

# Active Learning for Regression Problems with Ensemble Methods

Bjarne Jaster, Martin Kohlhase

Hochschule Bielefeld, Center for Applied Data Science Gütersloh

Interaktion 1, 33619 Bielefeld

E-Mail: {bjarne.jaster, martin.kohlhase}@hsbi.de

## 1 Introduction

Traditional machine learning paradigms depend on the availability of labeled data, a luxury that is not often the reality in real-world scenarios. In domains such as industry, healthcare, autonomous systems and finances a massive amount of unlabeled data is produced every day. As the demand for accurate and robust models to deal with this data grows, the inefficiency and the cost of manual labeling motivates the research field active learning [1].

Active learning describes an efficient and effective way of selecting the most valuable data samples for labeling. The value of a sample is defined by a criterion which is unique to each active learning strategy. This selection reduces the amount of manual labeling, optimizes resource allocation, and enhances model performance in real-world scenarios. This makes active learning a pivotal tool for domains where labeled data is scarce or costly. A common criterion for active learning method is the informativeness of a data sample. This is measured by the uncertainty of a Machine Learning (ML) model in its prediction [2]. The uncertainty estimation is dependent on the ML-model, which motivates this work to explore the quality of different ML-models and their uncertainty estimation methods for active learning. Additionally, the computational effort of different uncertainty estimators is explored, because there often is a trade-off between the accuracy and the computational effort of an uncertainty estimation method. For instance, probabilistic techniques such as *Fully Bayesian*

*Gaussian Processes (FB-GPs)* and *Bayesian Neural Networks* provide accurate uncertainty estimates at the cost of computational complexity due to parameter sampling from an intractable distribution, commonly done with *Markov Chain Monte Carlo (MCMC)* Sampling [3]. *Neural Networks* and their softmax-layer probabilities, on the other hand, provide less accurate and often overconfident uncertainty estimates [4] but add none additional computational effort.

## 2 Related Work

Active learning describes the process of selecting unlabeled data samples. An active learner is characterized by a predefined budget, representing the quantity of data samples it can actively select, and a criterion quantifying the value of a given data sample. The active learner chooses the most valuable data samples, measured by the criterion, and requests labels for these selections. The criterion typically relies on either the spatial distribution of data samples [5] to maximize the training set diversity or utilizes uncertainty estimates [2] to minimize regions in the input-space characterized by high predictive uncertainty. Leveraging uncertainty estimates requires the use of a ML-model capable of quantifying predictive uncertainty. By iteratively selecting new samples, the active learner enhances its performance, often achieving better results with fewer labeled examples compared to traditional passive learning methods.

The research field of active learning is divided into three subfields [6]: *Pool-based Sampling* is the most common subfield of active learning. It involves selecting instances for labeling from a fixed pool of unlabeled data. The algorithm ranks instances within this pool based on its selection criterion. The selected instances are then labeled and added to the training set. *Membership Query Synthesis* involves generating label-queries synthetically based on its current knowledge, instead of selecting instances from an existing dataset. These queries are designed to be informative and help the model to learn more effectively. *Stream-based Selective Sampling* focuses on scenarios where data arrives in a continuous stream, and labeling resources are limited. In this subfield, the active learning algorithm processes data instances one by one as

they arrive. It decides on-the-fly whether to label the current instance or wait for a more valuable one based on the selection criterion. This work focuses on the pool-based sampling approach to active learning.

The concept of uncertainty, which can be used as an active learning criterion, is commonly divided into two parts: Epistemic uncertainty refers to a systematic uncertainty and results from incomplete or missing knowledge. This uncertainty is caused by factors like small data sets and other influences arising from an incomplete and potentially faulty data source, as well as incomplete knowledge about the process being modeled. Aleatoric uncertainty represents the part of uncertainty that cannot be reduced, including statistical relationships such as noise or fundamentally random connections in the data [7]. When using uncertainty as a criterion for active learning, especially the epistemic uncertainty is of interest, because data samples with high epistemic uncertainty carry the information that can increase the performance of the model [8].

