COMPARATIVE ANALYSIS OF HYPERPARAMETER TUNING IN 3D PRINTING

[1]Michael Ogunsanya, [2]Salil Desai

[1,2]North Carolina A&T State University. Greensboro, North Carolina, USA
Email: [1] maogunsanya@aggies.ncat.edu, [2]sdesai@ncat.edu

Abstract: This paper presents hyperparameter tuning techniques for a deep learning predictive model with applications in additive manufacturing processes. Bioprinting is an additive manufacturing process which utilizes biomaterials, cells, and growth factors to build functional tissue constructs for biomedical applications. In this research, we evaluate the hyperparameter space using grid search technique to tune the perceptron deep learning hyperparameters for optimal prediction of additive manufacturing outcomes. Hyperparameter entities include number of neurons, learning rate, and number of epochs to run machine learning models. Five input parameters and three output variables were evaluated for a typical additive manufacturing process. A comparative analysis is conducted to demonstrate improved runtime and lower root mean squared error for additive manufacturing predictive models. The results from this research are extensible to several additive manufacturing processes including 3D bioprinting.

Index terms: Additive Manufacturing, Bioprinting, Deep Learning, Grid Search, Hyperparameter tuning

I. INTRODUCTION

Additive manufacturing (AM) commonly referred as 3D printing is a layer-by-layer fabrication process [1], [2]. Additive manufacturing has entailed enormous freedom to build parts with complex geometries and freeform structures [4]. In recent years, 3D printing has been conducted in a variety of materials including polymers [5], ceramics, metals, composites, and biomaterials [5] to name a few. Additive manufacturing has evolved from prototyping jobs to functional parts with applications ranging from electronics [6], [7], biomedical devices [8], [9], energy appliances [10], [11], automotive components [12], [13] to tissue engineering constructs [14], [15].

With the proliferation of additive manufacturing across different industry sectors, several 3D service bureaus have been created [16]. Herein, the objective is to identify optimal 3D process for each part design based on its geometrical characteristics and user inputs. This includes the selection of appropriate materials, tolerances and 3D printing process that can fulfill user requirements. In 3D printing service bureaus involve multiple AM equipment that are connected using high speed networks which can share vital processing information in real-time. These cyber-physical systems utilize smart algorithms to allocate input parts to specific equipment [17], [18].

Bioprinting is an emerging field of 3D printing wherein cellular materials, growth factors, extracellular matrices are orchestrated to build functional biological constructs. However, unlike traditional 3D printing methods, bioprinting methods are highly sensitive to several factors [19]–[21]. These include biomaterial rheology, bioprinting process parameters and microenvironmental variables such as ambient temperature and humidity. In addition, variations in growth factor conditions and nanoscale topological stimuli can largely affect tissue growth. Researchers have used computational molecular models to aid the design and manufacturing of biological constructs [22], [23]. Further, machine learning models are being applied in 3D printing to improve printing outcomes [24]. It is imperative that these models be optimized for their prediction and classification accuracies to be viable for practical purposes [25]. Moreover, it is critical to balance model accuracy versus algorithm run times when implementing these models for real-time feedback and control.

Deep learning has been used in different works for prediction and classification problems in additive manufacturing processes. Despite the vast applications, little or no efforts have been focused on hyperparameter tuning that aim to improve the loss alongside reduction in chosen model runtime. This research focuses on optimizing both the loss and runtime by considering a simple perceptron model. Our approach enables the selection of alternate model architecture by varying the selected hyperparameters.

For hyperparameter tuning (sometimes referred to as hyperparameter optimization), Bayesian optimization and its variants have been used in the literature [26]–[29]. Other models include manual search [30], grid search [31]–[33], random search [33], [34], particle swarm optimization [35], [36], genetic algorithm [37]–[39], and other optimization methods.

The remainder of the paper are organized as follows: Section II gives a detailed methodology of our chosen hyperparameter tuning technique, Section III has the results and discussion, Section IV

concludes and gives an extension of our work in the field of additive manufacturing.

II. METHODOLODY

Grid search was used in this research because it is widely used and every hyperparameter combination of the search is considered hyperparameter compared to other methods. With grid search, it is easier to see how other different deep learning model configurations are generated from which optimal model configuration can be chosen based on some selection criteria. In our case, both model prediction loss and model runtime were chosen.

