Optimizing Random Forest for Handwritten Digit Recognition Through Hyper-parameter Tuning

Yaqeen Saad Ali[1], Rihab Hazim Qasim[1] , Sura Mahroos Searan[1], Othman Mohammed Jasim[2,*], Ibaa Sadoon Jabbar Alzubaydı[3]

[1]Department of computer Science, College of computer Science and Information Technology , University of Anbar, Anbar, Iraq

[2]Department of Computer Engineering Techniques, College of Technical Engineering, University of Al Maarif, Al Anbar, 31001, Iraq

[3]Construction and Projects Department, University of Technology, Baghdad, Iraq

Emails: yaqeen.cs91@uoanbar.edu.iq; rehz1991@uoanbar.edu.iq; surasms917@uoanbar.edu.iq; othman.jaseem@uoanbar.edu.iq; Ibaa.s.Jabbar@uotechnology.edu.iq

Abstract

The significant increase in the volume of recently released records and multimedia news that is available presents fresh issues for pattern-recognition and machine-learning, particularly in addressing the longstanding issue of recognizing handwritten digits. Handwriting-recognition is a captivating area of research due to the uniqueness of each individual's handwriting style. It involves a computer's ability that automatically identify and comprehend handwritten (digit or character). Hyper parameters play a crucial role in the performance of machine learning algorithms, directly influencing the training process and significantly affecting the resulting model's performance. This work introduce a general automated hyper parameter tuning mechanics were used to optimize the random forest parameters, which are: grid- random search and Bayesian optimization applying on MNIST digit database (images) that have already been pre-processed. These proposed methods successfully identify optimal hyper parameters across a wide variety of ML models, taking into consideration the time cost of the search. This work shows the effectiveness and efficiency of used techniques, crucial for real-world applications. The results of this study show an accuracy rate of 99.3% for the Grid Search model, 98.8% for the Random Search model, and 96.0% for Bayesian Optimization on random forest algorithm.

Keywords: Handwritten-Recognition; Mnistdataset; Random-forest algorithm; Grid search; Random search; Bayesian Optimization

1. Introduction

With a plenty of information existing on paper, the cost-effectiveness of processing digital files surpasses that of traditional paper documents. The primary objective of (handwriting-recognition) structure is to translate characters written by hand into formats that are usable by machines. This technology finds application in various domains, including (vehicle license plate) appreciation, postal letter-sorting services, preservation of historical documents in archaeology departments, and automation of old documents in libraries and banks, among others[1]. These applications involve extensive databases, necessitating high recognition accuracy, low computational complexity, and consistent performance. Using a machine learning technique, handwritten digit recognition is the process of recognizing numbers written by humans. The machine performing the recognition task needs to be trained based on labled classes dataset. A lot of techniques like support vector machines , Artificial neural networks , deep learning-based classifiers and Neuro-Fuzzy Systems have been presented in the scientific literature [2],[3],[4],[5][6]. Despite reported decent recognition accuracy by these classifiers, handwritten digit recognition continues to be research problem, prompting the exploration of new techniques and methodologies to enhance

DOI: https://doi.org/10.54216/FPA.200112
Received: December 21, 2024 Revised: February 04, 2025 Accepted: April 01, 2025

performance in terms of recognition (accuracy-running time-and complexity of computation) [7], [8], [9] [10][11][12]. Consequently, this work introduces optimizing the parameters of random forest algorithm via choosing the best parameter by applying three model, which are grid, random search and Bayesian optimization on the mnist dataset to find the best testing accuracy.

Many instruments and techniques have been developed in recent decades to improve handwritten numerical recognition, with ongoing efforts to enhance accuracy. Mioulet (2015) [12] accomplished studies contrasting the effectiveness of various models, such as RBF-SVM hybrid systems, bagged RBF, bagged SVM, and support vector machines. The hybrid RBF-SVM classifier performed well in terms of classification accuracy, according to the results. The process of handwriting digit recognition was investigated in a study by Shamim et al. (2018) [14] utilizing a variety of algorithms and working methods, including :(ANN), (RF) ,(SVM), (KNN), and KMeans method. Using the same database, the effectiveness of these algorithms was investigated, and accuracy-based comparisons were conducted. Abdulrazzaq, Maiwan Bahjat, and Jwan Najeeb Saeed (2019) [15] matched three categorization algorithms, Multilayer Perceptron (MLP),Naive Bayes (NB) and K_Star algorithm using the NIST handwritten dataset in order to address the problem of identifying each individual handwriting uniqueness. Finding the best classifier with an acceptable accuracy level while using the fewest possible features was the goal. The recall, precision, and F-measure metrics showed that the K_Star method performed better than NB and MLP, yielding an accuracy rate of 82.36%. Shao et al. (2023) introduced a deep Convolutional Neural Network (CNN) model [16] in order to enhance the MNIST dataset handwritten digit recognition rate. The model had a multi-layer deep organization structure with fully connected layers for classification and convolution and activation layers for feature extraction. To effectively increase recognition performance, hyper parameters including learning rate, activation function, batch normalization, kernel sizes, and batch sizes were improved. The Key Contributions of the paper in bullet points.

