CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	BHO-MA: Bayesian Hyperparameter Optimization with Multi-objective Acquisition
Authors	Dogan, Vedat;Prestwich, Steven
Publication date	2024-02-01
Original Citation	Dogan, V. and Prestwich, S. (2024) 'BHO-MA: Bayesian Hyperparameter Optimization with Multi-objective Acquisition', in Pereira, A. I., Mendes, A., Fernandes, F. P., Pacheco, M. F., Coelho, J. P. and Lima, J. (eds) Optimization, Learning Algorithms and Applications. OL2A 2023. Communications in Computer and Information Science, vol 1981, pp 391-408. Springer, Cham. https://doi.org/10.1007/978-3-031-53025-8_27
Type of publication	Conference item
Link to publisher's version	10.1007/978-3-031-53025-8_27
Rights	© 2024 The Author(s), under exclusive license to Springer Nature Switzerland AG. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. - https://creativecommons.org/licenses/by/4.0/
Download date	2026-11-05 14:40:09
Item downloaded from	https://hdl.handle.net/10468/16861



UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

BHO-MA: Bayesian Hyperparameter Optimization with Multi-objective Acquisition*

Vedat Dogan[1][0000−0002−3807−5425] and Steven Prestwich[2][0000−0002−6218−9158]

[1] Confirm Centre for Smart Manufacturing
School of Computer Science and Information Technology, University College Cork, Ireland
vedat.dogan@cs.ucc.ie
[2] Insight Centre for Data Analytics
School of Computer Science and Information Technology, University College Cork, Ireland
s.prestwich@cs.ucc.ie

Abstract. Good hyperparameter values are crucial for the performance of machine learning models. In particular, poorly chosen values can cause under- or overfitting in regression and classification. A common approach to hyperparameter tuning is grid search, but this is crude and computationally expensive, and the literature contains several more efficient automatic methods such as Bayesian optimization. In this work, we develop a Bayesian hyperparameter optimization technique with more robust performance, by combining several acquisition functions and applying a multi-objective approach. We evaluated our method using both classification and regression tasks. We selected four data sets from the literature and compared the performance with eight popular methods. The results show that the proposed method achieved better results than all others.

Keywords: Bayesian Optimization · Multi-objective Optimization · Hyperparameter Tuning

1 Introduction

Machine learning (ML) algorithms have been used in various application domains such as recommender systems, natural language processing, behaviour analytics, and computer vision [18]. The construction of an effective ML model is a time-consuming and complex process. Effective ML models have achieved great success in a wide range of practical problems but the performance depends on correctly setting the many parameters. We can categorize the ML parameters considering the setting procedure. Model *parameters*, such as neural network weights, are learned during the training process. In contrast, *hyperparameters* (HPs), such as the number and size of neural layers, must be set before training. Tuning HPs is a crucial step when building an ML model. There is a wide range of possibilities that need to be explored to build an effective and optimal ML model. The process of HP tuning differs among different ML models according to their different types of HP, which can be categorical, discrete, or continuous. A traditional approach

* This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 16/RC/3918.

is manual tuning, which requires expertise and a deep understanding of the chosen ML algorithms. It is not effective for most problems because of the high number of HPs, model complexity, the time-consuming nature of evaluations, and the convexity of HP interactions. This has motivated research in the field of hyperparameter optimization (HPO), which aims to find optimal HPs for a given model. The main goal of HPO is to build automated systems and allow non-expert users to effectively apply ML models to practical problems. It also reduces human effort, as ML experts often spend a great deal of time on HP tuning, especially when working with complex ML models. Thus HPO improves the performance of ML models, as HPs might have different optima for different problems.

Many optimization techniques have been applied successfully to HPO problems. Bayesian optimization (BO) is an iterative global optimization method to optimize expensive-to-evaluate black-box functions [17]. BO uses a surrogate model, typically Gaussian process (GP) [25], and an acquisition function to evaluate future points based on previous observations. While the surrogate model aims to fit all current observations into the objective function and obtain a probability distribution, the acquisition function determines the next optimal evaluation by balancing exploration and exploitation. It works by updating the surrogate model after each iteration until reaching a near-optimal solution. BO can usually detect near-optimal HP configurations within a few iterations [11]. Several BO approaches focused on surrogate models have been developed for HPO. Besides GP, commonly used surrogate models include Random forest (RF) [33] and the tree Parzen estimator (TPE) [12]. As we shall discuss in Section 2, there has been much recent progress in the HPO field and the methods have strengths and weaknesses according to their constraints. However, surprisingly few works have focused on improving the next point selection process by modifying the acquisition function of the HPO problem.

The main contribution of this paper is to combine the strengths of several methods to obtain more robust results. Ensemble learners [8] have proven to be effective in a wide variety of ML applications, especially when diverse methods are combined. We apply this idea to BO by using more than one acquisition function, to obtain a more intelligent search strategy for hyperparameter optimization. As stated in [14] no single acquisition function always outperforms others. We propose a new Bayesian hyperparameter optimization method using a multiobjective acquisition function, which we call BHO-MA. Our proposed BHO-MA method seeks robust selection by trading off acquisition functions with different search strategies for hyperparameter optimization tasks. To the best of our knowledge, this idea has not previously been applied to HPO. Our contributions can be considered as three-fold. First, we use multiple acquisition functions for hyperparameter tuning and solve the multi-objective problem with a genetic algorithm to obtain the best trade-off. Second, we show empirically how using the multi-acquisition approach affects HPO performance and obtains near-optimal configurations. Also, BHO-MA provides another insight into the benefit of combining multi-objective optimization and preference learning literature, as there is no single optimal solution for multiple objectives. It provides the user with the freedom of choosing the next selection according to his/her preference after the Pareto-front is obtained and explored by the automated machine learning regime while optimizing the hyperparameters of the preferred ML model.

The rest of the paper is organized as follows. Related works on HPO are surveyed in Section 2. The preliminaries of multi-objective problems and Bayesian optimization are given in Section 3. BHO-MA method is explained in Section 4. In Section 5 the experimental setup is described, and Section 6 follows with results and discussions. Finally, Section 7 concludes the paper and proposes future work.

