Artificial Intelligence Studies 8(2) 138-147 Research Article/Araştırma Makalesi

A I S ARTIFICIAL INTELLIGENCE STUDIES

CNN Hyperparameter Optimization for MNIST Dataset with Metaheuristic Algorithms

Gülistan Arslan[*a], **Hasan Temurtaş**[b]

ABSTRACT

In this study, we used nature-inspired metaheuristic algorithms for hyperparameter optimization, a key problem in deep learning. Specifically, the Grey Wolf Optimization (GWO) and Harris Hawk Optimization (HHO) algorithms were comparatively evaluated on the MNIST dataset, which is widely used for handwritten digit classification. The main objective of the study was to achieve high classification accuracy while simultaneously keeping the model structure as simple and computational cost low as possible. In this study, critical hyperparameters such as the number of layers, number of neurons, learning rate, dropout rate, and batch size were optimized. The findings show that both algorithms achieve high accuracy rates, but HHO, with a test accuracy of 98.1%, surpasses GWO's performance of 97.94%. Importantly, HHO achieved this success with fewer layers, a lower epoch count, and minimal regularization techniques. This demonstrates the advantage of HHO, especially under limited hardware resources and time constraints. In conclusion, our proposed study highlights that GWO and HHO algorithms provide effective solutions in hyperparameter optimization; moreover, HHO stands out with its low computational cost and high generalization ability.

[a,*] Kütahya Dumlupınar University,
Faculty of Engineering,
Department of Computer Engineering,
43100 – Kütahya, Türkiye
ORCID: 0000-0001-6498-1635

[b] Kütahya Dumlupınar University,
Faculty of Engineering,
Department of Computer Engineering,
43100 – Kütahya, Türkiye
ORCID: 0000-0001-6738-3024

[*] Corresponding author.
e-mail: gulistan.arslan@dpu.edu.tr

Keywords: Hyperparameter Optimization, Convolutional Neural Networks, Metaheuristic Algorithms, MNIST Dataset

Anahtar Kelimeler: Hiperparametre Optimizasyonu, Evrişimli Sinir Ağları, Metasezgisel Algoritmalar, MNIST Veri Kümesi

Submitted: 23.10.2025
Revised: 22.12.2025
Accepted: 25.12.2025

doi:10.30855/ais.2025.08.02.04

Metasezgisel Algoritmalar ile MNIST Veri Seti İçin CNN Hiperparametre Optimizasyonu

ÖZ

Bu çalışmada, derin öğrenmede önemli bir problem olan hiperparametre optimizasyonu için doğadan ilham alan metasezgisel algoritmalar kullandık. Özellikle, Grey Wolf Optimization GWO) ve Harris Hawk Optimization (HHO) algoritmaları, el yazısıyla yazılmış rakamların sınıflandırılmasında yaygın olarak kullanılan MNIST veri kümesi üzerinde karşılaştırmalı olarak değerlendirildi. Çalışmanın temel amacı, model yapısını mümkün olduğunca basit ve hesaplama maliyetini düşük tutarken aynı anda yüksek sınıflandırma doğruluğu elde etmekti. Bu çalışmada, katman sayısı, nöron sayısı, öğrenme oranı, bırakma oranı ve parti boyutu gibi kritik hiperparametreler optimize edildi. Bulgular, her iki algoritmanın da yüksek doğruluk oranlarına ulaştığını, ancak %98,1'lik bir test doğruluğu ile HHO'nun GWO'nun %97,94'lük performansını geçtiğini göstermektedir. Daha da önemlisi, HHO bu başarıyı daha az katman, daha düşük dönem sayısı ve minimum düzenleme teknikleriyle elde etmiştir. Bu, özellikle sınırlı donanım kaynakları ve zaman kısıtlamaları altında HHO'nun avantajını göstermektedir. Sonuç olarak, önerilen çalışmamız GWO ve HHO algoritmalarının hiperparametre optimizasyonunda etkili çözümler sağladığını vurgulamaktadır; ayrıca HHO düşük hesaplama maliyeti ve yüksek genelleme yeteneği ile öne çıkmaktadır.