## 2.1 Gaussian Processes

One machine learning method that provides an uncertainty estimate is a *Gaussian Process (GP)*. GPs are a good model choice for active learning because they are well-suited for smaller datasets, a crucial characteristic due to the limited amount of training samples during the active learning process. However, there are two issues associated with GPs, when used for active learning. Firstly, they rely on proper hyperparameter selection, which poses a challenge as the costly labeling of data samples makes it difficult to justify withholding samples for testing and hyperparameter tuning. Consequently, hyperparameters must often be chosen based on heuristics or expert knowledge, which may not be available or applicable in all cases. Secondly, GPs' uncertainty estimate do not differentiate between epistemic and aleatoric uncertainty.

These issues have motivated Riis et al. [2] to explore the use of FB-GPs for active learning. The concept involves sampling the hyperparameters, commonly noise and lengthscale, from a posterior distribution conditioned on the training samples. This directly addresses the first issue and enables the creation of an ensemble of GPs. Combined with the law of total variance  $V(y|x) =$

$V(E[y|x]) + E[V(y|x)]$ , this approach allows for the decomposition of total uncertainty into epistemic  $V(E[y|x])$  and aleatoric uncertainty  $E[V(y|x)]$ . This results in a significantly more accurate and useful uncertainty estimate for active learning.

## 2.2 Random Forests

*Random Forests* are a widely used ensemble learning technique, that uses multiple decision trees to make accurate and robust predictions. They are able to model diverse datasets without requiring extensive hyperparameter tuning [9]. Additionally, [10] shows that they are the best ML-method for small to medium sized real-world datasets. These features make Random Forest a good candidate for an ML-model in active learning problems.

In [11] different ways to estimate the uncertainty of a prediction of a Random Forest are described. First, they argue that the standard approach of estimating uncertainty for an ensemble, taking the variance of the individual predictions, is not suitable for Random Forest. This is due to the different training sets and feature selections in the individual trees. Then, two suitable methods for estimating the uncertainty are presented: The first method, denoted as the *Jackknife Estimate*, calculates the Leave-One-Out Error implicitly [12] and uses its average as the uncertainty. This is done by computing the difference between the average prediction made by trees not trained on a particular sample and the average prediction generated by the entire ensemble. The second method, referred to as the *Infinitesimal Jackknife estimator* [13], introduces a novel approach. It down-weighs each training sample by an infinitesimal amount and computes the variance of a prediction over all training samples. The variance can be interpreted as the uncertainty. To enhance the reliability and accuracy of uncertainty estimates from both the Jackknife and the Infinitesimal Jackknife estimator, the authors present unbiased versions of these methods. These enhancements correct the inherent upward bias observed in the initial estimations, ultimately resulting in better-calibrated uncertainty assessments. These estimates do not distinguish the two parts of uncertainty. Although a method exists to differentiate between aleatoric and epistemic uncertainty

specifically for Random Forests [14], it is designed for classification tasks and does not easily extend to regression problems.

## 2.3 Neural Networks

*Neural Networks* are widely used models, especially for large datasets [15]. They also provide probabilities in their predictions when trained on a classification problem, which can be used as an uncertainty estimate. However, the uncertainty (entropy of the predicted class probabilities) of these predictions is both poorly calibrated [4] and not applicable to regression problems, as Neural Networks trained on regression problems only provide point predictions. One approach to obtain uncertainty estimates is by training an ensemble of Neural Networks [16], each initialized with different weights. This results in different Neural Networks converging to various local minima of the loss function. However, training multiple Neural Networks can be computationally expensive. Thus, implicit ensembling techniques that require the training of only a single Neural Network and yield well calibrated uncertainty estimates are presented.

The first technique, known as *Dropout*, is a well-established regularization technique applied during the training [17]. Dropout operates by randomly setting the output of a fraction of neurons in each layer to zero with a specific probability, effectively excluding their contribution to the networks output. Importantly, this random dropout of neurons varies during each training iteration, encouraging the development of redundant representations and mitigating the risk of overfitting. Notably, during testing or inference, Dropout is deactivated to allow the model to utilize its full predictive capability. However, when Dropout is retained and applied during testing, it can be demonstrated that the mean and variance of multiple forward passes approximate the behavior of Bayesian Neural Networks [18]. The underlying concept of the uncertainty estimation is that the model has formed redundant representations for test samples closely aligned with the training data, resulting in a lower variance in the predictions. Conversely, for samples more distant from the training data, the model lacks this redundancy, leading to a higher variance in predictions as they become heavily reliant on specific neurons.

