SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Complex System Engineering For Scalable Architectures: A Structural Complexity Optimization Approach

Savinder Kaur[1], Sachin Sharma[2], Shanthi R[3], Kumkum Bala[4], Amol Bhilare[5], Budigi Prabhakar[6], Antonibiya S[7], Irbutaeva Lola Tashbekovna[8]

[1] Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, Punjab 140401, India. Email: savinder.kaur.orp@chitkara.edu.in
ORCID: 0009-0004-6109-0772
[2] Assistant Professor, Department of Computer Application, Dev Bhoomi Uttarakhand University, Dehradun, Uttarakhand, India. Email: socse.sachin@dbuu.ac.in ORCID: 0009-0000-7449-5809
[3] Assistant Professor, Department of Mathematics, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: shanthir@maher.ac.in
[4] Department of Computer Engineering, Bharati Vidyapeeth's College of Engineering, Lavale, Pune, Maharashtra, India. Email: itskumkum.bala@gmail.com
[5] Assistant Professor, Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra 411037, India. Email: amol.bhilare@vit.edu
[6] Department of Physics, Noida International University, Greater Noida, Uttar Pradesh 203201, India. Email: budigi.prabhakar@niu.edu.in
[7] Assistant Professor, Department of Mathematics, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: antonibiya@maher.ac.in
[8] Department of Medical Rehabilitation, Sports Medicine and Traditional Medicine, Samarkand State Medical University, Samarkand, Uzbekistan. Email: lola_tulypan@mail.ru ORCID: 0009-0007-3632-855X

Abstract

The increasing complexity of modern engineered systems arises from tightly coupled components and their interdependent interactions, necessitating robust approaches for scalable and efficient system design. Research presents a Complex System Engineering method that models product architectures as dependency networks, represented through adjacency matrices to capture structural relationships among components. A quantitative structural complexity metric is developed to evaluate and compare different architectural configurations. Data used for experimentation is collected from the Structural Complexity Design Dataset, which consists of 13000 data. The proposed approach integrates preprocessing using Min–Max normalization to scale interaction weights for consistent analysis, and Linear Discriminant Analysis (LDA) to extract the most discriminative structural features that enhance class separability among architectural configurations. The method further integrates multi-objective Non-dominated Sorting Genetic Algorithm II (NSGA-II) optimization to balance modularity, scalability, and complexity distribution across system modules. The method is validated using a gas turbine engine, where multiple architectural configurations were analyzed. Experimental result demonstrates an optimization cost of 0.65, Structural Complexity Index (SCI) of 0.60, Convergence Iterations of 350, Execution Time of 10.2s, and a Scalability Score of 0.89, which is implemented in Python. Furthermore, analysis reveals a non-linear relationship between structural complexity and development cost, emphasizing the need for optimal complexity allocation during early design stages. The proposed method provides a systematic foundation for designing scalable, modular, and cost-efficient complex systems, making it applicable to aerospace, manufacturing, and large-scale engineered infrastructures.

Keywords: Complex System Engineering, Scalable Architectures, Structural Complexity, Modularity Optimization, Dependency Network Modeling, Multi-Objective Optimization.

1. Introduction

There exist uses for gas turbines as well as large energy systems in aerospace, energy generation, and sustainable processes by means of efficient simulation and thermodynamics of cycles [1]. There are certain problems associated with the use of the traditional method that includes the application of physical and simulation models due to several uncertainties such as flow distortion, seasonality, manufacturing tolerances, and disturbances [1]. These are some of the problems associated with the scalability of cloud computing systems where there exists high variance of demands [2]. Similar challenges are posed by solar thermal energy generation facilities due to supply-demand mismatch, while thermoelectric and hydroelectric energy plants are faced with constraints from both economical and environmental aspects [3]. Artificial intelligence, machine learning, and deep learning technologies have been increasingly used in energy and propulsion systems. As far as gas turbine systems are concerned, machine learning assists in adapting parameters to make accurate predictions, while neural networks and convolutional neural networks assist in identifying the system and modeling its dynamics [4]. In the case of power systems, machine learning makes important contributions in predicting failures, diagnosing faults, and optimizing power through digital twins and real-time models [5]. The developments made in machine learning include graph neural networks and knowledge graphs for enhancing the forecasting and energy management under uncertain conditions in distributed energy systems [6]. Lastly, there is the use of intelligent technology in predicting combustor acoustic behavior and designing hydrogen turbines through additive manufacturing and AI optimizations. On the other hand, notwithstanding all the above innovations, a lot of problems still persist when using AI-based methods in energy systems. For instance, one of the issues that exist is that conventional approaches for adapting performance are expensive and inefficient, thereby making it difficult to use them in actual cases [7]. Furthermore, very few studies have been carried out on how to apply different machine learning approaches, particularly in terms of combustion acoustics and turbofan adaptation [8]. Furthermore, the current systems employing AI in distributed energy networks and power plants face difficulties with data collection, operational stability, and the lack of comprehensive AI platforms.