To cite this article: A.Gulistan , T. Hasan, "*CNN Hyperparameter Optimization for MNIST Dataset with Metaheuristic Algorithms* , vol.8, no.2, pp. 138-147, 2025. doi:10.30855/ais.2025.08.02.04

1. Introduction

One of the basic blocks of the modern technology includes machine learning and deep learning models [1]. These models have made notable theoretical and practical progress especially in problems like image processing, natural language processing, bioinformatics and robotics [2]. In particular, deep learning has shown that it can be used to extract valuable features out of sophisticated data structures [3]. Convolutional neural networks (CNNs) have been popular among deep learning algorithms because it offers a high level of accuracy and performance in visual data [4]. The big layered CNNs can get impressive performances even on large datasets [5]. Nonetheless, the correct choice of hyperparameters when designing the model is one of the most important factors that affect the performance of CNN-based models [6]. Hyperparameters are basic elements that influence the architecture of a CNN model as well as the learning procedure [7]. These include the number of layers. Other important parameters are the size and number of filters in each layer, dropout rate, learning rate, batch size, and number of epochs. Appropriate choices of hyperparameters increase the learning capacity and generalization of the model; conversely, inappropriate configurations may decrease the accuracy and adversely affect performance of a model [8],[9]. As an example, a small learning rate can cause the convergence time to be longer, or a very large learning rate can cause the model to become unstable and fail to reach the optimal solution. Likewise, high dropout rates can result in overfitting the information, whereas low dropout rates will result in underfitting. Thus, the ability to optimize hyperparameters is essential to increase the performance of CNN-based models [10]. One of the most important strategies is hyperparameter optimization that is created to address the challenges connected with deep learning models [11]. A number of methods have been offered in literature, such as random search, grid search, and Bayesian optimization [12]. Nevertheless, these methods are also time constrained and computationally costly, particularly when working with large and complicated hyperparameter spaces. This reason has led to the emergence of metaheuristic optimization algorithms as an alternative with importance since they are capable of giving faster and efficient solutions in high-dimensional search spaces [13]. In recent studies, metaheuristic algorithms have also been successfully integrated with classical machine learning methods to overcome local optimum limitations and improve optimization performance, demonstrating their effectiveness beyond deep learning architectures [14] [15].Recent studies have also shown that learning-based optimization approaches, such as reinforcement learning, can dynamically adapt to changing system conditions and resource constraints, offering efficient optimization solutions for intelligent systems operating under limited computational and energy resources [16][17].In this paper, we suggest a method of CNN hyperparameter optimization based on the metaheuristic optimization algorithms, Grey Wolf Optimization (GWO) algorithm is compared with Harris Hawks Optimization (HHO). The two algorithms have a number of benefits over traditional optimization methods. GWO was inspired by the social hunting methods of grey wolves because it can quickly find and chase global optima, as well as attack them [18]. HHO is based on the principles of chasing and attacking used by Harris hawks to build a dynamic balance between global and local optimization processes [19]. The two algorithms are good explorers, allow efficient searches in complex and high dimensional hyperparameter spaces, and allow finding globally optimal solutions. Tests conducted in this study were done on the MNIST dataset, a benchmark dataset commonly used in the research of image processing[20]. MNIST is a set of 28x28 grayscale images of handwritten digits which is widely used in training and testing deep learning models and classification algorithms. To perform our experiments, the hyperparameter configurations of CNN models were optimized with assistance of GWO and HHO and the results of this optimization on the validation sets of the models were evaluated. The findings were thoroughly analyzed based on test accuracy, convergence curves and confusion matrices. All in all, it can be assumed that this work will be able to significantly advance the sphere of hyperparameter optimization, as well as the overall adoption of nature-inspired algorithms in deep learning studies.

2. Related Works

The primary objective of this study is to examine and compare the performance of the GWO and HHO algorithms in CNN hyperparameter optimization.

GWO is a metaheuristic optimization algorithm that mathematically models the hunting strategies of grey wolves to solve global optimization problems effectively [18]. The hierarchy in GWO consists of alpha, beta, delta, and omega wolves, where the alpha wolf represents the best solution, followed by beta and delta wolves as the second and third best candidates, while the omega wolves constitute the rest of the population. During the hunting process, the positions of wolves are continuously updated to approximate the prey's location, enabling the algorithm to generate better solutions iteratively. Due to its exploration capabilities, GWO has been widely recognized as an effective method for solving complex, high-dimensional problems, including CNN hyperparameter optimization. For example, comparative researches with 23 functions that were standard tests showed that GWO was better than Particle Swarm Optimization (PSO) algorithm [21].

GWO has been used to tune the CNN parameters in a number of works. A study resulted in a GWO-based CNN model that was more accurate than the conventional techniques [22]. Likewise, a GWO-optimized CNN model has shown very high performance compared to models optimised with PSO and Genetic Algorithms in the field of skin cancer classification [23]. A different study has pitted CNN-GWO models against artificial neural networks, support vectors machines, and decision trees and indicated that CNN-GWO models were demonstrably superior in terms of accuracy and efficiency [24]. In addition, the hybrid Gaussian-transformed optimization algorithms on the MNIST dataset was able to efficiently optimize CNN hyperparameters, with the highest accuracy of 98% on the MNIST, 92% on the Fashion-MNIST, 76% on the CIFAR-10, and 70 percent on the CIFAR-100 [10].