In this research, five input parameters were chosen for additive manufacturing processes based on prominent parameters found in the literature. These include layer thickness, build orientation, extrusion temperature, build temperature, and print speed. Three output parameters (e.g., dimensional accuracy, porosity, and tensile strength) were predicted using a simple perceptron deep learning model as shown in Figure 1. A simple perceptron with a single hidden layer was used an illustrative case to demonstrate our work. A multi-layer perceptron (MLP) and variation of neurons across each hidden layer is shown in [40].

In this research, hyperparameter optimization was performed using the grid search method. The model was coded using custom Python script. The first step included creating a search space for the selected hyperparameters. The chosen hyperparameters include the number of neurons [3, 5, 8], learning rate [0.001, 0.0001, 0.00001], and number of epochs [5000, 15000, 25000]. Thus, creating a total of $3^3 = 27$ hyperparameter combination for the simple perceptron deep learning model. The output metrics include model prediction losses and runtimes. A model with the least prediction loss and runtime was chosen. In most cases, choosing optimal hyperparameter combination might not be as straight forward due to multi-objective optimization. The desired model configuration lies around a balance between lower model loss and lower runtime. This is essential especially for domains where prediction or classification cannot be compromised. In some cases, either model prediction loss, runtime, or other objective functions may be considered.

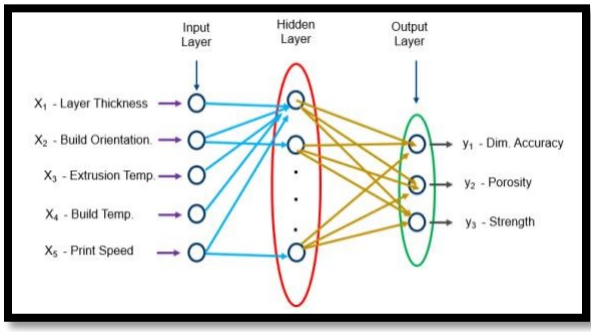

Figure 1: Perceptron deep learning architecture

Figure 2 shows a 3D printer which is based on Fused Deposition Modeling (FDM) technology. This 3D printer has a heating bed and sensors attached to capture real-time data during the printing operation. Furthermore, it has additional functionalities such as CNC milling and laser engraving capabilities.

Figure 2: Snap Maker 3D Printer with CNC milling and laser engraving capabilities.

Data was collected from the additive manufacturing process for the chosen input parameters. For the chosen process, a factorial design was performed at three different levels (low, medium, and high) with three replicates (n=3) for a total of 243 data points. The data imported into the chosen Python Environment were split into the training and test dataset in the ratio of 80 to 20, respectively. The split training and dataset were further split into input and output columns. At this stage, the hyperparameter list was created. Then, the perceptron deep learning model was trained using forward and backward propagation until the chosen stopping criteria was reached (in our case, it was the number of epochs). The trained model was evaluated with the test data and the root-mean-square error, RMSE, was computed as the model prediction loss. Also, the model runtime was recorded. Both the RMSE and the runtime were plotted to show where optimal or better model configurations lie.

The algorithm was run on a Lenovo Yoga 910-13IKB machine with the specifications: Intel® Core™ i7-7500U CPU @ 2.70GHz 2.90 GHz, installed RAM of 8.00 GB, and 64-bit operating system.

III. RESULTS

For each model configuration, the model was trained as stated in Section II. The learning status was tested by the prediction on the test data. If the prediction was all zeros or all having the same values for the test data, it clearly indicates that the model could not be trained at the given hyperparameter

combination, and it was disregarded. The remaining model configurations are potential candidates for the optimal selection based on their RMSE and runtime. Out of the 27 different model configurations for each output parameter prediction, only 18 were able to learn from the data at those hyperparameter combinations. Table 1 shows the prediction RMSE and runtime values at those model configuration where the perceptron deep learning model were able to learn from the training data for three output parameters. As can be seen lower learning rates (0.00001) and higher number of epochs (25000) for different neurons (3, 5 and 8) resulted in higher runtimes across all the three output variables. Thus, an optimal balance is to be attained between lower RMSE and runtime for different model configurations.

For the training dataset, all three output parameters learnt at the same hyperparameter combinations compared to when there are more hidden layers in the work of [40].

