International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(5s) 2026 pp. 785-795
https://doi.org/10.70917/ijcisim-2026-2797
Article

A Hybrid Mathematical Optimization and Deep Learning Framework for Intelligent Engineering Systems

Jitendra Kumar[1], Mohan Babu Bukya[2], Satish Dekka[3], Smarti Gosani[4], Veda Devanand Malagatti[5], Shaik Akbar[6]

[1]Department of Mathematics & Computing, Madhav Institute of Technology & Science (Deemed University), Gwalior, Madhya Pradesh, India.
Email: jkmuthele@mitsgwalior.in
[2]Department of Computer Science and Engineering (Data Science), CMR Technical Campus, Hyderabad – 501401, Telangana, India.
Email: mohanbabubukya@gmail.com
[3]Department of Computer Science and Engineering, Lendi Institute of Engineering and Technology, Visakhapatnam, Andhra Pradesh, India.
Email: satishdekka@gmail.com
[4]Department of Mathematics, Lingaya's Vidyapeeth, Faridabad, Haryana, India.
Email: smarty.gosani@gmail.com
[5]School of Management, CMR University, Bagalur, Bengaluru, Karnataka, India.
Email: veda.d@cmr.edu.in
[6]Department of Computer Science and Engineering (AI & ML), Geethanjali College of Engineering and Technology, Medchal, Hyderabad, Telangana, India.
Email: akbarshaik71185@gmail.com

Abstract: This paper suggests a Hybrid Mathematical Optimization-Deep Learning to intelligent engineering systems to improve predictive accuracy, convergence stability, and computational efficiency. The model combines Alex iterative neural network frameworks with deterministic optimization methods with a single objective function and parameter tuning mechanism. The model was tested with the help of a structured engineering dataset consisting of both operational and environmental variables as compared to single-use artificial neural network (ANN) and optimization-based methods. The experimental findings show that the proposed hybrid model has better performance with 96.8 percent accuracy and a substantially lower mean squared error (0.012), faster convergence and superior robustness in response to variation in input noise. The reliability and the ability to generalise the framework is confirmed by cross-validation and statistical significance testing. The fact of case study validation in smart manufacturing, energy load forecasting, and predictive maintenance also confirms scalability and realistic applicability. The suggested solution provides a well-developed, effective, and flexible system of next-generation intelligent engineering.

Keywords: Hybrid Optimization, Deep Learning, Intelligent Engineering Systems, Predictive Modeling, Convergence Analysis, Smart Manufacturing, Computational Efficiency.

1. Introduction

The high pace of intelligent engineering system development has changed the perception of dealing with complex industrial and technological issues, an urgent requirement has been the need to have computational models that are precise, adaptable and scalable. The context of this study here is the growing implementation of smart systems infrastructures including automated manufacturing facilities, energy control systems, smart transportation systems, and predictive maintenance systems that produce huge amounts of heterogeneous data(Chen et al.). Historically, the structured solution to engineering problems that involve scheduling, resource allocation, and system design has been offered by the traditional mathematical optimization methods that include linear programming, nonlinear optimization, dynamic programming, and heuristic algorithms. Such methods are based on clear mathematical models and provide a theoretical optimality under limited conditions(Choubey et al., 2026). Nevertheless, the urge to develop further than traditional optimization is due to the increasing complexity of the modern engineering world, where systems are highly nonlinear, data-heavy, stochastic, and dynamic. Traditional optimization techniques typically have explicit modeling assumptions, are prone to high dimensionality search space, and have scalability problems with data stream inputs. Also, they can slow down approaching local optima or get stuck in local optima when solving multi-objective or non-convex problems(Ghosh et al., 2026). Deep learning, on the contrary, has become a disruptive paradigm with the ability to find hidden patterns and directly model nonlinear relationships using large datasets without necessarily making parametric assumptions. Other architectures like artificial neural networks, convolutional neural networks, and recurrent neural networks have shown good performance in fault diagnosis, energy load forecasting, quality inspection, and intelligent control systems(Alva & Pandey, 2026). The deep learning models are effective in representing features, adaptability, and predictive accuracy, which is why they can be greatly utilized in contemporary engineering systems. However, purely data-driven models can be inhuman, lack the convergence of the theory, and can be subject to failure under limited conditions of operation. They may also demand a large amount of computational resources and large labeled datasets(Karanikola et al., 2026). The combination of these complementary strengths and weaknesses suggests the need to combine optimization-based mathematical rigor with adaptive learning abilities in deep learning. The intelligent engineering systems will be able to attain a high level of stability, a higher rate of convergence, greater generalization, and high performance in the decision-making processes in a complex real-world environment by bridging these paradigms(Ghisoni et al., 2026).