- Addressed the handwritten digit recognition problem, a well-known problem in the pattern recognition and machine learning.
- To focus on individuality of the handwriting, and use the MNIST dataset of preprocessed digit images.
- Highlighted the role of hyperparameter tuning for improving machine-learning algorithm.
- Suggested a general automated hyperparameter tuning system designed for the Random Forest model.
- Evaluated three hyperparameter search methods: Grid Search, Random Search, Bayesian Optimization
- To analysed the balance between how close the results are to the game's perfect balance and the efficiency of the method (time used).

This paper is formed as follows: section 2 include survey for hyper parameter tuning with its techniques that applied to tune the algorithm parameter, section 3 provide an overview of algorithm survey which include random forest algorithm, Section 4 introduce experiments in details , Section 5 include the results of this work (The results of three techniques of hyperparameter tuning) , Section 6 include conclusions.

2. Hyper-Parameter Tuning

Hyperparameter tuning is a crucial phase in determining the best machine (ML) settings. This procedure can take a long time, especially if there are many parameters that need to be adjusted or if assessing objective functions is costly [17]. Given its significance, hyper parameter tuning has become an interesting topic in the machine learning community. Algorithms for hyper parameter tuning fall into two categories: Model free and Model based. Model-free: algorithms do not leverage knowledge about the solution space during optimization, and examples include manual search, random search, and grid search. Grid search (this method estimate all the possible combinations of hyper-parameter values), Random search (this approach randomly chooses hyper-parameter values for the hyper-parameter space) and Bayesian (this method employs a Bayesian-model to estimate the optimal hyper parameter values) [18],[19],[20],[21].

2.1 Grid Search

A methodical method thoroughly explores predefined section of the hyper-parameter space in relation to the intended algorithm. A conventional method for determining the optimal value involves conducting a grid search, which involves executing experiments or procedures under various conditions. It is more useful to use grid search especially if the model has a limited total number of parameters of the grid [22][23]. The grid's bounds can be defined by knowing that the answer falls inside a particular range of values. The grid search method faces several challenges. The sheer number of experiments can become impractical when dealing with multiple factors, significant experimental errors may occur, resulting in different responses under identical conditions. Consequently, selecting the best point on the grid can be misleading, particularly when the optimum is relatively flat [24],[25].

DOI: https://doi.org/10.54216/FPA.200112
Received: December 21, 2024 Revised: February 04, 2025 Accepted: April 01, 2025

2.2 Random Search

Random search is an approach that supersedes the exhaustive exploration of all combinations by randomly selecting them. While it can be straightforwardly implemented in discrete cases, this method can also be extended to continuous and mixed spaces. Random search has demonstrated its ability to outperform grid search, particularly in situations where just few number of hyper-parameters significantly influence the n the machine learning algorithm's performance [26],[27].

2.3 Bayesian Optimization

This algorithm introduces an additional dynamic approach compared to grid search. It comprises two essential components: The acquisition function and the probabilistic surrogate model. This algorithm operates interactively, with each iteration involving fitting the surrogate model to all previous examination of the aim function. Subsequently, the acquisition function identifies the most promising parameters to enhance the search and directs attention to them, aiming to discover the optimal set of hyper parameters. The Bayesian-based model endeavors to predict the performance of untested combinations [28],[29].

3. Random Forest algorithm

The random-forest algorithm is multilateral and viable to both regression and classification trouble, similar to decision trees. Its functionality involves constructing multiple decision trees and averaging their outcomes. The term "random forest" is derived from the introduction of randomness during the tree structure's creation. Instead of directly seeking the better feature when splitting a node, the algorithm identifies the best attribute within a randomly selected subset of attributes (see figure 1). This injection of randomness results in more diverse trees, ultimately enhancing the overall performance of the algorithm [30].