Table 1: RMSE and runtime values for the eighteen (18) perceptron deep learning model configurations for each of the three output parameters (dimensional accuracy, porosity, and tensile strength)

No of Neurons	Learning Rate	Epoch	HPO Combination	Dim_Acc		Porosity		Strength	
				RMSE	Runtime (s)	RMSE	Runtime (s)	RMSE	Runtime (s)
3	0.0001	5000	3-0.0001-5000	1.96	1.327319	3.19	1.301461	5.37	1.47904
3	1.00E-05	5000	3-0.00001-5000	1.9	2.070057	3.24	2.076302	5.39	2.257595
3	0.0001	15000	3-0.0001-15000	1.97	5.660765	3.19	5.848238	5.37	5.897032
3	1.00E-05	15000	3-0.00001-15000	1.92	7.462006	3.21	7.741441	5.38	7.71358
3	0.0001	25000	3-0.0001-25000	1.97	13.2233	3.19	13.82609	5.37	13.48801
3	1.00E-05	25000	3-0.00001-25000	1.92	16.17036	3.19	16.7026	5.37	16.28902
5	0.0001	5000	5-0.0001-5000	1.96	1.409673	3.19	1.425759	5.37	1.367751
5	1.00E-05	5000	5-0.00001-5000	1.9	2.239215	3.24	2.219268	5.39	2.145727
5	0.0001	15000	5-0.0001-15000	1.97	5.985354	3.19	6.08929	5.37	5.985843
5	1.00E-05	15000	5-0.00001-15000	1.92	7.893492	3.21	8.014984	5.37	7.895362
5	0.0001	25000	5-0.0001-25000	1.97	13.99194	2.15	14.16252	5.37	13.93173
5	1.00E-05	25000	5-0.00001-25000	1.92	17.02709	3.19	17.1575	5.37	16.99023
8	0.0001	5000	8-0.0001-5000	1.97	1.477919	3.19	1.500125	5.37	1.43581
8	1.00E-05	5000	8-0.00001-5000	1.9	2.386956	3.25	2.332991	5.39	2.257712
8	0.0001	15000	8-0.0001-15000	1.97	6.39182	3.19	6.329442	5.37	6.247316
8	1.00E-05	15000	8-0.00001-15000	1.92	8.435162	3.21	8.312044	5.38	8.264749
8	0.0001	25000	8-0.0001-25000	1.97	14.68035	3.19	14.71367	5.37	14.54946
8	1.00E-05	25000	8-0.00001-25000	1.92	17.84134	3.2	17.90441	5.37	17.67085

Figure 3 shows the RMSE and runtime parameters for dimensional accuracy output variable. Dimensional accuracy had the least prediction loss at 1.9 which was obtained at three different model configurations at 2s runtime. The best runtime occurred at 1.33s with higher RMSE of 1.96. At a learning rate of 0.001, all models were unable to learn from the training dataset. At a learning rate of 0.0001, learning was recorded and RMSE are either 1.96 or 1.97 and the runtime increased with as the number of epochs increased. Best RMSE and runtime combinations were obtained at 5000 epochs. The number of neurons in the hidden layer had no significant impact on the RMSE and runtime.

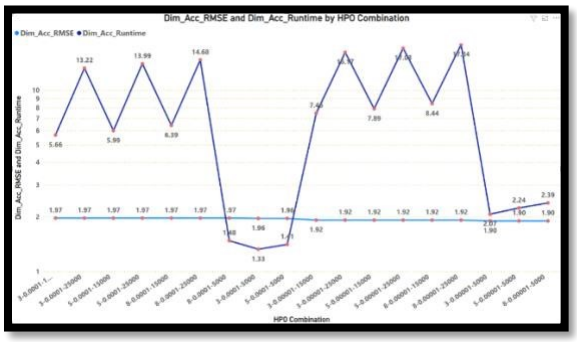

Figure 3: Plot of RMSE and Runtime against the 18 hyperparameter combination for dimensional accuracy

Figure 4 shows the RMSE and runtime parameters for the porosity output variable. Porosity had greater RMSE values compared to the RMSE values obtained from dimensional accuracy with most around 3.19. Learning rate plays an important role in predicting the porosity for different model configurations. A learning of 0.0001 gave an RMSE value of 3.19 at all instances with an increased runtime as the number of epochs increase. Although, the best model configuration with the least RMSE value is 2.15 but the runtime increased to about eight times compared to RMSE value of 3.19. Like dimensional accuracy, the number of neurons play no importance for different model configurations.