2 Related Work

Several methods have been successfully applied to the HPO problem over the years. Grid Search (GS) [2] and Random Search (RS) [3] are relatively simple decision-theoretic approaches. While RS samples hyperparameters randomly from a configuration space, GS evaluates the Cartesian product of a set of values for each hyperparameter. Neither scales well to large-scale search spaces, and each HP configuration is treated independently. Because of computational resource limitations, multi-fidelity optimization algorithms are improved to solve these problems. Hyperband [20] is a bandit-based optimization technique that generates small data sets and sets the specific budget for each HP combination. It can be considered an improved version of RS. Metaheuristic techniques have been successfully applied to large-scale, complex HPO problems, including genetic algorithm (GA) [32] and particle swarm optimization (PSO) [22]. They can locate near-optimal solutions well but are expensive in terms of function evaluations. Bayesian optimization (BO) determines the next sample of HP value based on previously observed HPs [21], thus reducing function evaluations compared to other methods. BO uses the distribution of the objective function as the surrogate function. Examples of surrogates are Gaussian process (BO-GP), random forest (BO-RF) [33], and tree-structured Parzen estimators (BO-TPE) [12].

In recent years, the research community has focused on building systems that continuously improve over time, especially in automated HPO tasks [16]. In some works, reinforcement learning is successfully employed in solving HPO problems [36, 9]. Meta-learning techniques received increased attention in recent works by combining other optimization techniques for HPO problems [7, 24, 15]. Applying HPO techniques to tune hyperparameters of ML models can greatly improve their performance on practical problems. [6] worked on the health estimation problem of lithium-ion batteries and improved the estimation accuracy by optimizing convolutional neural network (CNN) hyperparameters with BO. Another HPO application of CNN hyperparameters can be found in [29]. They worked on the facial emotion recognition problem and improved the algorithm performance by applying the RS algorithm to the HPO problem. HPO of artificial neural networks was used by [5] for designing software sensors. For the prediction problem of passengers at a metro station using deep neural networks, [28] explored different HPO techniques such as GD and PSO. In [13] heuristic techniques for the HPO problem worked well for deep learning models for genomic prediction problems.

We believe BO is a very promising global optimization method, especially after the interesting work of [31] for applying BO in high-dimensional space. In this work, we focus on improving BO by using a multi-objective acquisition approach. We shall describe our BHO-MA method and compare it with various HPO methods in the literature.

3 Preliminaries

We now provide some necessary background.

3.1 Multi-objective Optimization

Multi-objective optimization (MOO) aims to optimize a vector-valued objective function $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^n$ over a bounded set $\mathcal{X} \subset \mathbb{R}^d$. There are multiple objectives to optimize when we consider the multi-objective optimization problems. It is formulated as

$$\underset{\mathbf{x} \in X}{\text{minimize}} \;\; \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_n(\mathbf{x})). \tag{1}$$

Because of different possible trade-offs between objectives, there is no single best solution. The main goal is to approximate the *Pareto-optimal solution set* which is a set of optimal objective trade-offs. The *Pareto-optimal* solution set is a set of non-dominated solutions that trade-off objectives. Let us say that $\mathbf{f}(\mathbf{x})$ *dominates* another solution $\mathbf{f}(\mathbf{x}')$ if $\mathbf{f}^{(i)}(x) \succ \mathbf{f}^{(i)}(x')$ for all $i = 1, 2, \ldots, M$ and there exists $i' \in \{1, 2, \ldots, M\}$ such that $f^{i'}(x) \succ f^{i'}(x')$. So we can express the *Pareto-optimal* by $P^* = \{\mathbf{f}(\mathbf{x})$ *s.t.* $\nexists \mathbf{x}' \in X : \mathbf{f}(\mathbf{x}') \succ \mathbf{f}(\mathbf{x})\}$ and $X^* = \{\mathbf{x} \in \mathbf{X}$ *s.t.* $\mathbf{f}(\mathbf{x}) \in P^*\}$. A solution set is Pareto-optimal if it is not dominated by any other point and it dominates at least one point. The Pareto-set the set of all Pareto-optimal points, and a set of Pareto-optimal points is called a Pareto-front. The existence of multiple feasible optima, namely the Pareto-front, provides freedom to the decision-maker to choose an objective trade-off according to their preferences. *Hypervolume* is a metric for measuring the quality of the frontier which is bounded from below by a reference point that bounds a region of interest in objective space [30]. It is commonly assumed to be known by the decision-maker.

3.2 Bayesian Optimization

Bayesian optimization (BO) is a method to optimize expensive-to-evaluate black-box functions. BO uses a probabilistic surrogate model, typically Gaussian Process (GP) [25], $p(f|\mathcal{D})$ to model the objective function f based on previously observed data points $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$. GPs are models that are specified by a mean function $\mu(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta) : \mathbb{R}^d \to \mathbb{R}$ and predictive variance function $\sigma(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ where $\mathbf{x} \in \mathbb{R}^d$. Surrogate model $p(f|\mathcal{D})$ is assisted by an acquisition function $\alpha : \mathcal{X} \to \mathbb{R}$. We represent an acquisition function depending on the previous observations by $\alpha(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta)$ where θ is Gaussian parameters such as kernel for the model. Because the objective function is expensive to evaluate and the surrogate-based acquisition function is not, it can be optimized much more quickly than the true function to yield $\mathbf{x}_{new}$. Gaussian process uses the acquisition function to select the $\mathbf{x}_{new}$ Then it evaluates the objective function $y_{new} = f(\mathbf{x}_{new})$ and updates the data set with new observations $\mathcal{D} \leftarrow \mathcal{D} \cup (\mathbf{x}_{new}, y_{new})$. It continues until the budget criterion is met.

4 Method

We now describe our proposed method which uses multi-objective acquisition optimization to take advantage of different search strategies of different acquisition functions during Bayesian optimization. First, we will present the problem statement and then the adaptation of multi-objective acquisition optimization to hyperparameter optimization.

