated model to the true distributions.

Several methods exist including std scaling [18] (which multiplies the STD of each predicted distribution by a constant scaling factor s), isotonic regression [17] and histogram binning.

A recalibration example can be seen example of Fig. 11.

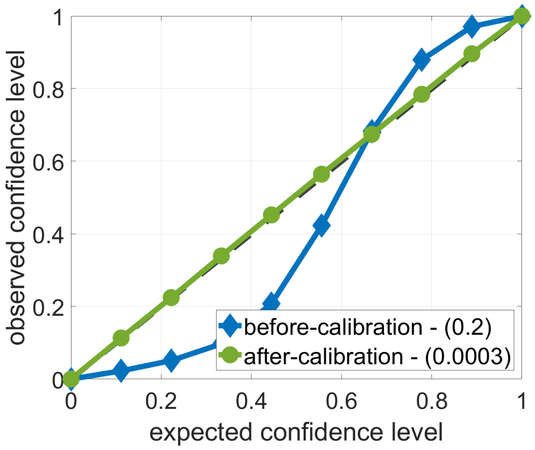


Figure 11. Recalibration of predicted uncertainty distributions.

3 An illustrative example

In the following, we investigate the application of uncertainty estimation using BDT machine learning algorithms in the context of muon energy reconstruction, specifically focusing on an example case within the ANNIE [2] experiment at Fermilab.

3.1 The experiment

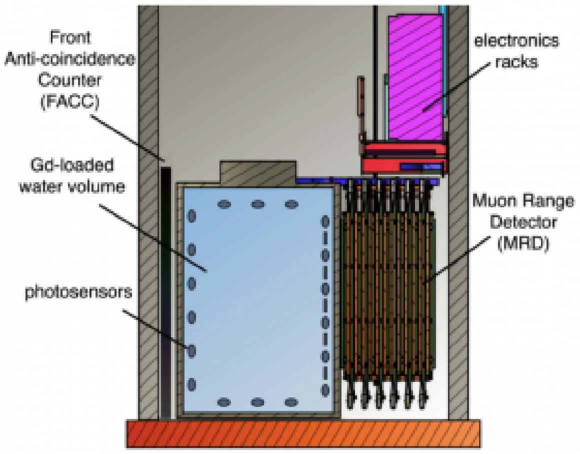


Figure 12. ANNIE detector schematic

Neutrinos from the muon neutrino beam (BNB) of Fermilab enter the ANNIE detector water tank where they interact with water (neutrino-nucleus interactions) and produce various particles including muons.

The produced muons travel through the water tank, leaving behind tracks. As these muons move faster than the speed of light in water, they emit Cherenkov radiation—a type of electromagnetic radiation characterized by the production of photons in a cone-like pattern along the path of the muon.

The Cherenkov photons, emitted at specific angles relative to the muon track (see Fig. 13), propagate through the water tank. This emitted light is then detected by the photodetectors—such as Photomultiplier Tubes (PMTs) and Large-Area Picosecond Photodetectors (LAPPDs)—that are strategically positioned on the walls of the water tank.

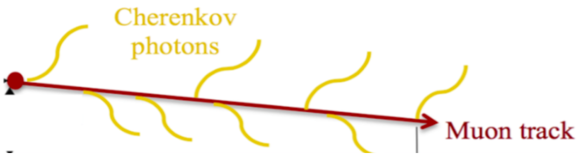


Figure 13. Cherenkov photon emissions used for muon tracking

In conjunction with the photodetectors, an external Muon Range Detector (MRD) records the position and timing of the muon tracks throughout this subdetector.

Diagrams of neutrino-nucleus interactions that produce muon leptons can be seen in Figs. 14, 15.

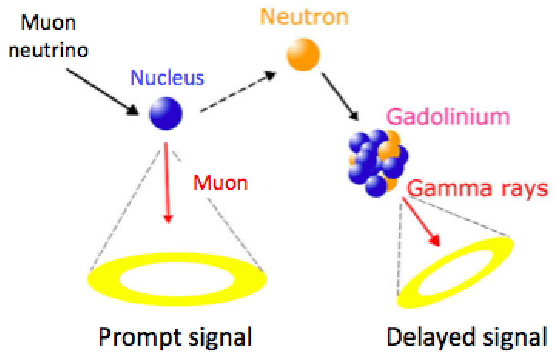


Figure 14. A neutrino interaction

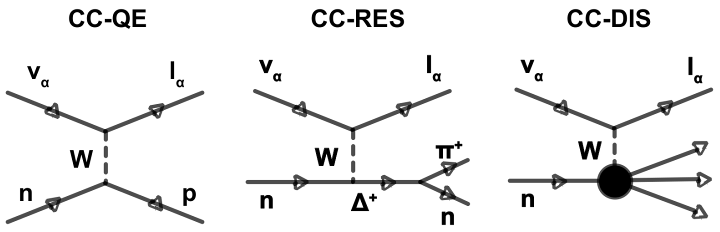


Figure 15. Feynman diagrams of CC neutrino-nucleon scattering channels. Source: [20].

