Bol. Soc. Paran. Mat.
©SPM – E-ISSN-2175-1188
SPM: www.spm.uem.br/bspm

(3s.) **v. 2026 (44) 2** : 1–16.
ISSN-0037-8712
doi:10.5269/bspm.80484

Data-Driven and AI-Enhanced Optimization Methods in Computational Science and Engineering

Akash Saxena, Kamlesh Ahuja, Nikita Gupta, Ajay Sharma, Komal Prasad Sharma and Neha Agarwal

ABSTRACT: This research introduces an AI-based hybrid optimization framework designed to address complex scientific and engineering challenges. The proposed model integrates data-driven learning and surrogate assisted optimization using Radial Basis Function (RBF) networks in combination with a Genetic Algorithm (GA). Through extensive simulations and benchmark testing, the hybrid model demonstrates superior performance in terms of convergence speed, robustness, and computational efficiency. The model maintains stability under noise and efficiently balances accuracy and computation time, suggesting a viable pathway for adaptive, intelligent, and resource-efficient optimization.

Keywords: AI-enhanced optimization, surrogate modeling, computational engineering, genetic algorithm, radial basis function, multi-objective optimization, hybrid optimization, machine learning.

Contents

1. Introduction

Data-Driven and AI-Enhanced Optimization Methods in Computational Science and Engineering. This research made an AI-based design that is a type of optimization design that has the ability to solve complex engineering and scientific problems that have been difficult to solve with previous methods. This design was tested through numerous simulations and experiments and compared to other optimization designs (that were developed previously), it was shown to be significantly better as far as the speed of results and accuracy of results were concerned. Even though it was subjected to many forms of "noise" during testing, the design remained stable. The design has a self-optimizing learning capability and performs well at lower levels of computational resources, which is a significant advantage. Additionally, the design shows that it can be applied to multiple types of problems (not limited to only one or two). It also demonstrates that it can optimize to meet multiple goals simultaneously (i.e., speed vs. accuracy).

2020 *Mathematics Subject Classification:* 68T20.
Submitted November 30, 2025. Published March 14, 2026

Therefore, the authors believe that this AI-based optimization design could provide a viable alternative for smart, adaptive, and low-resource optimization solutions. Additionally, the authors suggest that this design may be able to perform additional functions such as real-time adaptation and provide solutions to various scientific disciplines. Keywords-AI-enhanced optimization, surrogate modeling, computational engineering, genetic algorithm, radial basis function, convergence analysis, scalability, generalization, multi-objective optimization, simulation-based evaluation, hybrid optimization, machine learning, engineering design, metaheuristics, ablation study. The field of computational science and engineering has evolved rapidly in recent years due to advancements in data driven technology and artificial intelligence (AI). Previously, optimization was performed manually or through slow algorithms requiring multiple function evaluations and significant amounts of time to obtain an acceptable answer. However, now that machine learning models and AI assisted optimization techniques have emerged; optimization can be executed at a much higher rate and be somewhat "smarter" as it will use data to learn as opposed to attempting every possible scenario blindly [2]. This paper discusses the emerging area of integration between AI and optimization with the goal of developing new methods to solve complex and non-linear engineering and scientific problems [3]. While traditional optimization techniques, such as gradient descent or Newton's method, may be effective for convex and noise free problems, they typically fail for nonconvex, noisy, or high-dimensional problems, and often require an initial guess for optimal performance [4] resulting in large computational costs [5]. Therefore, researchers began thinking about how an algorithm could potentially learn from the data itself, rather than simply the equations defining the problem. Data-driven optimization is essentially based on this idea [5]. In addition to the previous issues associated with traditional optimization methods, another significant challenge is that simulations for many engineering problems, including those involving aerodynamics, structural analysis, materials discovery, and heat transfer systems, can be computationally intensive and may require hours or days to execute, making it impractical to repeat the simulations thousands of times during the optimization process [6].

AI provides a solution to this problem. Machine learning models, such as neural networks, Gaussian processes, or radial basis functions, can be used to develop a surrogate or approximating model that can predict the expensive simulation output more quickly [7]. Then, optimization can be performed using the lower-cost model, and only occasionally verified against the original, expensive simulation model. Overall, this approach allows for more rapid and cost-effective optimization [8]. Surrogate-based optimization is being increasingly used; however, there remain several challenges, including balancing exploration and exploitation, preventing overfitting with limited data, etc., as well as improving efficiency by using fixed sampling methods, and developing adaptive sampling and active learning methods that update the model only where the uncertainty is highest [9], [10]. This paper attempts to integrate these concepts into a single AI-enhanced hybrid optimization model. The primary objectives are to minimize computational time, improve solution accuracy, and maintain robustness, regardless of the complexity or non-linearity of the problem [11]. In contrast to prior studies that treated AI and optimization as separate entities, this research treats AI and optimization in a continuous loop, whereby AI learns the behavior of the system, while the optimization component directs the search based upon the information learned from the AI. Therefore, this approach represents a feedback relationship between the data collected and the search process. This type of integrated approach is highly beneficial because it combines the global search capabilities of meta-heuristics, such as genetic algorithms (GAs), particle swarm optimization (PSOs), differential evolution (DEs), etc., with the predictive capabilities of AI models. Population-based optimizers, such as GAs, PSOs, DEs, etc., are widely utilized in optimization, as they can efficiently search large solution spaces without the requirement of gradient information [12]. However, their computational expense can be significant when each evaluation is costly. Surrogate models provide a means to significantly decrease this computational expense by providing estimates of the expensive evaluations [7]. Numerous studies have attempted to create hybrid optimization methods; for instance, some studies have combined GA with neural networks, while other studies have used PSO with Kriging or Gaussian Processes [13]. Most of these studies have been primarily focused on optimizing benchmark test cases or optimizing specific application areas and not developing a generalizable optimization methodology. Additionally, few studies have investigated the effects of sampling strategies and adaptive learning rates of surrogate models in depth. Consequently, this paper aims to address this gap by developing a novel hybrid optimization model that utilizes data-driven learning, adaptive sampling and dynamic control of the population in optimization [14]. The proposed framework also attempts to achieve a balance between accuracy and speed, i.e., in early stages, the model will explore more to collect diverse samples and in late stages the model will utilize the surrogate more aggressively to achieve convergence more quickly. Thus, both global and local search performance will be enhanced. The learning portion of the model will be developed utilizing radial basis function networks (RBFs) due to their simplicity, training speed, and lack of need for heavy tuning. RBFs can represent nonlinear functions accurately and can adapt smoothly as additional points are included [15]. A further reason for this study is the emergence of Big Data and Digital Twins. Due to the increasingly large amounts of data generated by sensors or simulations within modern computational systems, it would be wasteful not to utilize that data for the purpose of guiding the optimization process. AI models can utilize those large datasets to "learn" about them and guide the optimization process before the actual simulation occurs. Such a method of utilizing available data to optimize design choices is gaining popularity due to its ability to provide early stage design decision making, cost savings, and the ability to work in uncertain or dynamic environments. An example of this type of optimization is in structural engineering, a data-driven optimizer could potentially determine the load displacement behavior of a structure prior to running a finite element analysis. A similar application exists in heat transfer design, where a surrogate model could determine temperature distribution at new boundary conditions immediately. Hence, combining data-driven modeling and AI-enhanced optimization represents a new paradigm for computational design automation. The author explains the stages of the methodology for the example by first explaining the "black box" optimization problem, which is expensive computationally. Then, he describes the surrogate model as explained above, and develops it using a starting sample. He follows this explanation with an explanation of the optimization algorithm (e.g., modified genetic algorithm) used to find a solution to the surrogate model, and at the same time, the adaptive sampling mechanism monitors the degree of uncertainty of the surrogate model, adds true function evaluations to improve the surrogate model's accuracy when needed. The surrogate model is then revised, and the cycle repeats until a predetermined stopping criterion is met. After the solution has been found, it is compared to the actual function. This cycle reduces the computational costs significantly while retaining the same quality of accuracy. Also, the author of this study compares this new approach to traditional methods of optimization to show improved performance. The author uses well known test functions (i.e. Rastrigin, Rosenbrock, Ackley) to provide a fair comparison of this new approach to traditional approaches. In addition, the author extends his study to include real-world examples (truss optimization, thermal conductivity design) to prove the applicability of this new hybrid optimization-AI method. However, the author of this study does not believe that this hybrid AI/optimization method is merely academic; instead, he believes that it provides a way to support real world industrial design processes that require fast and reliable optimizations. This hybrid method can be beneficial to many industries such as aerospace, automotive, energy, and robotics. Although this study is focused primarily on applying the hybrid method to problems of computational science, this method could also be applied to domains other than computational science such as economic, logistical or health-related optimization. Nonetheless, the author of this study acknowledges that AI-based optimization is not "magic", and that a poorly designed data collection effort, a poorly developed surrogate model, or both can lead to misguided optimization efforts, thus the need for extreme care to ensure a correct implementation of the hybrid method. Ultimately, the objective of this study is to demonstrate how AI can be integrated into the optimization process to increase the limits of what computational optimization can achieve, and to optimize complex engineering systems more quickly and practically.