HHO is a metaheuristic optimization algorithm that was proposed rather recently in 2019 and is based upon the cooperative chasing and attacking behavior of Harris hawks [19]. The algorithm uses dynamic behaviors simulated, including surprise pounce and energy of escape, and allows balancing between global exploration and local exploitation. The ability of the prey to avoid predators is modelled using the concept of escaping energy; the algorithm moves towards exploitation rather than exploration with each iteration. Also, HHO is a combination of two modes of hunting (soft siege and hard siege) to improve optimization strategy [25]. These aspects render HHO an adaptable and robust solution to solving complicated optimization problems as in CNN hyperparameter optimization.

The effectiveness of HHO in deep learning application has been reported by several studies. As an example, HHO was employed in optimizing CNN hyperparameters in brain tumor classification with MRI scans where the chaotic and local search strategies were used to enhance the accuracy of the method to a score above 95% on two test sets [26]. In addition to deep learning applications, Harris Hawks Optimization has also demonstrated strong performance in medical data classification tasks, where it has been effectively employed to optimize model parameters and achieve high classification accuracy in disease diagnosis [27]. Instead, another study presented HHOForSkin, a new CNN architecture that is optimized using HHO to classify skin cancer. This model used 26 layers of CNN, and its F1-score was astonishing 98.93 percent on a set of published data [28]. In the other article, metaheuristic algorithms, including simulated annealing, differential evolution, and harmony search, were used on CNNs on MNIST and CIFAR and, although they needed more computational effort, the optimized models achieved much better results compared to conventional CNNs [29]. On the same note, the ResNet-18 model was reported to perform well with a reported accuracy of 96% on MNIST [30].

In general, the literature proves that both GWO and HHO are effective tools to optimize hyperparameters in CNNs with high accuracy and better performance than conventional optimization strategies. Similarly, metaheuristic optimization algorithms have also been successfully applied to classical machine learning models, improving classification performance through effective parameter optimization [31]. The results highlight why metaheuristic optimization algorithms are potential alternatives to boosting deep learning models.

3.Material and Methods

The experiments in this study were conducted on the MNIST dataset to evaluate the effectiveness of hyperparameter optimization for deep learning models using the GWO and HHO algorithms. These methods were applied in a multidimensional hyperparameter search space with the goal of maximizing model performance. This section presents the dataset and preprocessing steps, the proposed of the CNN model, the hyperparameter optimization process, and the mathematical formulations of the applied algorithms in detail.

3.1. Dataset and Preprocessing

The MNIST dataset consists of grayscale images of handwritten digits ranging from 0 to 9, each represented by a 28×28 pixel resolution. A total of 60,000 images were used for training and 10,000 images for testing. For preprocessing, normalization was applied, transforming pixel values originally ranging from 0–255 into continuous values between 0 and 1, as expressed in Equation (1).

$$x_{norm} = \frac{x}{255} \tag{1}$$

Class labels for the classification process were encoded using the one-hot encoding method, where each digit is represented by a 10-dimensional vector. For instance, the digit “2” is represented as shown in Equation (2).

$$y = [0,0,1,0,0,0,0,0,0,0] \tag{2}$$

3.2.Proposed Model Architecture

The CNN model used in this study consists of an input layer and a hidden layer. The hidden layer is a smoothing layer that converts the 28x28 input image into a one-dimensional vector. Each hidden layer contains a number of neurons determined by the hyperparameters. As expressed in Equation (3), the Rectified Linear Unit (ReLU) activation function is used.

$$f(x) = max(0, x) \tag{3}$$

To reduce overfitting and achieve stable optimization during training, dropout and batch normalization techniques were applied to selected layers. A softmax activation function was used to generate probability distributions across 10 classes, as defined in Equation (4).

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{10} e^{z_j}} \tag{4}$$

The cross-entropy function is used as the loss function of the model and its mathematical formulation is given in Equation (5). In this expression, $y\,i$ is the one-hot encoded vector representing the true class label, $y\hat{}i$ is the probability vector predicted by the model and N is the total number of training examples.

$$\mathcal{L} = -\sum_{i=1}^{N} y_i\ .log(\hat{y_i}) \tag{5}$$

3.3. Hyperparameters and Their Boundaries

The hyperparameters explored during optimization and their boundary values are summarized in Table 1.

Table 1. Parameters and threshold values used for training.

Hyperparameter	**Lower Bound**	**Upper Bound**
Number of Layers (n_{layers})	1	7
Number of Neurons per Layer ($n_{neurons}$)	4	128
Learning Rate (η)	0.001	0.01
Mini-Batch Size ($batch\ size$)	32	256
Number of Epochs ($epochs$)	10	50
Number of DropoutLayers ($dropout\ layers$)	0	7
Dropout Rate ($dropout\ rate$)	0.1	0.5
Number of Batch Normalization Layers ($dropout\ layers\ batch\ norm\ layers$)	0	7

3.4. Updating Outputs and Obtaining the Best Configuration

Optimization algorithms define a hyperparameter space (H) and a fitness function (f) as inputs. H is a multidimensional space of hyperparameters consisting of continuous and discrete values. The fitness function measures the accuracy of the model created with the hyperparameters, and this accuracy value is returned as the output of the algorithms at each iteration.