3.2 Event generation and data

In order to train a machine learning prediction model for the muon energy reconstruction task, true labels (energy values) are needed. For this reason, we use simulated data to train and evaluate our models. ANNIE uses a neutrino event generator called `GENIE`(Generates Events for Neutrino Interaction Experiments), that predicts the properties(*e.g.* energy) of all the particles(*e.g.* muons) produced in the primary neutrino interactions.

The generated interactions are then propagated through a simulation of the ANNIE detector using `WCSim`. In this way, we simulate what is recorded by the detector electronics during these interactions (*e.g.* PMT-LAPPD charges and timing).

Before undertaking the energy reconstruction task, multiple reconstructions steps have been employed (*e.g.* vertex, track), so the data available for this task are muon track lengths, some of its relative coordinates within the detector and the number of hits on the detector parts.

3.3 Feature analysis and selection

In the process of employing a machine learning model, it is useful to perform exploratory data analysis (EDA) by investigating the data features that will be used. In this way, we can verify and validate our theoretical assumptions about the variables, identify correlations and anomalies within data. In the case that the model does not perform well on the primary features, we can construct more advanced features that can capture better the formulation of the problem. We examine the permutation importances as well as the SHAP values of our features.

3.3.1 Feature permutation importance

In this feature investigation technique, features are ranked by importance depending on the drop in model score when they are removed. In Fig. 16, we observe that some features seem to have near zero permutation importance while they should have more based on the physics problem intuition. This can be attributed to the strict reconstruction selection cuts applied in previous reconstruction steps(vertex reconstruction and track reconstruction), that limit the impact of some features.

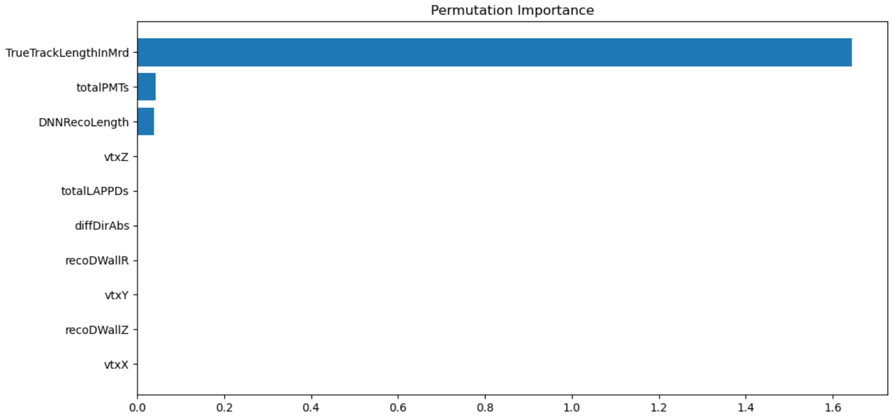


Figure 16. Permutation importances

3.3.2 Shapley values

SHAP (SHapley Additive exPlanations) is a game theory approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions.

In essence, the Shapley value of a feature is its contribution to the outcome, weighted and summed over all possible contributions. It can be employed in feature importance studies in both regression and classification tasks.

The distribution of the impacts each feature has on the model output can be seen in Fig. 17.

We observe that SHAP values of the features the are similar to their permutation importances.

We tried to train and test the model with only the first 5 features (with the highest importances and SHAP values). However, we observed that the mean square error in the validation set increased from 850 to 887. So, we decided to keep all the features.

3.4 Tree structure - best subtree

In Fig. 18, we visualise the best iteration(subtree) of a Catboost with uncertainty model trained on our data. Notice that the tree depth is 2 as we have selected it in the hyperparameter tuning process.

