Global Journals LaTeX JournalKaleidoscope™

Artificial Intelligence formulated this projection for compatibility purposes from the original article published at Global Journals. However, this technology is currently in beta. *Therefore, kindly ignore odd layouts, missed formulae, text, tables, or figures.*

Machine Learning Model Optimization with Hyper Parameter Tuning Approach

Md Riyad Hossain[1] and Dr. Douglas Timmer[2]

[1] University of Texas Rio Grande Valley

Received: 5 June 2021 Accepted: 4 July 2021 Published: 15 July 2021

Abstract

Hyper-parameters tuning is a key step to find the optimal machine learning parameters. Determining the best hyper-parameters takes a good deal of time, especially when the objective functions are costly to determine, or a large number of parameters are required to be tuned. In contrast to the conventional machine learning algorithms, Neural Network requires tuning hyperparameters more because it has to process a lot of parameters together, and depending on the fine tuning, the accuracy of the model can be varied in between 25

Index terms— machine learning, hyper parameter optimization, grid search, random search, BO-GP.

1 Introduction

n the era of Machine learning, performance (based on accuracy and computing time) is very important. The growing number of tuning parameters associated with the Machine learning models is tedious and timeconsuming to set by standard optimization techniques. Researchers working with ML models often spend long hours to find the perfect combination of hyperparameters [1]. If we think w, x, y, z as the parameters of the model, and if all of these parameters are integers ranging from 0.0001 to say 5.00, then hyperparameter tuning is the finding the best combinations to make the objective function optimal.

One of the major difficulties in working with the Machine learning problem is tuning hyperparameters. These are the design parameters that could directly affect the training outcome. The conversion from a nontuned Machine learning model to a tuned ML model is like learning to predict everything accurately from predicting nothing correctly [2]. There are two types of parameters in ML models: Hyperparameters, and Model parameters. Hyperparameters are arbitrarily set by the user even before starting to train the model, whereas, the model parameters are learned during the training.

The quality of a predictive model mostly depends on the configuration of its hyperparameters, but it is often difficult to know how these hyperparameters interact with each other to affect the final results of the model [14]. To determine accuracy and make a comparison between two models it is always better to make comparisons between two models with both of the models' parameters tuned. It would be unfair to compare a Decision Tree model with the best parameter against an ANN model whose hyperparameters haven't been optimized yet.

2 II.

3 Literature Review

The hyperparameter tuning, due to its importance, has changed to a new interesting topic in the ML community. The hyperparameter tuning algorithms are either model-free or model-based.

Model-free algorithms are free of using knowledge about the solution space extracted during the optimization; a few of this category includes manual search [4], random search [2,[6][7], and grid search [5]. In the Manual search categories, we assume the values of the parameters based on our previous experience. In this technique, the user allows some sets of hyperparameters based on judgments or previous experience, trains the algorithm by them, observes the performance, keeps repeating to train the model until achieving a satisfactory accuracy and then selects the best set of hyperparameters that gives the maximum accuracy. However, this technique is heavily dependent on the judgment and previous expertise and its reliability is dependent on the correctness of the previous knowledge [3]. Some of the few of the main parameters used by Random forest classifiers are: criterion, max_depth, n_estimators, min_samples_split etc.

In the Random search, we train and test our model based on some random combinations of the hyperparameters. This method is better used to identify new combinations of the parameters or to discover new hyperparameters. Although it may take more time to process, it often leads to better performance. Bergstra et al. (2012) in their work mentioned that, over the same domain, random search is able to find models that are as good as or even better in a reduced computation time. After granting the same computational budget for the random search, it was evident that random search can find better models by effectively searching for larger and less promising configuration spaces [16]. Random Search, which is developed based on grid research, sets up a grid of hyper-parameter values and selects random combinations to train the algorithm; [2].

In the grid search, the user sets a grid of hyperparameters and trains the model based on each possible combination. Amirabadi et al. (2020) proposes two novel suboptimal grid search techniques on the four separate dataset to show the efficiency of their hyperparameter tuning model and later compare it with some of the other recently published work. The main drawback of the grid search method is its high complexity. It is commonly used when there are a few numbers of hyperparameters to be tuned. In other words, grid search works well when the best combinations are already determined. Some of the similar works of grid search applications have been reported by Zhang et al. (2014) [17], Ghawi et al. (2019) [18], and Beyramysoltan et al. (2013) [19].