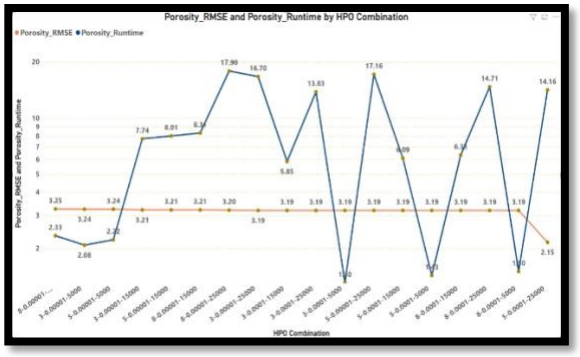

Figure 4: Plot of RMSE and Runtime against the 18 hyperparameter combination for porosity

Figure 5 shows the RMSE and runtime parameters for the tensile strength output variable. Strength prediction had the highest loss compared to the RMSE values from both dimensional accuracy and porosity. The RMSE hovered in the range 5.37 to 5.39. The lowest RMSE of 5.37 was obtained at smallest runtime of 1.37 seconds for the 5-0.0001-5000 model configuration. Also, the runtime increases with an increase with both learning rate and number of epochs. For the strength, number of neurons play a significant role although the RMSE value stays at 5.37 but optimal runtime was obtained with 5 neurons and at 1.37 s compared to 1.48 s with 3 neurons and 1.44 s with 8 neurons.

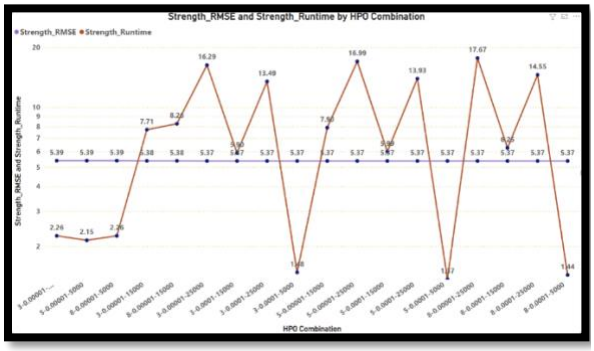

Figure 5: Plot of RMSE and Runtime against the 18 hyperparameter combination for tensile strength

Figure 6 shows the RMSE and runtime parameters for all the three output variables. The HPO combination of 5-0.0001-25000 gave the lowest RMSE for all three output variables. These include dimensional accuracy – 1.97, porosity – 2.45 and strength – 5.37. These combinatorial plots can aid in selecting optimal settings that can satisfy multiple output variables. However, these settings need to be validated with moderate runtimes for 3D printing process which may require real-time feedback. Furthermore, this model is applicable to 3D bioprinting which is highly sensitive to material, process and microenvironmental variables as compared to traditional 3D printing processes. Herein, a hyperparameter optimization would involve a function of multiple input-output variable combinations. The output variables can vary from bioprinting variables such as print accuracy to biological outcomes such as histology and cellular viability.

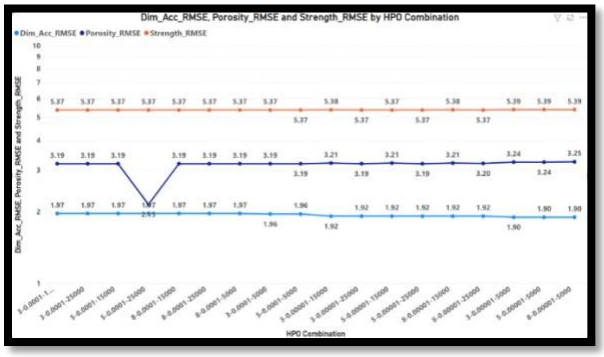

Figure 6: Plot of the 18 nodel configurations with their RMSE values for dimension accuracy, porosity, and tensile strength