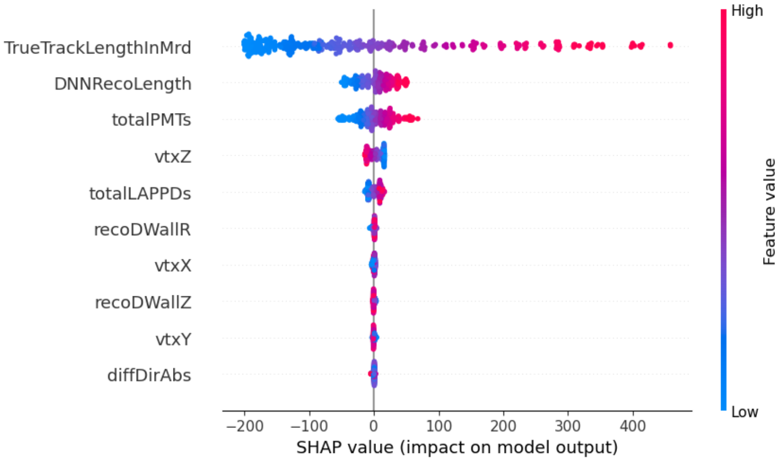


Figure 17. Distribution of the impacts each feature has on the model output

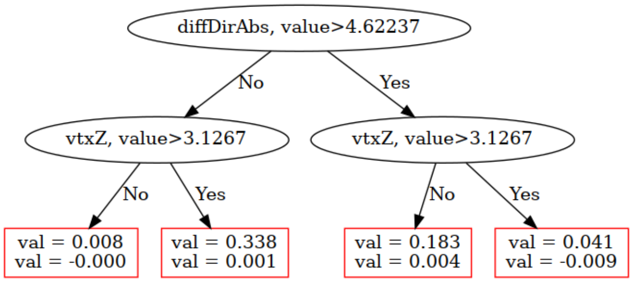


Figure 18. Best CatBoost subtree(with the best result of the loss function)

The values of the splits are scaled. The output of each leaf contains the contribution of the subtree to the point prediction and expected variance(data uncertainty).

Since we have 800 subtrees in each CatBoost model, we cannot find the important features and margins by viewing just one subtree (like we would in a classification model). However, we find it useful to visualize a subtree, to understand its substructure better.

4 Experiments

4.1 Methodologies

We compare probabilistic and point predictions

In specific, the models we test are:

1. CatBoost with uncertainty (CBU)
2. IBUG on CatBoost
3. IBUG on XGBoost
4. IBUG+CBU (averaging 1. and 2., as they have shown to be complementary in [4])

The general methodology we use to train and evaluate the models can be seen in Fig. 19 :

After training the model, (we firstly add IBUG extender if needed, and) we train a calibration model using the validation data. In prediction mode, the prediction variances are calibrated by the trained calibration model to produce the final predictive distributions. In evaluation mode, we use proper scoring rules (*i.e*. NLL, CRPS) to evaluate our predictions as they include both coverage and sharpness. We also calculate the error of point-predictions (distributional mean) using RMSE and MAE metrics, as accuracy is almost as equally important as uncertainty estimation performance.

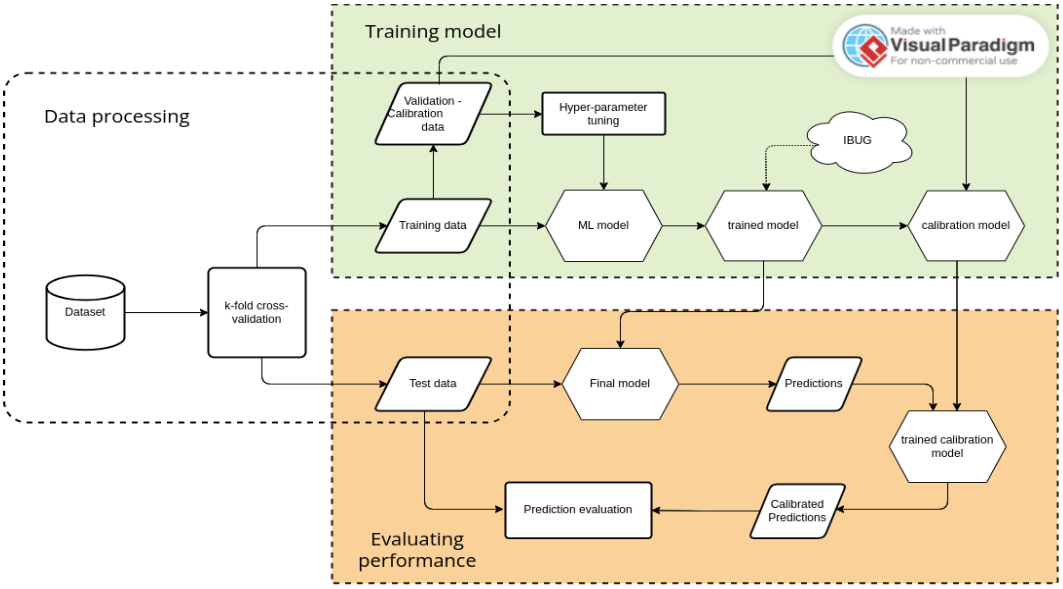


Figure 19. The training, calibrating and evaluating process used.