Zhang et al. (2019) [20] in their work reported a few of the drawbacks of the existing hyperparameter tuning methods. In their work, they mentioned grid search as an ad-hoc process, as it traverses all the possible combinations, and the entire procedure requires a lot of time. Andradóttir (2014) [13] shows that Random Search (RS) eradicates some of the limitations of the grid search technique to an extent. RS can reduce the overall time consumption, but the main disadvantage is that it cannot converge to the global optimal value.

The combination of randomly selected hyperparameters can never guarantee a steady and widely acceptable result. That's why, apart from the manually tuning methods, automated tuning methods are becoming more and more popular in recent times; snoek et al. (2015) [10]. Bayesian Optimization is one of the most widely used automated hyperparameter tuning methods to find the global optimum in fewer steps. However, Bayesian optimization's results are sensitive to the parameters of the surrogate model and the accuracy is greatly depending on the quality of the learning model; Amirabadi et al. (**??**020) [3].

To minimize the error function of hyperparameter values, Bayesian optimization adopts probabilistic surrogate models like Gaussian processes. Through precise exploration and development, an alternative model of hyperparameter space is established; Eggensperger et al. (2013) [8]. However, probabilistic surrogates need accurate estimations of sufficient statistics of error function distribution. So, a sizable number of hyperparameters is required to evaluate the estimations and this method doesn't work well when there is to process myriad hyperparameters altogether.

4 III.

5 Methodology a) Dataset description

Denier: Denier is a weight measurement usually refers to the thickness of the threads. It is the weight (grams) of a single optical fiber for 9 kilometers. If we have a 9 km fiber weighs 1 gram, this fiber has a denier of 1, or 1D. A fiber with less than 1 gram weight calls Microfibers [22]. Microfibers become a new development trend in the synthetic polymer industry. The higher the denier is, the more thick and strong the fiber is. Conversely, less denier means that the fiber/fabric will be softer and more transparent. Fine denier fibers are becoming a new standard and are very useful for the development of new textiles with excellent performance [21].

Breaking Elongation (%): Elongation at break is one of the few main quality parameters of any synthetic fiber [24]. It is the percentage of elongation at break. Fiber elongation partly reflects the extent of stretching a filament under a certain loading condition. Fibers with high elongation at break are determined to be easily stretched under a predetermined load. Fibers showing these characteristics are known to be flexible. The elongation behavior of any single fiber can be complex because of its multiplicity of structural factors affecting it. Moreover, a cotton fiber comes up with a natural crimp, which is important for fibers to stick together while undergoing other production processes [23]. If L is the length of the fiber, then the equation for the percentages of the breaking elongation would be:?? ???????????????? ???????????????????? = ??? ?????????? ?? 0 * 100%

Breaking elongation for the cotton fiber might be varied from 5% to 10%, which is significantly lower than that of wool fibers (25%-45%), and much lower than polyester fibers (typically over 50%).

Breaking force (cN) and Tenacity (cN/tex): Breaking tenacity is the maximum load that a single fiber can withstand before breaking. For the Polypropylene and PET staple fibers, 10 mm lengths sample filaments is drawn until failure. Breaking tenacity is measured in grams/denier. Very small forces are encountered when evaluating fiber properties, so an instrument with gramlevel accuracy is required [25]. The tenacity of virgin PP fibers is about 5-8 g/den, and the elongation at break is about 100%. At the same time, the tenacity of recycled PET is about 3.5-5.7 g/den; the elongation at break usually exceeds 100%. Draw Ratio: The drawing ration is the ratio of the diameter of the initial blank form to the diameter of the drawn part. The limiting drawing ratio

(Capstan speed/Nip reel speed) for the extruder section is between 1.6 and 2.2 [26], whereas, for the stretching section it is in between 3 and 4.