3.4.1. Grey Wolf Optimization

In the GWO algorithm, alpha (α), beta (β), and delta (δ) wolves are assigned as leaders, and the positions of other wolves are updated under the guidance of these leaders. The relevant update equations are defined below, and the distance to the prey's location is shown in Equation (6).

$$\vec{D} = |\vec{C} \cdot \vec{X}_{prey} - \vec{X}| \tag{6}$$

According to the expressions given in the equation, $\vec{D}$ represents the distance vector between the wolf and the prey; $\vec{C}$ represents a random vector ($\vec{C} = 2 \cdot \vec{r}_1$, where $\vec{r}_1$ is a random number in the interval [0,1]); $\vec{X}_{prey}$ represents the position vector of the prey (the best solution); and $\vec{X}$ represents the current position vector of the wolf.

The process of updating the new position is given mathematically in Equation (7).

$$\vec{X}_{new} = \vec{X}_{prey} - \vec{A} \cdot \vec{D} \tag{7}$$

Here, $A = 2 \cdot \vec{a} \cdot \vec{r}_1 - \vec{a}$ where $\vec{r}_1$ is a random vector that generates a value in the interval [0,1]. The parameter $\vec{a}$ represents a coefficient that decreases with the number of iterations and is defined as $\vec{a} = 2 - \frac{2 \cdot t}{T}$. In this formulation, t denotes the current iteration number, while T denotes the maximum number of iterations.

3.4.2. Harris Hawks Optimization

The HHO algorithm operates in both high-attack and low-attack modes by modeling the hunting strategies of hawks. Depending on the prey's energy (E), different update strategies are applied:

$$\vec{X}_{new} = \vec{X}_{best} - E \cdot |\vec{J} \cdot \vec{X}_{best} - \vec{X}_i| \tag{8}$$

In the equation, $\vec{X}_{\text{best}}$ represents the position of the best solution found so far, $\vec{J}$ [-1,1] denotes a random vector in the interval and $\vec{X}_i$ represents the current position of the hawk.

Low Attack Mode when prey energy is high (E ≥0.5), hawks search more widely.

$$\vec{X}_{\text{new}} = \vec{X}_{\text{best}} + \vec{S} \cdot (\vec{X}_{\text{best}} - \vec{X}_i) \tag{9}$$

Here, S is a random expansion factor, and $\vec{X}_i$ denotes the current position of the hawk.
The energy level (E) is calculated according to Equation (10), where t represents the current iteration number and T represents the maximum iteration number.

$$E = 2 \cdot (1 - \frac{t}{T}) \tag{10}$$

3.4.3. Implementation of Hyperparameter Selections

Hyperparameters are selected from continuous values and converted to discrete values. This transformation is implemented to match the physical structure of the model and to meet memory and data processing requirements. For example, the number of layers (n_{layers}) is obtained from a continuous value, and rounding to an integer is calculated using Equation (11).

$$n_{layers} = round\,(x) \tag{11}$$

The number of neurons in each layer (n_{neurons}) was chosen in the form of 2^k , and the mini-batch size was also determined in the form of 2^k.

$$k \in Z, 2 \leq 2^k \leq 128 \tag{12}$$

During the optimization process, hyperparameter values may sometimes exceed the limits of the search space. The necessary adjustment for this situation is implemented in Equation (13).

$$\mathrm{x}' = \begin{cases} \mathrm{l} & \text{if } \mathrm{x} < \mathrm{l} \\ \mathrm{u} & \text{if } \mathrm{x} < \mathrm{u} \\ \mathrm{x} & \text{else} \end{cases} \tag{13}$$

Equation 14 was used for the hyperparameter set h given in the Definition and Application of Fitness Function.

In defining and applying the fitness function, Equation (14) is used for the given hyperparameter set *h*.

$$(\mathrm{h}) = \mathrm{accuracy}(\mathrm{M}(\mathrm{h}, \mathrm{X}_{\mathrm{train}}, \mathrm{y}_{\mathrm{train}}), \mathrm{X}_{\mathrm{val}}, \mathrm{y}_{\mathrm{val}}) \tag{14}$$

4. Results and Discussion

This section, we evaluate the results obtained by using the GWO and HHO algorithms to optimize the deep learning model on the MNIST dataset. The impact of hyperparameter selection on the model, accuracy results, and the training process are discussed in detail.
Table 2 summarizes the optimization results of the GWO and HHO algorithms. Table 2 shows the best test accuracy of both algorithms, the selected hyperparameter values (number of layers, number of neurons, learning rate, batch size, number of epochs, dropout layers and their ratios, and batch normalization layers).