4.1 Problem Statement

Let us assume that we have an ML algorithm $\mathcal{A}$ with M hyperparameters. If we denote the m-th hyperparameter by Λ_m, the configuration space can be denoted by $\mathbf{\Lambda} = \Lambda_1 \times \Lambda_2 \times \ldots \times \Lambda_M$. Also, define the set of hyperparameters as a vector and denote by $\lambda \in \mathbf{\Lambda}$ and $\mathcal{A}_\lambda$ represents the algorithm $\mathcal{A}$ with hyperparameters λ. For a given data set $\mathcal{D}$, our goal is to find [16]:

$$\lambda^* = \underset{\lambda \in \mathbf{\Lambda}}{\operatorname{argmin}}\ \mathbb{E}_{(\mathcal{D}_{tr}, \mathcal{D}_{vl}) \sim \mathcal{D}} \mathbf{V}(\mathcal{L}, \mathcal{A}_\lambda, \mathcal{D}_{tr}, \mathcal{D}_{vl}), \tag{2}$$

where $\mathbf{V}(\mathcal{L}, \mathcal{A}_\lambda, \mathcal{D}_{tr}, \mathcal{D}_{vl})$ represents the loss of the model generated by algorithm $\mathcal{A}$ with HPs λ on training data $\mathcal{D}_{tr}$ and evaluated on validation data $\mathcal{D}_{vl}$.

4.2 Acquisition Functions

There are several popular choices of acquisition functions. Different acquisition functions have different characteristics according to their structure and point selection strategy. Improvement-based strategies rely on the best selection so far at each iteration. For example, the probability improvement (PI) function value decreases when the difference between the mean function and the best objective value is far below zero. The expected improvement (EI) function value at sampled points would always be worse than the EI values at pending decision points. Uncertainty-based acquisition functions, for instance, upper-confidence-bound (UCB), increase as $\sigma(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta)$ increases. The formulations of the acquisition functions used in this work are provided as follows. $\Phi(\cdot)$ will denote the cumulative distribution function of the standard normal, and $\phi(\cdot)$ will denote the standard normal density function.

Probability of Improvement (PI) The strategy of the PI acquisition function is to maximize the probability of improvement over the best current value. Given the minimum objective function value τ in the data set, the formulation is given in Equation 3 as in [19].

$$\alpha_{PI}(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta) = \Phi(\gamma(\mathbf{x})), \quad \gamma(\mathbf{x}) = \frac{\tau - \mu(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta)}{\sigma(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta)} \tag{3}$$

Expected Improvement (EI). The EI strategy maximizes the expected improvement over the current best observation [27]. The corresponding formulation can be expressed as in Equation 4.

$$\begin{aligned} \alpha_{EI}(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta) = \sigma(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta)(\gamma(\mathbf{x})\Phi(\gamma(\mathbf{x})) + \phi(\gamma(\mathbf{x}))), \\ \gamma(\mathbf{x}) = \frac{\tau - \mu(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta)}{\sigma(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta)} \end{aligned} \tag{4}$$

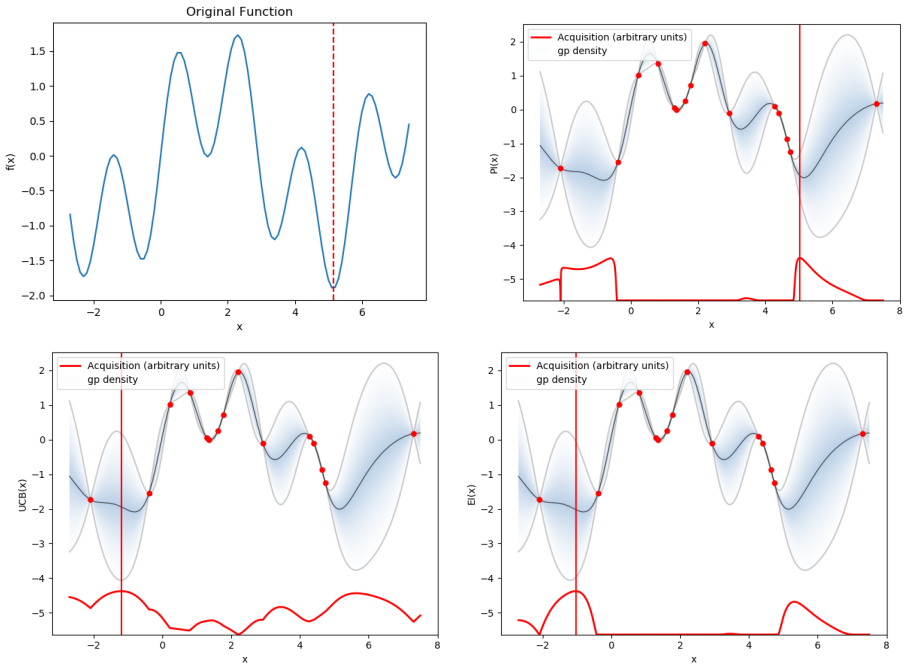


Fig. 1. Illustration of the difference of acquisition function search strategies.

Upper Confidence Bound (UCB). This is not an improvement-based strategy like EI and PI. Instead, it tries to guide the search from an optimistic perspective. The formulation is given in Equation 5,

$$\alpha_{UCB}(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta) = \mu(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta) + \beta\sigma(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta) \tag{5}$$

where β is a parameter that represents the exploration-exploitation trade-off. We fix $\beta = 0.1$.

4.3 Multi-objective Acquisition Adaptation to Hyperparameter Optimization

For an illustration of the acquisition function search strategy difference, we use a multimodal problem in Fig. 1. Multimodal problems can have more than one global optimal point with several local optima. The red line on the top-first graph represents the optimal point. The PI, UCB and EI acquisition function strategies can be seen on the other three graphs. As shown in Fig. 1, different acquisition functions suggest different candidates after initializing on 15 data points. While the PI acquisition function is trying to make the next decision around global optima, the EI and UCB acquisition functions are exploring the other points at the current iteration.

In the proposed BHO-MA method, we use the MOO approach during the next point selection process, according to the different selection strategies explained above

Algorithm 1 The Bayesian optimization algorithm via multi-objective acquisition function for HPO task.

Initialize the number of iterations *N*. Initialize the size of sample size *n* for beginning the algorithm. Select random hyperparameters $\lambda_1, \ldots, \lambda_m$ with the size of m from the configuration space. Obtain the evaluation results of the loss function of the selected ML model $f(\lambda_1), \ldots, f(\lambda_m)$ that is defined in Equation 2.