6 b) Hyper-parameter Optimization (HPO)

The purpose of hyperparameter optimization is to find the global optimal value ?? * of the objective function f(x) can be evaluated for any arbitrary ?? ? ??, ?? * = arg ?????? ????? ð ??”ð ??”(??), and X is a hyperparameter space that can contain categorical, discrete, and continuous variables [27].In order to construct the design of different machine learning models, the application of effective hyperparameter optimization techniques can simplify the process of identifying the best hyperparameters for the models. HPO contains four major components: First, an estimator that could be a regressor or any classifier with one or more objective functions, second: a search space, Third: an optimization method to find the best combinations, and Fourth: a function to make a comparison between the effectiveness of various hyperparameter configurations [28]. Some of the common hyperparameter techniques is discussed below: Grid Search: Grid search is a process that exhaustively searches a manually specified subset of the hyperparameter space of the target algorithm [30]. A traditional approach to finding the optimum is to do a grid search, for example, to run experiments or processes on a number of conditions, for example, if there are three factors, a 15 × 15× 15 would mean performing 3375 experiments under different conditions. [32]. Grid search is more practical when [31]: (1) the total number of parameters in the model is small, say M <10. The grid is M-dimensional, so the number of test solutions is proportional to L M , where L is the number of test solutions along each dimension of the grid. (2) The solution is known to be within a specific range of values, which can be used to define the limits of the grid. (3) The direct problem d = g (m) can be computed quickly enough that the time required to compute L M from them is not prohibitive. (4) The error function E (m) is uniform on the scale of the grid spacing, Î?”m, so that the minimum is not lost because the grid spacing is too coarse.

There are many problems with the grid search method. The first is that the number of experiments can be prohibitive if there are several factors. The second is that there can be significant experimental error, which means that if the experiments are repeated under identical conditions, different responses can be obtained; therefore, choosing the best point on the grid can be misleading, especially if the optimum is fairly flat. The third is that the initial grid may be too small for the number of experiments to be feasible, and it could lose characteristics close to the optimum or find a false (local) optimum [32].

Random Search: Random search [33] is a basic improvement on grid search. It indicates a randomized search over hyper-parameters from certain distributions over possible parameter values. The searching process continues till the predetermined budget is exhausted, or until the desired accuracy is reached. This methods are the simplest stochastic optimization and are very useful for certain problems, such as small search space and fast-running simulation. RS finds a value for each hyperparameter, prior to the probability distribution function. Both the GS and RS estimate the cost measure based on the produced hyperparameter sets. Although RS is simple, it has proven to be more effective than Grid search in many of the cases [33].

Random search has been shown to provide better results due to several benefits: first, the budget can be set independently according to the distribution of the search space, therefore, random search can work better especially when multiple hyper-parameters are not uniformly distributed [34]. Second: Because each evaluation is independent, it is easy to parallelize and allocate resources. Unlike GS, RS samples a number of parameter combinations from a defined distribution, which maximizes system efficiency by reducing the likelihood of wasting a lot of time in a small, underperforming area. In addition, this method can detect global optimum values or close to global if given a sufficient budget. Third, although getting optimal results using random search is not promising, more time consumption will lead to a greater likelihood of finding the best hyperparameter set, whereas longer search BO is more efficient than GS and RS because it can detect optimal combinations of hyperparameters by analyzing previously tested values, and running the surrogate model is usually much cheaper than running the objective function as a whole. However, because Bayesian optimization models are run based on previously tested values, it is difficult to belong to them with parallel sequential methods; but they are generally able to detect optimal close hyperparameter combinations in a few iterations [36]. Common substitution models for BO include the Gaussian process (GP) [37], random forest (RF) [38], and Parzen estimator (TPE) **??**39]. Therefore, there are three main BO algorithms based on their substitution models: BO-GP, BO-RF, BO-TPE. GP is an attractive reduced order model of BO that can be used to quantify forecast uncertainty. This is not a parametric model and the number of its parameters depends only on the input points. With the right kernel function, your GP can take advantage of the data structure. However, the GP also has disadvantages. For example, it is conceptually difficult to understand with BO theory. In addition, its low scalability with large dimensions or a large number of data points is another important issue [36]. Applying HPO in ML Models

In order to put the theory into practice, several experiments have been performed on an industrialbased synthetic polymer model. This section describes experiments with four different HPO techniques on three general and representative ML algorithms. In the first part of the section, we discussed the experimental setup and the main HPO process. In the second part, we compare and analyze the results of the application of different HPO methods. The use of random search is recommended in the early stages of HPO to narrow the search space quickly, before using guided algorithms to get better results. The main drawback [28] of RS and GS is that each evaluation in its iteration does not depend on previous evaluations; thus, they waste time evaluating underperforming areas of the search space.