Table 1: Synthetic Datasets from [23]

Name	No. of Features	Noise	Input Space
Gramacy1d	1	0.1	$[0.5, 2.5]$
Higdon	1	0.1	$[0, 20]$
Gramacy2d	2	0.01	$[-2, 6]^2$
Branin	2	11.32	$[-5, 10] \times [0, 15]$
Ishigami	3	0.187	$[-\pi, \pi]^3$
Friedman	5	0.1	$[0, 1]^5$
Hartmann	6	0.01	$[0, 1]^6$

The second uncertainty estimation method utilizes *DropConnect*, which is conceptually similar to Dropout. Initially developed as a regularization technique for Neural Networks [19] too, DropConnect sets connections between neurons to zero, instead of the output of the neurons. The authors of [20] experimentally show that the variance of multiple DropConnect-Neural Networks predictions provides better calibrated uncertainty estimates than the variance of a Dropout-Neural Network.

The third method, *Local Ensembles* [21], approximates the variance of an ensemble of equally competent predictors without explicitly constructing the ensemble. To achieve this, the weights of the Neural Networks are perturbed in the direction of the smallest eigenvectors of the Hessian of the loss function. These eigenvectors represent directions of low curvature, indicating flat regions on the loss landscape, where weight-perturbations have minimal impact on the loss. To efficiently approximate these eigenvectors, *Lanczos iteration* is employed [22].

## 3 Experimental Procedure

### 3.1 Datasets

We use a comprehensive set of datasets, comprising seven synthetic functions (Tab. 1) and eleven real-world datasets (Tab. 2). Six of the seven synthetic

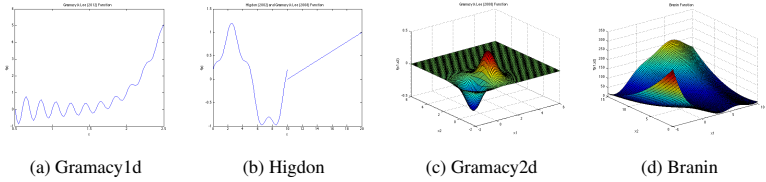


Figure 1: Visualizations of the 1d and 2d synthetic functions (taken from [23])

Table 2: Real World Datasets from [26]

Name	No. of Samples	No. of Features
auto-mpg	397	7
concrete-data	1030	8
cps-wages	534	18
housing	452	13
no2	500	7
pm10	397	7
real-estate-valuation	414	6
slump-test-slump	103	7
slump-test-flow	103	7
slump-test-compressive-strength	103	7
winequality-red	1599	11
winequality-white	4898	11
yacht-hydrodynamics	308	6

functions, described in [23], are employed in benchmarking FB-GP based active learning methods [2]. These synthetic functions serve to demonstrate the models’ capabilities in addressing common challenges: The function *Gramacy1d* (Fig. 1a) assesses the models’ ability to distinguish noise from signal. *Higdon* (Fig. 1b) and *Gramacy2d* (Fig. 1c) feature both linear and non-linear regions. The other synthetic functions are of higher dimension and some are characterized by strong non-linear behavior, enabling the evaluation of the models performance in complex scenarios. The *Friedman* function has a well-established reputation in regression problems, having been previously employed by Friedman et al. [24] and Breiman [25]. The datasets for every

function are created by drawing 2000 samples uniformly from the defined input space and by adding Gaussian white noise to the function output.

While synthetic datasets offer valuable insights into specialized problem scenarios, the evaluation of active learning methods on real-world datasets is also relevant, because real-world dataset either combine multiple synthetic scenarios or exhibit behavior not covered with synthetic functions. To this end, eleven real-world datasets that are consistent with those used by Wu et al. [5] are incorporated. These datasets are sourced from the UCI and CMU StatLib repository [26]. An overview over the datasets and their number of data samples and features is given in Tab. 2

The inputs and outputs of all datasets are normalized to have zero mean and standard deviation one. Categorical features are one-hot-encoded for compatibility with the used ML-models. In the *slump-test* dataset, three target values are present. To deal with that each target is viewed as an individual dataset.