1: Construct *Gaussian model* with the observations
2: **for** i = 0 : *N* **do**
3: Construct the acquisition functions according to the Equation 6
4: Find non-dominated solutions of multi-objective acquisition problem using the NSGA-II algorithm
5: Make a random selection from the non-dominated solution set
6: Calculate the evaluation results of the loss function
7: Update Gaussian Process model with new observations
8: **end for**
9: **Return** Best hyperparameter λ^* and corresponding evaluation results

and in Section 4.2. We aim to find the best trade-off between acquisition functions with different search strategies. This is called the Pareto-front solutions of acquisition functions. Acquisition functions are the key point in Bayesian optimization. It is expected to be cheap to solve compared with the true objective function, and easier to optimize. It also includes information on uncertainty and estimated values everywhere in the search space. So modifying the acquisition optimization step on the Gaussian process improves the global optimization process significantly. This is the main reason that the literature has worked on creating various acquisition functions such as q-knowledge gradient (qKG) [34], and predictive entropy search (PES) [26]. Each acquisition function may not agree to sample the same points as each of them has a different search strategy. Reconstructing the acquisition function with multiple acquisition functions gains us the information to use the benefits of different acquisition search strategies.

For the purpose declared above, we constructed the data set $\mathcal{D}$ with the finite collection of hyperparameters $\lambda_1, \lambda_2, \ldots, \lambda_n \in \mathbf{\Lambda}$ and $f(\lambda_1), f(\lambda_2), \ldots, f(\lambda_n) \in \mathbb{R}$ is the evaluation results of the loss function defined in Equation 2, where n represents the number of initial data points. Let us define the variable $\mathbf{x}_n = [\lambda_1, \ldots, \lambda_n]$ and $y_n = [f(\lambda_1), \ldots, f(\lambda_n)]$ with the GP parameters θ, the optimization problem of the multi-objective acquisition function is constructed as follows:

$$\min_{\mathbf{x} \in \mathcal{X}} \begin{Bmatrix} -\alpha_{EI}((\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta)) \\ -\alpha_{PI}((\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta)) \\ -\alpha_{UCB}(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta) \end{Bmatrix} \tag{6}$$

Solving MOO problems is computationally more difficult and time-consuming. But as acquisition functions are easy-to-solve functions, the multi-objective structure of the problem does not have a significant effect on algorithmic performance, especially when the learner being tuned is applied to a large problem. The objective of HPO is to minimize the number of executions of the learner. However, wall-clock times are provided in Table 4.

Instead of a random selection of acquisition functions during the optimization process, the Pareto-front allows us to use trade-offs between these different search strategies. It allows the decision maker to pick the most preferred solutions from this set. The advantage is the decision maker does not have to specify the preferences before optimization, which is an acquisition function in BO literature. The MOO problem defined in Equation 6 is solved by the popular NSGA-II algorithm [10]. The NSGA-II algorithm is a successful algorithm for solving MOO problems. At each iteration, we constructed the multi-objective acquisition function by using Equation 6. We constructed the population of 100 with current acquisition values and terminated the algorithm after 100 generations. After we obtained the final population, we used random sampling for the next iteration. The details can be found in Algorithm 1.

5 Experiments

This section presents experiments applying the popular HPO techniques on ML algorithms on two benchmark data sets. Firstly, the experimental setup is discussed. Secondly, the results of the BHO-MA are compared with those of other popular methods and analyzed.

5.1 Experimental Setup

We evaluated BHO-MA's performance on four tasks selected from the sklearn library [23]. Firstly, the Modified National Institute of Standards and Technology (MNIST) and IRIS data sets are selected for the classification problem. MNIST is a hand-written digit recognition data set which has 1797 instances with 64 attributes, used for multi-classification problems. Iris data set is a classification data set which has 150 instances with 4 attributes, that contain three classes and each class refers to a type of iris plant. The Boston housing and diabetes data sets are chosen as regression problem benchmarks where the Boston housing includes 506 instances with 13 attributes while diabetes data set has 442 instances with 10 attributes.

Support vector machines (SVM), K-nearest neighbour (KNN), and random forest (RF) are three ML algorithms that are chosen as targets for HPO problems according to the type of their hyperparameters. They can be applied to regression and classification problems and are popular in the ML community to solve practical problems. We implemented 3-fold cross-validation to evaluate the HPO methods for each experiment. For the classification problems, the accuracy metric is selected for performance comparison while the mean squared error (MSE) is chosen for the regression problems. Accuracy measures the proportion of correctly classified data. MSE calculates the average squared difference between predicted and actual values. The highest accuracy and the lowest MSE value are returned for each experiment. The maximum number of iterations for all HPO methods is set to 50 for the selected ML models. Thus, all learners are tuned based on 50 executions, for a fair comparison with the results presented in [35]. All experiments are repeated 10 times with different random seeds, and report the average results with standard deviations. We compared our results with 8 different popular approaches in the literature. We presented the details of our approaches in Section 2. The algorithm implementation is made by considering the settings mentioned in Section 5.1.

Table 1. Configuration space for the HPs of tested ML models.

Model	Hyper-parameter	Type	Search Space
RF Classifier	n_estimators	Discrete	[10,100]
	max_depth	Discrete	[5,50]
	min_samples_split	Discrete	[2,11]
	min_samples_leaf	Discrete	[1,11]
	criterion	Categorical	['gini', 'entropy']
	max_features	Discrete	[1,64]
SVM Classifier	C	Continuous	[0.1,50]
	kernel	Categorical	['linear', 'poly', 'rbf', 'sigmoid']
KNN Classifier	n_neighbors	Discrete	[1,20]
RF Regressor	n_estimators	Discrete	[10,100]
	max_depth	Discrete	[5,50]
	min_samples_split	Discrete	[2,11]
	min_samples_leaf	Discrete	[1,11]
	criterion	Categorical	['mse','mae']
	max_features	Discrete	[1,13]
SVM Regressor	C	Continuous	[0.1,50]
	kernel	Categorical	['linear', 'poly', 'rbf', 'sigmoid']
	epsilon	Continuous	[0.001,1]
KNN Regressor	n_neighbors	Discrete	[1,20]

5.2 Parameters

All experiments are conducted on a single core of 1.4 GHz Quad Core i5, 8Gb 2133 MHz LPDDR3 RAM. We constructed the selected ML models using the *sklearn* library [23] by using the parameters in Table 1. To make the comparison fair, we used 3-fold cross-validation for each experiment. The number of iterations is selected as the budget for each ML model. For the stopping criterion, the default tolerance value 10^{-3} is used for all algorithms. For scoring the final performance, *neg_mean_squared_error* is used for regression and *accuracy* is used for classification tasks. We set the *n_jobs* parameter to -1 cause being able to use the all processors. Other parameters are left as default.