IBUG method calibrates itself during the tuning of k parameter so it will not need an additional calibration step. While CBU method is not explicitly calibrated, it seems that in our case its results in test set are calibrated enough as shown in Fig. 20, so we will not use an additional calibration step.

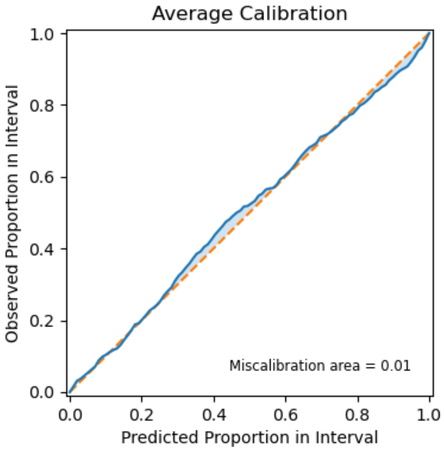


Figure 20. Reliability diagram of CBU method: no recalibration needed.

4.2 Results

Table 2 summarizes the probabilistic performance of each method. We observe that the averaging method (IBUG+CBU) and the CBU method have the best scores, with averaging method being slightly better at the CRPS proper scoring rule and at the sharpness metric. This observation suggests that combining these distinct perspectives on uncertainty leads to better uncertainty estimation while making sharper(more confident) predictions

Similarly, table 3 shows the point prediction performance of each method. We observe that the averaging method (IBUG+CBU) and the CBU method have the least error. Both base methods using CatBoost (CBU, IBUG on CatBoost)have similar results here as expected, since their difference (on point-predicting) is the loss function they optimise (RMSEwithUncertainty and RMSE). CBU is slightly better than simple IBUG methods since it uses an ensemble of models.

Table 2. Probabilistic performance comparison of each uncertainty quantification method.

Method	CRPS	NLL	RMSCE	Sharpness
CBU	16.62	**4.81**	**0.016**	31.6
IBUG on CatBoost	16.97	5.03	0.053	**29.7**
IBUG on XGBoost	18.21	4.91	0.061	40.8
IBUG + CBU	**16.31**	**4.79**	**0.016**	**29.8**

Table 3. Point prediction performance comparison of each method

Method	RMSE	MAE
CBU	**29.78**	**23.17**
IBUG on CatBoost	30.03	23.26
IBUG on XGBoost	32.84	25.39
IBUG + CBU	**29.85**	**22.85**

Overaly, we find that averaging IBUG on CatBoost and CBU(CatBoost with Uncertainty) methods generally outperform the other methods, but only marginally.

More diagrams about the uncertainty distribution and prediction accuracy resolution per energy can be found in the appendix.

5 Conclusion

Uncertainty estimation is crucial to understand a machine learning model's prediction weaknesses. IBUG wrapper and CatBoost with Uncertainty methods can effectively capture both aleatoric and epistemic uncertainty, but a combination of both methods is the optimal approach.

Limitations This work is based on using BDTs for energy reconstruction after multiple reconstruction steps. In the near future, sequential models like RNNs and LSTMs or even topology-specific models like GNNs could be used to predict the energy directly from charge-timing data. In this case, other methods for uncertainty quantification should be employed. A versatile model-agnostic uncertainty quantification method would be ideal given the fast pace of machine learning research.

Here we only discuss uncertainties as viewed by a machine learning model that is invariant to known systematic uncertainties. To enhance the performance of our models we should incorporate these uncertainties in the learning process to create uncertainty-aware models, as studied in [9].

References

[1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2021.05.008. URL https://www.sciencedirect.com/science/article/pii/S1566253521001081. 3

[2] I Anghel, JF Beacom, M Bergevin, C Blanco, E Catano-Mur, F Di Lodovico, A Elagin, H Frisch, J Griskevich, R Hill, et al. Letter of intent: The accelerator neutrino neutron interaction experiment (annie). *arXiv preprint arXiv:1504.01480*, 2015. 9

[3] Ivanna Baturynska and Kristian Martinsen. Prediction of geometry deviations in additive manufactured parts: comparison of linear regression with machine learning algorithms. *Journal of Intelligent Manufacturing*, 32, 01 2021. doi: 10.1007/s10845-020-01567-0. 2