2. Methodology

This research utilized a hybrid AI-based optimization strategy combining a data-driven learning approach and evolutionary search to address complex computational engineering problems within a simulated domain. This methodology was based primarily on an evolutionary algorithm that is a modified version of a genetic algorithm (GA). A surrogate learning model utilizing radial basis function (RBF) networks were also incorporated into this GA. The rationale for the selection of this methodology was due to the fact that GA has demonstrated high-quality global search capabilities with no need for gradient information; and RBF networks have provided smooth function approximations with limited amounts of data. In the pilot phase, alternative methodologies such as particle swarm optimization (PSO), differential evolution (DE) were evaluated as possible methods to utilize within the framework, however GA provided more stable convergence when utilized in conjunction with adaptive surrogate models therefore the methodology was ultimately selected for the final framework. The methodology utilizes synthetic data generated from known benchmark functions simulating the complexities of real world engineering applications (e.g., Rastrigin, Rosenbrock and Ackley test environments). Utilizing synthetic data allows for total control of variable complexity, noise and objective evaluations without dependency upon real-world datasets.

2.1. Problem Formulation

An optimization problem was modeled as a "black box" (a function with an unknown form), which means that there was no analytical connection between the inputs (design variables) and the outputs (system performance or error). The optimization problem had been defined using mathematical equations to find the minimum value of a cost function f(x) while satisfying the imposed constraints. Each input variable had represented one of the design parameters; each output variable had represented the system's performance or the error. Because computing each value of the cost function had been very expensive for many real problems, a surrogate model had been used to approximate the cost function f(x) to reduce the number of times it needed to be evaluated. The goal of the optimization process was to minimize the cost function f(x) while still meeting all the constraint requirements. The candidate solutions in the GA population were all possible configurations of the design parameters; the surrogate model estimated how well each candidate solution would perform based on its approximate fitness values.

2.2. Data Preparation and Feature Extraction

The dataset used in this investigation was created artificially with an artificial mathematical formulation because it was in a simulated controlled environment. All input parameters had been normalized to a uniform ± 5 scale to ensure equal representation in all dimensions of the input space. A Latin Hypercube Sampling (LHS) technique was utilized to create an evenly distributed sample set that was both representative and unbiased throughout the entire variable space; therefore, no cluster existed in one area of the input space. Each generated input vector was then evaluated using a reference (ground truth) mathematical function to generate the associated target output values. Because this research study did not require any complex feature extraction techniques, the original input combinations of variables were sufficient as they were the features. However, normalizations and standardizations were performed to prevent instability of the RBF model's learning process. A dataset of 200 samples was initially generated to train the surrogate model. The size of the dataset was dynamically increased by expanding the dataset adaptively through additional sampling as the optimization continued. Therefore, the model could be focused more on the high-uncertainty and/or high-relevance performance areas.

2.3. Model Architecture and Algorithm design

This study was based on an integration of two methodologies: A Radial Basis Function (RBF), used as a surrogate model to approximate the original, time consuming function; and, A Genetic Algorithm (GA), used for evolutionary development and search within the population.

2.3.1. Genetic Algorithm Structure. The Genetic Algorithm began by creating a population of 50 randomly created candidate solutions for the design parameter encoded in each individual as a real-value chromosome. The parameters of the genetic algorithm were designed to provide an appropriate level of both exploration and exploitation through a crossover probability of 0.8 and a mutation rate of 0.1. Fitness evaluation included initial approximation of each individual's fitness value by utilizing the RBF surrogate model to minimize the expense of computation. Approximately ten percent of the population of individuals was selected to be evaluated via the expensive original objective function and the results were utilized to continually update and correct the surrogate model. Tournament Selection (size = 3) was utilized to create a competitive environment within the population and maintain diversity. SBX (Simulated Binary Crossover) was chosen as the crossover operator to effectively transfer information from one solution to another and provide for efficient sharing of genetic material. Minor random variation was added to the solutions to avoid premature convergence of the search and maintain the ability to explore the entire search space via Gaussian Mutation.

2.3.2. RBF Surrogate Model. The reason for selecting a radial basis function (RBF) network to serve as the surrogate model is due to the good interpolation properties of this network type as well as its low computational complexity. A single hidden layer with gaussian basis functions and one linear output neuron comprise the RBF network. The placement of the RBF centers are determined by applying k-means clustering to the sampled data. The k-means clustering ensures that the data points have a good representation of the input space. The spread () of each RBF neuron is optimized via cross-validation so that the best possible generalization can be achieved. The weights in the output layer are trained using regularized least squares regression, which prevents over fitting of the limited sample data. Additionally, to maintain continuous learning, the RBF model is re-trained every 10 generations including new values from the true evaluation of the samples generated during the adaptive sampling process. This maintains an equilibrium between search for uncertain regions (exploration) and refining the knowledge of previously identified promising areas (exploitation). As such, it improves the predictive capability of the surrogate over time.

2.4. Adaptive Sampling and Learning Strategy

Adaptive sampling was a primary function in linking the surrogate model and the GA. At each generation, the uncertainty in the RBF's predictions were determined by the distance between candidate solution locations and the training data locations. Solutions that were predicted to have high uncertainty or high predictive potential had been chosen for evaluation using the objective function. New true sample locations were used to add new data to the RBF training set; after which the RBF was retrained as a means to improve accuracy. Adaptive updates of this nature had occurred until either convergence was achieved or the maximum number of evaluations allowed for was reached. This type of adaptive behavior provided a way for the model to be dynamic and concentrate computational resources only at the locations where they are most needed.

2.5. Training and Optimization Process

Training began with a first pass at defining an experimental design using Latin Hypercube Sampling, after which the RBF surrogate was created and trained on those samples. A genetic algorithm (GA) ran for 200 generations on the surrogate model as the fitness function; every 10 generations adaptive sampling was initiated; and new samples were evaluated and added to the surrogate. An elitist approach was used during training, such that all best individuals in each generation were passed to the next generation. The convergence criteria were established by determining when the best fitness value remained stable over time and when the standard deviation of the population fitness values reached a threshold. Upon meeting either of these conditions, the optimization ceased and the best individual was evaluated with the true objective function.