The *BoTorch* library is used for Bayesian optimization [1]. The method initialized 20 Sobol points to construct the initial GP model. *MaternKernel* is used to compute the covariance matrix with parameter $\nu = 2.5$ and other parameters taking default values. After constructing the GP model, the acquisition functions are initialized with the parameters below. The current best selection is important to the EI acquisition function to select the next point successfully. So in each iteration, the selections are analyzed and set to the *best_f* parameter of EI. The selection of β is important for UCB acquisition function tuning and we set it to 0.1. The PI acquisition function is constructed with the default hyperparameters as well.

After we obtained the multi-objective acquisition function with the parameters above, we used a genetic algorithm called Non-dominated Sorting Genetic Algorithm (NSGA-II) to solve the MOO problem. For the NSGA-II algorithm [10] we used the *PyMOO* library [4]. *pop_size* is selected as 100 at each iteration and *n_gen* as 100. We

set *eliminate_duplicates=True* to discard unnecessary function evaluations. After we obtained the Pareto-optimal solution set, we used a random selection process.

Pre-processing is important during the machine learning process, but we did not focus on it for our experiments as it is not our main aim.

6 Results and Discussion

We compared our results with those of 8 different HPO methods in the literature and present our results in Table 3 and Table 2. We also shared the wall-clock time comparison of the best performances of all Bayesian-related algorithms in Table 4. It can be seen that our algorithm is relatively slower than others from Table 4. We believe the main reason is its use of MOO during acquisition optimization. However, the difference will be negligible when the classification or regression tasks involve big data. The aim of BO is to optimise an objecting while minimising the number of black-box function evaluations, and a modest increase in runtime is not a drawback Please see Fig. 2 and Fig. 3 for more details of experimental results.

The KNN search space of HPs has discrete, RF has categorical and continuous, and SVM has discrete and categorical variables. The SVM regressor's parameters are the regularization parameter C, the kernel type, and ϵ which determines the width of the tube around the estimated function (hyperplane). The selected parameter for the KNN regressor is the discrete-valued *n_neighbors* as it is the most important parameter for this ML model. The RF algorithm has the largest parameter space. The mean value of accuracy and MSE scores are presented with the standard deviations of ten runs. We can see that using the default HP configurations does not yield the best model performance in our experiments which shows that optimizing hyperparameters are crucial.

The Models Performance on Boston-housing Dataset. In this experiment, we evaluated the BHO-MA performance for optimizing the three selected regression ML models for predicting housing prices. We experimented with the BHO-MA method for optimizing the KNN, RF and SVM regressors. As we can see from the MSE score which is shared in Table 2, the proposed BHO-MA method for optimizing the ML models selected from the literature reached the minimum and the best score compared to the other algorithms. The closest algorithm to optimize the regressor's parameters is BOHB which combines the Bayesian optimization and Hyperband method. As the black-box approach works well with low-dimensional feature spaces, it makes sense to have similar results with them. In terms of the whole performance analysis of the selected three ML algorithms, the best result was obtained from the RF model which has the minimum MSE score compared with all. It also has the biggest parameter spaces including discrete and categorical variables. Also, we can observe that all of the methods outperformed GS at the end of the budget. It is clear that the BHO-MA technique for the HPO problem worked well among the other techniques for the Boston housing data set for the regression task.

The Models Performance on MNIST. The popular MNIST is an image-processing data set that is created for recognizing handwritten digits. It has been worked on with various classification algorithms over the years. In our experiment, we evaluated the accuracy of

Table 2. The performance of optimizing KNN, RF and SVM algorithms for regression and classification tasks on Boston-housing and MNIST data sets over 10 different runs.

Hyperparameter Optimization Algorithm	**Regressor's MSE Mean ± Std.** (Boston-housing data set)			**Classifier's Accuracy (%) Mean ± Std.** (MNIST data set)		
	KNN	RF	SVM	KNN	RF	SVM
BHO-MA	**80.75±0.0300**	**25.24±0.2100**	**58.45±0.0217**	**97.60±0.0264**	**93.96±0.0412**	**97.45±0.0032**
BOHB	80.77±0.1322	25.56±0.7812	59.67±1.1484	97.44±0.0034	93.38±0.0054	97.44±0.0136
GA	80.77±0.0014	26.95±0.4355	60.17±0.8733	96.83±0.0003	93.83±0.0032	97.44±0.0004
Hyperband	80.87±0.2654	26.14±0.5471	73.44±3.4956	96.22±0.0034	93.38±0.0005	97.44±0.0023
BO-GP	80.99±0.3365	26.44±0.6812	59.70±1.1694	96.54±0.0024	93.35±0.0075	97.40±0.0002
BO-TPE	81.06±0.3137	26.38±0.6258	62.09±4.2994	96.29±0.0025	92.92±0.0072	97.41±0.0003
PSO	81.01±0.2835	26.73±0.4541	60.26±1.0675	95.54±0.0024	92.46±0.0058	97.39±0.0001
GS	81.49±0.0004	28.06±0.3970	67.07±0.0030	96.82±0.0122	93.60±0.0015	97.38±0.0142
RS	80.99±0.2731	26.80±0.4438	60.88±2.0768	96.64±0.0021	92.98±0.0046	97.40±0.0004
Default HPs	81.48±0.0010	29.57±0.3762	77.43±0.0008	96.27±0.0014	92.97±0.0363	96.99±0.0018

Table 3. The performance of optimizing KNN, RF and SVM algorithms for regression and classification tasks on Diabetes and Iris data sets over 10 different runs.