Research aim: To design a systematic complex system engineering method that would enable modeling and optimization of scalable system architectures using dependency networks to represent product structures and measures complexity based on network analysis, as well as the multi-objective NSGA-II to determine the optimal architectural designs.

Research organization: Research is structured in a certain way. In Section 1, the basics of complex system engineering are discussed, with an emphasis on the issue of scalable design that is important for complex engineered systems due to dependency networks and structural complexities. In Section 2, the literature related to the modeling of system architecture, as well as the analysis based on network complexity, is discussed. The methodology proposed within this research paper is introduced in Section 3, which discusses the ways in which the system is represented, using data preprocessing, feature selection, and modular decomposition of system architectures. Section 4 provides details about optimization, with emphasis put on the results achieved using the NSGA-II in relation to the gas turbine engine. Section 5 discusses the key findings and future research.

2. Related work

Table 1 presents an overview of the latest research on the topic of complex systems modeling, optimization, and intelligent engineering designs. Almost all works have emphasized the use of advanced modeling methods, evolutionary approaches, graph-based learning, and hybrid optimization approaches to enhance the performance of complex systems. Nonetheless, scalability problems, low generalizability, high computational complexity, and a lack of application in real-time scenarios are major weaknesses of most of these approaches.

Table 1: Recent research on complex systems engineering for scalable architectures

Ref	Objective	Technique	Result	Limitation / Future Work
[9]	To improve the complexity metric using Systems Modeling Language + Design Structure Matrix	SysML modeling, modified Design Structure Matrix (DSM), multi-level decomposition	Better complexity quantification than DSM; similar trends	Absolute value mismatch; needs validation
[10]	To improve robustness under attacks/failures	Multifactorial Evolutionary Algorithm with Cross-task Resource-allocation (MFEA-CR) evolutionary multitasking	Better robustness than baseline methods	Limited generalization to broader failure models

[11]	To enhance community detection	Contrastive Evolutionary Community Clustering Network (CECC-Net), feature-based evolutionary clustering	Improved clustering performance	Scalability and diversity issues remain
[12]	Improve fault detection & interpretability	Hierarchical Graph Convolutional Attention Network (Graph Convolutional Network + attention + causal discovery)	Higher accuracy + interpretability	Poor generalization across domains
[13]	Build a scalable gas turbine ML model	ML regression on simulator data	High accuracy (<0.1% error)	Limited to gas turbines only
[14]	To improve the co-simulation of systems	Cognitive Twin + Functional Mock-up Interface Version 2.0 (FMI 2.0) + ontology	Better interoperability & integration	Needs more scalability & automation
[15]	Optimize thermal power plant operations	Artificial Neural Network + constrained Multi-Ant Colony Optimization for Continuous Domains (MAD-OPT) optimization	>95% accuracy, efficiency gain	Not fully real-time or deployable
[16]	Solve complex, constrained, non-smooth optimization problems under No-Free-Lunch constraints.	Hybrid metaheuristic: success-history intelligent optimizer + Differential Evolution + Gaussian Transformation	Consistently high performance on 2022 IEEE Congress on Evolutionary Computation benchmarks and engineering problems with strong stability.	Computational cost and need for further testing on larger-scale real-world problems

Research gap: Optimization techniques for complex engineered systems often fail to address conflicting objectives like modularity, scalability, and structural complexity. Current methods typically rely on single-objective formulations or heuristics that ignore the interdependence of network components and their effects on scalability. Additionally, they struggle with global design space analysis and trade-off optimization. The proposed technique utilizes adjacency matrices to depict system architecture and develops a structural complexity indicator, for employing NSGA-II to enhance trade-off optimization in the design space.

3. Methodology

To model system architectures, the suggested approach is represented through dependency networks by employing adjacency matrices. Interaction data is first normalized through Min-Max normalization, after which features are extracted through LDA. Then, both the structural complexity and modular decomposition were calculated to provide values for the properties of the system architecture. To optimize the parameters of scalability, modularity, and structural complexity, a multi-objective optimization algorithm such as NSGA-II was employed, in which optimal dominance and crowding distance were utilized as seen in Figure 1.