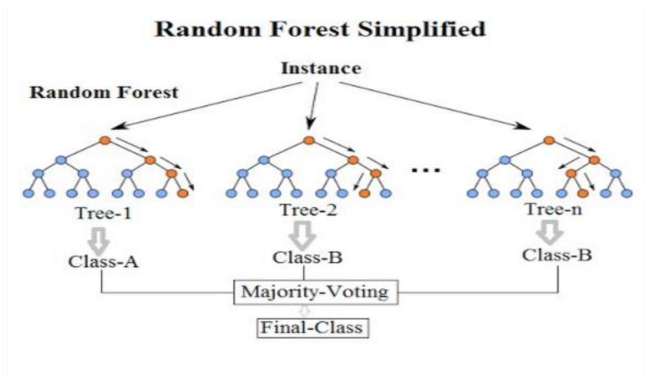

Figure 1. Random Forest Classifier [31].

4. Experiment

This work centered on optimizing the random forest (RF) algorithm's key parameters and examining the relation among the algorithm’s performance and parameters' values. Initially, an elaborating were performed on the parameters that were under scrutiny and detail the tuning process.

4.1 Mnist dataset

Classification of handwriting digit is a field in the computer-vision concentrated on handwritin digit classification. This field introduce challenges due to the inherent variability in handwriting styles. however, it holds key applications like signature verification, document digitization, and the development of mobile input methods. The dataset (MNIST) is closely working with in HCR research, this dataset contain images which are 70.000 grouped between 60.000 for training set and 10.000 for testing set image featuring handwritten digits from 0 to 9, each presented in 28x28 pixel (As in Table 1) HCR remains a formidable task, offering ample opportunities for continued research and advancements [28][29].

DOI: https://doi.org/10.54216/FPA.200112
Received: December 21, 2024 Revised: February 04, 2025 Accepted: April 01, 2025

Figure 2. Selected image from MNIST test dataset

Figure 2 display subset of selected digit from mnist data-set. Using a method called anti-aliasing; adjacent pixels of different colors are blended to smooth out an image's jagged edges. Through this process, images become more realistic and recognizable. When it comes to the MNIST dataset, anti-aliasing helps models meant to identify handwritten numbers perform more accurately. By smoothing out the jagged edges present in digit images, anti-aliasing facilitates a more precise distinction between different digits, thereby enhancing the performance of recognition models [29]. Table 1 Discusses mnist-dataset contain the most features of images such as image size, class number, number of images and image format.

Table 1: Discuss mnist dataset.

Feature	**Value**
Number of images	70000
Image size	28*28 pizels
Number of classes	10
Distribution of classes	60000 images in training set , 10000 images in the test set
Image format	Grayscale

4.2 Implementation

In this work (mnist) dataset is used, it is international accepted group of handwritten digits, for both train and test motivation. In order to applying the machine learning system, Python library present a man and various range of ML such as a scikit learn [31]. Before, the dataset need preparation before training a machine-learning model. The pre-processing steps may change depend on the variety of model being employed. However, some common pre-processing steps for the MNIST dataset include: [32].

4.2.1 Normalize Pixel Values

It was feasible to normalizing the pixel values via scaling them within a given range, like 0 to 1 or -1 By divid each pixel value by the maximum pixel value.

4.2.2 Resize Images

Resize the images to a consistent size, as certain machine learning algorithms may require standardized input sizes. Resize can be accomplished through cropping or padding the images.

4.2.3 Convert to Gray Scale

Convert the images to gray scale; while not necessary for all algorithms, it can enhance performance in some cases. This can be done by averaging the red, green, and blue channels of each pixel. Once the MNIST dataset has undergone these preprocessing steps, it becomes suitable for train the machine-learning model. The bar graph clarify the division of training model samples across digit labels in the MNIST dataset, with the 'x axis' representing the digit label and the 'y axis' showing the number of training samples for each label (see figure 3).