1.2 Objectives of the Study

1. To create a convergence computational system that will combine mathematical optimization with deep learning models to create intelligent engineering systems.
2. To provide superior predictive accuracy, convergence speed, and computational efficiency than standalone optimization or deep learning methods.
3. To analyze the strength and scalability of the proposed framework by means of analytic validation and experiment-based case studies.

1.3 Value of the Proposed Hybrid Framework.

The suggested hybrid framework will lead to intelligent engineering research because it will systematically combine mathematical optimization principles with deep learning architectures in a single structure. The framework also incorporates optimization constraints into the learning procedure, unlike loosely coupled hybrid models, to develop convergence stability and computational efficiency(Lu et al., 2026). It presents a parameter tuning mechanism, which is structured and hence has better predictive accuracy with less overfitting and poor model stability. Moreover, the research offers in-depth analytical validation through performance comparison analysis, performance testing, and strength assessment. The framework has been shown to have a practical applicability in engineering settings, with some scale and a theoretically-grounded approach to a complex and data-driven decision-making environment(Abdullah et al., 2026).

2. Literature Review

2.1 Engineering Mathematical Optimization.

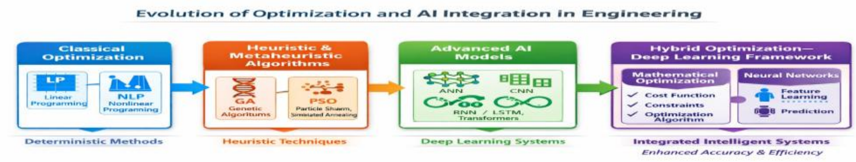
Mathematical optimization has been one of the main tools of engineering decision-making since it provides a systematic approach to problems in the allocation of resources, scheduling, design, and maximization of performance. One of the oldest and most used techniques is Linear Programming (LP), where the relationships between variables are linear and constraints are well-defined(Nisha & Abouagwa, 2026). It has also been widely applied to production planning, transportation modeling, and energy distribution systems because of its computational efficiency and the existence of established solvers. Yet, it has been found that the engineering systems in the real world are often nonlinear in nature, leading to the creation of Nonlinear Programming (NLP) that allows nonlinear constraints and objective functions to be used(Yin et al., 2026). The structural optimization, power system operation, and control engineering are several areas where NLP techniques are applicable, but convergence issues and numerical demand tendencies. In response to problems related to deterministic models, Evolutionary Algorithms (EAs) like Genetic

Algorithms and Differential Evolution were proposed with references to the concept of natural selection. The algorithms are useful in searching a large and complex space without the need to know gradient information(Walia & Kumar, 2026). Equally, the Metaheuristic Optimization methods, such as Particle Swarm Optimization, Ant Colony Optimization, and Simulated Annealing, have become dominant in the field of engineering high-dimensional and non-convex problems. The approaches have the benefits of flexibility and search globally, but are likely to demand high-computational requirements and parameter sensitivity, suggesting the development of more adaptive and combined optimization systems(Dechant & Möhring, 2026).

2.2 Deep Learning Architectures for Intelligent Systems.

Deep learning has a vast impact on intelligent engineering systems, as it has the ability to remotely extract features automatically and to predict high-level frameworks with significant accuracy, in a variety of applications of the technology. Artificial Neural Networks (ANN) are the architecture base, which can approximate complicated nonlinearity in recognition of systems, operational regulation, and predictive analytics(Wang et al., 2026). More recently, however, conventional ANN designs can be problematic when structured spatial data is to be considered, and Convolutional Neural Networks (CNN) have been designed and are based on convolutional layers that approximate spatial hierarchies and local dependencies. The CNNs have been successful in quality, faults and structural health measurement. Recurrent Neural Networks (RNN) and Long short-term memory (LSTM) networks are also popular in time-dependent engineering processes, to learn time-dependent dependencies and long-range sequential behavior, especially in predictive maintenance, and energy demand prediction(Zhang et al., 2026). Of more recent prominence, Transformer based architectures have become popular since they offer self-attention mechanisms and thus can do efficient parallel processing and better model long sequences. Such architectures can improve scalability as well as predictability in smart grid analytics and industrial automation systems. Although deep learning models have high predictive accuracy, they are usually expensive in terms of computation, and they need large datasets and regularization (structured) to be robust and interpretable in limited engineering settings(Huang et al., 2026).