[4] Jonathan Brophy and Daniel Lowd. Instance-based uncertainty estimation for gradient-boosted regression trees. *ArXiv*, abs/2205.11412, 2022. URL https://api.semanticscholar.org/CorpusID:248986210. 6, 12

[5] 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. ACM, 2016. 1

[6] Tony Duan, Avati Anand, Daisy Yi Ding, Khanh K Thai, Sanjay Basu, Andrew Ng, and Alejandro Schuler. Ngboost: Natural gradient boosting for probabilistic prediction. In *International conference on machine learning*, pages 2690–2700. PMLR, 2020. 1

[7] Jerome Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 10 2001. doi: 10.2307/2699986. 2

[8] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiao Xiang Zhu. A survey of uncertainty in deep neural networks, 2022. 5

[9] Aishik Ghosh, Benjamin Nachman, and Daniel Whiteson. Uncertainty-aware machine learning for high energy physics. *Physical Review D*, 104(5), September 2021. ISSN 2470-0029. doi: 10.1103/physrevd.104.056026. URL http://dx.doi.org/10.1103/PhysRevD.104.056026. 14

[10] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL https://doi.org/10.1198/016214506000001437. 6, 7

[11] Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery. Probabilistic Forecasts, Calibration and Sharpness. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 69(2):243–268, 03 2007. ISSN 1369-7412. doi: 10.1111/j.1467-9868.2007.00587.x. URL https://doi.org/10.1111/j.1467-9868.2007.00587.x. 6

[12] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks, 2017. 9

[13] Stephen C. Hora. Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. *Reliability Engineering & System Safety*, 54(2):217–223, 1996. ISSN 0951-8320. doi: https://doi.org/10.1016/S0951-8320(96)00077-4. URL https://www.sciencedirect.com/science/article/pii/S0951832096000774. Treatment of Aleatory and Epistemic Uncertainty. 3

[14] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110:457–506, 2021. 3, 4

[15] 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 I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf`. 1

[16] Balazs Kegl, CecileGermain, ChallengeAdmin, ClaireAdam, David Rousseau, Djabbz, fradav, Glen Cowan, Isabelle, and joycenv. Higgs boson machine learning challenge, 2014. URL `https://kaggle.com/competitions/higgs-boson`. 1

[17] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International conference on machine learning*, pages 2796–2804. PMLR, 2018. 9

[18] Dan Levi, Liran Gispan, Niv Giladi, and Ethan Fetaya. Evaluating and calibrating uncertainty prediction in regression tasks. *Sensors*, 22(15), 2022. ISSN 1424-8220. doi: 10.3390/s22155540. URL `https://www.mdpi.com/1424-8220/22/15/5540`. 6, 9

[19] Andrey Malinin, Liudmila Prokhorenkova, and Aleksei Ustimenko. Uncertainty in gradient boosting via ensembles. *arXiv preprint arXiv:2006.10562*, 2020. 5

[20] Michael Thomas Nieslony. *Towards a neutron multiplicity measurement with the Accelerator Neutrino Neutron Interaction Experiment*. PhD thesis, Mainz U., 11 2022. 10

[21] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 2015:2901–2907, 04 2015. doi: 10.1609/aaai.v29i1.9602. 6

[22] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf`. 1

[23] Byron P Roe, Hai-Jun Yang, Ji Zhu, Yong Liu, Ion Stancu, and Gordon McGregor. Boosted decision trees as an alternative to artificial neural networks for particle identification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 543(2-3):577–584, 2005. 1

[24] Xiao Wang, Hongrui Liu, Chuan Shi, and Cheng Yang. Be confident! towards trustworthy graph neural networks via confidence calibration, 2022. 9

[25] Hai-Jun Yang, Byron P Roe, and Ji Zhu. Studies of boosted decision trees for miniboone particle identification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 555(1-2):370–385, 2005. 1