DOI: https://doi.org/10.54216/FPA.200112
Received: December 21, 2024 Revised: February 04, 2025 Accepted: April 01, 2025

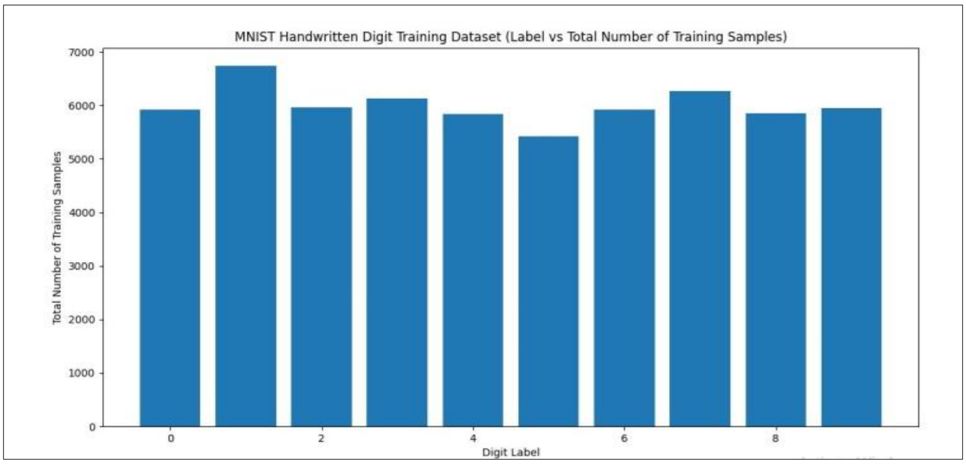

Figure 3. Shows each digit label in the MNIST dataset, (total number of training samples).

Figure 4 display the implementation of applying hyper parameter tuning techniques in order to obtain the best parameter of RF model on mnist dataset, started by choose and process dataset to the next step for choose hyper-technique inorder to tune RF model parameters to give the final result.

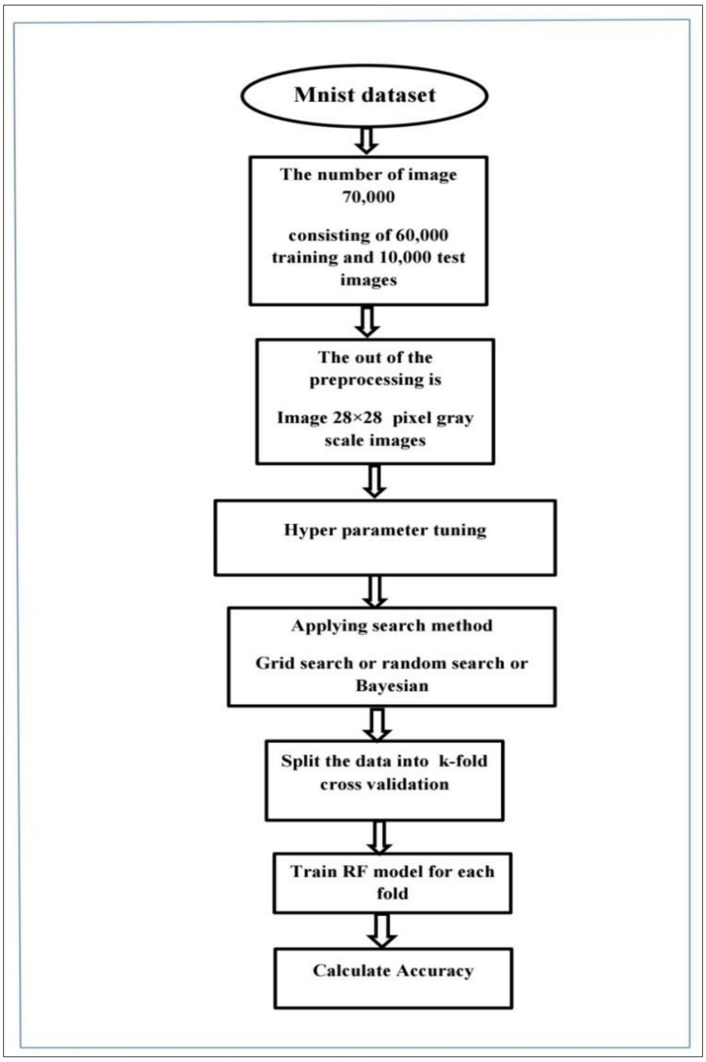

Figure 4. Block diagram of model implementation

DOI: https://doi.org/10.54216/FPA.200112
Received: December 21, 2024 Revised: February 04, 2025 Accepted: April 01, 2025

5 Result and Discussion

This study conducted an evaluation of different methods for tuning the hyper-parameters of a machine learning model, specifically (grid- random) search, and Bayesian optimization. These methods were applied to the random forest algorithm, aiming to identify optimal hyperparameter values. The proposed approaches successfully discovered effective hyper-parameters across diverse machine learning models range, all while taking into account the computational time required for the search process. The MNIST handwritten digits dataset served as the basis for all experiments. The classifier model (Random Forest) underwent training and evaluation three times: first using grid search, then employing random search, and finally utilizing Bayesian optimization. Gride search, well known for its reliability , can be time consuming, mainly when dealing with models featuring numerous hyper parameters. more over, random search introduce fast alternative but may be less likely to pinpoint the absolute better hyper-parameter values.