## 3.2 Active Learning Methods

Our evaluation focuses on three primary ML-models: FB-GPs, Random Forests, and Neural Networks. For FB-GPs, three distinct uncertainty estimation approaches are employed:

- Mean of the predicted variances: Represents aleatoric uncertainty
- Variance of the predicted means: Represents epistemic uncertainty
- Combination of the mentioned criteria: Represents total uncertainty

Random Forests are evaluated using the Jackknife and Infinitesimal-Jackknife estimators, along with their bias-corrected counterparts. Neural Networks undergo assessment with three presented uncertainty estimation methods: Dropout, DropConnect, and Local Ensembles. Additionally, a passive learning baseline for each machine learning method is provided, which randomly selects the same amount of data samples as the active learning methods.

Each method starts with an initial training set. The size of this set is equal to the dimensionality of the dataset. In case of a 1d- or 2d-dataset it consists



of three samples. The samples of the initial set are selected as follows: The first sample is selected as the one closest to the mean of the input of the data. Subsequently, the remaining initial training samples are chosen based on their maximum distance to their nearest training sample. This is done to provide a deterministic initial training set with good diversity to the ML-models. This way, the starting conditions are equal and not dependent on random selection of initial data samples. Next, we train an ML-model and compute uncertainties for each unlabeled data sample. The sample with the highest uncertainty is then added to the training set, and the ML-model is retrained. This selection process is iterated 50 times.

### 3.2.1 Training of ML-Methods

Hyperparameter tuning for machine learning methods typically relies on a separate test or validation set. However, in the active learning scenario with limited data availability, this approach is impractical. Therefore, hyperparameters are chosen based on established rules of thumbs or taken from other work that dealt with similar problems.

For FB-GPs, the hyperparameters lengthscale and noise are sampled from a distribution, requiring the user to specify only the ensemble size, which is set to 800 to manage computational resources. For Random Forests, hyperparameters such as tree size and the number of input variables considered in each split are configured in alignment with Breiman’s original Random Forests paper [9], which shows minimal sensitivity to the second hyperparameter. Neural Networks depend on a multitude of hyperparameters, with the network-size being the most critical. Given the limited number of training samples in active learning (at most 50 to 70), we chose one-hidden-layer networks with 50 neurons, inspired by Tohme [27]. The hyperbolic tangent is used as the activation function, and networks are trained with the ADAM optimizer (learning-rate: 0.01). To promote stability of the training results, network weights from the previous active learning iteration are used for initialization.

To mitigate the risk of overfitting, regularization techniques are employed. Specifically, Dropout and DropConnect are incorporated, which are already

integrated into their respective uncertainty estimation methods. Dropout is also applied to the Neural Networks that are used for the Local Ensembles-method. The proportion of neurons or connections that are dropped is set to 0.05, similar to findings from [17]. Additionally, we implement early stopping based on training error to accelerate the training process when convergence is reached. This approach is particularly advantageous when combined with the weight initialization from the preceding iteration, as the model is likely near a local minimum.

### 3.3 Evaluation Process

In other work the quality of an active learning method is commonly assessed by providing learning-curves, which illustrate the final performance of the ML-model as well as the speed and the stability of the learning process. Due to space constraints we only show the most important metric, which is the quality of the ML-model after the final amount of training samples is acquired. For regression problems, the quality is commonly assessed by the Root Mean Square Error (RMSE). We use the normalized-RMSE which allows comparisons over multiple dataset:

$$\text{normalized-RMSE} = \frac{\sqrt{\sum_{i=1}^n (y_i - \hat{y}_i)^2}}{\text{Var}(y)}$$

The normalized-RMSE is computed over the remaining unlabeled data samples from the pool, demonstrating the generalization capabilities of the model. The normalized-RMSE is also computed for the passive learning baselines, which randomly select the same amount of training samples as the active learning methods. This allows for an evaluation of the uncertainty based selection criterion. All active and passive learning methods are run ten times per dataset.

In addition to comparing the quality of the different active learning methods, the computational effort of each method is considered. The average time it takes to acquire the final amount of data samples per method is calculated, ensuring uniform hardware conditions for all experiments to allow a comparison.

