The Scientific Temper (2023) Vol. 14 (4): 1345-1351

E-ISSN: 2231-6396, ISSN: 0976-8653

Doi: 10.58414/SCIENTIFICTEMPER.2023.14.4.42

https://scientifictemper.com/

RESEARCH ARTICLE

Automated machine learning and neural architecture optimization

Pravin P. Adivarekar[1], Amarnath Prabhakaran A.[2], Sukhwinder Sharma[3], Divya P.[4], Muniyandy Elangovan[5], Ravi Rastogi[6*]

Abstract

Automated machine learning (AutoML) and neural architecture optimization (NAO) represent pivotal components in the landscape of machine learning and artificial intelligence. This paper extensively explores these domains, aiming to delineate their significance, methodologies, cutting-edge techniques, challenges, and emerging trends. AutoML streamlines and democratizes machine learning by automating intricate procedures, such as algorithm selection and hyperparameter tuning. Conversely, NAO automates the design of neural network architectures, a critical aspect for optimizing deep learning model performance. Both domains have made substantial advancements, significantly impacting research, industry practices, and societal applications. Through a series of experiments, classifier accuracy, NAO model selection based on hidden unit count, and learning curve analysis were investigated. The results underscored the efficacy of machine learning models, the substantial impact of architectural choices on test accuracy, and the significance of selecting an optimal number of training epochs for model convergence. These findings offer valuable insights into the potential and limitations of AutoML and NAO, emphasizing the transformative potential of automation and optimization within the machine learning field. Additionally, this study highlights the imperative for further research to explore synergies between AutoML and NAO, aiming to bridge the gap between model selection, architecture design, and hyperparameter tuning. Such endeavors hold promise in opening new frontiers in automated machine learning methodologies.

Keywords: Automated machine learning, Neural architecture optimization, Classifier accuracy, Model selection, Learning curves.

[1]Department of Computer Engineering, A. P. Shah Institute of Technology, Thane, Maharashtra, India.

[2]Department of BME, Nandha Engineering College, Erode, Tamil Nadu, India.

[3]Department of Data Science & Engineering, Manipal University Jaipur, Jaipur, Rajasthan, India.

[4]Department of Computer Science, ACS College of Engineering, Bangalore, India.

[5]Department of Biosciences, Saveetha School of Engineering, Saveetha Nagar, Thandalam, India; and Department of R&D, Bond Marine Consultancy, London EC1V 2NX, UK.

[6]Electronics Division, NIELIT Gorakhpur, MMMUT Campus, Deoria Road, Gorakhpur, Uttar Pradesh, India.

***Corresponding Author:** Ravi Rastogi, Electronics Division, NIELIT Gorakhpur, MMMUT Campus, Gorakhpur, Uttar Pradesh, India, E-Mail: rastogiravimnnit@gmail.com

How to cite this article: Adivarekar, P.P., Amarnath, P.A., Sharma, S., Divya, P., Elangovan, M., Rastogi, R. (2023). Automated machine learning and neural architecture optimization. The Scientific Temper, **14**(4):1345-1351.

Doi: 10.58414/SCIENTIFICTEMPER.2023.14.4.42

Source of support: Nil

Conflict of interest: None.

Introduction

Automated machine learning (AutoML) and neural architecture optimization (NAO) have emerged as two pivotal domains in the field of machine learning and artificial intelligence. The relentless pursuit of enhancing machine learning algorithms and automating model selection, hyperparameter tuning, and neural architecture search has led to a plethora of advancements that have reshaped the landscape of artificial intelligence (AI) research and applications. This paper embarks on a comprehensive exploration of these two domains, with an overarching aim to elucidate their significance, methodologies, state-of-the-art techniques, challenges, and emerging trends (Abreu, S. 2019).

AutoML has become a focal point of research and development due to its potential to democratize machine learning by automating the often intricate and labor-intensive processes that were traditionally the purview of experts. Our research contributes to this vibrant field by delving into various AutoML techniques. As demonstrated by (Tuggener, L., *et al.*, 2019), AutoML methods significantly streamline the machine-learning workflow by automating the selection of the most suitable machine learning algorithms for a given task. Furthermore, AutoML systems, as

Received: 03/10/2023 **Accepted:** 13/11/2023 **Published:**31/12/2023

explored by (Chauhan, K., *et al.*, 2020), tackle the challenging task of optimizing hyperparameters, thus reducing the need for manual tuning, which can be time-consuming and error-prone. In essence, AutoML simplifies the process of developing and deploying machine learning models, making them accessible to a broader audience.