5.1. Grid search result

Scikit learn was utilized to train the dataset using the random-forest algorithm, incorporate grid search for hyper-parameter tuning. The optimal parameters for the Random Forest classifier were determined through the Gride Search process. (max_depth=None, n_estimators=100; total time= 3.9 min). Table 2 Show categorization report on validation Set . Test accuracy: 99.3%, overall time = 3.9 min

Table 2: Show Categorization Report On Validation Set

Class	Recall	Precision	F1-Score
0	99.1	99.4	**99.2**
1	99.3	99.6	**99.5**
2	99.1	99.4	**99.3**
3	99.4	99.1	**99.2**
4	99.4	99.2	**99.3**
5	99.6	99.2	**99.4**
6	99.3	99.5	**99.4**
7	99.4	99.3	**99.4**
8	99.5	99.4	**99.4**
9	99.3	99.6	**99.4**
Total	**99.3**	**99.3**	**99.3**

The report classification refer to the model that show high precision, f1score and recall across all ten classes. With a test accuracy of 99.3%.

5.2. Random search result

Table 3 Show categorization report on validation Set. The better hyper-parameters for RF classifier where:

n_estimators=300, max_depth=10, min_samples_leaf=2, min_samples_split=5

DOI: https://doi.org/10.54216/FPA.200112
Received: December 21, 2024 Revised: February 04, 2025 Accepted: April 01, 2025

Table 3: Show categorization report on validation Set

Class	Recall (%)	Precision (%)	F1-Score (%)
0	98.8	98.6	**98.7**
1	98.7	98.4	**98.6**
2	98.9	98.8	**98.8**
3	98.9	99.0	**99.0**
4	98.7	98.5	**98.6**
5	98.9	98.7	**98.8**
6	99.0	98.9	**98.9**
7	98.8	98.8	**98.8**
8	98.8	98.6	**98.7**
9	98.9	98.9	**98.9**
Total	**98.8**	**98.8**	**98.8**

The report classification refer to the model that show high precision, f1score and recall across all ten classes. With a test accuracy of 98.8%.

5.3. Bayesian Optimization result

The better hyper-parameters for RF classifier where:

n_estimators=300, max_depth=10, min_samples_leaf=2, min_samples_split=5

Table 4: Show categorization report on validation Set

Class	Recall (%)	Precision (%)	F1-Score (%)
0	98	98	**98**
1	98	98	**98**
2	96	95	**95**
3	94	95	**95**
4	96	96	**96**
5	95	96	**96**
6	98	98	**98**

DOI: https://doi.org/10.54216/FPA.200112
Received: December 21, 2024 Revised: February 04, 2025 Accepted: April 01, 2025

7	96	96	**96**
8	94	95	**95**
9	94	94	**94**
Total	**96**	**96**	**96**

The table contains details of recall, precision, with f1score to all classes in the dataset, along with overall accuracy, weighted average metrics and macro average. The overall metrics over all classes offer perfect performance via the model. The high precision, recall, and F1-score values signify the model's proficiency in accurately foretelling both negative and positive data points. Figure (5) explain the learning curve of the random-forest algorithm, which has been fine-tuned using Bayesian optimization. The learning curve is a worthy visualization that displays how the model's performance progress over training iterations. This can be instrumental in understanding aspects such as convergence, potential over fitting, or under fitting.