Table ??: Performance evaluation of applying HPO methods to the regressor on the synthetic polymer dataset

7 Discussion & Conclusion

Machine learning has become the primary strategy for dealing with data problems and is widely used in various applications. To apply ML models to practical problems, hyperparameters must be tuned to handle specific datasets. However, as the size of the generated data increases greatly in real life, and manual tuning of hyperparameters is extremely computationally expensive, it has become essential to optimize the hyperparameters by an automatic process. In this work, we used hyperparameter techniques in the ML model to find the best set of hyperparameters. Our data set was small, and in this small datset we can see that the randomly selected subsets are very representative for the given data set, as they can effectively optimize all types of hyperparameters. Our future work would be to test our model on a much larger data set and see the feedback.

1

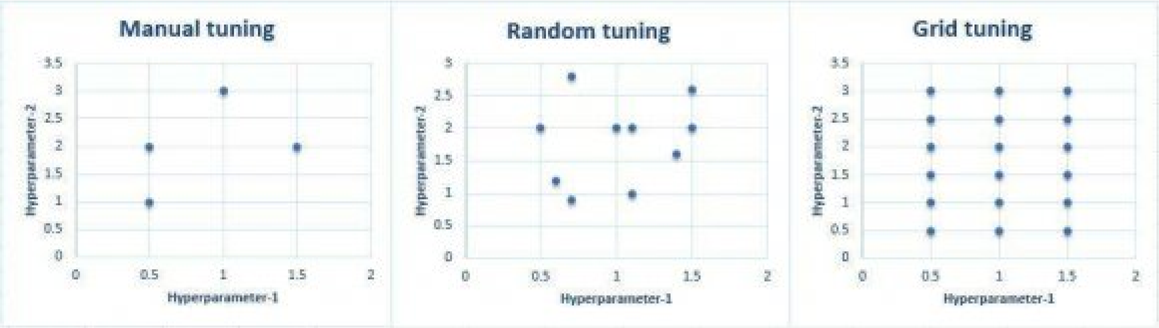

Figure 1: Figure 1 :

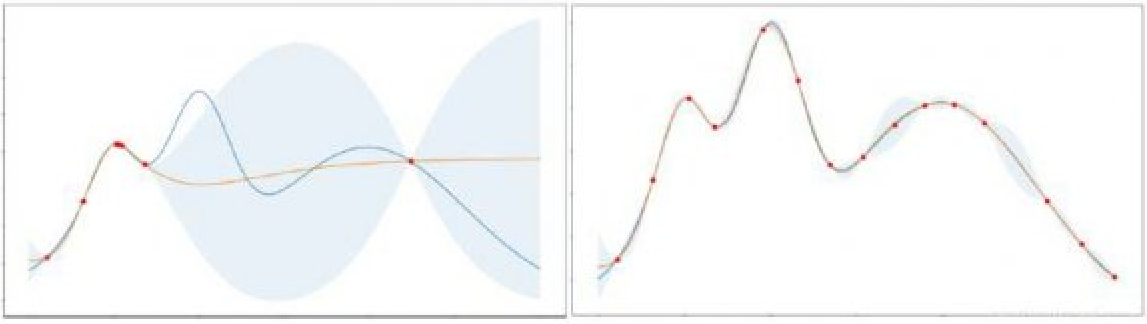

Figure 2:

2

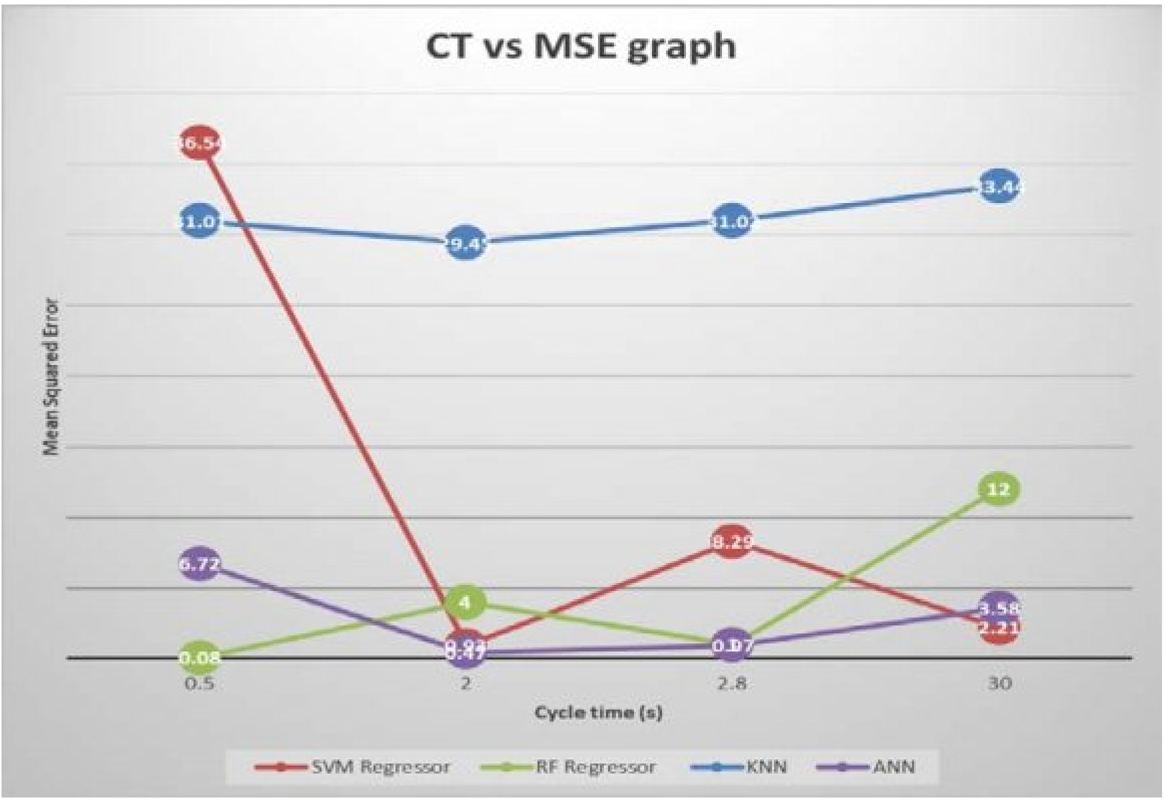

Figure 3: Figure 2 :

1

Machine Learning Model Optimization with Hyper Parameter Tuning Approach
Year 2021
10
( ) D

ML Model	Hyper-parameter
RF Regressor	n_estimators, max_depth, min_samples_split, min_samples_leaf, criterion, max_features
SVM Regressor	C, kernel, epsilon
KNN Regressor	n neighbors

[Note: © 2021 Global JournalsGlobal Journal of Computer Science and TechnologyVolume XXI Issue II Version I]

Figure 4: Table 1 :

.1 Conflict of Interest:

The authors whose names are listed in this work certify that they have no affiliations with or involvement in any organization or entity with any financial interest, or non-financial interest in the subject matter or materials discussed in this manuscript.

[Neurocomputing] , 10.1016/j.neucom.2020.07.061. *Neurocomputing* 415 p. .

[Springer] , Springer . 10.1007/978-3-642-25566-3_40. `https://doi.org/10.1007/978-3-642-25566-3_40` Berlin, Heidelberg.

[Li et al. ()] 'A Novel Bandit-Based Approach to Hyperparameter Optimization'. L Li , K Jamieson , G Desalvo , A Rostamizadeh , A Talwalker , Hyperband . *Journal of Machine Learning Research* 2018. 18 p. .

[Andrad_Ottir ()] 'A review of random search methods'. S Andrad_Ottir . *Handbook of Simulation Optimization*, 2015. Springer. p. .

[Tyagi ()] *Advances in Yarn Spinning Technology // Yarn structure and properties from different spinning techniques*, G K Tyagi . doi:10. 1533/9780857090218.1.119. 2010. p. .