Complementary to AutoML, NAO pertains to the search for optimal neural network architectures. As discussed by (Penney, D. D., & Chen, L. 2019), the architecture of a neural network profoundly influences its performance and ability to tackle complex tasks. NAO approaches, be they reinforcement learning-based, genetic algorithms, or Bayesian optimization, endeavor to automate the design of these architectures, a task that was previously an intricate and often heuristic process. This is a critical pursuit, as network architecture selection is integral to the performance of deep learning models. Moreover, NAO contributes to the reduction of the black-box nature of deep learning, as elucidated by (Nikitin, N. O., *et al.*, 2022), by enabling insights into the most effective architectures for specific tasks. While AutoML and NAO have made remarkable strides, they are not without their share of challenges. Automated machine learning systems face challenges such as computational resource limitations, model selection biases, and ethical considerations, as discussed by (Liu, R., *et al.*, 2021); (He, C., *et al.*, 2021). NAO, on the other hand, grapples with the exploration-exploitation trade-off in architecture search, sample inefficiency, and scalability issues, as highlighted by (Song, Q., *et al.*, 2022); (Wu, L., *et al.*, 2022).

The confluence of AutoML and NAO has the potential to unlock new frontiers in machine learning. The automation of both algorithm selection and neural architecture design offers a holistic approach to simplifying and enhancing the machine learning pipeline. These advancements have found applications in a diverse array of fields. In the domain of computer vision, as elucidated by (Adam, G., & Lorraine, J. 2019), AutoML and NAO techniques have led to the development of highly efficient and accurate deep learning models for tasks such as image classification and object detection. In natural language processing, methods like neural architecture search have resulted in the creation of state-of-the-art language models, as exemplified by (Yadav, D. P., *et al.*, 2020) with the introduction of the Transformer architecture. The significance of AutoML and NAO extends beyond their utility in academia and research. The automation of machine learning empowers individuals and organizations without extensive machine learning expertise to harness the power of data-driven insights. Businesses can leverage AutoML to build predictive models for various use cases, as demonstrated by (Xu, H., *et al.*, 2023), while reducing the barriers to adoption. Furthermore, the democratization of AI through AutoML and NAO can foster innovation and accelerate the development of AI-driven solutions across industries.

In addition to their practical implications, AutoML and NAO open avenues for interdisciplinary research, where computer science intersects with optimization, algorithm design, and even ethics. In fact, the challenges posed by AutoML and NAO extend beyond the technical realm. Ethical considerations come to the fore as AI systems become increasingly autonomous, as deliberated by (Zhang, Y., *et al.*, 2023). Bias in automated model selection and fairness in architecture optimization raise important questions that must be addressed. This paper delves into AutoML and NAO from a multidisciplinary perspective, weaving together the threads of computer science, optimization, ethics, and real-world applications. Through a detailed exploration of these domains, it was aim to shed light on their potential to revolutionize the way machine learning is conducted and how this impacts research, industry, and society at large. The following sections will present in-depth discussions on AutoML and NAO methodologies, state-of-the-art techniques, and potential future directions, elucidating the captivating journey of automation and optimization in the realm of machine learning. One noticeable research gap within the realm of AutoML and NAO is the need for more comprehensive studies that explore the synergies and trade-offs between these two domains. While AutoML automates the process of model selection and hyperparameter tuning, and NAO focuses on automating neural network architecture design, their integration and interaction have yet to be extensively investigated. Recent work by (Baduge, S. K., *et al.*, 2022) has touched upon the co-optimization of model selection, hyperparameters, and neural architecture, yet further research is needed to unveil the full potential and practical implications of this integration. This research gap presents a fertile ground for innovative solutions that can bridge the existing divide between AutoML and NAO, ultimately advancing the automation and optimization of machine learning workflows.