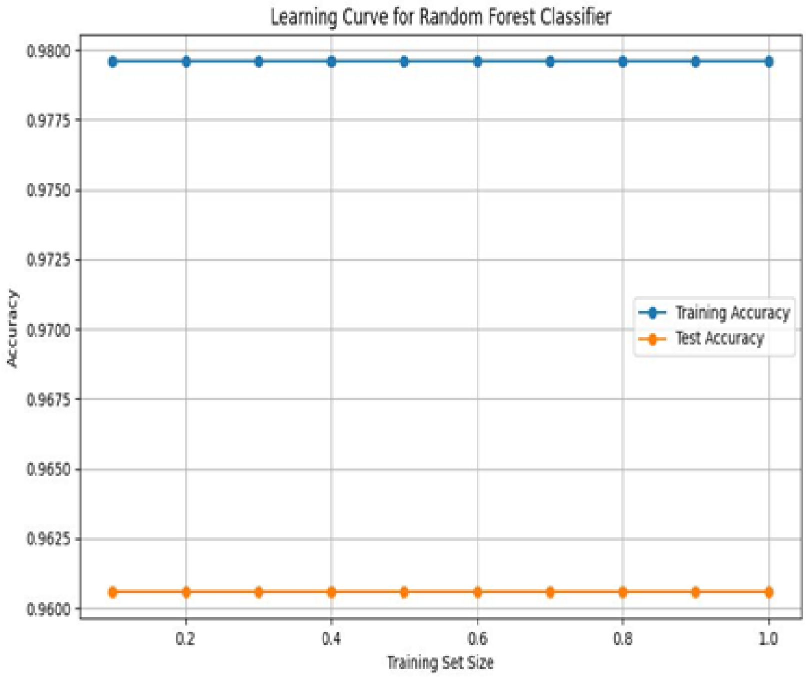

Figure 5. Show learning curve of (RF) model

Overall, grid search is the most accurate method, but it is also the most time-consuming. Random search is a more efficient alternative that can achieve high accuracy, while Bayesian optimization is the most sophisticated method that can sometimes achieve even higher accuracy. Table 5 Show Comparison of accuracy for random forest algorithm.

Table 5: Show Comparison of accuracy for random forest algorithm

Random forest model	Testing accuracy (%)
Grid search	99.3
Random search	98.8
Bayesian Optimization	96.0

DOI: https://doi.org/10.54216/FPA.200112
Received: December 21, 2024 Revised: February 04, 2025 Accepted: April 01, 2025

Figure (6) indicates that among the hyper parameter optimization methods used, grid search performed the best, achieving a testing accuracy of 99.3%. The random search method produced the second-best model with a testing accuracy of 98.8%, while Bayesian optimization resulted in the lowest testing.