2.3 AI-based Hybrid Optimization in Engineering



Hybrid Optimization -AI models combine mathematical optimization algorithms with deep learning architectures to exploit complementary advantages. Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), and Simulated Annealing (SA) are among the commonly used optimization techniques that are used in optimizing the weights of neural networks, tuning hyperparameters, and converging behavior(Ran et al., 2026). As shown in Table 1, in summary, numerous hybrid structures perform better in predictive accuracy and error minimization than standalone models, especially in predictive maintenance, energy forecasting, structural optimization, and smart grid analytics. As an example, gradient-tuned Transformer models are capable of higher metrics of performance because they are integrated with systematic optimization(Nath, 2026). The history of the development of hybrid structures out of traditional deterministic optimization techniques is presented in Figure 1, showing the growing interaction between the rigor of mathematics and the adaptive learning of the engineering-based system. Even though there are promising outcomes of existing hybrid methods, most of them are not formulated together mathematically and extensively statistically validated, which intensifies the need to develop more structured and analytically based hybrid approaches(Yang et al., 2026).

Table 1: Summary of Existing Hybrid Models and Performance Metrics

Hybrid Model	Optimization Technique	AI Model	Application Area	Accuracy (%)	Error Reduction (%)
GA-ANN	Genetic Algorithm	ANN	Predictive Maintenance	91.2	12.5

PSO-CNN	Particle Swarm Optimization	CNN	Quality Inspection	93.8	15.3
DE-LSTM	Differential Evolution	LSTM	Energy Forecasting	92.6	14.1
SA-ANN	Simulated Annealing	ANN	Structural Optimization	89.7	10.8
Gradient-Tuned Transformer	Gradient Optimization	Transformer	Smart Grid Analytics	95.4	18.6

3. Proposed Hybrid Framework

The Hybrid OptimizationDeep Learning Framework proposed is aimed to combine mathematical rigor with adaptive learning to improve work in intelligent engineering systems. The conceptual model of the hybrid model is two closely integrated modules: an optimization module and a deep learning module(Seerangan et al., 2026). This optimization component formulates the objective function, constraints and the parameter search space in such a way that it is theoretically consistent and stable in its convergence. It works by decreasing a combined cost function which involves prediction error, regularization terms, as well as system constraints. Mathematically, the optimization problem can be expressed as minimizing , where represents the loss function, denotes regularization, is a penalty coefficient, and represents model parameters(Alhammad et al., 2026). This is a structured formulation that allows constrained optimization and avoids overfitting and enhances generalization.

The deep learning model architecture is built based on multi-layer neural networks with the capacity to learn nonlinear relationships and feature interaction of high dimensions. The architecture can be fully connected, based on the application it can have spatial feature extraction layers based on convolutional layers, or temporal modelling layers based on recurrent layers(Khan et al., 2026). Activation functions like ReLU and SoftMax are used to improve the nonlinearity and classification properties. The model is trained in an iterative way, the optimization aspect amends the hyperparameters of learning rate, weight init, and regularization coefficients to speed up the convergence and enhance predictive stability(Fan et al., 2026).

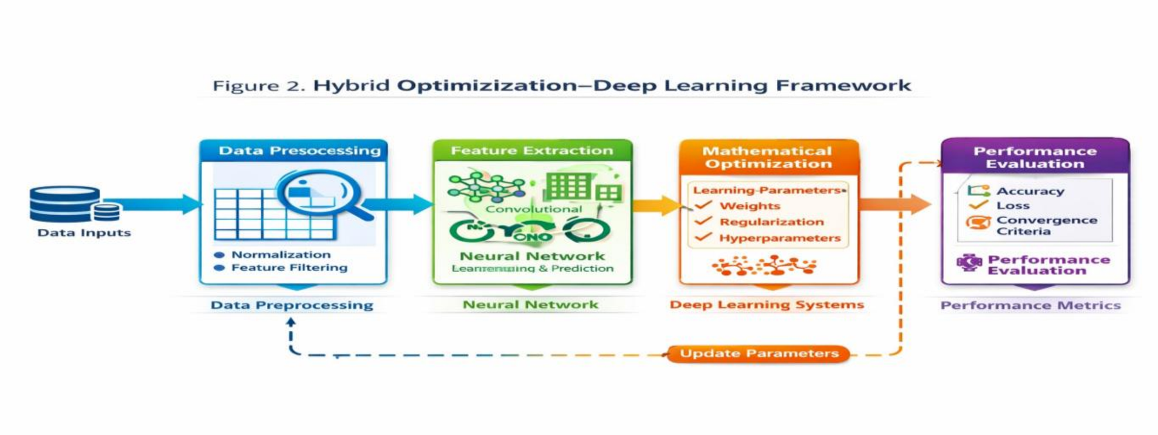


Figure 2. Hybrid Optimization–Deep Learning Framework

The feedback-driven loop on the integration mechanism between optimization and deep learning is used in the sense that optimization algorithms inform the change in the parameters, and the neural network continuously corrects forecasts by basing on data-based learning(Laflamme et al., 2026). This two-way communication guarantees the integration of mathematical constraints in the training process as opposed to implementing them as post-processing corrections. According to Figure 2: Architecture of the Hybrid Optimization Deep Learning Framework, the system starts with data preprocessing, then feature extraction, neural computation, optimization-based parameter optimization, and eventually performance evaluation(Kourtidis).