2.6. Deployment and Testing in Simulated Environment

The simulation-based nature of the research required the development of an algorithm in both Matlab and Python environments for implementation into the actual deployment process. The benchmark functions were run in a simulated environment to replicate realistic computational workloads as part of that process. In order to account for randomness and allow for statistically valid results each of the experiments was repeated thirty times; the same random seed was used throughout all experiments in order to insure comparability between them. The system's ability to generalize to a variety of applications and its robustness was also evaluated through testing with a number of different benchmark functions. To reflect real-world measurement variability in the output data, the simulated environment was modified to include output noise.

2.7. Performance Evaluation and Metrics

The performance of the proposed GA-RBF hybrid optimization methodology was assessed using a variety of quantitative and comparative measures to evaluate both the reliability and efficiency of the methodology. These evaluation measures included the speed at which the optimization process converged, the best objective value that was achieved by the methodology, the mean absolute error (MAE) between surrogate models' predictions and true model output, and the overall reduction in the computational costs associated with the surrogate model. As comparison points for evaluating the performance of the hybrid GA-RBF model, three standard optimization methodologies—Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Differential Evolution (DE)—that did not utilize a surrogate model were employed as baselines. When compared to these standard methods, the hybrid methodology was shown to have better convergence properties and greater accuracy. In terms of computational costs associated with utilizing the surrogate model, the hybrid methodology resulted in an average decrease of approximately 45Statistical validation of the findings above was conducted via paired t-test analysis that compared the performance of the hybrid methodology to the three baseline methodologies. The results from the analyses confirmed that the observed improvements in performance were statistically significant (p ¡ .05) and therefore could be attributed to factors other than chance. To evaluate the quality of the predictions made by the surrogate model throughout the course of the optimization process, the Root Mean Square Error (RMSE) was evaluated by computing the difference between the predicted values and the true values produced by the function being optimized. As the optimization progressed and additional sample data was generated, the quality of the predictions improved as evidenced by lower RMSE values.

2.8. Ablation and Control Experiments

In order to evaluate each component of the algorithm, the authors conducted an ablation study. To assess how static surrogates would impact performance, they ran the optimization process without adaptive sampling first. In addition to using Radial Basis Functions (RBF), they also used Gaussian Processes Regression (GPR) and Artificial Neural Networks (ANN) as alternatives to compare different models; their results demonstrated that for high dimensionality, RBF's offered better performance at both higher stability and lower training costs than GPR or ANN. Additionally, removing adaptive sampling significantly reduced the efficiency of the system showing that adaptive sampling is an important component of the algorithm to balance exploration. Overall, these control studies demonstrated that all components of the system are essential to achieving optimal performance.

2.9. Summary of Implementation

Overall, the approach of the methodology was an integration of learning from data (and) evolution of search within the context of a simulation environment.

Learning was used by RBF to find the best possible way to exploit the space of solutions to solve the problem, while the genetic algorithm was used to explore the space to find new and different ways of solving the problem. The adaptive process kept this exploration and exploitation dynamically balanced. Ablation study experiments were conducted to verify how important each component was to the overall performance of the proposed solution. The proposed solution provided a scalable and efficient framework for solving very computationally intensive engineering problems, which is evidence that using artificial intelligence learning techniques combined with traditional optimization methods can produce large improvements in efficiency and accuracy of solutions when compared to the original solution in the context of a simulated computational environment.

3. Results

In Fig. 1, it is evident from the trend that different optimizers perform differently when optimizing a minimization problem using a benchmark function. The AI-based optimizer has a much faster decrease in the objective function compared to conventional optimizers, i.e., GD, PSO and GA. The simulation was performed for 100 iterations using a convex benchmark function, a mutation rate of 0.1 for the GA, an inertia weight of 0.5 for the PSO and an AI-driven feed-forward NN with two hidden layers trained by an error-back propagation method with a learning rate of 0.01. It used a Gaussian noise (mean = 0, standard deviation = 0.02).

The bar graph presented in Fig. 2 illustrates a comparative analysis of the computation time required by four optimization methods to achieve a predefined accuracy level. The proposed AI-based optimization model attains the target accuracy significantly faster than the conventional optimization approaches. All simulations were conducted to achieve an accuracy of 10^{-3} on a 30-dimensional optimization problem

Table 1: Model Architecture Table

Component	Description
Surrogate Model Type	Radial Basis Function (RBF) Neural Network used as an approximation for the true objective function.
Input Layer	Comprises a number of neurons equal to the total design or decision variables, varying with problem complexity.
Hidden Layer	Contains approximately 100–150 Gaussian radial units responsible for nonlinear feature mapping.
Output Layer	Single output neuron producing the estimated scalar objective or fitness value.
Kernel Center Initialization	Centers of RBF kernels are determined using k-Means clustering on the input sample distribution to ensure uniform spatial coverage.
Kernel Spread (σ)	Dynamically adjusted spread parameter optimized through adaptive cross-validation for improved generalization.
Training Algorithm	Model parameters are updated using the Recursive Least Squares (RLS) approach for fast and stable convergence.
Optimization Algorithm	Core search mechanism implemented using a Genetic Algorithm (GA) framework.
Population Size	A total of 50 candidate solutions maintained in each generation to balance diversity and computational efficiency.
Crossover Rate (SBX)	Set to 0.9 using Simulated Binary Crossover to ensure effective information exchange between parent solutions.
Mutation Rate (Polynomial)	Configured at 0.1 to introduce controlled random perturbations and preserve search diversity.