A Supplemental material

CRPS formula explanation Source:link

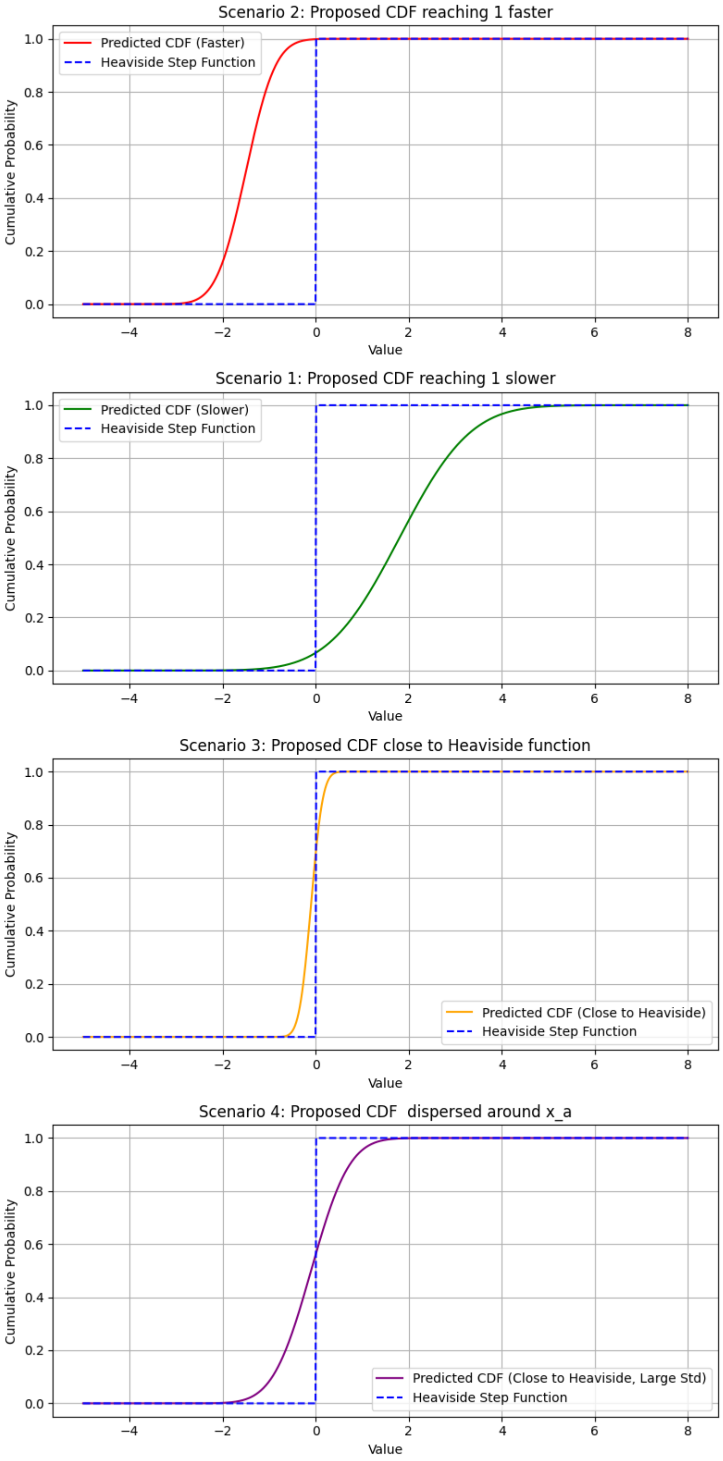
$$CRPS(P, x_a) = \int_{-\infty}^{\infty} \|P(x) - H(x - x_a)\|_2 dx, \tag{5}$$

where: x_a is the true value, x the predicted value, $P(x)$ the predicted cumulative distribution of x and $H(x)$ is the Heaviside step function $H(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$

The distribution that corresponds to a Heaviside CDF is the delta function $\delta(x - x_a)$. What this score is calculating is the difference between our distribution and a delta function. If we have a model that minimizes CRPS, then we are looking for a distribution that is close to the delta function. In other words, we want our distribution to be large around x_a.

To illustrate what the integrand $\|P(x) - H(x - x_a)\|_2$ means, we consider several scenarios, as seen in Fig. 21.

The shade areas between the two CDFs determine the integrand of the integral in CRPS. The only way to get a small score is to choose a distribution that is focused around x_a.