Table 3: normalized-RMSE after the final active learning iteration on real-world datasets, averaged over 10 runs. The best method for each dataset is printed bold. The last row shows the average score over all datasets (RF - Random Forest, NN - Neural Network)

	FB-GP		RF		NN	
	Random	Best	Random	Best	Random	Best
auto-mpg	0.18	<b>0.12</b>	0.18	0.16	0.23	0.25
concrete	0.28	<b>0.25</b>	0.40	0.48	0.33	0.41
cps-wages	1.04	0.90	0.85	<b>0.78</b>	1.06	1.04
housing	0.27	0.12	0.17	<b>0.08</b>	0.29	0.39
no2	0.74	<b>0.60</b>	0.64	0.65	0.72	0.74
pm10	1.06	0.88	0.84	<b>0.81</b>	1.11	1.12
real-estate	0.46	<b>0.38</b>	0.41	<b>0.38</b>	0.59	0.60
slump-slump	0.83	0.58	0.75	<b>0.45</b>	0.69	0.76
slump-flow	0.63	0.51	0.63	<b>0.48</b>	0.67	0.68
slump-strength	0.01	<b>0.00</b>	0.38	0.49	0.12	0.10
wine-red	0.93	0.91	0.76	<b>0.75</b>	0.98	1.11
wine-white	0.97	0.94	<b>0.80</b>	0.83	1.04	1.23
yacht	0.01	<b>0.00</b>	0.19	0.05	0.05	0.03
<i>average</i>	0.57	0.48	0.54	0.49	0.61	0.65

While assessing complexity using  $\mathcal{O}$ -notations is of interest, it poses challenges for the MCMC-methods, which are used for FB-GPs, as shown in [28].

## 4 Results

### 4.1 Real-World Datasets

Tab. 3 shows the normalized-RMSE of the model after the final active learning iteration for the real-world datasets. For every ML-method the normalized-RMSE of the passive learning baseline (Random) and the best uncertainty estimator per dataset are shown. This is done to make the results more clear and to enable the comparison of the maximum potential of each ML-model for active learning. In the last row the average RMSE per method is shown for an overall comparison.

Table 4: Average RMSE per uncertainty estimator on real-world datasets (J - Jackknife estimator, IJ - Infinitesimal Jackknife estimator)

FB-GP				
aleatoric	epistemic	total	Random	
0.57	0.51	0.49	0.57	
RF				
J	unbiased J	IJ	unbiased IJ	Random
0.52	0.52	0.51	0.51	0.54
NN				
DropConnect	Dropout	LocalEnsemble	Random	
0.74	0.73	0.69	0.61	

The average score illustrates a comparable performance between active learning with Random Forests and FB-GPs, while active learning with Neural Networks performs worst. The RMSE per dataset also reinforces this observation, as for no dataset the Neural Networks perform best, whereas Random Forests performs best roughly on the same amount of datasets as FB-GPs. The performance of both methods is also very similar for every datasets, with two major exceptions: For the *concrete* and the *slump-strength* dataset the FB-GPs outperform the Random Forests severely. This could be a result of poor hyperparameter choice or because Random Forests are a bad model choice for those particular datasets. Additionally, the random baseline (passive learning) of Random Forests performs better in both cases, indicating sub-optimal uncertainty estimates unable to pinpoint regions benefiting from additional data samples. This trend of poor uncertainty estimates is further evident for Neural Networks, where, barring three datasets, the passive learning consistently outperforms the best active learning strategy. Because the papers introducing these methods show that they produce reliable uncertainty estimates, this suggests the inadequacy of the use of these uncertainty estimators for active learning. A potential explanation for this unsuitability could be the general poor generalization of the Neural Networks due to suboptimal hyperparameter selection. However, to proof the hypothesis that a poorly fitted model causes poor uncertainty estimates, dedicated experiments need to be conducted.

Tab. 4 shows the average scores per uncertainty estimator for the real-world datasets. This enables the comparison between the different uncertainty estimators of the specific ML-models. The top section emphasizes the advantage of distinguishing the uncertainty into its epistemic and aleatoric components, because active learning with FB-GPs based on the epistemic uncertainty demonstrates better results than the aleatoric-based approach. Interestingly, the total uncertainty criterion proves most effective for active learning, even though both aleatoric and epistemic uncertainties contribute equally.

Comparing random baselines indicates Random Forests as the optimal model choice for real-world datasets, consistent with findings by [10]. However, FB-GP uncertainty estimates exhibit a greater performance increase (compared to the passive learning baseline) than Random Forests. This indicates that active learning with Random Forests could benefit from a better uncertainty estimate that differentiates between epistemic and aleatoric uncertainty.