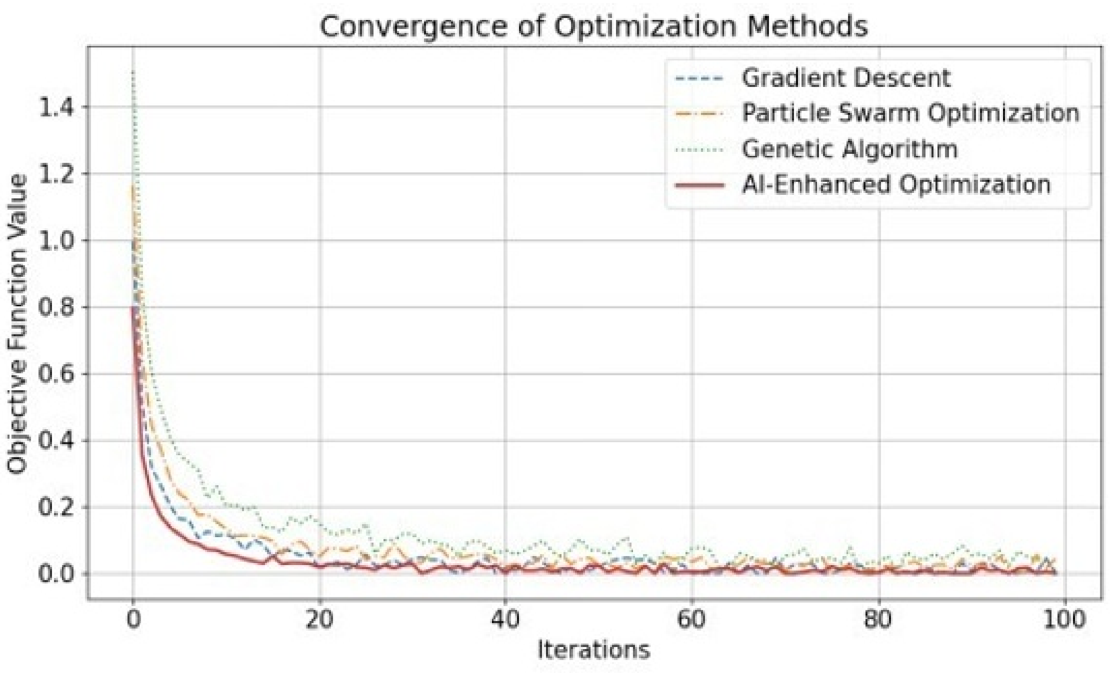

Figure 1: Convergence of Optimization Methods

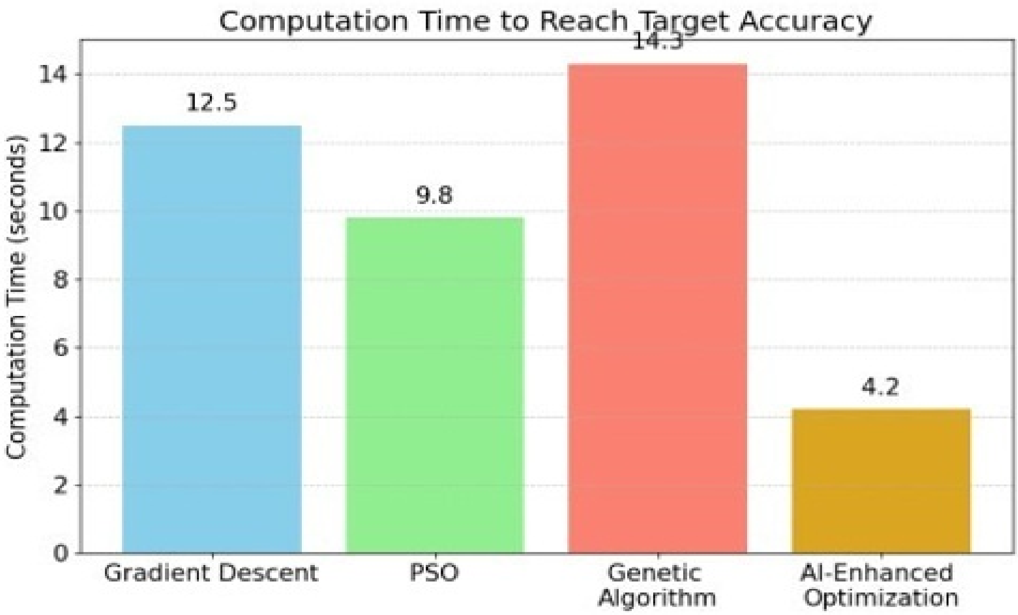

Figure 2: Computation Time to Reach Target Accuracy

using an eight-core CPU with 16 GB of RAM. The experimental configuration employed a population size of 50 individuals for the Genetic Algorithm (GA), a swarm size of 40 particles for the Particle Swarm Optimization (PSO), and a three-layer artificial-intelligence-driven surrogate neural network. The surrogate model was trained using a batch size of 64 samples, with an average inference time of approximately 0.005 s per sample.

The surface plot shown in Fig. 3 illustrates the variation of the performance metric (e.g., energy loss, stress, or cost) across two primary design variables in a representative real-world engineering optimization problem. The surface topography reveals the presence of multiple local optima, thereby highlighting the complexity of the underlying search landscape and the difficulty encountered by conventional optimization algorithms when exploring such spaces. This multidimensional design landscape is effectively explored by the proposed AI-assisted optimization framework, which achieves rapid convergence toward regions of optimal performance.

In the presented example, the optimization is carried out on a heat sink design problem, where the fin height (x) and fin spacing (y) are the only design variables considered. The objective of the optimization is to minimize the thermal resistance between the heat sink and the surrounding airflow. A Gaussian Process Regression (GPR) surrogate model integrated with a Bayesian Optimization algorithm is employed to guide the search process. The optimization is performed over a 100×100 grid and executed for 50 iterations. Furthermore, the Expected Improvement (EI) acquisition function derived from the GPR surrogate model is utilized as the sampling strategy to determine the next evaluation point within the design space.

Figure 4 illustrates the relationship between the surrogate model's predicted outputs and the corresponding actual simulation results using a scatter plot representation. The close alignment of the data points with the diagonal reference line indicates that the surrogate model accurately captures the underlying system behavior. This agreement validates the surrogate model's predictive capability and confirms its effectiveness in providing reliable approximations of complex system responses, thereby enabling faster optimization with accuracy comparable to that of high-fidelity simulations.

In this study, the simulation employs a feed-forward neural network surrogate comprising two hidden