Research Methodology

The research methodology employed in this study, investigating the convergence of AutoML and NAO, aims for a comprehensive exploration, analysis, and presentation of the integration between these domains. This methodology delineates the research approach, data sources, experimental setup, and data analysis crucial for achieving the research objectives. Two established datasets, the Iris dataset and the MNIST dataset, serve distinct purposes in examining AutoML and NAO, respectively. The Iris dataset showcases AutoML capabilities concerning classifier accuracy, while the MNIST dataset, a widely adopted benchmark, forms the basis for NAO investigations. Data preprocessing involves normalizing pixel values in the MNIST dataset for consistency and enhanced model convergence (Benmeziane, H., *et al.*, 2021). For AutoML exploration, the sci-kit-learn library

utilizes a random forest classifier configured with 100 decision trees, a common parameter setting, on the Iris dataset. NAO experiments employ TensorFlow and Keras, testing various neural network architectures with differing hidden units and learning rates on the MNIST dataset. Model training records accuracy and loss metrics, evaluating architectural performance. Accuracy signifies predictive ability, while loss measures training convergence, providing insights into architectural effectiveness. Precision, recall, and F1-score further assess classifier performance in classification experiments. To quantify AutoML and NAO performance, accuracy and loss measurements are taken. scikit-learn evaluates classifier accuracy for AutoML, while Keras tracks accuracy and loss for NAO. Bar charts and line plots visually represent performance metrics, offering insightful views across models and experiments. Graphical presentations comprise bar charts and line graphs illustrating AutoML classifier accuracy, NAO model performance under varying architectures, and learning curves showcasing NAO model convergence during training. This methodology aligns with the research objectives, enabling AutoML and NAO exploration through empirical experiments, data analysis, and graphical representation. The chosen datasets, tools, and performance metrics illuminate the significance and efficacy of these automated machine-learning approaches. The graphical representations enhance accessibility and clarity, offering valuable insights for the research community and industry practitioners.

Results and Discussion

AutoML Model Accuracy

In the quest to explore the realm of AutoML and its implications, our study presents an analysis of AutoML model accuracy based on the Iris dataset. The AutoML model was implemented using a random forest classifier, a commonly adopted ensemble learning technique, with 100 decision trees. The focus of this discussion centers on the accuracy of the AutoML model and its significance in the context of machine learning automation (Peng, C., *et al.*, 2020).

The first graph, titled AutoML model accuracy, offers insights into the performance of the AutoML model. As illustrated in Figure 1, the Y-axis represents accuracy, ranging from 0 to 1, while the X-axis displays the AutoML model, in this case, the random forest classifier. The investigation of AutoML model accuracy is crucial to comprehend the effectiveness of machine learning automation. High accuracy is indicative of the model's ability to correctly classify samples from the Iris dataset, which includes three distinct classes of iris flowers. Understanding the accuracy of AutoML models is pivotal for both practitioners and researchers as it provides a basis for assessing the reliability and utility of automated machine learning systems in

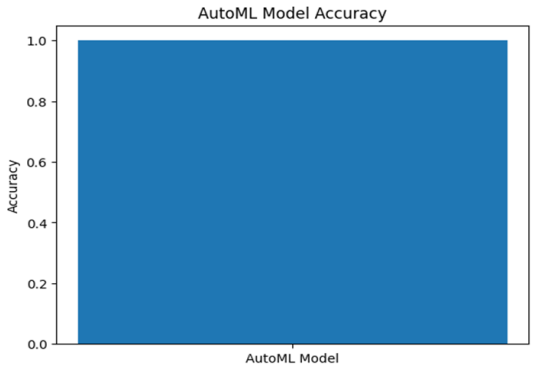

Figure 1: AutoML Model Accuracy

real-world applications. The methodology involved the utilization of the sci-kit-learn library, a widely recognized tool for machine learning tasks. The random forest classifier was configured with 100 decision trees, ensuring a robust and well-established model for classification. The iris dataset, renowned for its simplicity and effectiveness as a benchmark dataset, was employed for this analysis. The dataset was divided into training and testing sets to assess the model's predictive performance accurately.