Table 2. Optimization results of the algorithms and the best hyperparameter values found.

Parameter	GWO	HHO
Best Test Accuracy	0.9794	0.9810
Number of Layers	4	2
Number of Neurons	[64, 64, 64, 64]	[64, 64]
Learning Rate	0.0055	0.003
Batch Size	128	128
Number of Epochs	38	25
Dropout Layers	4	2
Dropout Rate	0.22	0.15
Batch Normalization Layers	4	0

Table 2 shows the best test accuracies achieved by both enhancements using optimized hyperparameter settings. Moreover HHO's results exceeded GWO's accuracy with 98.1% accuracy. In addition this small difference is significant because HHO achieved accuracy with fewer epochs and fewer layers. HHO employed a lower learning rate and used minimal dropout layers. GWO achieved accuracy with a deeper model and more regularization methods (dropout and batch normalization). This demonstrates that HHO can effectively configure simpler models, while GWO achieves similar results with more complex models.
Figure 1 compares the best test accuracies achieved by the HHO and GWO algorithms. These accuracies were used to measure the effectiveness of each algorithm in hyperparameter optimization.

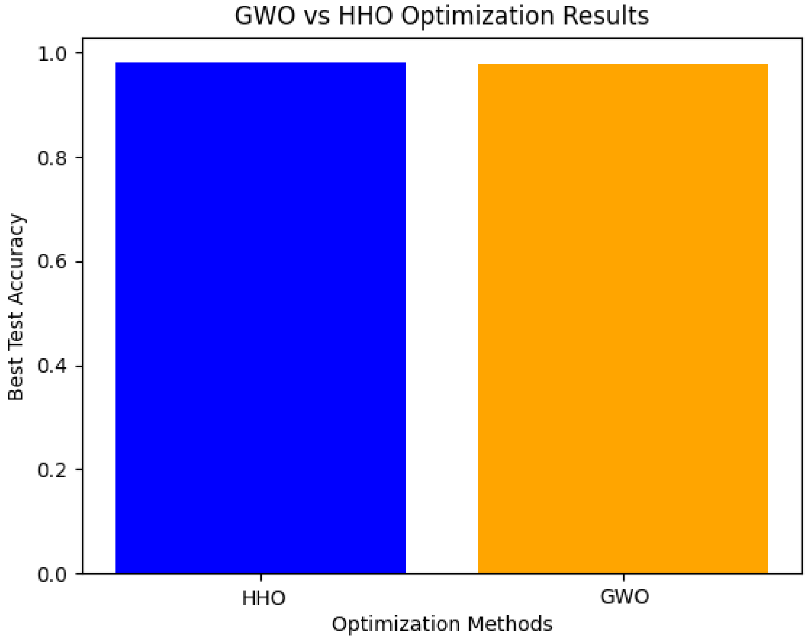

Figure 1. Comparison of the best test accuracies achieved by the HHO and GWO algorithms.

Figure 1 shows a comparison of HHO and GWO in terms of test accuracy. While both algorithms achieved quite high accuracy, the model trained with the parameters specified by HHO showed higher test accuracy than GWO. Although this difference is small, the fact that HHO achieves this accuracy using fewer layers, epochs, and regularizations demonstrates its high optimization efficiency. It can be argued that HHO offers an advantage, especially for models with more limited resources.

Figure 2 shows the training and validation processes of the best hyperparameters obtained with the GWO algorithm. The graph shows how the training accuracy and validation accuracy change in each epoch.