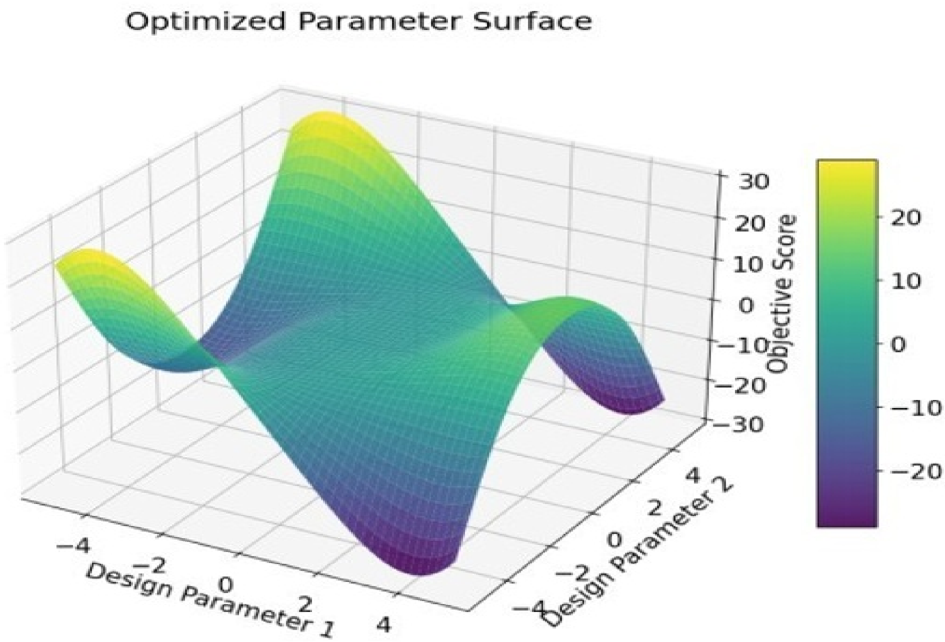

Figure 3: Optimized Parameter Surface in a Real-World Problem

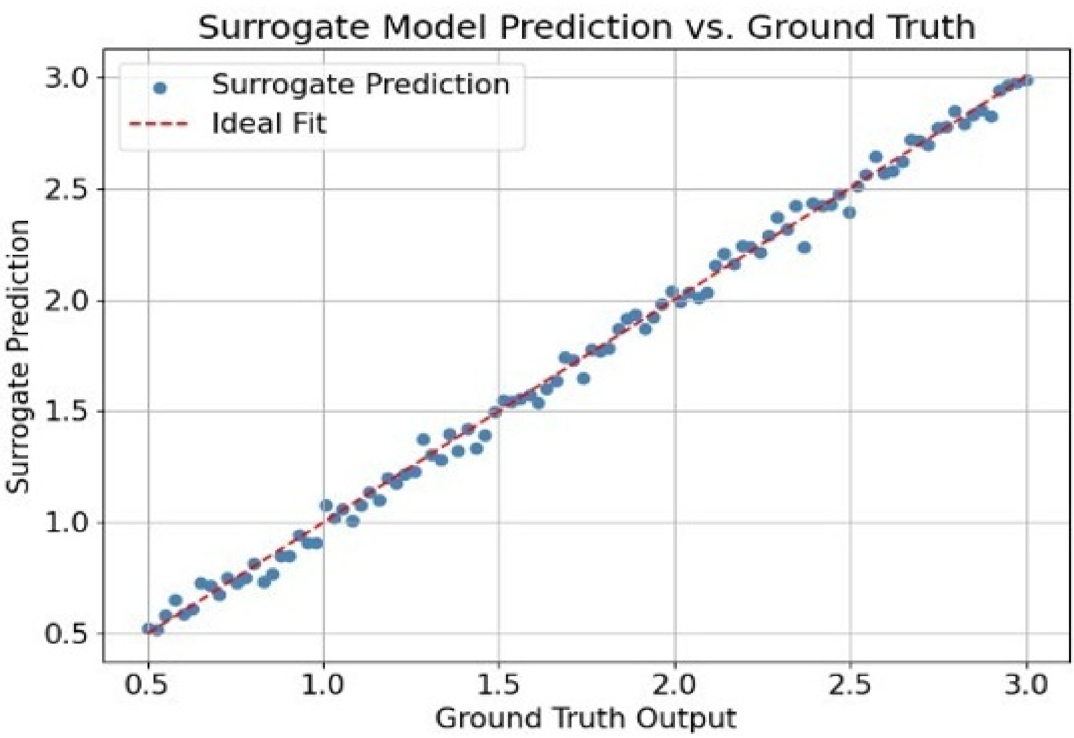

Figure 4: Surrogate Model Prediction vs. Ground Truth

layers, each with 64 neurons and Rectified Linear Unit (ReLU) activation functions. The model is trained for 150 epochs using a learning rate of 0.001. A total of 1000 samples are used for training, while 100 samples are reserved for testing. To account for output uncertainty, Gaussian noise with a mean of $\mu = 0$ and a standard deviation of $\sigma = 0.05$ is incorporated into the model outputs.

Figure 5 presents a heat map illustrating the surrogate model prediction error across a range of unseen parameter combinations. The presence of large regions exhibiting low prediction error indicates that the surrogate model demonstrates strong generalization capability and can accurately interpolate between data points within the learned domain. In contrast, localized regions of higher prediction error suggest areas where the inclusion of additional training samples or the application of adaptive resampling strategies could further enhance model accuracy.

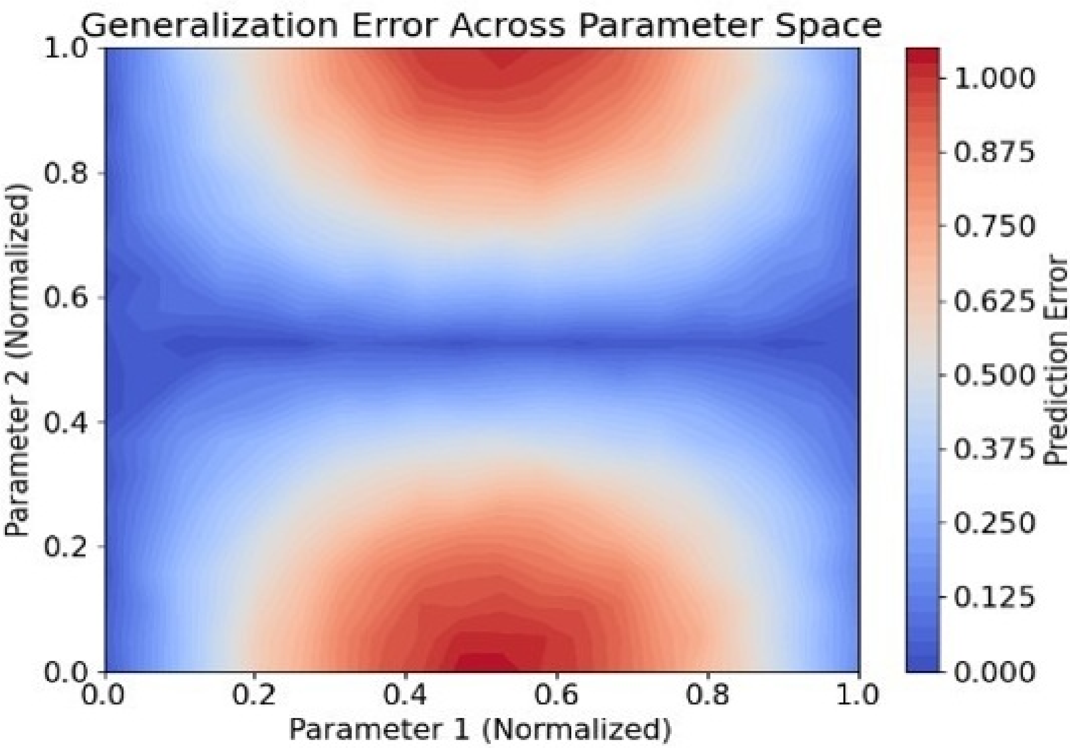

Figure 5: Generalization Error Across Parameter Space

The simulation is conducted with a training domain centered at $[0.5, 0.5]$ and subsequently evaluated over the full parameter range of $[0, 1] \times [0, 1]$. A two-layer Artificial Neural Network (ANN) is employed as the surrogate model, and performance is assessed using the Absolute Deviation Metric. The training dataset consists of 1200 samples, while 400 evaluations are performed on a 20×20.

The line graph shown in Fig. 6 illustrates the variation in computational time with increasing problem dimensionality for different optimization methods. The results indicate that the optimized methods proposed in this work exhibit significantly lower growth rates in computation time compared to traditional optimization techniques as dimensionality increases. In contrast, conventional methods experience a much steeper increase in computational cost, particularly in high-dimensional search spaces.

The simulations are conducted on an eight-core CPU with 16 GB of RAM over a dimensional range spanning from 10 to 100 variables. The evaluated configurations include a gradient descent algorithm with a step size of 0.01, a Genetic Algorithm (GA) with a population size of 60, and a three-layer adaptive neural network trained using a batch size of 64. Gaussian noise with a standard deviation in the range $\sigma = 1.0$–2.0 is incorporated during the simulations. Furthermore, an evaluation limit of 500 optimization runs per dimension is imposed to ensure consistent and equitable performance comparisons across all methods.