[Bergstra et al. ()] 'Algorithms for hyper-parameter optimization'. J Bergstra , R Bardenet , Y Bengio , B Kégl . *Adv Neural Inf Process Syst (NIPS)* 2011. 24 p. .

[Bergstra et al. ()] 'Algorithms for hyperparameter optimization'. J S Bergstra , R Bardenet , Y Bengio , B Kégl . *Advances in Neural Information Processing Systems*, 2011. p. .

[Larochelle et al. ()] 'An empirical evaluation of deep architectures on problems with many factors of variation'. H Larochelle , D Erhan , A Courville , J Bergstra , Y Bengio . *Proceedings of the 24th International Conference on Machine Learning*, (the 24th International Conference on Machine Learning) 2007. ACM. p. .

[Hutter et al. ()] *Automated Machine Learning Methods, Systems*, F Hutter , L Kotthoff , J &vanschoren . 2019. Springer International Publishing.

[Cho et al. ()] 'Basic Enhancement Strategies When Using Bayesian Optimization for Hyperparameter Tuning of Deep Neural Networks'. H Cho , Y Kim , E Lee , D Choi , Y Lee , W Rhee . 10.1109/access.2020.2981072. *IEEE Access* 2020. 8 p. .

[Cho et al. ()] 'Basic Enhancement Strategies When Using Bayesian Optimization for Hyperparameter Tuning of Deep Neural Networks'. H Cho , Y Kim , E Lee , D Choi , Y Lee , W Rhee . 10.1109/access.2020.2981072. *IEEE Access* 2020. 8 p. .

[Hutter et al. ()] 'Beyond manual tuning of hyperparameters'. F Hutter , J Lücke , L Schmidt-Thieme . *DISKI* 2015. 29 (4) p. .

[Chan and treleaven ()] *Continuous Model Selection for Large-Scale Recommender Systems. Handbook of Statistics Big Data Analytics*, S Chan , P &treleaven . 10.1016/b978-0-444-63492-4.00005-8. 2015. p. .

[Zhang et al. ()] 'Crystalline behaviors and phase transition during the manufacture of fine denier PA6 fibers'. C Zhang , Y Liu , S Liu . 10.1007/s11426-009-0242-5. `https://doi.org/10.1007/s11426-009-0242-5` *Sci. China Ser. B-Chem* 2009. 52 p. 1835.

[Zhang et al. ()] 'Deep Neural Network Hyperparameter Optimization with Orthogonal Array Tuning'. X Zhang , X Chen , L Yao , C Ge , M Dong . 10.1007/978-3-030-36808-1_31. *Computer and Information Science Neural Information Processing*, 2019. p. .

[Jones et al. ()] 'Efficient global optimization of expensive black-box functions'. D R Jones , M Schonlau , W J Welch . *J. Glob. Optim* 1998. 13 p. .

[Ghawi and Pfeffer ()] 'Efficient Hyperparameter Tuning with Grid Search for Text Categorization using kNN Approach with BM25 Similarity'. R Ghawi , J Pfeffer . 10.1515/comp-2019-0011. *Open Computer Science* 2019. 9 (1) p. .

[Friedrichs and Igel ()] 'Evolutionary tuning of multiple SVM parameters'. F Friedrichs , C Igel . *Neurocomputing* 2005. 64 p. .

[Swift and Booker ()] *Forming Processes. Manufacturing Process Selection Handbook*, K Swift , J Booker . 10.1016/b978-0-08-099360-7.00004-5. 2013. p. .

[Seeger ()] 'Gaussian Processes For Machine Learning'. M Seeger . 10.1142/s0129065704001899. *International Journal of Neural Systems* 2004. 14 (02) p. .

[Elmogahzy ()] *Handbook of Properties of Textile and Technical Fibres // Tensile properties of cotton fibers*, Yehia Elmogahzy . 10.1016/B978-0-08-101272-7.00007-9. 2018. p. .

[Hazan et al. ()] E Hazan , A Klivans , Y Yuan . arXiv:1706.00764. `https://arxiv.org/abs1706.00764` *Hyperparameter optimization: a spectral approach*, 2017. (arXiv preprint)

[Yu and Zhu ()] *Hyper-Parameter Optimization: A Review of Algorithms and Applications*, T Yu , H Zhu . `https://arxiv.org/abs/2003.05689` 2020.