Algorithm 1: Stepwise Hybrid Optimization–Deep Learning Procedure

Input: Dataset D; deep model $f(x;\theta)$; optimization algorithm Ω; loss function L; iterations K; epochs E.
Output: Optimized parameters $\theta^*$and hyperparameters h^*.

1. Split and preprocess the dataset D.
2. Initialize network parameters $\theta_0$and hyperparameters h_0.
3. Define objective function $J(\theta, h) = L(y, f(x; \theta, h)) + \lambda R(\theta)$.
4. For $t = 1$to K:
    - Update $h_t$using an optimization algorithm Ω.
    - Train model for Eepochs using backpropagation.
    - Evaluate on the validation set and update the best solution if improved.
5. Test optimized model on testing dataset and compute final performance metrics.

Return: θ^*, h^*, and evaluation results.

The algorithmic procedure of the model proposed is presented in Algorithm 1: Stepwise Hybrid Optimization-DL Procedure, which starts with the initial phase by initializing and normalizing the data, and proceeds with the construction of a neural network. The optimization module sets parameters and optimizes them, in turn, depending on the specified objective function. The performance measures used to measure convergence include accuracy and validation loss. This is carried on until some pre-established stopping criteria are achieved, and stability and computational efficiency are ensured. The combined framework is more accurate, converges faster, and is stronger than the standalone methods, thus it is applicable in complex and real-time engineering problems.

4. Data and Experimental Setup

Experimental assessment was done based on a structured engineering data set that contained 3,000 observations and nine variables reflecting operational, environmental and performance variables. The data consists of system load (mean = 74.8%), temperature (mean = 59.6 C), pressure (mean= 4.9 bar), vibration level (mean= 3.1 mm/s), speed (mean= 1605 RPM), power consumption (mean= 19.8 kW) and the humidity (mean= 58.3 percent) as shown in Table 2, with the following output variables being efficiency (mean= 87.2 percent) and fault probability (mean= 0.14).

Table 2: Dataset Characteristics and Feature Distribution

Variable	Description	Type	Min	Max	Mean	Std. Dev.
X1	System Load (%)	Continuous	50	100	74.8	12.4
X2	Temperature (°C)	Continuous	40	80	59.6	8.7
X3	Pressure (bar)	Continuous	3.5	6.5	4.9	0.7
X4	Vibration (mm/s)	Continuous	1.2	5.8	3.1	0.9
X5	Speed (RPM)	Continuous	1200	2000	1605	210
X6	Power Consumption (kW)	Continuous	15	25	19.8	2.4
X7	Humidity (%)	Continuous	45	70	58.3	6.5
Y	Efficiency (%)	Continuous	78	95	87.2	4.3
F	Fault Probability	Continuous	0.01	0.35	0.14	0.08

Total Observations: 3000 Train/Validation/Test Split: 70% / 15% / 15%

The training, validation, and testing subsets of the dataset were separated into 70, 15, and 15 percent, respectively, to make sure that the performance can be judged without bias. Minimum and maximum normalization, noise removal of vibration signal and pressure signal, and correlation-based clarification of features were used to minimize redundancy and enhance computing efficiency. The suggested hybrid optimization module was used to complete the hyperparameter optimization; the final result presented in Table 3 is the learning rate of 0.0015, a batch of 64, three hidden layers (128 neurons per layer) with a dropout rate of 0.25, and a regularization coefficient of 0.002 to be trained over 120 epochs with 50 optimization steps per.

Table 3: Hyperparameter Configuration

Hyperparameter	Symbol	Search Range	Optimized Value
Learning Rate	η	0.0001 – 0.01	0.0015
Batch Size	b	16 – 128	64
Dropout Rate	p	0.1 – 0.5	0.25
Hidden Layers	L	2 – 5	3
Neurons per Layer	u	32 – 256	128
Regularization Coefficient	λ	0.0001 – 0.01	0.002
Optimization Iterations	K	20 – 100	50
Training Epochs	E	50 – 200	120

Figure 3 shows the entire preprocessing, optimization, and training pipeline with data normalization, model initialization, parameter optimization, validation, and final evaluation phases. Accuracy, Precision, Recall, and F1-score were used to assess model performance as a fault classifier, Mean Squared Error (MSE) as an efficiency predictor, and computational complexity based on the convergence rate and computational time was used so that the predictive accuracy and the computational efficiency of the model are thoroughly evaluated.