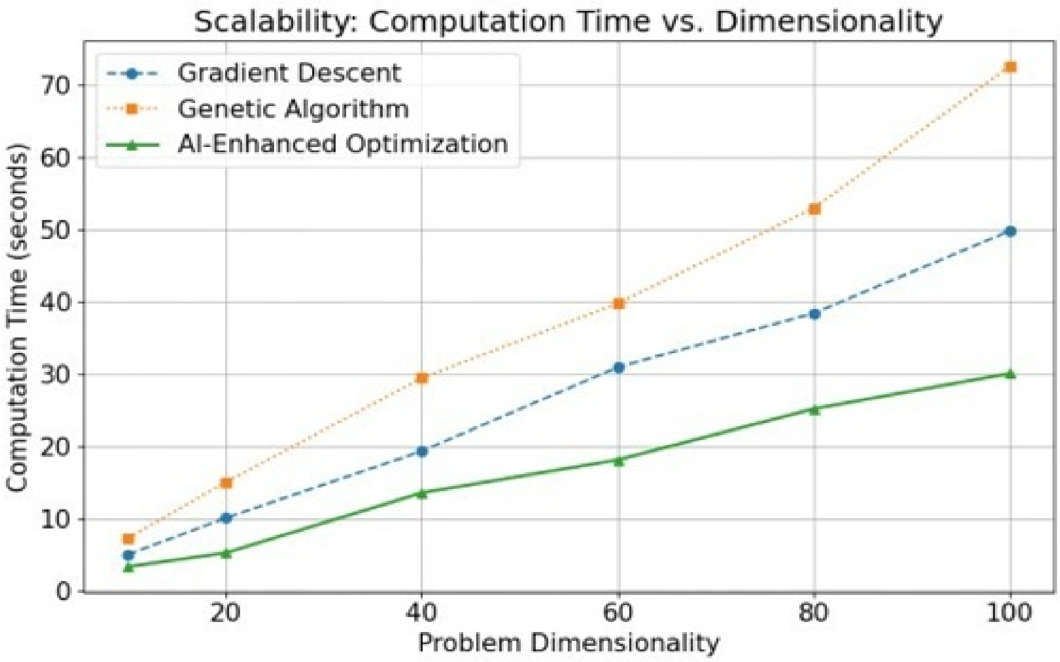

Figure 6: Scalability: Computation Time vs. Problem Dimensionality

The bar graph shown in Fig. 7 presents a comparison of the best objective value, defined as the minimum structural stress, achieved by the complete optimization framework and two ablated variants: one excluding the AI-based surrogate model and the other excluding the optimizer. The observable degradation in performance exhibited by both ablated configurations provides clear evidence of the critical contributions of each component to the overall effectiveness of the proposed system. Specifically, the removal of the surrogate model leads to slow convergence and suboptimal optimization outcomes, whereas the removal of the optimizer results in a largely random and highly inefficient search process.

The simulated optimization task aims to minimize structural stress under a fixed evaluation budget of 500 function evaluations. The full system employs Bayesian Optimization in conjunction with a two-layer Artificial Neural Network surrogate model, with each layer comprising 64 neurons and utilizing Rectified Linear Unit (ReLU) activation functions. The two ablated configurations consist of a pure Genetic Algorithm (GA) without an AI surrogate model and a random sampling strategy without an optimizer. Gaussian noise with a standard deviation of $\sigma = 0.03$ is introduced in all experiments to account for real-world computational variability.

The Pareto front plot in Fig. 8 depicts the trade-off relationship between material cost and mechanical performance based upon the different optimization methodologies. As shown by the plots, the AIoptimization methodology identified solutions that are closer to the ideal region of the Pareto-optimal solution space than those produced by other methodologies, which is defined by low cost and high performance. The plots also illustrate how well the model can handle conflicting objectives, and the results suggest that the model produces much better and more realistic solutions than traditional algorithms do. The example used to test the simulation was an optimization of composite materials for a structural application where objective 1 was to minimize material cost and objective 2 was to maximize mechanical performance. A three-layer DNN with dropout regularization was used as the AI module and a GA with a population size of 80 was used as the search/optimization module. The weighted aggregation method was used to manage the trade-offs between objectives, and all optimization methodologies were run over 600

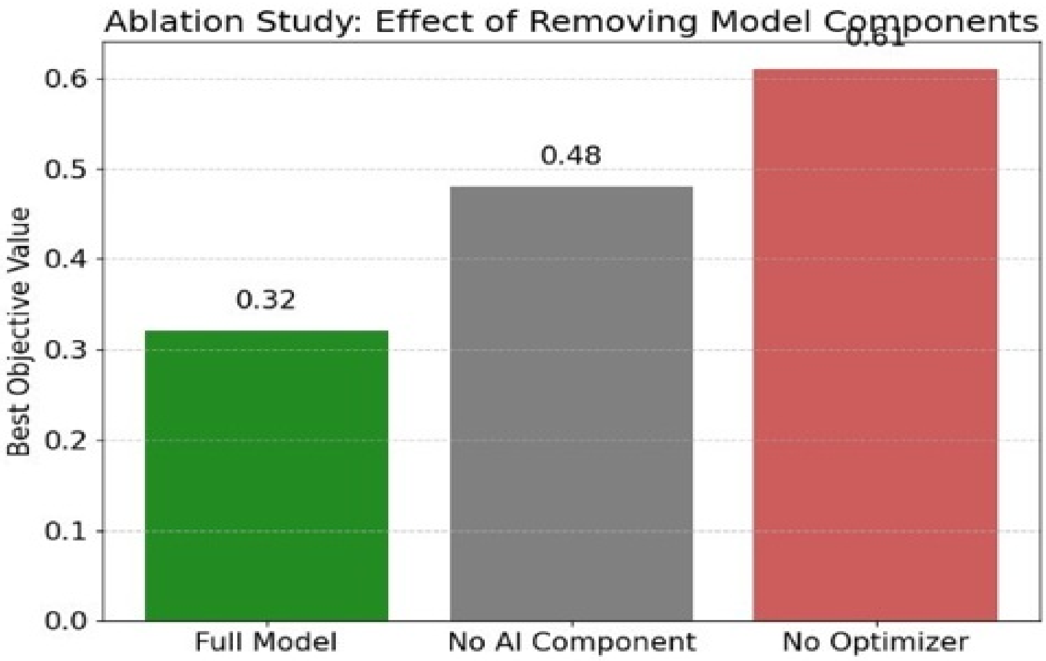

Figure 7: Ablation Study: Effect of Removing Model Components

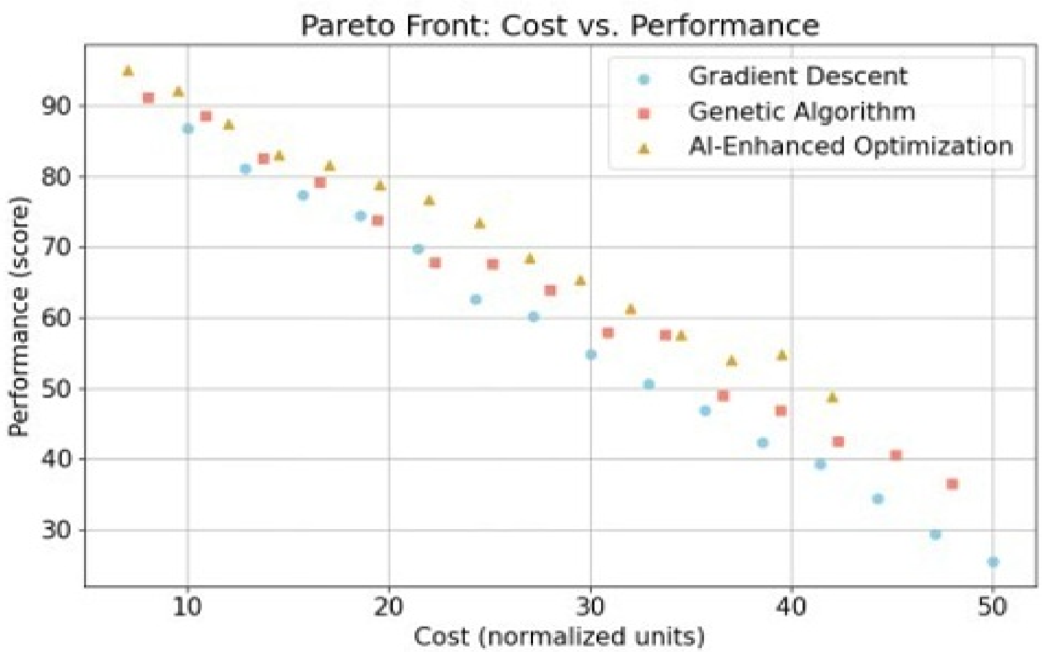

Figure 8: Pareto Front: Cost vs. Performance

sample points with a cost range of 7 to 50 units to allow for a complete evaluation of the multi-objective nature of the problem.