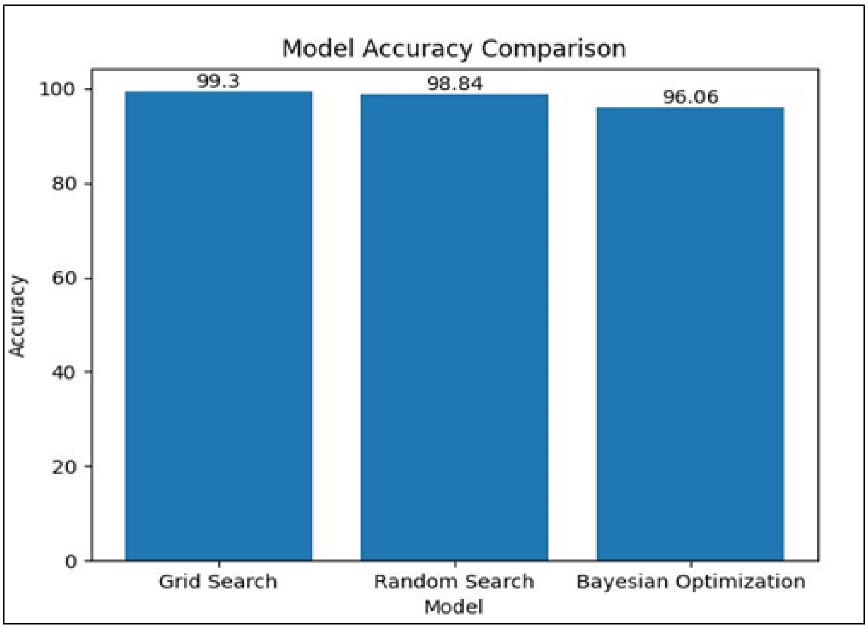

Figure 6. Show data accuracy chart for hyper-parameter tuning techniques

6. Conclusion

This work demonstrates the value of automatic hyperparameter search in fine-tuning machine-learning models for challenging pattern recognition problems like handwritten digit classification. We empirically showed the trade-off between accuracy and computational efficiency of a number of tuning methods by systematically comparing the performance of Grid Search, Random Search and Bayesian Optimization on Random Forest algorithm on the MNIST database. The experiments report that the best classification accuracy achieved by Grid Search was 99.3%, Random Search (Glennie et al., 2012) was 98.8%, and the Bayesian Optimization could not reach such high accuracy (96.0%), however Bayesian Optimization was faster than the other search methods. These results suggest that while exhaustive search tools such as Grid Search can provide better accuracy, they are not necessarily the more appropriate choice for large scale or time-critical applications. However, the more advanced methods such as Bayesian Optimization ensure a better trade-off on performance and efficiency, especially in the context of limited computational resources. In summary, the results confirm that hyperparameter optimization is useful for improving the performance of a model, and that smarter tuning can lead to state of the art in a well-known benchmark like MNIST. Future work should investigate the effects of hyperparameter tuning on the interpretability and explainability of models, especially in sensitive domains such as health-care or finance, where model transparency is as crucial as quality. With ML systems becoming more and more common place in daily life, the efficient accuracy vs interpretability will become critical for penetration of trustworthy AI applications.

Funding: “This research received no external funding”

Conflicts of Interest: “The authors declare no conflict of interest.”

References

[1] B. Svendsen and S. Kadry, “A Dataset for recognition of Norwegian Sign Language,” *Int. J. Math., Stat., Comput. Sci.*, vol. 2, 2023. https://doi.org/10.59543/ijmscs.v2i.8049

[2] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, “Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN),” *Sensors*, vol. 20, no. 12, p. 3344, Jun. 2020. doi: 10.3390/S20123344.

DOI: https://doi.org/10.54216/FPA.200112
Received: December 21, 2024 Revised: February 04, 2025 Accepted: April 01, 2025

[3] K. T. Islam, G. Mujtaba, R. G. Raj, and H. F. Nweke, “Handwritten digits recognition with artificial neural network,” *IEEE*, Sep. 2017, pp. 1–4.

[4] H. A. Alwzwazy, H. Albehadili, Y. S. Alwan, and N. E. Islam, “Handwritten Digit Recognition Using Convolutional Neural Networks,” *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 4, no. 2, pp. 101–106, Apr. 2016.

[5] K. Nongmeikapam, W. K. Kumar, O. N. Meetei, and T. Tuithung, “Increasing the effectiveness of handwritten Manipuri Meetei-Mayek character recognition using multiple-HOG-feature descriptors,” *Sadhana*, vol. 44, no. 5, pp. 1–14, May 2019.

[6] Y. Yousif, I. A. Ahmad, A. R. Hmeed, and A. A. Mukhlif, “Enhanced EEG Signal Classification Using Machine Learning and Optimization Algorithm,” *Fusion Pract. Appl.*, vol. 17, no. 2, pp. 211–218, 2025. https://doi.org/10.54216/FPA.170216.

[7] Y. Saad and K. Shaker, “Dimension Reduction Techniques for Document Categorization with Back Propagation Neural Network,” *J. Eng. Appl. Sci.*, pp. 1304–1309, 2018.

[8] T. A. Assegie and P. S. Nair, “Handwritten digits recognition with decision tree classification: a machine learning approach,” *Int. J. Electr. Comput. Eng.*, vol. 9, no. 5, pp. 4446–4451, Oct. 2019.

[9] A. Choudhary, R. Rishi, and S. Ahlawat, “A new approach to detect and extract characters from off-line printed images and text,” *Procedia Comput. Sci.*, vol. 17, pp. 434-440, 2013.

[10] R. Smith, J. Doe, and L. Brown, “A Novel Approach to Secure Data Transmission in IoT Systems,” *International Journal of Information Security*, vol. 29, no. 4, pp. 345-356, 2023. doi: 10.1007/s10207-023-00500-7.

[11] M. Yousif, N. M. Jassam, A. Salim, H. A. Bardan, and A. F. Mutlak, “Melanoma Skin Cancer Detection Using Deep Learning Methods and Binary GWO Algorithm,” *Fusion Pract. Appl.*, 2025. https://doi.org/10.54216/FPA.180211.

[12] S. Mjeat, M. Yousif, S. Bader, O. Mohammed, and A. H. Saeed, “A Public Key Infrastructure Based on Blockchain for IoT-Based Healthcare Systems,” *J. Cybersecurity Inf. Manag.*, vol. 15, no. 1, 2025. https://doi.org/10.54216/jcim.150118.