Hyperparameter Optimization Algorithm	**Regressor's MSE Mean ± Std.** (Diabetes data set)			**Classifier's Accuracy Mean ± Std.** (Iris data set)		
	KNN	RF	SVM	KNN	RF	SVM
BHO-MA	**3190.66±0.0003**	**3074.17±2.1634**	**3030.71±12.626**	**0.9893±0.0046**	**0.9683±0.0028**	**0.9819±0.0002**
BOHB	3198.38±5.3243	3103.67±6.5465	3047.37±14.722	0.9733±0.0087	0.9667± 0.0017	0.9801±0.0065
GA	3193.54±2.5412	3099.82±3.6587	3039.88±1.4529	0.9766±0.0023	0.9666±0.0054	0.9803±0.0006
Hyperband	3194.07±1.4476	3098.32±5.4284	3038.68±2.4354	0.9820±0.0043	0.9670±0.0089	0.9810±0.0002
BO-GP	3197.04±7.4451	3112.87±22.979	3049.30±21.657	0.9806±0.0058	0.9660±0.0021	0.9813±0.0042
BO-TPE	3195.07±7.9000	3120.69±18.835	3056.60±26.128	0.9801±0.0062	0.9644±0.0066	0.9770±0.0035
PSO	3194.09±3.6494	3121.81±9.6953	3052.89±5.4383	0.9813±0.0061	0.9613±0.0048	0.9803±0.0031
GS	3241.33±0.0000	3102.64±0.0000	3040.91±0.0000	0.9802±0.0000	0.9666±0.0000	0.9733±0.0000
RS	3200.33±19.332	3135.33±26.122	3048.28±10.857	0.9840±0.0046	0.9646±0.0032	0.9806±0.0037
Default	3619.27±0.0000	3354.91±46.426	3895.66±0.0000	0.9805±0.0000	0.9633±0.0035	0.9609±0.0000

three ML models explained in Section 5. The accuracy of the KNN, RF and SVM models is shared as a percentage the Table 2. As we can see from Table 2, the BHO-MA method reached the highest accuracy compared with the other algorithms implemented. The second best HPO algorithm for KNN and SVM is BOHB and GA for RF. As the RF has the largest parameter space, it is understandable that the genetic algorithm works better than the black-box approach in this experiment. The worst algorithm performance is with the default HPs for all algorithms which shows the importance of the HPO problem in ML literature. Among all ML algorithms, the KNN classifier got the best accuracy.

The Models Performance on Diabetes Dataset. The diabetes data set includes information about patients who have diabetes, such as age, sex, blood pressure, etc. We used the data set for the regression problem and the same ML models for the HPO task. We shared the MSE score in Table 3 with the standard deviation over ten runs. As we can see, the proposed BHO-MA method optimized the ML models better than other HPO methods. The best ML model is SVM which has the minimum score compared with RF

Table 4. The best surrogate-assisted accuracy results and wall-clock time duration (s) for regression and classification tasks on all data sets over 10 different runs.

Hyperparameter Optimization Algorithm	**Regressor's MSE Mean ± Std.** (Boston-housing data set)		**Classifier's Accuracy (%) Mean ± Std.** (MNIST data set)		**Regressor's MSE Mean ± Std.** (Diabetes data set)		**Classifier's Accuracy (%) Mean ± Std.** (Iris data set)	
	RF	Time (s)	KNN	Time (s)	SVM	Time (s)	KNN	Time (s)
BHO-MA	**25.24±0.2100**	6.1	**97.60±0.0264**	7.6	**3030.71±12.626**	10.8	**0.9893±0.0046**	11.1
BOHB	25.56±0.7812	**4.9**	97.44±0.0034	3.4	3047.37±14.722	9.8	0.9733±0.0087	6.9
Hyperband	26.14±0.5471	6.2	96.22±0.0034	4.8	3038.68±2.4354	**7.6**	0.9820±0.0043	**5.4**
BO-GP	26.44±0.6812	12.3	96.54±0.0024	**2.6**	3049.30±21.657	9.4	0.9806±0.0058	7.9
BO-TPE	26.38±0.6258	5.9	96.29±0.0025	3.7	3056.60±26.128	11.3	0.9801±0.0062	2.3

and KNN. The genetic algorithm and hyperband method seem the second and third-best HPO methods to optimize the HPs. We found the worst performance with SVM using default HPs. As we can observe from the difference between the best and the worst score from all experiments, the proposed HPO method significantly improves the final MSE score.

The Models Performance on Iris Dataset. The iris data set is relatively small compared with other data sets selected for the experiments. It is maybe the best-known data set in pattern recognition literature and a multi-class classification data set, and we used it for classification problems with the same three ML models to optimize their HPs. The target is classifying the iris plants according to predictive attributes. The accuracy for the classification problem is shared in Table 3. As we can observe, the BHO-MA method reached the best accuracy among all ML models selected for the HPO task. The best accuracy is obtained by the KNN algorithm with optimized parameters by the proposed BHO-MA method. As the KNN algorithm is successfully studied over the years for classification problems, we are not surprised by this result. The variation in the SVM classifier accuracy score shows how important HPO is in ML. The performance improvement is more than 2% between the default hyperparameters and the optimized model.

Our approach reached the best scores for both classification and regression tasks. It also gives the freedom to the user to choose the next preferred hyperparameter values from the Pareto-optimal solution set according to the trade-off between three different acquisition functions. Several popular acquisition functions have been described in the literature, and the proposed application can be extendable by using multiple acquisition function search strategies.

We shared the box plots of the experiments for each data set in Fig. 2 and Fig. 3. We can observe from the figures the mean and standard deviations of the performance metrics over 10 different runs.

7 Conclusion

We introduced BHO-ML, a hybrid and effective method for hyperparameter optimization. To the best of our knowledge, the first approach uses a multi-objective acquisition

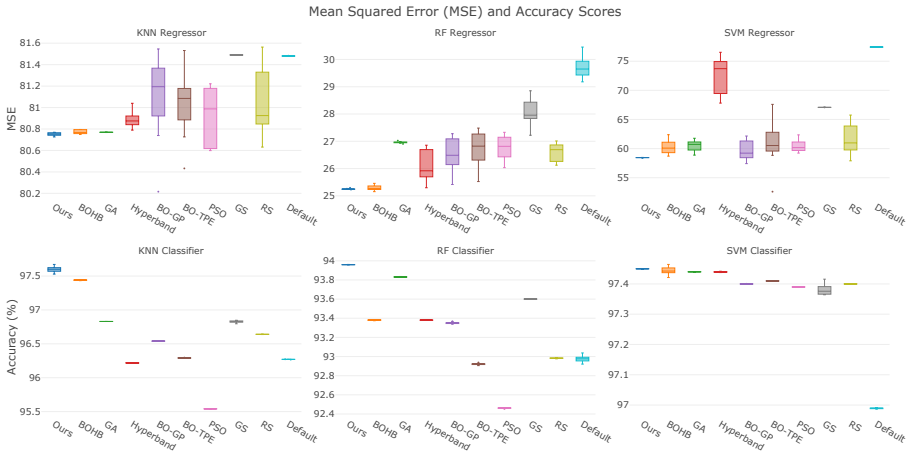


Fig. 2. Comparisons of the accuracy and mean squared errors between ours and the other methods on 2 different data sets with 3 different algorithms over 10 different runs.