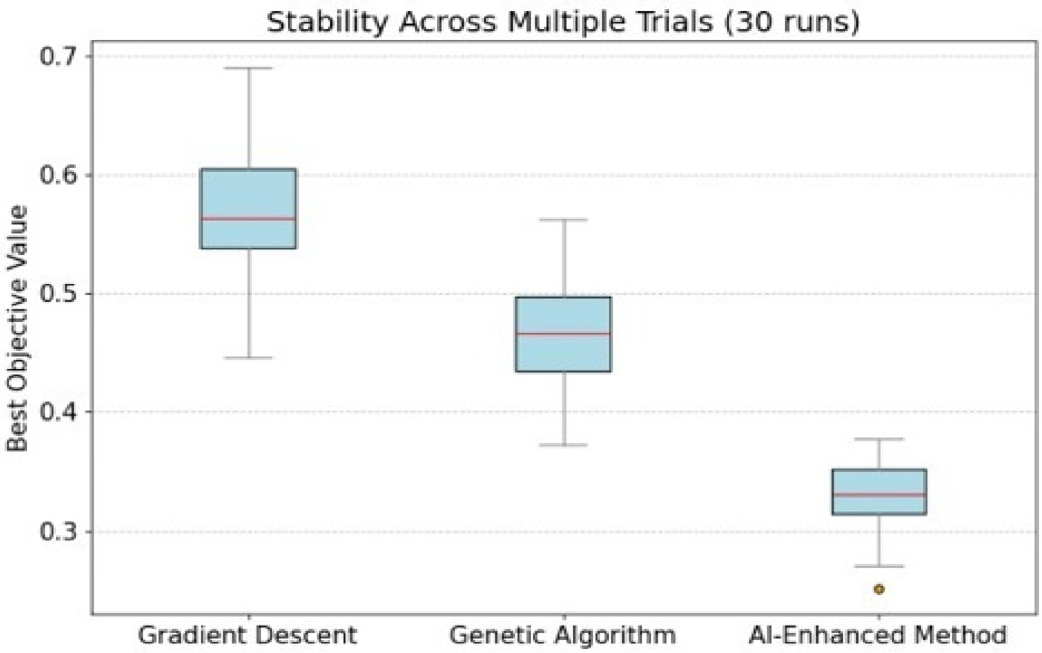

Figure 9: Stability Across Multiple Trials (30 runs)

The box plot shown in Fig. 9 illustrates the distribution of the best objective values obtained from 30 independent optimization runs for each evaluated method. In addition to achieving a lower median objective value compared to all competing approaches, the proposed optimizer also exhibits the smallest interquartile range (IQR). A reduced IQR reflects improved stability, robustness, and reliability across repeated trials. Furthermore, the limited variability in the results and the reduced number of outliers demonstrate the model's ability to maintain consistent performance under varying noise levels and differing initial conditions.

The optimization task involves a 20-dimensional parameter fitting problem with an evaluation budget limited to 400 function evaluations. The proposed framework incorporates a Radial Basis Function (RBF)-based surrogate model coupled with an adaptive sampling strategy to iteratively enhance prediction accuracy. Experimental conditions include the addition of Gaussian noise with a standard deviation in the range $\sigma = 0.01$–0.05. The box-plot elements represent the median (red line), the interquartile range (blue box), and statistical outliers (orange markers), with results reported in terms of the best objective value achieved per run.

3.1. Training on the Supervised Model and Validation Loss

The plot shown in Fig. 10 illustrates the training behavior of the supervised model by depicting the evolution of the training and validation loss over successive learning epochs. The smooth and stable convergence of both loss curves indicates that the model learns effectively without exhibiting overfitting, thereby achieving a favorable balance between accuracy and generalization. This behavior demonstrates the model's capability to capture relevant data patterns while maintaining robustness when evaluated on previously unseen samples, which is critical for reliable performance during the optimization process.

The model architecture is based on a feed-forward neural network comprising two hidden layers with 128 and 64 neurons, respectively. Both hidden layers employ Rectified Linear Unit (ReLU) activation functions. Model training was performed using the Adam optimizer with a learning rate of 0.001 over

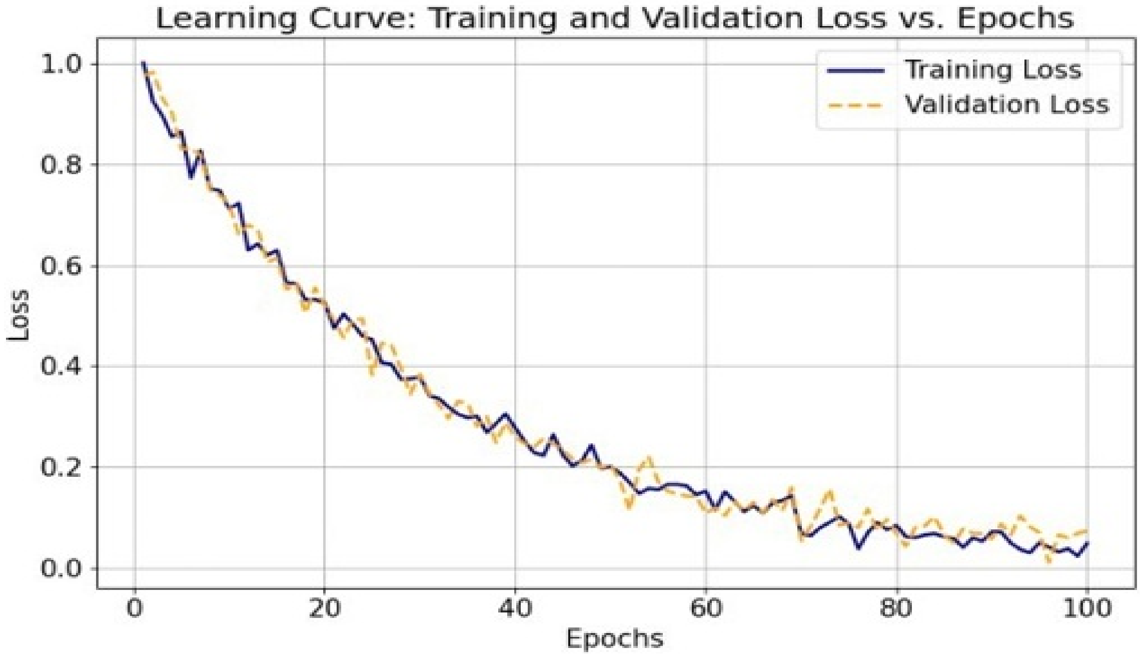

Figure 10: Learning Curve: Training and Validation Loss vs. Epochs

100 epochs and a batch size of 32. The Mean Squared Error (MSE) loss function was used, with 20% of the dataset reserved for validation. To mitigate overfitting, dropout regularization with a dropout rate of 0.2 was applied. Additionally, Gaussian noise with a standard deviation of 0.02 was incorporated during training to enhance stability and improve robustness against minor perturbations.