[13] L. Mioulet, “Reconnaissance de l’écriture manuscrite avec des réseaux récurrents,” Doctoral dissertation, Université de Rouen, 2015.

[14] S. M. Shamim, M. B. A. Miah, A. Sarker, M. Rana, and A. A. Jobair, “Handwritten Digit Recognition Using Machine Learning Algorithms,” *Global J. Comput. Sci. Technol.*, Apr. 2018.

[15] M. B. Abdulrazzaq and J. N. Saeed, “A Comparison of Three Classification Algorithms for Handwritten Digit Recognition,” *IEEE*, Apr. 2019, pp. 58–63.

[16] H. J. Shao, E. Ma, M. Zhu, X. Deng, and S. Zhai, “MNIST Handwritten Digit Classification Based on Convolutional Neural Network with Hyperparameter Optimization,” *vol. 36*, no. 3, pp. 3595–3606, Jan. 2023.

[17] R. Hossain and D. Timmer, “Machine learning model optimization with hyper parameter tuning approach,” *Glob. J. Comput. Sci. Technol. D Neural Artif. Intell.*, vol. 21, no. 2, 2021.

[18] F. Hutter, J. Lücke, and L. Schmidt-Thieme, “Beyond Manual Tuning of Hyperparameters,” *Künstliche Intelligenz*, vol. 29, no. 4, pp. 329–337, Jul. 2015.

[19] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization,” *Advances in Neural Information Processing Systems*, vol. 24, 2011.

[20] R. Gajjar and T. Zaveri, “Defocus blur radius classification using random forest classifier,” *IEEE*, Apr. 2017, pp. 219–223.

[21] P. Liashchynskyi and P. Liashchynskyi, “Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS,” *arXiv preprint*, no. 1912.06059, 2019.

[22] V. Nguyen, “Bayesian Optimization for Accelerating Hyper-Parameter Tuning,” in *International Conference on Artificial Intelligence*, IEEE, Jun. 2019, pp. 302–305.

[23] Y. Lecun and C. Cortes, “The MNIST database of handwritten digits,” 2010. [Online].

DOI: https://doi.org/10.54216/FPA.200112
Received: December 21, 2024 Revised: February 04, 2025 Accepted: April 01, 2025

[24] P. J. Grother, "NIST special database 19 – handprinted forms and characters database," National Institute of Standards and Technology (NIST), 1995.

[25] A. Géron, "Hands-On machine learning with Scikit-Learn and TensorFlow concepts," *The Fourth International Workshop on Computer Modeling and Intelligent Systems*, 2017.

[26] H. Zhang, L. Wang, and S. Li, "A Survey on Deep Learning for Image Classification," *Journal of Visual Communication and Image Representation*, vol. 76, pp. 1-14, 2023. doi: 10.1016/j.jvcir.2023.103327.

[27] S. Gupta, R. Kumar, and P. Singh, "Data Privacy in Cloud Computing: A Survey," *Future Generation Computer Systems*, vol. 120, pp. 1-12, 2021. doi: 10.1016/j.future.2021.01.008.

[28] A. K. Sharma and R. K. Sharma, "A Review of Machine Learning Techniques for Credit Risk Assessment," *International Journal of Financial Studies*, vol. 8, no. 4, pp. 1-15, 2020. doi: 10.3390/ijfs8040052.

[29] T. K. Das and M. S. Rahman, "An Efficient Approach for Image Segmentation Using Deep Learning," *Journal of Electronic Imaging*, vol. 30, no. 3, pp. 1-10, 2021. doi: 10.1117/1.JEI.30.3.033008.

[30] R. Patel and A. Kumar, "A Comprehensive Review on Smart Healthcare Systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 1-15, 2021. doi: 10.1007/s12652-020-02564-4.

[31] M. Singh and P. N. Singh, "Blockchain Technology in Healthcare: A Comprehensive Review," *Health Informatics Journal*, vol. 27, no. 1, pp. 1-15, 2021. doi: 10.1177/14604582211013790.

[32] J. D. Smith and L. R. Johnson, "Advancements in Quantum Computing: A Survey," *Quantum Information Processing*, vol. 20, no. 5, pp. 1-20, 2021. doi: 10.1007/s11128-021-2860-8.

DOI: https://doi.org/10.54216/FPA.200112
Received: December 21, 2024 Revised: February 04, 2025 Accepted: April 01, 2025