CONCLUSION

This research explores a perceptron deep learning configuration by optimizing key hyperparameters. Five input parameters (layer thickness, build orientation, extrusion temperature, build temperature, and print speed) were considered to predict three outputs (dimension accuracy, porosity, and tensile strength). Data were generated in which 80% of the data were used to train the perceptron and tested on the remaining 20%. The grid search method was used on the hyperparameter search space to determine the optimal hyperparameter combination at both best model prediction loss of root-mean-square error and the lowest model runtime. Overall, the dimension accuracy perceptron was able to make prediction with the least RMSE value of 1.9 at a runtime of about 1.41 seconds given the hyperparameter combinations as 5 neurons in the hidden layer, learning rate as 0.0001, and a number of epochs to be 5000. For porosity, the RMSE value was 3.19 and a runtime of 1.30 s with a hyperparameter combination of 3 neurons, 0.0001 learning rate, and 5000 epochs. For dimension accuracy and porosity, number of neurons in the hidden has no significant effect on the RMSE value and runtime as it is for the strength prediction. Strength prediction has the greatest RMSE value of 5.37 at a runtime of 1.37 s. All the strength prediction RMSE values are either 5.37, 5.38, or 5.39 but at different runtimes which is directly proportional to the number of epochs.

Multi parameter HPO was performed yielding ranges of RMSE for dimensional accuracy, porosity, and tensile strength. Our work clearly indicates that it is insufficient to rely on the default or randomly choosing hyperparameter combination for a deep learning model. It also provides a hyperparameter design space for choosing different

model variants that yields better prediction loss and even at reduced runtime. This work will serve as foundation for additive manufacturing processes where quality cannot be compromised or processes that needs to strike a balance between prediction loss and runtime.

ACKNOWLDEGMENTS

The authors would like to express their gratitude for funding support from the National Science Foundation Grant (NSF Awards 1663128, 2100739, 2100850, 2200538) Center of Excellence in Product Design and Advanced Manufacturing at North Carolina A&T State University.

REFERENCES

[1] S. K. Parupelli and S. Desai, "A Comprehensive Review of Additive Manufacturing (3D Printing): Processes, Applications and Future Potential," *Am. J. Appl. Sci.*, vol. 16, no. 8, pp. 244–272, 2019, doi: 10.3844/ajassp.2019.244.272.

[2] S. Desai, "Methods and apparatus of manufacturing micro and nano-scale features," *Google Patents*, vol. US Patent, 2013, Accessed: Aug. 11, 2022. [Online]. Available: https://patents.google.com/patent/US8573757B2/en.

[3] S. Parupelli, S. D.-A. journal of applied sciences, and undefined 2019, "A comprehensive review of additive manufacturing (3d printing): processes, applications and future potential," *par.nsf.gov*, Accessed: Apr. 04, 2022. [Online]. Available: https://par.nsf.gov/biblio/10124832.

[4] G. Haeberle and S. Desai, "Investigating Rapid Thermoform Tooling Via Additive Manufacturing (3d Printing)," *Am. J. Appl. Sci.*, vol. 16, no. 8, pp. 238–243, Oct. 2019, doi: 10.3844/ajassp.2019.238.243.

[5] S. Desai, B. Bidanda, and P. J. Bártolo, "Emerging Trends in the Applications of Metallic and Ceramic Biomaterials," in *Bio-Materials and Prototyping Applications in Medicine*, Springer International Publishing, 2021, pp. 1–17.

[6] S. K. Parupelli and S. Desai, "Hybrid additive manufacturing (3D printing) and characterization of functionally gradient materials via in situ laser curing," *Int. J. Adv. Manuf. Technol.*, vol. 110, no. 1–2, pp. 543–556, Sep. 2020, doi: 10.1007/s00170-020-05884-9.

[7] S. K. Parupelli and S. Desai, "Understanding Hybrid Additive Manufacturing of Functional Devices," *Am. J. Eng. Appl. Sci.*, vol. 10, no. 1, pp. 264–271, Jan. 2017, doi: 10.3844/ajeassp.2017.264.271.

[8] F. K. Aldawood, S. X. Chang, and S. Desai, "Design and manufacture of a high precision personalized electron bolus device for radiation therapy," *Med. DEVICES SENSORS*, vol. 3, no. 6, Dec. 2020, doi: 10.1002/mds3.10077.

[9] E. Adarkwa, R. Kotoka, and S. Desai, "3D printing of polymeric Coatings on AZ31 Mg alloy Substrate for Corrosion Protection of biomedical implants," *Med. DEVICES SENSORS*, vol. 4, no. 1, Feb. 2021, doi: 10.1002/mds3.10167.

[10] M. Yang, Z. Xu, S. Desai, D. Kumar, and J. Sankar, "Fabrication of Micro Single Chamber Solid Oxide Fuel Cell Using Photolithography and Pulsed Laser Deposition," *J. Fuel Cell Sci. Technol.*, vol. 12, no. 2, 2015, doi: 10.1115/1.4029094.