For Random Forests one can see that the unbiased versions of the Jackknife and Infinitesimal Jackknife do not achieve increased performance regarding active learning compared to the non-bias-corrected versions. This is an expected result as the bias correction is achieved by dividing the uncertainty estimate by a constant. For active learning, the data sample with the highest uncertainty is added to the training data, which does not change when all values are divided by a constant. The difference between the Jackknife and the Infinitesimal Jackknife is not significant on average.

The uncertainty estimator for Neural Networks are all, as already mentioned, not suitable for active learning without further research and adaptation. The average results show that Local Ensembles outperform Dropout and Drop-Connect, which have similar results, which is explainable due to their similar nature.

## 4.2 Synthetic Datasets

The outcomes obtained from the synthetic datasets are presented in Tab. 5. They showcase substantial distinctions from the results observed in real-world datasets. Notably, Random Forests and Neural Networks demonstrate similar

Table 5: normalized-RMSE after the final active learning iteration on synthetic datasets, averaged over 10 runs. The best method for each dataset is printed bold. The last row shows the average score over all datasets.

	FB-GP		RF		NN	
	Random	Best	Random	Best	Random	Best
Gramacy1d	0.07	<b>0.02</b>	0.05	0.05	0.08	0.09
Higdon	0.06	<b>0.04</b>	0.06	0.05	0.09	0.09
Gramacy2d	0.46	<b>0.02</b>	0.62	0.79	0.79	1.01
Branin	0.13	<b>0.08</b>	0.37	0.50	0.16	0.31
Ishigami	<b>0.42</b>	0.54	0.57	0.60	0.67	0.91
Friedman	0.04	<b>0.02</b>	0.33	0.60	0.22	0.28
Hartmann	0.58	<b>0.46</b>	0.86	0.81	1.16	0.86
<i>average</i>	0.25	0.17	0.41	0.49	0.45	0.51

performances, while active learning based on FB-GPs significantly outperforms both. FB-GPs exhibit superior performance for each dataset, solidifying their efficacy. Furthermore, the passive learning approach outperforms the active learning approaches for Random Forests and Neural Networks, indicating poor uncertainty estimates. Conversely, active learning with FB-GPs achieves superior results compared to passive learning for all datasets except the *Ishigami* dataset, which is characterized by strong non-linearities. This suggests that complex problems for which an ML-Model achieves poor generalization, may introduce uncertainty estimates that are not useful for active learning.

Analyzing the passive learning scores shows that, for the synthetic datasets, FB-GPs emerge as the optimal model choice. This deviates from the results observed with real-world datasets. This difference can be attributed to distinct characteristics of the synthetic datasets, notably the presence of homoscedastic noise and the continuous nature of their underlying functions. These features inherently favor FB-GPs, as they are able to model continuous output well due to their probabilistic distribution over functions. Additionally, FB-GPs employ a single noise parameter for the entire input space, predicated on the assumption of homoscedastic noise. While this assumption contributes to strong performance when homoscedastic noise prevails, it poses challenges in

Table 6: Average RMSE per uncertainty estimator on synthetic datasets

FB-GP				
aleatoric 0.34	epistemic 0.17	total 0.19	Random 0.25	
RF				
J 0.53	unbiased J 0.59	IJ 0.56	unbiased IJ 0.52	Random 0.41
NN				
DropConnect 0.69	Dropout 0.56	LocalEnsemble 0.58	Random 0.45	

effectively modeling heteroscedastic noise. However, it is essential to note that these favorable characteristics of synthetic datasets are not necessarily given for real-world datasets, requiring further experiments to validate the hypothesis that the performance of FB-GPs relies on the characteristics of the dataset.

The average results per uncertainty estimator (Tab. 6) strengthen the suitability of epistemic uncertainty as an active learning criterion over aleatoric uncertainty, with the latter performing worse than passive learning. In contrast to the real-world datasets, the total uncertainty does not result in a performance improvement. The uncertainty estimator performance for Random Forests and Neural Networks deviates from the results observed on real-world datasets. For Random Forests, the Infinitesimal Jackknife yields better results, with a slight advantage for the unbiased version, while the unbiased version of the Jackknife estimator outperforms the standard version. In the previous section, we argued that the unbiased and non-bias-corrected uncertainty estimates for Random Forests should theoretically yield similar results. However, due to the highly stochastic nature of Random Forests, caused by bootstrapping and random feature selection, the sample size of ten active learning runs might not suffice to ensure comparable results. This is further underscored by the observation that, for the Jackknife estimator, the unbiased version performs worse than the biased one, whereas for the Infinitesimal Jackknife estimator, the unbiased version fares better. If a significant difference in performance resulted from the bias correction, one would expect the difference to be consistent across both

Table 7: Average Time per uncertainty estimator in seconds

	FB-GP	RF	NN
Real-World	2851	6	137
Synthetic	1579	9	202

estimators, which is not the case. Regarding Neural Networks, DropConnect performs notably worse, while Dropout and Local Ensembles deliver roughly equivalent performance.