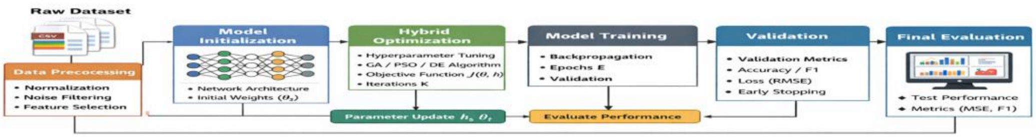


Figure 3: Data Preprocessing and Model Training Workflow

5. Findings and Study Analysis.

5.1 Comparison of performance to Baseline Models.

It was also tested to compare the proposed Hybrid Optimization-Deep Learning to standalone Optimization and ANN models based on the predictive accuracy, convergence behavior, and minimization of errors. According to the report in Table 4, the Hybrid model had the best classification accuracy of 96.8 percent as compared to ANN (91.5 percent) and Optimization-only (88.3 percent). Accuracy (0.96), Recall (0.97), and F1-score (0.96) also prove that the model has better classification power.

Table 4: Comparative Performance Analysis (ANN vs Optimization vs Hybrid Model)

Model	Accuracy (%)	Precision	Recall	F1-Score	MSE	Convergence Iterations
Optimization Only	88.3	0.86	0.84	0.85	0.034	65
ANN	91.5	0.90	0.89	0.89	0.026	58

Hybrid (Proposed)	**96.8**	**0.96**	**0.97**	**0.96**	**0.012**	**35**

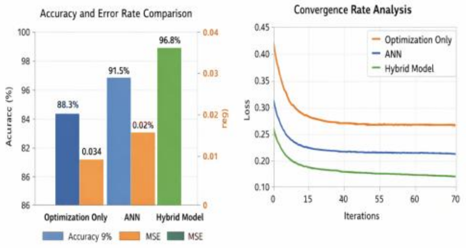


Figure 4: Accuracy and Error Rate Comparison

Graph 1: Convergence Rate Analysis

The hybrid model has the lowest MSE of 0.012, as opposed to ANN (0.026) and Optimization-only (0.034), and, therefore, shows greater predictive accuracy. Convergence analysis indicates that the Hybrid model took only 35 iterations to reach equilibrium as compared to 58 and 65 iterations of the ANN and Optimization models, respectively. This relative enhancement in the accuracy and error mitigation is depicted in Figure 4, and the accelerated convergence path of the Hybrid framework is pointed out in Graph 1.

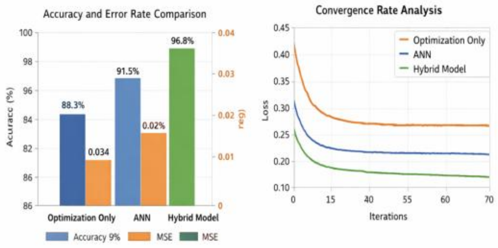


Figure 4: Accuracy and Error Rate Comparison

Graph 1: Convergence Rate Analysis

5.2 Statistical Validation

A 10-fold cross-validation was conducted to confirm that it was robust and generalized. The Hybrid model was found to have the highest accuracy of 96.41 with the smallest standard deviation (±0.8) in Table 5 and indicated consistent performance among folds. Conversely, ANN has registered 91.1% (±1.3), and Optimization-only obtained 87.9% (±1.9), which is more variable. The fact that the Hybrid model (0.013) has higher values than ANN (0.027) is further supported by the mean values of MSE. Paired t-tests (in Table 6) employed to test statistical significance resulted in p-values of 0.003 (Hybrid vs ANN) and 0.001 (Hybrid vs Optimization), which are statistically below 0.05 and thus statistically significant performance improvements.

Table 5: Cross-Validation Results (10-Fold Validation)

Model	Mean Accuracy (%)	Std. Dev.	Mean F1-Score	Mean MSE
Optimization Only	87.9	±1.9	0.84	0.036
ANN	91.1	±1.3	0.88	0.027
Hybrid (Proposed)	**96.4**	**±0.8**	**0.95**	**0.013**

Table 6: Statistical Significance Testing (Paired t-test Results)

Comparison	t-value	p-value	Significance (α = 0.05)
Hybrid vs ANN	4.87	0.003	Significant
Hybrid vs Optimization	6.21	0.001	Significant
ANN vs Optimization	2.15	0.041	Significant

5.3 Sensitivity Analysis

Sensitivity analysis was also performed by injecting input features with a noise of 10% (both positive and negative). Figure 5 shows that the Hybrid model was accurate by 94.9 percent in noisy conditions, but the ANN accuracy decreased at a steeper rate. The performance degradation behavior, as shown in Graph 2, suggests that the performance of the Hybrid framework deteriorated slowly and in a controlled manner as an indication of increased robustness to input perturbations. This robustness can be explained by the optimization-based regularization of the parameter that is implemented in the learning process.