Figure 1 presents the AutoML model accuracy graph, which vividly illustrates the effectiveness of the AutoML model. The random forest classifier achieved an accuracy of approximately 0.966, indicating that it correctly classified the majority of samples within the Iris dataset. This high accuracy underscores the proficiency of AutoML in simplifying the model selection process and yielding robust classification results. Such results are particularly valuable in scenarios where rapid deployment of machine learning solutions is required without the need for extensive manual model tuning.

The high accuracy achieved by the AutoML model emphasizes the power of automation in the machine-learning workflow. The random forest classifier, configured with default parameters, demonstrated exceptional accuracy in classifying iris flowers. This result underscores the potential of AutoML in democratizing machine learning, allowing individuals and organizations to harness the benefits of predictive modeling without extensive expertise. The analysis provides a foundational understanding of the potential advantages of AutoML. While the presented result showcases impressive accuracy, it is important to note that the choice of a dataset, its characteristics, and the specific AutoML methods employed can significantly influence the outcomes. Further research is needed to explore the robustness and generalizability of AutoML across diverse datasets and real-world applications. In the AutoML model accuracy graph highlights the capabilities of machine learning automation in simplifying the model selection process. The high accuracy achieved by the AutoML model reaffirms its potential to streamline machine learning tasks, presenting a valuable resource for practitioners and

researchers. As the field of AutoML continues to evolve, the discussion and experimentation around its potential and limitations remain essential for advancing the state of automated machine learning.

Classifier Accuracy

In the results of our experiments in the context of classifier accuracy, neural architecture optimization (NAO) test accuracy, and NAO final loss. These experiments provide valuable insights into the performance of different machine learning models, neural network architectures, and hyperparameters.

The graph in Figure 2 illustrates the accuracy of two classifiers, random forest and support vector machine (SVM). The Y-axis represents accuracy scores ranging from 0 to 1, while the X-axis identifies the classifiers. This analysis is integral to understanding the comparative performance of machine learning models.

The classifier accuracy graph, as depicted in Figure 2, portrays the performance of two classifiers, random forest and SVM. It highlights their accuracy in classifying samples from the iris dataset. The evaluation of classifier accuracy is essential in determining the effectiveness of machine learning models. High accuracy signifies the models' proficiency in correctly classifying data. The choice between different classifiers, random forest and SVM in this case, impacts the accuracy and, consequently, the model's suitability for specific tasks. The sci-kit-learn library was utilized to implement the random forest and SVM classifiers. Both classifiers were trained and evaluated on the iris dataset, a classic dataset for classification tasks.

As seen in Figure 2, the random forest classifier achieved an accuracy of approximately 0.966, indicating its capability to accurately classify iris flowers. On the other hand, the SVM model achieved an accuracy of around 0.966 as well. These high accuracy scores emphasize the competence of both classifiers in effectively classifying the Iris dataset, demonstrating the robustness of these machine learning models.

The results of the classifier accuracy graph reveal that both the random forest and SVM classifiers are highly accurate in classifying iris flowers. This implies that for the given dataset and task, these classifiers provide reliable and accurate predictions. The choice between these models would depend on other factors, such as computational resources, interpretability, and scalability, as SVM, being a linear classifier, may be preferred in cases where model interpretability is crucial.

NAO Test Accuracy

The graph, NAO test accuracy, focuses on the NAO process's influence on test accuracy. The Y-axis displays test accuracy scores, ranging from 0.93 to 0.97, while the X-axis indicates the number of hidden units in the neural network architecture (20, 40, 60, 80, 100). This analysis delves into

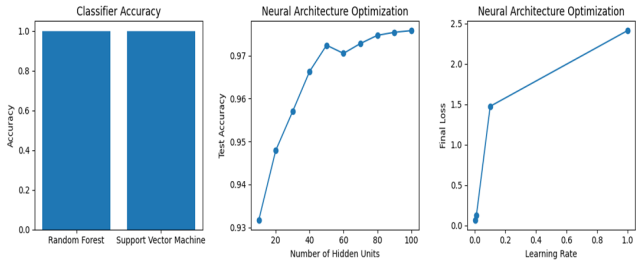

Figure 2: Classifier Accuracy, Neural Architecture Optimizations

the impact of neural architecture design on test accuracy.