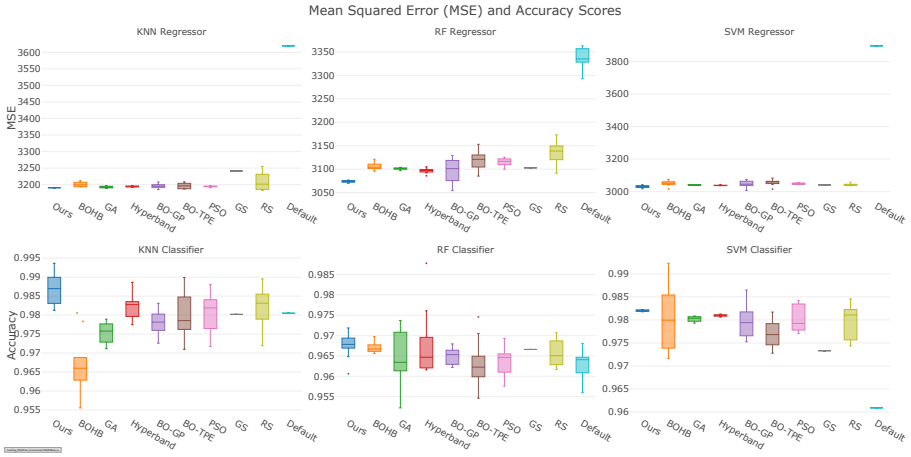


Fig. 3. Comparisons of the accuracy and mean squared errors between ours and the other methods on 2 different data sets with 3 different algorithms over 10 different runs.

function for hyperparameter optimization tasks. It achieved strong performance on a diverse set of benchmarks including both classification and regression tasks, and also we compared it with several state-of-the-art approaches.

Our work can be extended in various directions. It would be interesting to consider exploring the preferences of decision-makers during Bayesian optimization in terms of multiple acquisition options. Various kinds of multi-criteria decision-making methods can be applied to our work to improve hyperparameter optimization tasks for point selection during the Gaussian process. This could potentially extend our approach into

the domain of preference learning. Furthermore, there are various kinds of acquisition functions in the literature and they can be applied to our presented idea for the hyperparameter optimization task.

Finally, a more in-depth analysis of the relationship between the acquisition functions and tuned machine learning rate could provide insight into how acquisition function selection improves performance.

Limitations and Future Work. MOO problems are expensive to evaluate and time-consuming most of the time. It might cause a lack of performance during the acquisition optimization process of the proposed algorithm. Also, the Pareto-optimal solution set does not always provide the optimal solution for all objectives but still exploring the trade-off between acquisition functions improves the performance compared with various existing methods as we can see from Section 5. In future work, we shall explore the use of preferences during the optimization process. Moreover, we will look to gain some information by analyzing the relationship between acquisition selection and optimization results.

Acknowledgements This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 16/RC/3918 which is co-funded under the European Regional Development Fund. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

1. Balandat, M., Karrer, B., Jiang, D.R., Daulton, S., Letham, B., Wilson, A.G., Bakshy, E.: Botorch: Programmable bayesian optimization in pytorch. CoRR **abs/1910.06403** (2019), http://arxiv.org/abs/1910.06403
2. Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K. (eds.) Advances in Neural Information Processing Systems. vol. 24. Curran Associates, Inc. (2011), https://proceedings.neurips.cc/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf
3. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**(null), 281–305 (feb 2012)
4. Blank, J., Deb, K.: pymoo: Multi-objective optimization in python. IEEE Access **8**, 89497–89509 (2020)
5. Blume, S., Benedens, T., Schramm, D.: Hyperparameter optimization techniques for designing software sensors based on artificial neural networks. Sensors **21**(24) (2021). https://doi.org/10.3390/s21248435, https://www.mdpi.com/1424-8220/21/24/8435
6. Bockrath, S., Lorentz, V., Pruckner, M.: State of health estimation of lithium-ion batteries with a temporal convolutional neural network using partial load profiles. Applied Energy **329**, 120307 (2023). https://doi.org/https://doi.org/10.1016/j.apenergy.2022.120307, https://www.sciencedirect.com/science/article/pii/S0306261922015641
7. Bohdal, O., Yang, Y., Hospedales, T.: Evograd: Efficient gradient-based meta-learning and hyperparameter optimization. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information

Processing Systems. vol. 34, pp. 22234–22246. Curran Associates, Inc. (2021), https://proceedings.neurips.cc/paper/2021/file/bac49b876d5dfc9cd169c22ef5178ca7-Paper.pdf