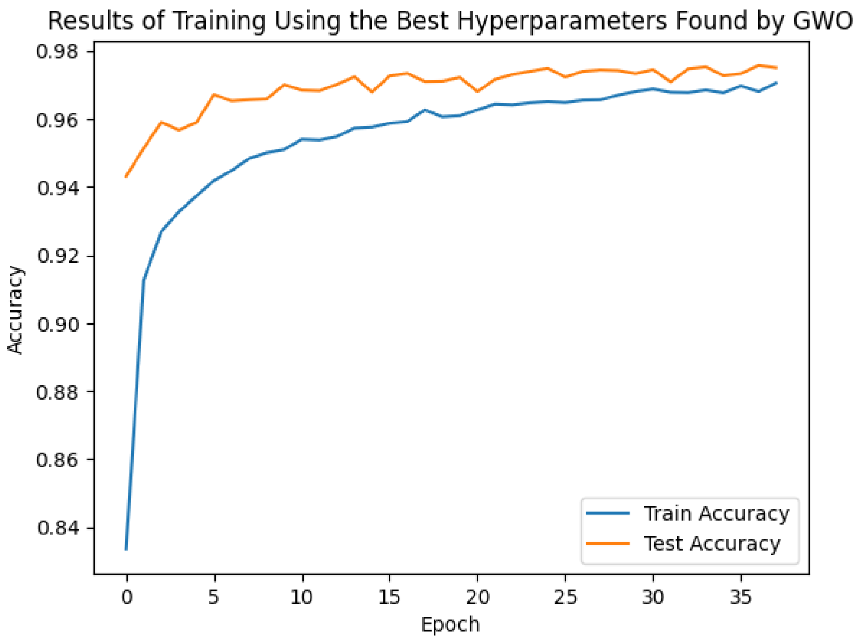

Figure 2. Training and validation accuracy curves for the GWO optimized model.

Figure 3 shows the results of the training and validation processes for the optimal hyperparameters obtained with the HHO algorithm. The graph shows the change in training accuracy and validation accuracy depending on the number of epochs.

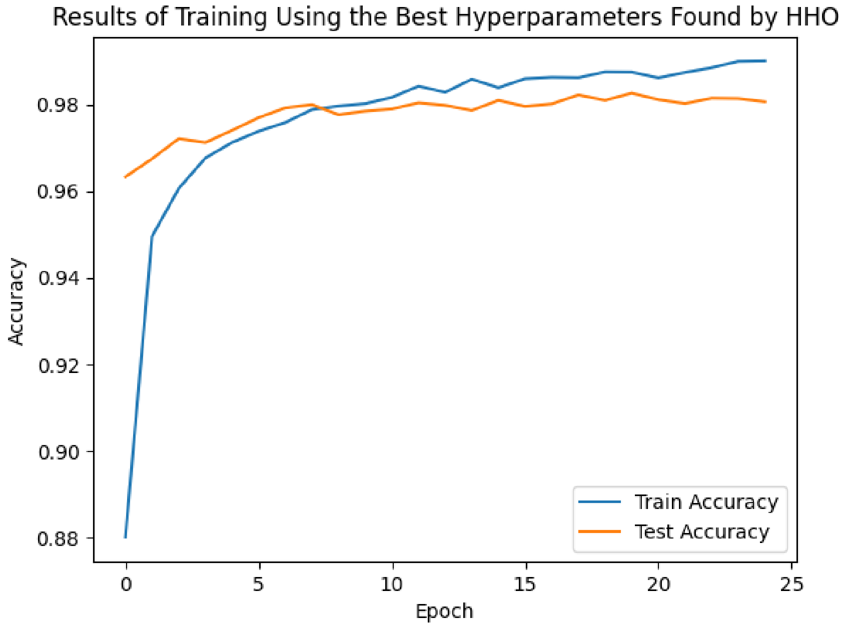

Figure 3. Training and validation accuracy curves for the HHO optimized model.

Figure 3 shows the training and validation processes for the best hyperparameters obtained with the HHO algorithm. HHO's training accuracy plateaued quickly with fewer epochs. Validation accuracy improved in line with training accuracy, demonstrating the model's high generalization capacity. Furthermore, HHO's validation accuracy fluctuated less, indicating that the model optimized more consistently during training. The fact that HHO achieved this accuracy with fewer layers and epochs demonstrates the algorithm's effectiveness in hyperparameter selection.
When the results obtained are compared with studies in the literature, it is seen that high accuracy is achieved with a small architecture and low computational cost. Table 3 compares the results obtained in this study with two different studies in the literature.

Table 3. Comparison of results with studies in the literature.

Study	**Method**	**Accuracy (%)**
Seng vd. [30]	ResNet-18	96
Fakhouri vd. [10]	Gaussian Transform Optimization	98
This Study (HHO)	Harris Hawk Optimization	98.1
This Study (GWO)	Grey Wolf Optimization	97.94

The proposed model, as presented in Table 3 is based on the classic foundations that have been established in the literature, in addition to the legacy results of models with complex designs and high model capacity, such as ResNet-18. This is a major strength, as a smaller model not only allows a reduction in the computational cost but also in the computational overhead and at the same time, it maintains the accuracy (or at most improves it).

Empirical evidence shows that the HHO is more accurate and at the same time with a relatively small model complexity. HHO is a more favorable choice of optimization strategy, as it reduces the number of layers and epochs to operate on smaller hardware resources and a restricted amount of time. The GWO on the other hand is promising with more profound and complex architectures, which achieve the same degree of accuracy.

Many studies in the literature have reported similar accuracy on MNIST dataset. However, the overall goal of the current study is to design a network topology that would provide the highest possible precision, as well as use the smallest possible model size, and the least possible computation costs. These findings confirm that this goal has been achieved since the optimised networks prove to be efficient both in accuracy and use of resources. In particular, the ability of the HHO algorithm to achieve high accuracy at low computational costs highlights its efficiency in its operations.