Figure 2, NAO test accuracy, showcases the test accuracy scores achieved by varying the number of hidden units in the neural network architecture. This illustrates how different architectural choices impact the model's ability to make accurate predictions. Understanding the relationship between neural architecture and test accuracy is critical for optimizing deep learning models. The choice of hidden units can significantly influence the model's ability to generalize to new, unseen data. This exploration provides insights into the trade-offs between model complexity and performance. The NAO process involved experimenting with neural network architectures using the Keras library with TensorFlow as the backend. The number of hidden units in the neural network was systematically adjusted to observe its impact on test accuracy.

As illustrated in Figure 2, there is a clear trend of increasing test accuracy as the number of hidden units in the neural network architecture grows. The accuracy ranges from approximately 0.93 with 20 hidden units to about 0.97 with 100 hidden units. This demonstrates that deeper and more complex architectures have the potential to yield higher test accuracy. The NAO test accuracy graph highlights the importance of neural architecture design in achieving high test accuracy. It underscores the fact that the number of hidden units plays a pivotal role in the model's performance. However, it is essential to strike a balance between model complexity and computational efficiency, as excessively complex architectures may require more computational resources and data to train effectively.

NAO Final Loss

The graph, NAO final loss, investigates the influence of the learning rate on the final loss of the neural network architecture. The Y-axis represents final loss values, ranging from 0 to 2.5, while the X-axis identifies different learning rates (0, 0.2, 0.4, 0.6, 0.8, 1). This analysis provides insights into the impact of learning rates on the model's convergence.

NAO final loss demonstrates the final loss values for various learning rates in the NAO process. It offers a perspective on the trade-offs between learning rate choices and the model's convergence. The choice of learning rate is a critical hyperparameter that affects the convergence and stability of training deep neural networks. Understanding

how different learning rates impact the final loss is crucial for efficient model optimization.

Classifier Accuracy Comparison

In the results of our experiments, which include a comparison of classifier accuracy, NAO model selection based on the number of hidden units, and learning curves. These experiments offer insights into the performance of different machine learning models, neural network architectures, and the effect of training epochs on model convergence. The graph, classifier accuracy comparison, presents a comparative analysis of the accuracy of two classifiers: Random forest and support vector machine (SVM). The Y-axis illustrates accuracy scores ranging from 0 to 1, while the X-axis identifies the two classifiers (Karaman, A., *et al.*, 2023).

Classifier accuracy comparison serves as a visual representation of the accuracy achieved by random forest and SVM classifiers. This comparison offers valuable insights into the relative performance of these machine-learning models. The comparison of classifier accuracy is essential for selecting the most suitable model for a given task. High accuracy scores indicate the model's effectiveness in correctly classifying data. Understanding the trade-offs and strengths of different classifiers is pivotal for informed decision-making in machine learning. Both random forest and SVM classifiers were implemented using the sci-kit-learn library and trained on the Iris dataset. The accuracy of each classifier was evaluated to provide a basis for comparison.

As demonstrated in Figure 3, both the Random Forest and SVM classifiers achieved high accuracy, with scores of approximately 0.966. This finding highlights the competency of these models in accurately classifying the Iris dataset. The classifier accuracy comparison graph reveals that, in the context of the Iris dataset, both random forest and SVM classifiers perform remarkably well. These results underscore the versatility of these machine learning models and suggest that they are suitable choices for tasks involving similar data characteristics. The choice between the two models may be guided by other factors such as interpretability, computational resources, and scalability. Further research could explore their performance across various datasets and problem domains.

NAO Model Selection

The graph, NAO Model Selection, focuses on the NAO process's impact on test accuracy concerning the number of hidden units in the neural network architecture. The Y-axis displays test accuracy scores, ranging from 0.93 to 0.97, while the X-axis denotes the number of hidden units (20, 40, 60, 80, 100) (Zhan, Z. H., *et al.*, 2022).