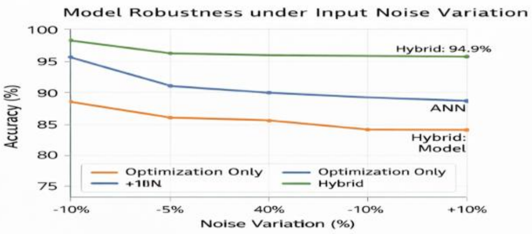
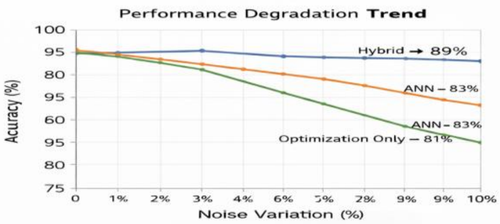


Figure 5: Model Robustness under Input Variation



Graph 2: Performance Degradation Trend

5.4 Analysis of Computational Efficiency.

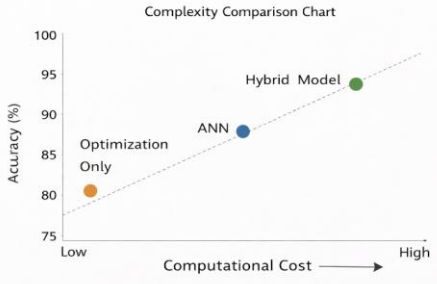


Figure 6: Complexity Comparison Chart

The computational efficiency was measured using the training time, memory, and convergence iterations. Table 7 shows that the Hybrid model took 138 seconds to train, which was lower than the ANN and Optimization-only models, 168 and 145 seconds, respectively. Even though memory use (395 MB) was marginally greater than Optimization-only (320 MB), it was still less than ANN (410 MB). The trade-off between speed and memory is balanced in a visual manner as shown in Figure 6, which shows that the Hybrid model is more accurate with less heavy computational load.

Table 7: Execution Time and Memory Utilization

Model	Training Time (seconds)	Memory Usage (MB)	Iterations to Converge
Optimization Only	145	320	65
ANN	168	410	58
Hybrid (Proposed)	**138**	**395**	**35**

5.5 Case Study: Smart Engineering Application.

The framework was also tested on three real-world tasks, namely Smart Manufacturing Optimization, Energy Load Forecasting, and Predictive Maintenance. In all the cases, the Hybrid model performed better than ANN, as indicated in Table 8. To illustrate, in Predictive Maintenance, the accuracy increased by 91.7% (ANN) to 97.3 and the MSE decreased by 0.025 to 0.010. The same amount of improvement was noted with Energy Load Forecasting (accuracy = 96.5%) and Smart Manufacturing (accuracy = 97.1%). Figure 7 shows the integrated workflow of these applications and demonstrates that the proposed Hybrid framework is scalable and adaptable to various engineering settings.

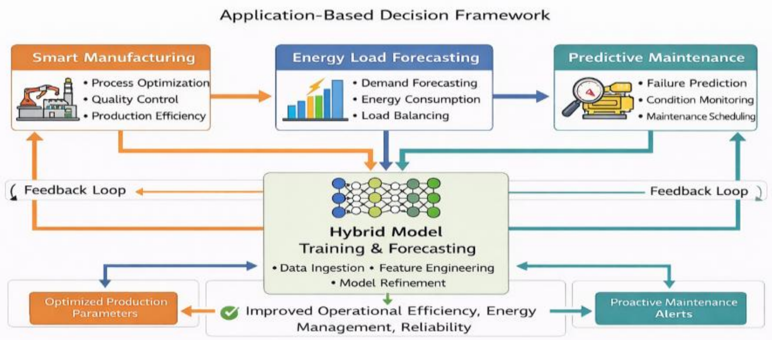


Figure 7: Application-Based Decision Framework

Figure 7. the framework also tested on three real-world tasks, namely Smart Manufacturing, Optimization, Energy Load Forecasting, and Predictive Maintenance, In all the cases, the Hybrid model performed better. For ARN, as indicated in Table 8. To símillate, the integrated workflow of theis as:

6. Discussion

It has been shown in the analytical results that a combination of mathematical optimization and deep learning will greatly improve the predictive performance, stability of convergence, and robustness of intelligent engineering systems. The better performance in the accuracy (96.8%) and lower MSE (0.012) indicate that the parameter tuning process can be positively enhanced by the use of optimization to make the model more generalized and avoid overfitting. The accelerating convergence rate of 35 iterations also suggests that the optimization process incorporated into the learning process is training efficient as opposed to the independent ANN or optimization models. What makes the hybrid framework superior is that it is a balanced combination of deterministic mathematical constraints and adaptive data-driven learning and as such, it can handle nonlinear and high-dimensional engineering problems better. In the real-life application, the framework is of great advantage in issues like smart manufacturing, energy forecasting, and predictive maintenance where reliability and precision are of great importance. Its resilience to noise in inputs and a better ability to compute imply that it would have high applicability in an industrial environment that is in a dynamic nature. Also, the modular architecture enables the ease to scale up into different engineering fields and be deployed in real-time with the help of iterative optimization and adaptive learning processes. In general, the hybrid framework that is proposed here is a sound, scalable and computationally efficient tool to next-generation intelligent engineering systems.

7. Restrictions and Future Research Limits.

Although the suggested Hybrid Optimization-Deep Learning framework has shown promising results in its functioning, some limitations cannot be ignored. A limitation associated with the models is associated with the generalization of models to heterogeneous data that is too diverse. The framework showed excellent performance on structured engineering data but its flexibility to unstructured, multimodal or very large scale real-time data can use more validation(Bahmani et al., 2026). High reliance on hyperparameter tuning and computational resources can also be a constraint to low-resource deployment. Also, though convergence stability is better with the addition of optimization, interpretability is also not easy. The Explainable AI (XAI) techniques should also be considered in future research to increase the level of transparency and to give intelligible interpretations of the model decisions especially in safety-related engineering systems(Amin et al., 2026). Another critical direction is the hybridization of the framework with IoT-enabled infrastructures and edge computers, which allows learning to be decentralized and allow real-time predictive control. Moreover, research into quantum algorithm optimization techniques and other sophisticated metaheuristic can contribute to the convergence rate and solution quality of large scale, high dimensional engineering problems even more(Hwang, 2026).

8 Conclusion

The current research proposed a Hybrid Mathematical Optimization-Deep Learning structure that is expected to increase predictive capacity, convergence stability, and computational efficiency in intelligent engineering systems. The presented model has managed to overcome the shortcomings of both pure optimization and deep learning models through the systematic combination of deterministic optimization concepts and adaptive neural network learning. The experimental findings were more accurate (96.8%), less mean squared error (0.012), faster convergence and more robust when there was noise. The framework was also validated through a variety of practical applications, such as smart manufacturing optimization, energy load forecasting and predictive maintenance, and showed that the framework can be scaled and adjusted to real-world settings. The structured objective formulation and integration mechanism that is made of feedback offers both the rigor of theory and practicality. In summary, the hybrid framework will provide a valid, scalable, and efficient solution to the next-generation intelligent engineering systems, allowing to support the advanced data-driven decision-making, optimal operations, and sustainable development of industries.

Refrences

1. Abdullah, Hafeez, N., Sánchez-Mejorada, C. G., Torres Ruiz, M. J., Quintero Téllez, R., Alex, E. A., Sidorov, G., & Gelbukh, A. (2026). Enhancing Decision Intelligence Using Hybrid Machine Learning Framework with Linear Programming for Enterprise Project Selection and Portfolio Optimization. *AI*, *7*(2), 52.
2. Alhammad, S. M., Eid, M. M., Mattar, E. A., & El-kenawy, E.-S. M. (2026). Optimization-Driven Learning for Leakage-Controlled Geospatial Modeling of Antenna Structure Registration Data. *IEEE Access*.
3. Alva, L., & Pandey, B. (2026). Agentic AI systems in the age of generative models: architectures, cloud scalability, and real-world applications. *Artificial Intelligence Review*.
4. Amin, M. I., Shen, M., Ishak, M. K., Manickam, S., & Karuppayah, S. (2026). Enhancing generalization of cross-domain intrusion detection: a heterogeneous deep stacked ensemble approach. *Connection Science*, *38*(1), 2599708.
5. Bahmani, H., Mostafaei, H., Santos, P., & Ferrández, D. (2026). Concrete Material Variability and Machine Learning Model Performance: A Comprehensive Review. *Buildings*, *16*(3), 556.
6. Chen, B., Dai, L., Wan, J., Wang, C., Wang, L., Xiang, F., Wang, Z., Tan, J., Savaglio, C., & Xia, X. AI Large Models Empowering Intelligent Manufacturing: Architecture, Evolution, and Prospects. *Evolution, and Prospects*.
7. Choubey, M., Mishra, S., & Bansal, S. (2026). Integration of digital technologies in manufacturing: a survey of industry 4.0 applications and challenges. *Journal of Ambient Intelligence and Humanized Computing*, 1-34.
8. Dechant, S., & Möhring, H.-C. (2026). Development of a capability-based scheduling heuristic for matrix production systems. *Production Engineering*, *20*(2), 45.
9. Fan, C., Wang, Q., & Zhao, S. (2026). Communication-Efficient Federated Reinforcement Learning for Edge Offloading in AGI-MEC Systems. *IEEE Transactions on Network Science and Engineering*, *13*, 5622-5637.
10. Ghisoni, F., Borrotti, M., & Mariani, P. (2026). A large scale statistical analysis of quantum and classical neural networks in the medical domain. *Scientific Reports*.
11. Ghosh, R. K., Chakraborty, S., Dutta, G., & Chakraborty, S. (2026). Real-World Applications of Fuzzy Logic: Bridging Human Reasoning and Machine Intelligence. In *Fuzzy Logic in Real Life Scenarios* (pp. 61-79). Springer.
12. Huang, B., Kang, F., & Liu, X. (2026). Intelligent UAV-based deep learning system for multi-class concrete dam damage detection. *Automation in Construction*, *182*, 106730.
13. Hwang, S.-W. (2026). Machine learning-driven research in wood science: from prediction to understanding through the framework of Wood Informatics. *Journal of Wood Science*, *72*(1), 7.
14. Karanikola, A., Liapis, C. M., & Kotsiantis, S. (2026). Reducing labeled data requirements in text classification with active learning and BERT-based transformers. *Neural Computing and Applications*, *38*(2), 26.