The results of the current paper determine that the two algorithms have the capacity to accomplish the process of hyper-parameter optimization effectively, whereas the choice of the relevant algorithm can change depending on the application contingencies. Future investigations can include comparisons of those algorithms with larger and more complex data collections. In addition, a hybrid optimization algorithm integrating GWO and HHO is also possible and may possibly be the solution that will pool the strengths of both paradigms and provide schemes of solutions to a wider range of problems. The type of methodological approach outlined below is not limited to classification tasks as it can be also generalized to other fields of artificial intelligence, such as regression, clustering, and real-time optimization.

5. Results

This paper has carefully compared the results of the use of the GWO and the HHO in the light of hyperparameter optimization of deep learning models used on the MNIST dataset. The main aim was to suggest an architecture that can attain great predictive accuracy with decreased model complexity and low computational cost. Empirical evidence shows that both optimization algorithms can fulfill this purpose: the HHO obtained an accuracy of 98.1% better than 97.94% of GWO. It is important to note that the HHO achieved this performance with a significantly simpler architecture, which is defined by fewer layers and training epochs, and thus provides significant benefits to systems with limited hardware resources or with high-frequency optimization cycles.

On the other hand, the GWO, used with a more complex network architecture, using regularization methods (like dropout or batch normalization) achieved similar levels of accuracy. This indicates that GWO is still a viable option in the event of more complicated and more intricate models. Though the same accuracy rates are mentioned in the literature on MNIST, the characteristic feature of the current investigation is that it was demonstrated to be able to reach high accuracy with a very small model complexity and computational costs. Specifically, the fact that HHO has been able to achieve a similar level of accuracy and save on the use of resources highlights the effectiveness and feasibility of the offered methodology.

The results also indicate that the two algorithms did not affect generalization during the training process. Again consistency in both training and validation accuracies indicates that overfitting was effectively countered and the overall levels of model performance were strong. Though GWO had a more gradual and smooth learning curve over longer period of time, HHO achieved high accuracy within a significantly shorter period, which implies that convergence dynamics between the two optimisers vary.

Altogether, the findings confirm the position that both GWO and HHO are efficient and promising substitutes of hyperparameter optimization in deep learning pipelines. The presented methodology is not specific to MNIST, but it is a flexible framework that can be extended to new datasets and different tasks of deep learning. Lastly, the possibility to combine the advantages of GWO and HHO in the future research can lead to the further opportunities to create more complete and scaled-up solutions in the area of automated machine learning.

Acknowledgment
This study did not receive any dedicated funding from public, commercial, or non-profit organizations

Conflict of Interest Statement

No conflict of interest was declared by the authors.

References

[1] P. Sajitha, A. D. Andrushia, N. Anand, and M. Z. Naser, "A review on machine learning and deep learning image-based plant disease classification for industrial farming systems," *J Ind Inf Integr*, vol. 38, p. 100572, 2024, doi: https://doi.org/10.1016/j.jii.2024.100572.
[2] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, "A survey of deep learning and its applications: a new paradigm to machine

learning," *Archives of Computational Methods in Engineering*, vol. 27, pp. 1071–1092, 2020.

[3] S. Arslan and D. Özdemir, "A Deep Ensemble Reinforcement Learning Based Approach For Stock Trading," *Artificial Intelligence Studies*, vol. 7, no. 2, pp. 83–96, 2024.

[4] J.-E. Kim, N.-E. Nam, J.-S. Shim, Y.-H. Jung, B.-H. Cho, and J. J. Hwang, "Transfer learning via deep neural networks for implant fixture system classification using periapical radiographs," *J Clin Med*, vol. 9, no. 4, p. 1117, 2020.

[5] M. M. Taye, "Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions. Computers, 12 (5), 91," 2023.

[6] J.-Y. Li, Z.-H. Zhan, J. Xu, S. Kwong, and J. Zhang, "Surrogate-assisted hybrid-model estimation of distribution algorithm for mixed-variable hyperparameters optimization in convolutional neural networks," *IEEE Trans Neural Netw Learn Syst*, vol. 34, no. 5, pp. 2338–2352, 2021.

[7] S. Altun and M. F. Talu, "Derin sinir ağları için hiperparametre metodlarının ve kitlerinin incelenmesi," *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, vol. 12, no. 2, pp. 187–199, 2021.

[8] N. Del Buono, F. Esposito, and L. Selicato, "Methods for hyperparameters optimization in learning approaches: an overview," in *Machine Learning, Optimization, and Data Science: 6th International Conference, LOD 2020, Siena, Italy, July 19–23, 2020, Revised Selected Papers, Part I 6*, Springer, 2020, pp. 100–112.