### 4.3 Computational Effort

As emphasized in the introduction, computational efficiency is as well an aspect as predictive performance when comparing various active learning methods. Tab. 7 outlines the time in seconds required to select a specific quantity of data samples, in this case fifty, averaged across all active learning runs for each respective ML-model. The results are quite apparent, with active learning using Random Forests proving to be the fastest. In contrast, Neural Networks are approximately 20 times slower, and FB-GPs are notably slower, ranging from approximately 100-500 times slower depending on the dataset. Although these results may vary with the amount of parallelization or optimized implementations, the trend of the computational effort for the different ML-methods is evident given the substantial differences observed.

The difference between the real-world and synthetic datasets is shown in Tab. 7, because it is notable with approximately 50%. For Random Forests and Neural Networks, the synthetic datasets are slower, whereas for FB-GPs, the real-world datasets display slower computation. This difference can be attributed to the pool size; the synthetic datasets comprise 2000 data samples, exceeding the size of most real-world datasets. The ML-model’s predictions are important for uncertainty estimation, and processing becomes slower with a larger pool size. While this holds true for FB-GPs, their slower performance on real-world datasets is primarily due to the higher dimensionality of these datasets. The FB-GPs sample one lengthscale-parameter for each input dimension resulting



in more parameters drawn by the overall time-consuming process of MCMC-sampling.

## 5 Conclusion and Future Work

Our work compares active learning strategies with three Machine Learning models and different uncertainty estimates for them. We apply a comprehensive set of datasets with real-world and synthetic problems to present individual strengths and weaknesses of the ML-models and their uncertainty estimates.

A key results is the superior reliability of uncertainty estimates provided by the FB-GPs. Their ability to differentiate aleatoric and epistemic uncertainty contributes to a high-quality active learning performance without the need for hyperparameter selection. The excellent results on different synthetically created problems, like noise-signal differentiation, makes them a good model choice when time and computational resources are not of the essence. Random Forests demonstrate performance comparable to FB-GPs across real-world datasets and present a significant speed advantage over them. Additionally, Random Forests are robust regarding the choice of hyperparameters, an important feature for active learning models due to the expensive or not feasible tuning process. This makes them the best model-choice for efficient active learning on real-world data. Conversely, Neural Networks are the least favorable ML-model, emphasizing the crucial role of hyperparameters that are challenging to select heuristically and ultimately impact the uncertainty estimation quality. Despite their relatively efficient processing, they fall behind Random Forests in terms of computational efficiency.

Moreover, our investigation underscores the importance of epistemic uncertainty for active learning. This motivates further research particularly for uncertainty estimates of Random Forests as there is to the best of our knowledge no method that differentiate epistemic and aleatoric uncertainty for regression-problems. Lastly, further research into the interplay between model characteristics and dataset features is motivated by the fact that FB-GPs excel significantly on synthetic data.

## Acknowledgements

The author acknowledges financial support by the project “SAIL: SustAInable Life cycle of Intelligent Socio Technical Systems” (Grant ID NW21 059B), which is funded by the program “Netzwerke 2021” of the Ministry of Culture and Science of the State of Northrhine Westphalia, Germany.