[Beyramysoltan et al. ()] 'Investigation of the equality constraint effect on the reduction of the rotational ambiguity in threecomponent system using a novel grid search method'. S Beyramysoltan , R Rajkó , H Abdollahi . 10.1016/j.aca.2013.06.043. *Analytica Chimica Acta* 2013. 791 p. .

[Klein et al. ()] 'Learning curve prediction with Bayesian neural networks'. A Klein , S Falkner , J T Springenberg , F Hutter . *International Conference On Learning Representation (ICLR*, 2017.

[Calandra et al. (2016)] 'Manifold Gaussian processes for regression'. R Calandra , J Peters , C E Rasmussen , M P Deisenroth . *Proceedings of the 2016 International Joint Conference on Neural Networks*, (the 2016 International Joint Conference on Neural NetworksVancouver, BC, Canada) July 2016. p. .

[Mantovani et al. ()] R G Mantovani , A L Rossi , J Vanschoren , B Bischl , A C De Carvalho . *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015. p. . (Effectiveness of random search in SVM hyper-parameter tuning)

[Blair ()] 'Materials and design for sports apparel'. K Blair . 10.1533/9781845693664.1.60. *Materials in Sports Equipment* 2007. p. .

[Menke ()] W Menke . 10.1016/b978-0-12-397160-9.00009-6. *Nonlinear Inverse Problems. Geophysical Data Analysis: Discrete Inverse Theory*, 2012. p. .

[Amirabadi et al. ()] *Novel suboptimal approaches for hyperparameter tuning of deep neural network*, M Amirabadi , M Kahaei , S Nezamalhosseini . 10.1016/j.phycom.2020.101057. 2020. Physical Communication. 41 p. 101057. (under the shelf of optical communication)

[Yang and shami ()] *On hyperparameter optimization of machine learning algorithms: Theory*, L Yang , A &shami . 2020.

[Li and Talwalkar ()] *Random search and reproducibility for neural architecture search*, L Li , A Talwalkar . arXiv:1902.07638. 2019. (arXiv preprint)

[Bergstra and Bengio ()] 'Random Search for Hyper-Parameter Optimization'. J S Bergstra , Y Bengio . *Journal of Machine Learning Research* 2012. 13 p. .

[Bergstra and Bengio ()] 'Random search for hyper-parameter optimization'. J Bergstra , Y Bengio . `http://dl.acm.org/citation.cfm?id=2188385.2188395` *J. Mach. Learn. Res* 1532-4435. 2012. 13 p. .

[Hutter et al. ()] 'Sequential Model-Based Optimization for General Algorithm Configuration'. F Hutter , H H Hoos , K Leyton-Brown . Lecture Notes in Computer Science Coello C.A.C. (ed.) 2011. 2011. 6683. ((eds) Learning and Intelligent Optimization)

[Snoek et al. ()] J Snoek , O Rippel , K Swersky , R Kiros , N Satish , N Sundaram . *Scalable bayesian optimization using deep neural networks, in: International conference on machine learning*, 2015. p. .

[Brereton ()] 'Steepest Ascent, Steepest Descent, and Gradient Methods'. R Brereton . 10.1016/b978-044452701-1.00037-5. *Comprehensive Chemometrics* 2009. p. .

[Zhang et al. ()] 'Support Vector Regression Based on Grid-Search Method for Short-Term Wind Power Forecasting'. H Zhang , L Chen , Y Qu , G Zhao , Z Guo . 10.1155/2014/835791. *Journal of Applied Mathematics* 2014. 2014. p. .

[Eggensperger et al. ()] 'Towards an empirical foundation for assessing bayesian optimization of hyperparameters'. K Eggensperger , M Feurer , F Hutter , J Bergstra , J Snoek , H Hoos , K Leyton-Brown . *NIPS workshop on Bayesian Optimization in Theory and Practice*, 2013. 10 p. 3.

[Joe (2020)] 'What Is Denier Rating? Why Does It Matter To You?'. Joe . `https://www.digitravelist.com/what-is-denier-rating/` *Digi Travelist* 2020. May 5.