[9] F. Aydemir and S. Arslan, "A system design with deep learning and IoT to ensure education continuity for post-COVID," *IEEE Transactions on Consumer Electronics*, vol. 69, no. 2, pp. 217–225, 2023.

[10] H. N. Fakhouri, S. Alawadi, F. M. Awaysheh, and F. Hamad, "Novel hybrid success history intelligent optimizer with gaussian transformation: Application in CNN hyperparameter tuning," *Cluster Comput*, vol. 27, no. 3, pp. 3717–3739, 2024.

[11] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020.

[12] B. Bischl *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *Wiley Interdiscip Rev Data Min Knowl Discov*, vol. 13, no. 2, p. e1484, 2023.

[13] J. Nayak, B. Naik, A. K. Das, and D. Pelusi, "Nature inspired optimization and its application to engineering," *Evol Intell*, vol. 14, no. 1, pp. 1–3, 2021, doi: 10.1007/s12065-021-00586-x.

[14] S. Dörterler, H. Dumlu, D. Özdemir, and H. Temurtaş, "Hybridization of meta-heuristic algorithms with k-means for clustering analysis: Case of medical datasets," *Gazi Mühendislik Bilimleri Dergisi*, vol. 10, no. 1, pp. 1–11, 2024.

[15] E. Şahin, D. Özdemir, and H. Temurtaş, "Multi-objective optimization of ViT architecture for efficient brain tumor classification," *Biomed Signal Process Control*, vol. 91, p. 105938, 2024.

[16] S. Arslan, S. Dörterler, and F. Aydemir, "Reinforcement learning for energy optimization in IoT based landslide early warning systems," *Journal of Scientific Reports-A*, no. 059, pp. 32–57, 2023.

[17] S. Dörterler, S. Arslan, and D. Özdemir, "Unlocking the potential: A review of artificial intelligence applications in wind energy," *Expert Syst*, vol. 41, no. 12, p. e13716, 2024.

[18] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.

[19] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future generation computer systems*, vol. 97, pp. 849–872, 2019.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[21] A. Köybaşı and İ. Yazici, "Solution of test problems with grey wolf optimization algorithm and comparison with particle swarm optimization," *Sakarya University Journal of Science*, vol. 24, no. 6, pp. 1252–1264, 2020.

[22] S. K. Ladi, G. K. Panda, R. Dash, P. K. Ladi, and R. Dhupar, "A novel grey wolf optimisation based CNN classifier for hyperspectral image classification," *Multimed Tools Appl*, vol. 81, no. 20, pp. 28207–28230, 2022.

[23] R. Mohakud and R. Dash, "Designing a grey wolf optimization based hyper-parameter optimized convolutional neural network classifier for skin cancer detection," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, Part B, pp. 6280–6291, 2022, doi: https://doi.org/10.1016/j.jksuci.2021.05.012.

[24] G. A. Kumar, A. Katiyar, and K. A, "Optimizing CNN Performance for Heart Attack Detection through Grey Wolf Algorithm," in *2024 Asia Pacific Conference on Innovation in Technology (APCIT)*, 2024, pp. 1–9. doi: 10.1109/APCIT62007.2024.10673435.

[25] T. Dokeroglu, A. Deniz, and H. E. Kiziloz, "A comprehensive survey on recent metaheuristics for feature selection," *Neurocomputing*, vol. 494, pp. 269–296, 2022.

[26] J. Basha *et al.*, "Chaotic harris hawks optimization with quasi-reflection-based learning: An application to enhance cnn design," *Sensors*, vol. 21, no. 19, p. 6654, 2021.

[27] N. Yagmur, I. Dag, and H. Temurtas, "Classification of anemia using Harris hawks optimization method and multivariate adaptive regression spline," *Neural Comput Appl*, vol. 36, no. 11, pp. 5653–5672, 2024.

[28] W. N. Ismail and H. A. Alsalamah, "Efficient Harris Hawk optimization (HHO)-based framework for accurate skin cancer prediction," *Mathematics*, vol. 11, no. 16, p. 3601, 2023.

[29] L. M. R. Rere, M. I. Fanany, and A. M. Arymurthy, "Metaheuristic algorithms for convolution neural network," *Comput Intell Neurosci*, vol. 2016, no. 1, p. 1537325, 2016.

[30] L. M. Seng, B. B. C. Chiang, Z. A. A. Salam, G. Y. Tan, and H. T. Chai, "MNIST handwritten digit recognition with different CNN architectures," *J. Appl. Technol. Innov*, vol. 5, no. 1, pp. 7–10, 2021.

[31] N. Yağmur, "A hybrid approach to obesity level determination with decision tree and pelican optimization algorithm," *Journal of Scientific Reports-A*, no. 057, pp. 97–109, 2024.