## References

- [1] Tharwat, A. & Schenck, W. Balancing Exploration and Exploitation: A novel active learner for imbalanced data. *Knowledge-Based Systems*. 210 pp. 106500 (2020,12)
- [2] Riis, C., Antunes, F., Boe, F., Carlos, H., Azevedo, L. & Pereira, F. Bayesian Active Learning with Fully Bayesian Gaussian Processes. *Advances In Neural Information Processing Systems*. pp. 12141-12153 (2022)
- [3] Lampinen, J. & Vehtari, A. Bayesian approach for neural networks - review and case studies. *Neural Networks*. 14, 257-274 (2001)
- [4] Gal, Y. & Others Uncertainty in deep learning. (PhD thesis, University of Cambridge,2016)
- [5] Wu, D., Lin, C. & Huang, J. Active Learning for Regression Using Greedy Sampling. *Information Sciences*. 474 pp. 90-105 (2019)
- [6] Tharwat, A. & Schenck, W. A Novel Low-Query-Budget Active Learner with Pseudo-Labels for Imbalanced Data. *Mathematics*. 10, 1068 (2022,3)
- [7] Hüllermeier, E. & Waegeman, W. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*. 110, 457-506 (2021,3)

- [8] Nguyen, V., Destercke, S. & Hüllermeier, E. Epistemic Uncertainty Sampling. *Discovery Science: 22nd International Conference, DS 2019*. pp. 72-86 (2019,10)
- [9] Breiman, L. Random Forests. *Machine Learning*. 45 pp. 5-32 (2001)
- [10] Fernández-Delgado, M., Cernadas, E., Barro, S. & Amorim, D. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?. *Journal Of Machine Learning Research*. 15 pp. 3133-3181 (2014)
- [11] Wager, S., Hastie, T. & Efron, B. Confidence Intervals for Random Forests: The Jackknife and the Infinitesimal Jackknife. *Journal Of Machine Learning Research*. 15 pp. 1625-1651 (2014)
- [12] Efron, B. Jackknife-after-bootstrap standard errors and influence functions. *Journal Of The Royal Statistical Society Series B: Statistical Methodology*. 54, 83-111 (1992)
- [13] Efron, B. Estimation and Accuracy After Model Selection. *Journal Of The American Statistical Association*. 109, 991-1007 (2014,7)
- [14] Shaker, M. & Hüllermeier, E. Aleatoric and Epistemic Uncertainty with Random Forests. *Advances In Intelligent Data Analysis XVIII*. pp. 444-456 (2020,4)
- [15] Sharma, P. & Singh, A. Era of deep neural networks: A review. *2017 8th International Conference On Computing, Communication And Networking Technologies (ICCCNT)*. pp. 1-5 (2017,7)
- [16] Lakshminarayanan, B., Pritzel, A. & Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *31st Conference On Neural Information Processing Systems (NIPS 2017)*. (2017)
- [17] Srivastava, N., Hinton, G., Krizhevsky, A. & Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal Of Machine Learning Research*. 15 pp. 1929-1958 (2014)

- [18] Gal, Y. & Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *33rd International Conference On Machine Learning, ICML 2016*. 3 pp. 1651-1660 (2015,6)
- [19] Wan, L., Zeiler, M., Zhang, S., Lecun, Y. & Fergus, R. Regularization of Neural Networks using DropConnect. *International Conference On Machine Learning*. pp. 1058-1066 (2013)
- [20] Mobiny, A., Yuan, P., Moulik, S., Garg, N., Wu, C. & Nguyen, H. DropConnect is effective in modeling uncertainty of Bayesian deep networks. *Scientific Reports*. 11, 5458 (2021,3)
- [21] Madras, D., Atwood, J. & D'Amour, A. Detecting Extrapolation with Local Ensembles. *International Conference On Learning Representations*. (2019)
- [22] Lanczos, C. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. (United States Governm. Press Office Los Angeles, CA,1950)
- [23] Surjanovic, S. & Bingham, D. Virtual Library of Simulation Experiments: Test Functions and Datasets. (Retrieved September 12, 2023, from <http://www.sfu.ca/ssurjano>)
- [24] Friedman, J. Multivariate adaptive regression splines. *The Annals Of Statistics*. 19, 1-67 (1991)
- [25] Breiman, L. Bagging predictors. *Machine Learning*. 24 pp. 123-140 (1996)
- [26] Kelly, M., Longjohn, R. & Nottingham, K. The UCI Machine Learning Repository. , <https://archive.ics.uci.edu>
- [27] Tohme, T., Vanslette, K. & Youcef-Toumi, K. Reliable neural networks for regression uncertainty estimation. *Reliability Engineering & System Safety*. 229 pp. 108811 (2023,1)

- [28] Matamoros, I. An introduction to computational complexity in Markov Chain Monte Carlo methods. *ArXiv Preprint ArXiv:2004.07083*. (2020,4)