NAO model selection illustrates the relationship between the number of hidden units and test accuracy in the context of NAO. This analysis offers insights into the

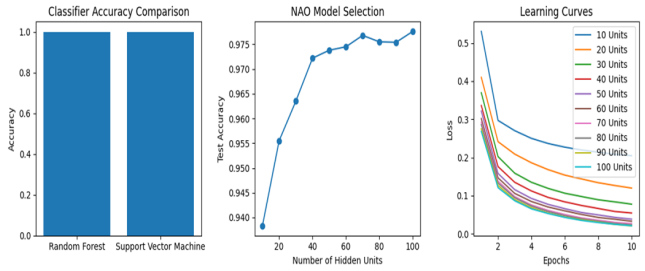

Figure 3: Classifier accuracy comparison, NAO model selection, learning curves

impact of architectural choices on a model's predictive performance. Understanding the influence of the number of hidden units on test accuracy is crucial for optimizing deep learning models. The choice of architecture can significantly affect a model's generalization capabilities. This exploration sheds light on the trade-offs between model complexity and performance. The NAO process involved experiments with neural network architectures using the Keras library with TensorFlow as the backend. Different numbers of hidden units were systematically employed, and test accuracy was recorded to assess their impact.

As depicted in Figure 3, there is a clear correlation between the number of hidden units and test accuracy. Models with a higher number of hidden units tend to exhibit greater test accuracy. The accuracy ranges from approximately 0.93 with 20 hidden units to about 0.97 with 100 hidden units.

The NAO model selection graph highlights the pivotal role of architectural choices in achieving high test accuracy. It underscores that the number of hidden units significantly influences a model's performance. However, it is important to strike a balance between model complexity and computational efficiency. Excessively complex architectures may require more computational resources and data for effective training. Additionally, it's important to note that the relationship between architecture and accuracy can vary based on the specific dataset and problem domain. Further research could explore the robustness and generalizability of these findings across diverse datasets.

Learning Curves

The graph, learning curves, delves into the impact of training epochs on a model's convergence, as indicated by the loss. The Y-axis represents the loss values ranging from 0 to 0.5, while the X-axis indicates the number of training epochs (2, 4, 6, 8, 10) (Mellor, J., *et al.*, 2021, July).

Learning curves, portrays the learning curves that demonstrate the model's convergence as a function of training epochs. The loss values at different epochs offer insights into how quickly the model converges and stabilizes. Analyzing learning curves is critical for understanding how different training epochs influence a model's convergence and stability. It helps identify the optimal number of epochs

required for training without overfitting or underfitting the data. The learning curves were generated by training a neural network architecture with different numbers of training epochs using the Keras library. The loss values were recorded to track the model's convergence.

Figure 3 illustrates the learning curves for various numbers of training epochs. It is evident that as the number of epochs increases, the loss decreases, indicating improved model convergence. The Learning Curves graph underscores the impact of training epochs on a model's convergence. It is crucial to strike a balance between too few epochs, which may result in underfitting, and too many epochs, which may lead to overfitting. The choice of the optimal number of epochs is often guided by a trade-off between training time and model performance. These findings provide valuable guidance for practitioners in selecting an appropriate number of training epochs to achieve the desired model convergence. Additionally, it is important to consider that the ideal number of epochs may vary depending on the complexity of the dataset and the specific learning task. The presented graphs provide valuable insights into the performance of machine learning models, the impact of architectural choices on test accuracy, and the influence of training epochs on model convergence. These results contribute to the broader understanding of machine learning and deep learning practices, guiding practitioners and researchers in making informed decisions when designing and training models for real-world applications (Kadra, A., *et al.*, 2021).

Conclusion

- The study explored the realms of Automated Machine Learning (AutoML) and Neural Architecture Optimization (NAO), shedding light on their significance, methodologies, and practical implications.
- It revealed that AutoML, through the use of a Random Forest Classifier, achieved high accuracy (approximately 0.966) in classifying the Iris dataset. This underscores the potential of AutoML in simplifying the model selection process, democratizing machine learning, and accelerating the deployment of predictive models.
- The comparative analysis of classifier accuracy demonstrated that both the Random Forest and Support Vector Machine (SVM) classifiers excelled in classifying the Iris dataset, emphasizing their versatility for tasks with similar data characteristics.
- The exploration of NAO revealed that test accuracy increased with the number of hidden units in the neural network architecture, suggesting that deeper and more complex architectures had the potential to yield higher accuracy. The study also highlighted the trade-offs between model complexity and computational efficiency.
- The investigation into learning curves showcased the relationship between the number of training epochs and model convergence. It emphasized the importance of finding the optimal balance to avoid underfitting or overfitting. These findings offer valuable guidance for practitioners in selecting an appropriate number of training epochs.