Scenario 2: Proposed CDF reaching 1 faster

Scenario 1: Proposed CDF reaching 1 slower

Scenario 3: Proposed CDF close to Heaviside function

Scenario 4: Proposed CDF dispersed around x_a

Figure 21. CRPS scenarios illustrations

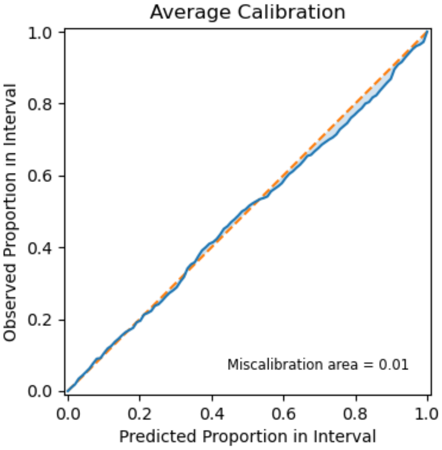
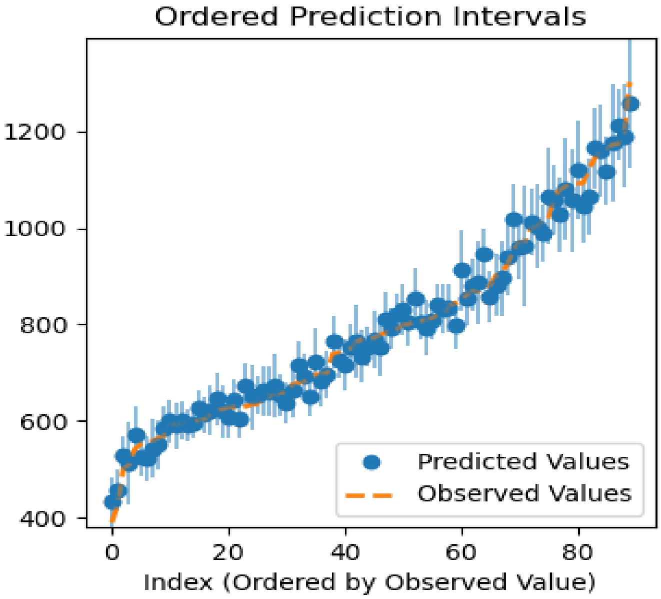
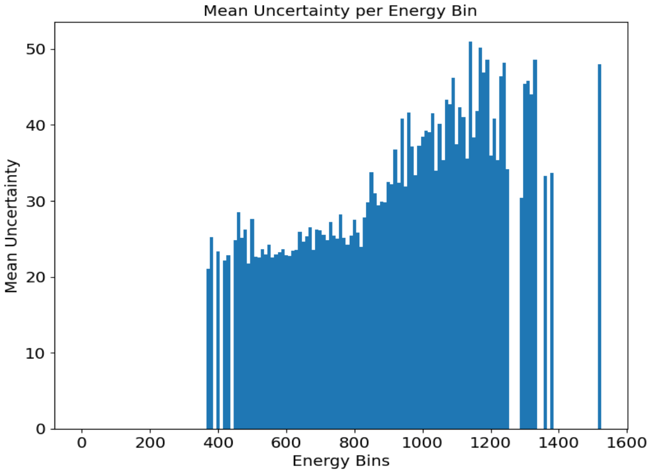