[11] S. Desai, M. Yang, Z. Xu, and J. Sankar, "Direct Write Manufacturing of Solid Oxide Fuel Cells for Green Energy," *J. Environ. Res. Dev.*, vol. 8, no. 3, pp. 477–483, 2014.

[12] S. Desai and M. Lovell, "Computational fluid dynamics analysis of a direct write manufacturing process," *Int. J. Nanomanuf.*, vol. 3, no. 3, pp. 171–188, 2009, doi: 10.1504/IJNM.2009.027424.

[13] J. McKenzie, santosh kumar Parupelli, S. S. Desai, S. Parupelli, D. Martin, and S. Desai, "Additive manufacturing of multiphase materials for electronics," *researchgate.net*, 2017, Accessed: Aug. 11, 2022. [Online]. Available: https://www.researchgate.net/profile/Santosh-Kumar-Parupelli/publication/347440627_Additive_Manufacturing_of_Multiphase_Materials_for_Electronics/links/5fdbcf3945851553a0c6ef7d/Additive-Manufacturing-of-Multiphase-Materials-for-Electronics.pdf.

[14] A. Aljohani and S. Desai, "3D Printing of Porous Scaffolds for Medical Applications," *Am. J. Eng. Appl. Sci.*, vol. 11, no. 3, pp. 1076–1085, 2018, doi: 10.3844/ajeassp.2018.1076.1085.

[15] S. Desai and B. Harrison, "Direct-Writing of Biomedia for Drug Delivery and Tissue Regeneration," pp. 71–89, 2010, doi: 10.1007/978-1-4419-1395-1_5.

[16] H. Elhoone, T. Zhang, … M. A.-I. J. of, and undefined 2020, "Cyber-based design for additive manufacturing using artificial neural networks for Industry 4.0," *Taylor Fr.*, vol. 58, no. 9, pp. 2841–2861, May 2020, doi: 10.1080/00207543.2019.1671627.

[17] N. Almakayeel, S. Desai, S. Alghamdi, and M. R. N. M. Qureshi, "Smart Agent System for Cyber Nano-Manufacturing in Industry 4.0," *Appl. Sci.*, vol. 12, no. 12, p. 6143, Jun. 2022, doi: 10.3390/app12126143.

[18] H. Almakaeel, A. Albalawi, and S. Desai, "Artificial neural network based framework for cyber nano manufacturing," *Manuf. Lett.*, vol. 15, pp. 151–154, 2018, doi: 10.1016/j.mfglet.2017.12.013.

[19] I. Marquetti and S. Desai, "Adsorption Behavior of Bone Morphogenetic Protein-2 on a Graphite Substrate for Biomedical Applications," *Am. J. Eng. Appl. Sci.*, vol. 11, no. 2, pp. 1037–1044, 2018, doi: 10.3844/ajeassp.2018.1037.1044.

[20] I. Marquetti and S. Desai, "Molecular modeling the adsorption behavior of bone morphogenetic protein-2 on hydrophobic and hydrophilic substrates," *Chem. Phys. Lett.*, vol. 706, pp. 285–294, 2018, doi: 10.1016/j.cplett.2018.06.015.

[21] I. Marquetti and S. Desai, "Orientation effects on the nanoscale adsorption behavior of bone morphogenetic protein-2 on hydrophilic silicon dioxide," *RSC Adv.*, vol. 9, no. 2, pp. 906–916, 2019, doi: 10.1039/C8RA09165J.

[22] I. Marquetti and S. Desai, "Nanoscale

Topographical Effects on the Adsorption Behavior of Bone Morphogenetic Protein-2 on Graphite," *Int. J. Mol. Sci.*, vol. 23, no. 5, 2022, doi: 10.3390/ijms23052432.

[23] I. Marquetti and S. Desai, "An Atomistic Investigation of Adsorption of Bone Morphogenetic Protein-2 on Gold with Nanoscale Topographies," *Surfaces*, vol. 5, no. 1, pp. 176–185, 2022, doi: 10.3390/surfaces5010010.

[24] M. Ogunsanya, J. Isichei, S. Parupelli, … S. D.-P., and undefined 2021, "In-situ droplet monitoring of inkjet 3D printing process using image analysis and machine learning models," *Elsevier*, Accessed: Apr. 04, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2351978921000524.