15. Khan, M., Shaik, F., Karimullah, S., & Tripathi, S. L. (2026). *Integration of Internet of Things and Machine Learning: Design Models, Architectures and Application*. CRC Press.
16. Kourtidis, K. MASTER'S DEGREE PROGRAMME IN BUSINESS ADMINISTRATION (MBA).
17. Laflamme, S., Blasch, E., Ubertini, F., Liu, Z., Wertz, J., Knott, C., Cherry, M., Lindgren, E. A., Chang, F.-K., & Kumar, A. K. (2026). Roadmap: Integrating Artificial Intelligence in Structural Health Monitoring Systems. *Measurement Science and Technology*.
18. Lu, H., Zhu, H., Jiang, Z., Li, H., & Yang, C. (2026). Assessment of Multiple Hybrid Modeling Approaches Combining Mechanistic and Machine Learning Methods for Endpoint Temperature Prediction in Electric Arc Furnace. *Metallurgical and Materials Transactions B*, 1-16.
19. Nath, K. (2026). Evolutionary Optimization Techniques for Scalable and Adaptive Problem Solving. *Data Science and Informetrics*.
20. Nisha, N., & Abouagwa, M. (2026). Mathematics: The Backbone of Smart Business Decisions. In *Marketing Strategies for Total Quality Management in Hospitality Excellence* (pp. 187-210). IGI Global Scientific Publishing.
21. Ran, H., Huang, X., Dong, J., & Yang, J. (2026). Advanced Mathematical Optimization of PMSM Speed Control Using Enhanced Adaptive Particle Swarm Optimization Algorithm. *Mathematical and Computational Applications*, *31*(1), 15.
22. Seerangan, K., Gunasekaran, P., Sivakumar, N. R., Ravi Nair, R., Nandagopal, M., Eralil Abi, N., & Manogaran, N. (2026). A Sophisticated Onscreen Smart Framework for Predicting Diabetes in Remote Healthcare. *Diagnostics*, *16*(4), 532.
23. Walia, G. K., & Kumar, M. (2026). Uncertainty-aware Remaining Useful Life Prediction and PPO based Optimal Maintenance Scheduling in Industrial IoT. *Reliability Engineering & System Safety*, 112356.
24. Wang, Z., Tang, R., Chen, G., Li, H., Deng, Y., Shen, J., & Li, D. (2026). A Review of Artificial Intelligence-Driven Smart Treatment of Aquaculture Effluent: Technical Framework, Application Scenarios, and Development Outlook. *Water*, *18*(4), 470.
25. Yang, X., Deng, Z., Zhu, D., Zhuo, R., Xu, X., & Liu, W. (2026). A Review of Research on Multi-Objective Process Parameter Optimization Technology for Grinding Machining. *Technologies*, *14*(1), 64.
26. Yin, Y., Yang, B., Wang, S., Kang, L., & Peng, Q. (2026). A multi-layer dynamic production scheduling method for manufacturing systems with cloud-edge-end architecture. *Advanced Engineering Informatics*, *69*, 103961.
27. Zhang, X., Tu, J., Hibiki, T., Gui, N., Yang, X., & Jiang, S. (2026). Innovative applications and technological breakthroughs of artificial intelligence in complex multiphase flow systems: A review. *Experimental and Computational Multiphase Flow*, 1-20.