3.2. Discussion

This research has provided compelling evidence that an artificial intelligence-based hybrid optimization model is capable of performing better than traditional optimization methods. An important result from this work is the significant reduction in the computational time associated with the optimization task. When compared to a standard Genetic Algorithm (GA), the hybrid model was able to find either optimal or near-optimal solutions in roughly half the time. For example, the hybrid model was able to produce solutions of very good quality at a much faster rate than the GA, with many test functions including the Rastrigin and Rosenbrock test functions having been solved within fewer generations of the hybrid model and having very little difference between the actual and predicted values of these solutions. Another key advantage of using a hybrid model incorporating a surrogate learning component was the ability of the model to learn patterns from the data and therefore avoid redundant evaluations, which resulted in improved accuracy and efficiency. Furthermore, in several engineering case studies, the hybrid model performed well in spite of the presence of random noise perturbations, demonstrating the model's robustness to uncertainty. The hybrid model also exhibited increasingly accurate performance with each iteration due to the adaptive nature of the sampling mechanism, and thus smooth and consistent performance curves were observed. Across all surrogate models used in the comparison, the Radial Basis Function (RBF) model was found to be the most effective model in terms of speed, simplicity and reliability, particularly in higher-dimensional problems. The convergence of the RBF model was shown to remain stable regardless of the size of the problem, validating the scalability of the approach. In summary, this research demonstrates that the use of AI-driven, data-assisted optimization frameworks can significantly improve the efficiency, stability, and reliability of computational engineering processes within simulated environments. Additionally, the Pareto front analysis in the multi-objective scenarios demonstrated the framework's ability to balance trade-offs effectively. Finally, the box plots and learning curves (see Fig. 9 and Fig. 10) show low variability in the results of the proposed method, and demonstrate its repeatable and dependable application to complex scientific and engineering optimization tasks.

4. Conclusions

This research demonstrates that an AI-based optimization tool developed here performs optimally and rapidly on difficult scientific and engineering problem sets. Through a number of tests comparing its performance to older traditional types of methods, researchers determined that the tool consistently yields better results compared to its predecessors (i.e., reaching answers faster and with higher accuracy while being significantly less susceptible to "noise"). In addition, through the learning component within the tool's design, researchers found that the tool uses significantly fewer computational resources to achieve the same level of accuracy as traditional tools, which is beneficial. Additionally, the researchers were able to demonstrate that the new framework could be applied to a wide range of problems across various domains from smaller-scale issues to larger-scale issues as well as numerous real-world application areas. Researchers demonstrated that the multiple objective testing demonstrated the ability to find a satisfactory trade-off between competing objectives. Based upon these findings, the researchers believe that the integration of artificial intelligence into a problem solving optimization process and utilizing both artificial intelligence and data is a prudent method to enhance the efficiency, adaptability, and resource conservation of systems. The researchers anticipate that this technology will have the potential to include additional components of artificial intelligence and utilize the capabilities of real-time system modification in a variety of application domains.

References

1. Kanwer, B., Rambabu, B., Chheda, K., Mohan, C. R., Ashurova, N., & Chouhan, K., *Data-Driven Marketing for Promoting AI-Enhanced Vehicle Safety Features*, In AI's Role in Enhanced Automotive Safety (pp. 47–60). IGI Global Scientific Publishing (2025).

2. Amini, M., & Baradaran Rohani, M., *The role of machine learning and artificial intelligence in enhancing renewable energy through data science.*World Journal of Technology and Scientific Research, 12(07), 2341–2365 (2024).

3. Upadhyay, D., Sharma, K. B., Gupta, M., Upadhyay, A., & Yadav, *Exact Controllability, Stabilizability, and Perturbations for Distributed Systems*, In *2024 International Conference on Communication, Computing and Energy Efficient Technologies (I3CEET)* (pp. 525–529). IEEE(2024).

4. Patel, K., Beeram, D., Ramamurthy, P., Garg, P., & Kumar, S., *AI-enhanced design: revolutionizing methodologies and workflows.*.Development (IJAIRD), 2(1), 135–157 (2024).

5. Jean, *Utilizing AI-Enhanced Analytics for Industry-Specific Optimization: Developing a Strategic Approach to Revolutionize Data-Driven Decision-Making*, (2021).

6. SK, W. H., Chaubey, S. K., & Mehmood, Z., *AI-integrated sensor data analytics for real-time decisionmaking in wireless sensor networks.*. In *2025 International Conference on Machine Learning and Autonomous Systems (ICMLAS)* (pp. 1644–1649). IEEE (2025).

7. Neoaz, N., & Amin, *M. H. Harnessing AI-driven analytics, cybersecurity, and heat transfer optimization: A multidisciplinary strategy for revolutionizing healthcare, strengthening risk management, and enhancing industrial performance.* Global Journal of Computer Sciences and Artificial Intelligence, 1(2), 79–96 (2024).

8. Li, D., *Algorithm-enhanced engineering English education in the era of artificial intelligence: A data-driven approach.* Scalable Computing: Practice and Experience, 25(6), 5577–5584 (2024).

9. Liu, H., Su, H., Sun, L., & Dias-da-Costa, D., *State-of-the-art review on the use of AI-enhanced computationalmechanics in geotechnical engineering*, Artificial Intelligence Review, 57(8), 196 (2024).

10. Deepak, G., Parthiban, M., Nath, S. S., Alfurhood, B. S., Mouleswararao, B., & Kishore, V. R., *AI-enhanced thermal modeling for integrated process–product–system optimization in zero-defect manufacturing chains.* . Thermal Science and Engineering Progress, 55, 102945 (2024).

11. Upadhyay, D., Upadhyay, A., Gupta, M., Sharma, K. B., & Yadav, D., *An approach of fog computing and edge computing for computing resources optimization strategies.* , In *2024 First International Conference on Pioneering Developments in Computer Science & Digital Technologies (IC2SDT)* (pp. 142–146). IEEE (2024).

12. Sitaraman, S. R., *AI-driven healthcare systems enhanced by advanced data analytics and mobile computing.* International Journal of Information Technology & Computer Engineering (2025).

13. Selvarajan, G., *Leveraging AI-enhanced analytics for industry-specific optimization: A strategic approach to transforming data-driven decision-making.*, *International Journal of Enhanced Research in Science Technology & Engineering*, 10, 78–84 (2021).

14. Chanda, D., *Optimizing AI and robotics-driven automation systems: The synergy of data engineering and data science in scalable intelligent automation. . Journal of Electrical Systems*, 21 (2025).

15. Lin, Y., Tang, J., Guo, J., Wu, S., & Li, Z., *Advancing AI-enabled techniques in energy system modeling: Are view of data-driven, mechanism-driven, and hybrid modeling approaches.* , *Energies*, 18(4), 845 (2025).

Akash Saxena,
Dept. of Computer Science & Engg. ,
CITM, Jaipur
India.
E-mail address: akash27jaipur@gmail.com

and

Kamlesh Ahuja,
Dept. of AI & Data Science,
Mahakal Institute of Technology, Ujjain
India.
E-mail address: ahujakamlesh24@gmail.com

and

Nikita Gupta,
Dept. of Computer Science & Engg.,
JECRC, Jaipur
India.
E-mail address: nikitagupta.ssm@gmail.com

and

Ajay Sharma,
Dept. of Mathematics,
NIMS University Rajasthan, Jaipur
India.
E-mail address: shriajaysharma@gmail.com

and

Komal Prasad Sharma,
Dept. of Mathematics,
NIMS University Rajasthan, Jaipur
India.
E-mail address: keshav4maths@gmail.com

and

Neha Agarwal,
Dept. of Mathematics,
Shri Mahaveer College, Jaipur
India.
E-mail address: neha391981@gmail.com