[25] T. Akter, S. D.-M. & Design, and undefined 2018, "Developing a predictive model for nanoimprint lithography using artificial neural networks," *Elsevier*, Accessed: Apr. 04, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0264127518307597.

[26] P. I. Frazier, "Bayesian Optimization," *Recent Adv. Optim. Model. Contemp. Probl.*, pp. 255–278, Oct. 2018, doi: 10.1287/EDUC.2018.0188.

[27] K. Eggensperger *et al.*, "Towards an Empirical Foundation for Assessing Bayesian Optimization of Hyperparameters," *BayesOpt Work. @ NeurIPS*, vol. 15, pp. 1–5, 2013, Accessed: Aug. 04, 2022. [Online]. Available: https://www.cs.ubc.ca/~hoos/Publ/EggEtAl13.pdf.

[28] R. Martinez-Cantin, "BayesOpt: A Bayesian optimization library for nonlinear optimization, experimental design and bandits," *J. Mach. Learn. Res.*, vol. 15, pp. 3735–3739, 2015, Accessed: Aug. 04, 2022. [Online]. Available: https://www.jmlr.org/papers/volume15/martinezcantin14a/martinezcantin14a.pdf.

[29] Y. Yang, K. Deng, and M. Zhu, "Multi-level Training and Bayesian Optimization for Economical Hyperparameter Optimization," Jul. 2020, doi: 10.48550/arxiv.2007.09953.

[30] F. Hutter, J. Lücke, and L. Schmidt-Thieme, "Beyond Manual Tuning of Hyperparameters," *KI - Kunstl. Intelligenz*, vol. 29, no. 4, pp. 329–337, Jul. 2015, doi: 10.1007/s13218-015-0381-0.

[31] B. H. Shekar and G. Dagnew, "Grid search-based hyperparameter tuning and classification of microarray cancer data," 2019, doi: 10.1109/ICACCP.2019.8882943.

[32] R. Ghawi and J. Pfeffer, "Efficient Hyperparameter Tuning with Grid Search for Text Categorization using kNN Approach with BM25 Similarity," *Open Comput. Sci.*, vol. 9, no. 1, pp. 160–180, Jan. 2019, doi: 10.1515/comp-2019-0011.

[33] L. Zahedi, F. G. Mohammadi, S. Rezapour, M. W. Ohland, and M. H. Amini, "Search Algorithms for Automated Hyper-Parameter Tuning," *arxiv.org*, 2021, Accessed: Aug. 04, 2022. [Online]. Available: https://arxiv.org/abs/2104.14677.

[34] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012, Accessed: Jan. 18, 2022. [Online]. Available: https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.

[35] P. R. Lorenzo, J. Nalepa, M. Kawulok, L. S. Ramos, and J. R. Pastor, "Particle swarm optimization for hyper-parameter selection in deep neural networks," in *GECCO 2017 - Proceedings of the 2017 Genetic and Evolutionary Computation Conference*, Jul. 2017, vol. 8, pp. 481–488, doi: 10.1145/3071178.3071208.

[36] T. Kim, S. C.-2019 I. congress on evolutionary, and undefined 2019, "Particle swarm optimization-based CNN-LSTM networks for forecasting energy consumption," *ieeexplore.ieee.org*, Accessed: Aug. 04, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8789968/.

[37] A. S. Wicaksono and A. A. Supianto, "Hyper parameter optimization using genetic algorithm on machine learning methods for online news popularity prediction," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 12, pp. 263–267, 2018, doi: 10.14569/IJACSA.2018.091238.

[38] A. Gaspar, D. Oliva, E. Cuevas, D. Zaldívar, M. Pérez, and G. Pajares, "Hyperparameter Optimization in a Convolutional Neural Network Using Metaheuristic Algorithms," in *Studies in Computational Intelligence*, vol. 967, no. 6, 2021, pp. 37–59.

[39] H. Alibrahim and S. A. Ludwig, "Hyperparameter Optimization: Comparing Genetic Algorithm against Grid Search and Bayesian Optimization," in *2021 IEEE Congress on Evolutionary Computation, CEC 2021 - Proceedings*, 2021, pp. 1551–1559, doi: 10.1109/CEC45853.2021.9504761.

[40] M. Ogunsanya and S. Desai, "Predictive Modeling of Additive Manufacturing Process using Deep Learning Algorithm," in *Proceedings of the IISE Annual Conference & Expo 2022*, 2022.

★★★