Figure 22. Reliability diagram of IBUG+CBU averaging method

How uncertainty looks in IBUG+CBU





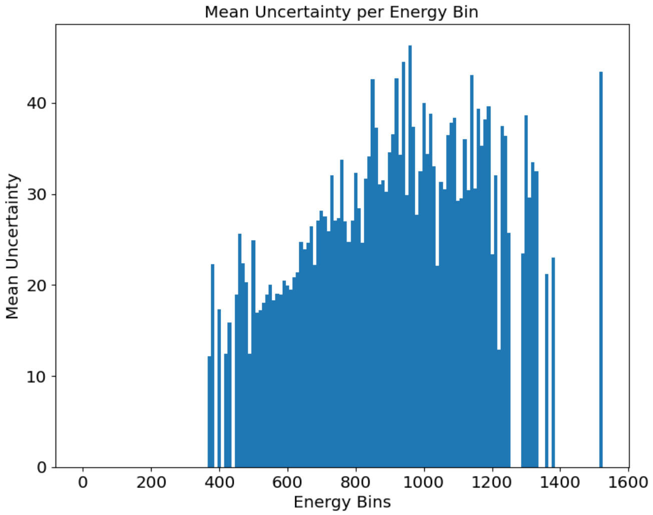


Figure 23. Uncertainty distribution per energy in IBUG on CatBoost

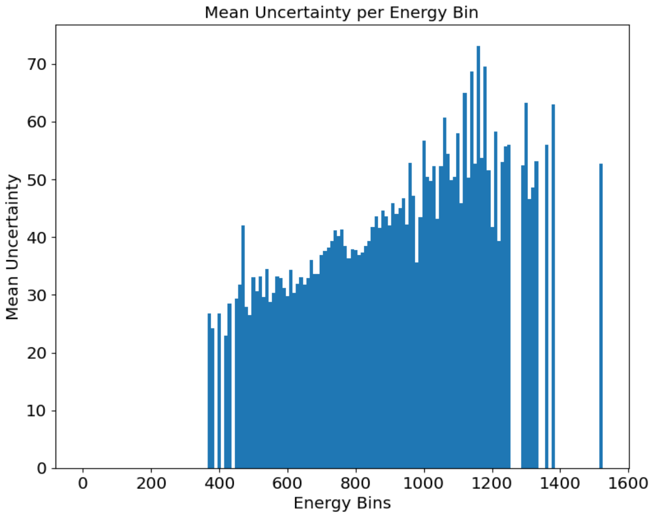


Figure 24. Uncertainty distribution per energy in IBUG on XGBoost

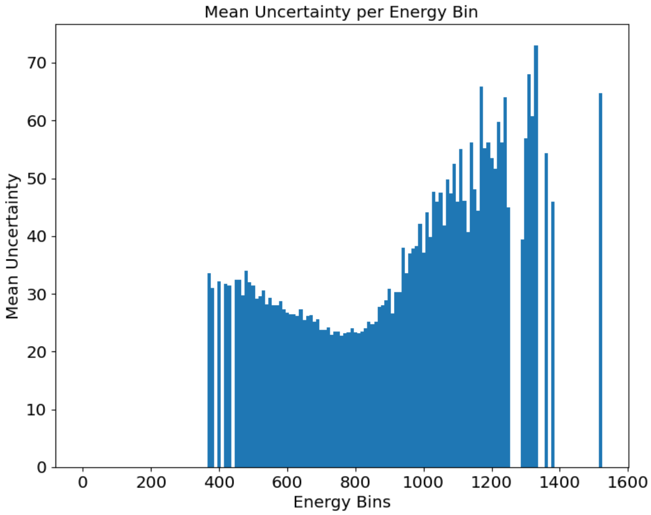


Figure 25. Uncertainty distribution per energy in CatBoost with Uncertainty (CBU)

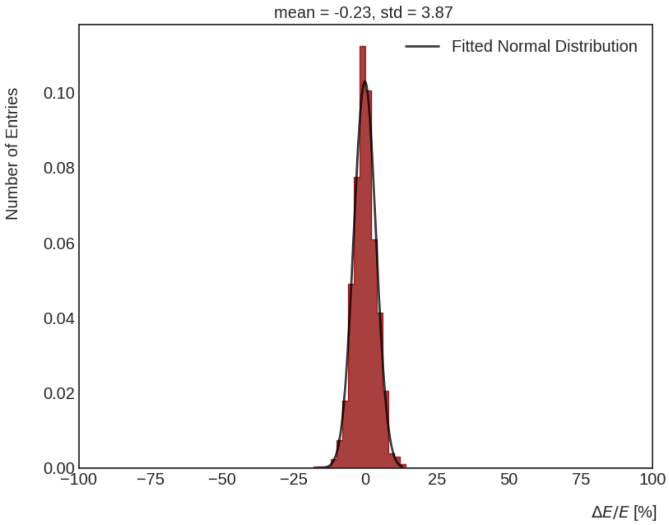


Figure 26

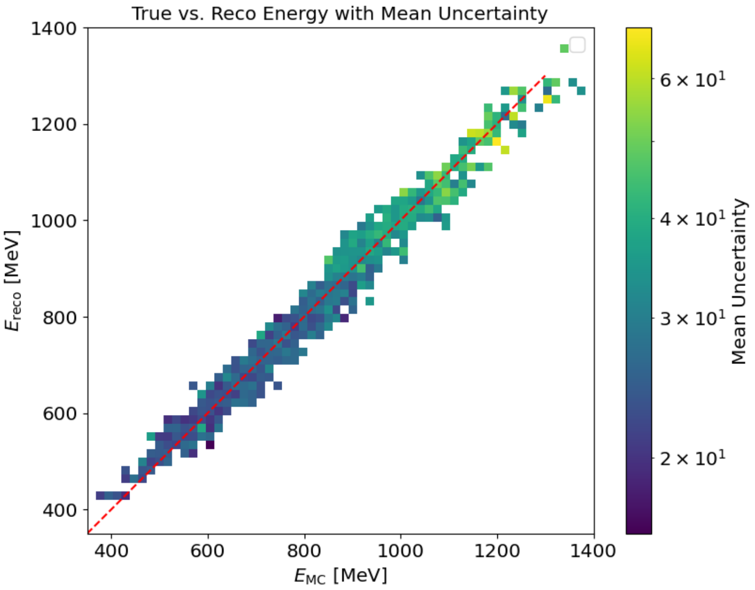

True vs. Reco Energy with Mean Uncertainty
E_{reco} [MeV]
E_{MC} [MeV]
400
600
800
1000
1200
1400
Mean Uncertainty
2×10^1
3×10^1
4×10^1
6×10^1

Figure 27