8. Bühlmann, P.: Bagging, Boosting and Ensemble Methods, pp. 985–1022. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-21551-3_33
9. Chen, S., Wu, J., Liu, X.: Emorl: Effective multi-objective reinforcement learning method for hyperparameter optimization. Engineering Applications of Artificial Intelligence **104**, 104315 (2021). https://doi.org/https://doi.org/10.1016/j.engappai.2021.104315, https://www.sciencedirect.com/science/article/pii/S0952197621001639
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (2002). https://doi.org/10.1109/4235.996017
11. DeCastro-García, N., Muñoz Castañeda, A.L., Escudero García, D., Carriegos, M.V., Sánchez Lasheras, F.: Effect of the sampling of a dataset in the hyperparameter optimization phase over the efficiency of a machine learning algorithm. Complex. **2019** (jan 2019). https://doi.org/10.1155/2019/6278908, https://doi.org/10.1155/2019/6278908
12. Eggensperger, K., Hutter, F., Hoos, H.H., Leyton-Brown, K.: Efficient benchmarking of hyperparameter optimizers via surrogates. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. p. 1114–1120. AAAI'15, AAAI Press (2015)
13. Han, J., Gondro, C., Reid, K., Steibel, J.: Heuristic hyperparameter optimization of deep learning models for genomic prediction. G3-Genes Genomes Genetics **11** (02 2021). https://doi.org/10.1093/g3journal/jkab032
14. Hoffman, M., Brochu, E., de Freitas, N.: Portfolio allocation for bayesian optimization. In: Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence. p. 327–336. UAI'11, AUAI Press, Arlington, Virginia, USA (2011)
15. Hospedales, T., Antoniou, A., Micaelli, P., Storkey, A.: Meta-learning in neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(9), 5149–5169 (2022). https://doi.org/10.1109/TPAMI.2021.3079209
16. Hutter, F., Kotthoff, L., Vanschoren, J. (eds.): Automated Machine Learning - Methods, Systems, Challenges. Springer (2019)
17. Injadat, M., Salo, F., Nassif, A.B., Essex, A., Shami, A.: Bayesian optimization with machine learning algorithms towards anomaly detection. In: 2018 IEEE Global Communications Conference (GLOBECOM). pp. 1–6 (2018). https://doi.org/10.1109/GLOCOM.2018.8647714
18. Jordan, M.I., Mitchell, T.M.: Machine learning: Trends, perspectives, and prospects. Science **349**(6245), 255–260 (2015). https://doi.org/10.1126/science.aaa8415, https://www.science.org/doi/abs/10.1126/science.aaa8415
19. Kushner, H.J.: A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. Journal of Basic Engineering **86**, 97–106 (1963)
20. Li, L., Jamieson, K.G., DeSalvo, G., Rostamizadeh, A., Talwalkar, A.: Efficient hyperparameter optimization and infinitely many armed bandits. CoRR **abs/1603.06560** (2016), http://arxiv.org/abs/1603.06560
21. Lindauer, M., Feurer, M., Eggensperger, K., Biedenkapp, A., Hutter, F.: Towards assessing the impact of bayesian optimization's own hyperparameters. CoRR **abs/1908.06674** (2019), http://arxiv.org/abs/1908.06674
22. Lorenzo, P.R., Nalepa, J., Kawulok, M., Ramos, L.S., Pastor, J.R.: Particle swarm optimization for hyper-parameter selection in deep neural networks. In: Proceedings of the Genetic and Evolutionary Computation Conference. p. 481–488. GECCO '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3071178.3071208, https://doi.org/10.1145/3071178.3071208

23. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in python. J. Mach. Learn. Res. **12**(null), 2825–2830 (nov 2011)
24. Raghu, A., Lorraine, J., Kornblith, S., McDermott, M., Duvenaud, D.K.: Meta-learning to improve pre-training. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 23231–23244. Curran Associates, Inc. (2021), https://proceedings.neurips.cc/paper/2021/file/c3810d4a9513b028fc0f2a83cb6d7b50-Paper.pdf
25. Rasmussen, C.E.: Gaussian Processes in Machine Learning, pp. 63–71. Springer Berlin Heidelberg, Berlin, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28650-9_4
26. Shah, A., Ghahramani, Z.: Parallel predictive entropy search for batch global optimization of expensive objective functions. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 28. Curran Associates, Inc. (2015), https://proceedings.neurips.cc/paper/2015/file/57c0531e13f40b91b3b0f1a30b529a1d-Paper.pdf
27. Srinivas, N., Krause, A., Kakade, S.M., Seeger, M.W.: Gaussian process bandits without regret: An experimental design approach. CoRR **abs/0912.3995** (2009), http://arxiv.org/abs/0912.3995
28. Tsai, C.W., Fang, Z.Y.: An effective hyperparameter optimization algorithm for dnn to predict passengers at a metro station. ACM Trans. Internet Technol. **21**(2) (mar 2021). https://doi.org/10.1145/3410156, https://doi.org/10.1145/3410156
29. Vulpe-Grigoraşi, A., Grigore, O.: Convolutional neural network hyperparameters optimization for facial emotion recognition. In: 2021 12th International Symposium on Advanced Topics in Electrical Engineering (ATEE). pp. 1–5 (2021). https://doi.org/10.1109/ATEE52255.2021.9425073
30. Wada, T., Hino, H.: Bayesian optimization for multi-objective optimization and multi-point search (2019). https://doi.org/10.48550/ARXIV.1905.02370
31. Wang, Z., Hutter, F., Zoghi, M., Matheson, D., de Freitas, N.: Bayesian optimization in a billion dimensions via random embeddings (2013). https://doi.org/10.48550/ARXIV.1301.1942, https://arxiv.org/abs/1301.1942
32. Wicaksono, A.S., Supianto, A.A.: Hyper parameter optimization using the genetic algorithm on machine learning methods for online news popularity prediction. International Journal of Advanced Computer Science and Applications **9**(12) (2018)
33. Wu, J., Chen, X.Y., Zhang, H., Xiong, L.D., Lei, H., Deng, S.H.: Hyperparameter optimization for machine learning models based on bayesian optimizationb. Journal of Electronic Science and Technology **17**(1), 26–40 (2019). https://doi.org/https://doi.org/10.11989/JEST.1674-862X.80904120, https://www.sciencedirect.com/science/article/pii/S1674862X19300047
34. Wu, J., Frazier, P.I.: The parallel knowledge gradient method for batch bayesian optimization (2016). https://doi.org/10.48550/ARXIV.1606.04414, https://arxiv.org/abs/1606.04414
35. Yang, L., Shami, A.: On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing **415**, 295–316 (11 2020). https://doi.org/10.1016/J.NEUCOM.2020.07.061
36. Zhang, B., Rajan, R., Pineda, L., Lambert, N., Biedenkapp, A., Chua, K., Hutter, F., Calandra, R.: On the importance of hyperparameter optimization for model-based reinforcement learning. In: Banerjee, A., Fukumizu, K. (eds.) Proceedings of The 24th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 130, pp. 4015–4023. PMLR (13–15 Apr 2021), https://proceedings.mlr.press/v130/zhang21n.html