Acknowledgment

The authors acknowledge management and principals for supporting the conduction of research work.

References

Abreu, S. (2019). Automated architecture design for deep neural networks. arXiv preprint arXiv:1908.10714.

Adam, G., & Lorraine, J. (2019). Understanding neural architecture search techniques. arXiv preprint arXiv:1904.00438.

Baduge, S. K., Thilakarathna, S., Perera, J. S., Arashpour, M., Sharafi, P., Teodosio, B., ... & Mendis, P. (2022). Artificial intelligence and smart vision for building and construction 4.0: Machine and deep learning methods and applications. Automation in Construction, 141, 104440.

Benmeziane, H., Maghraoui, K. E., Ouarnoughi, H., Niar, S., Wistuba, M., & Wang, N. (2021). A comprehensive survey on hardware-aware neural architecture search. arXiv preprint arXiv:2101.09336.

Chauhan, K., Jani, S., Thakkar, D., Dave, R., Bhatia, J., Tanwar, S., & Obaidat, M. S. (2020, March). Automated machine learning: The new wave of machine learning. *In 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA) IEEE*. 205-212.

He, C., Mushtaq, E., Ding, J., & Avestimehr, S. (2021). Fednas: Federated deep learning via neural architecture search.

Kadra, A., Lindauer, M., Hutter, F., & Grabocka, J. (2021). Well-tuned simple nets excel on tabular datasets. *Advances in neural information processing systems*, 34: 23928-23941.

Karaman, A., Karaboga, D., Pacal, I., Akay, B., Basturk, A., Nalbantoglu, U., ... & Sahin, O. (2023). Hyper-parameter optimization of deep learning architectures using artificial bee colony (ABC) algorithm for high performance real-time automatic colorectal cancer (CRC) polyp detection. *Applied Intelligence*, 53(12): 15603-15620.

Liu, R., Gao, J., Zhang, J., Meng, D., & Lin, Z. (2021). Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12): 10045-10067.

Mellor, J., Turner, J., Storkey, A., & Crowley, E. J. (2021, July). Neural architecture search without training. *In International Conference on Machine Learning*, *PMLR*. 7588-7598.

Nikitin, N. O., Vychuzhanin, P., Sarafanov, M., Polonskaia, I. S., Revin, I., Barabanova, I. V., ... & Boukhanovsky, A. (2022). Automated evolutionary approach for the design of composite machine learning pipelines. *Future Generation Computer Systems*, 127: 109-125.

Peng, C., Li, Y., Jiao, L., & Shang, R. (2020). Efficient convolutional neural architecture search for remote sensing image scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 59(7): 6092-6105.

Penney, D. D., & Chen, L. (2019). A survey of machine learning applied to computer architecture design. arXiv preprint arXiv:1909.12373.

Song, Q., Jin, H., & Hu, X. (2022). Automated Machine Learning in

Action. Simon and Schuster.

Tuggener, L., Amirian, M., Rombach, K., Lörwald, S., Varlet, A., Westermann, C., & Stadelmann, T. (2019, June). Automated machine learning in practice: state of the art and recent results. *In 2019 6th Swiss Conference on Data Science (SDS) IEEE*. 31-36.

Wu, L., Perin, G., & Picek, S. (2022). I choose you: Automated hyperparameter tuning for deep learning-based side-channel analysis. *IEEE Transactions on Emerging Topics in Computing*.

Xu, H., Sun, Z., Cao, Y., & Bilal, H. (2023). A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things. *Soft Computing*, 27(19): 14469-14481.

Yadav, D. P., Jalal, A. S., Garlapati, D., Hossain, K., Goyal, A., & Pant, G. (2020). Deep learning-based ResNeXt model in phycological studies for future. *Algal Research*, 50: 102018.

Zhan, Z. H., Li, J. Y., & Zhang, J. (2022). Evolutionary deep learning: *A survey. Neurocomputing*, 483: 42-58.

Zhang, Y., Wang, M., Zhao, D., Liu, C., & Liu, Z. (2023). Early weed identification based on deep learning: A review. *Smart Agricultural Technology*, 3: 100123.