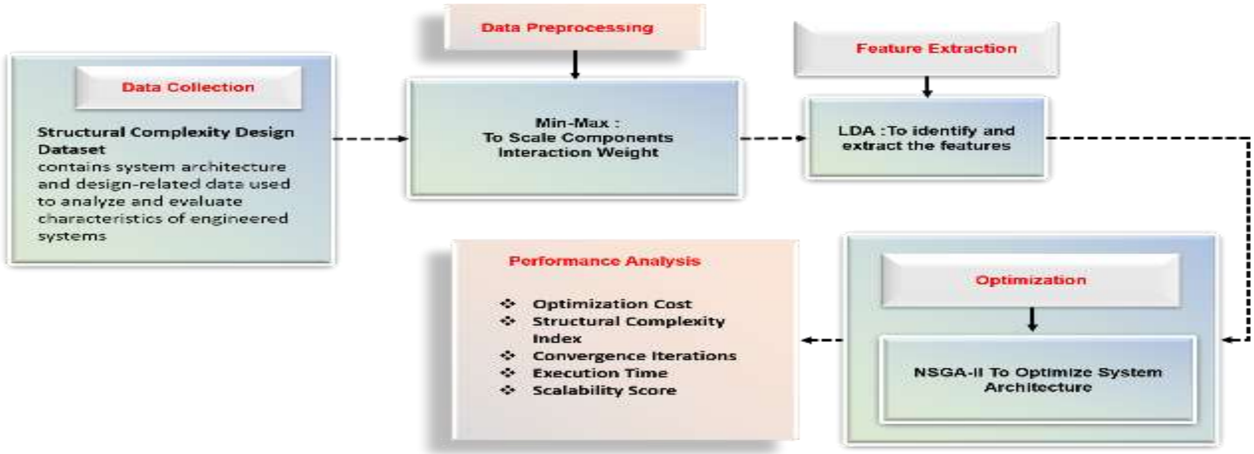


Figure 1: Methodology of the proposed model

3.1 Dataset description

The data used in this research was sourced from Kaggle platform (https://www.kaggle.com/datasets/colabsss/structural-complexity-design-dataset). This is a complex system engineering dataset containing 13,000 records and 33 columns related to scalable architecture design, dependency network behavior, modular structure, interaction strength, cost variation, scalability, flexibility, maintainability, and architecture suitability. Each record represents an architecture-level system design profile capturing measurable structural, operational, cost, and suitability-related attributes of engineered systems. This dataset supports analysis of structural complexity, modular architecture planning, scalable system design, architecture comparison, cost-aware design evaluation, and complex engineered system assessment. The dataset is suitable for studying architecture optimization and decision-making in complex engineered systems and is divided into 70% for training and 30% for testing.

3.2 Min-max normalization for feature scaling

Min–Max Normalization is a data preprocessing technique used to rescale input interaction and structural data into a fixed range (typically 0 to 1) while preserving the relative relationships between values. In this way, the contributions of all characteristics are equal during analysis because those with higher magnitudes will not have a disproportionate impact on the analysis of the system. The goal of normalization is the normalization of interactions between elements in the system design model in order to facilitate an even scaling of all relationships for subsequent analysis.

$$Y_{normalized} = \frac{Y - Y_{min}}{Y_{max} - Y_{min}} \quad (1)$$

Equation (1) represents the min-max normalization equation, where $Y_{normalized}$ is the scaled value of the original feature after transformation. Y is the original input feature value, Y_{max} and Y_{min} are the maximum and minimum values of the features.

3.3 Dimensionality reduction using LDA

LDA is a supervised method for reducing data dimensions while maintaining separation among different classes. To extract the most relevant structural features from system architecture data that enhance class separability among different configurations, enabling more effective evaluation and optimization of modularity, scalability, and structural complexity.

$$\mu_i = 1/m_i \sum_{e_{ji} g_i} e_j \quad (2)$$

$$T_A = \sum_{j=1}^{d} m_j(\mu_j - \mu)(\mu_j - \mu)S \quad (3)$$

$$T_x = \sum_{i=1}^{d} \sum_{j=1}^{m_i} (e_{ji} - \mu_i)(e_{ji} - \mu_i)S \quad (4)$$

$$C = {T_X}^{-1} T_A \quad (5)$$

Equation (2) represents the mean of the elements e_{ji} in the ith group g_i, where m_i is the number of elements in that group and $\sum_{e_{ji} g_i} e_j$ represents the sum of all elements e_j that belong to group i. In equation (3), T_A represents the between-group scatter matrix, measuring how each group's mean deviates from the overall mean, where d is the number of groups, S is the scatter matrix, and m_j is the number of samples in the j-th group, μ is the overall mean vector. Equation (4) ${T_X}^{-1}$ defines the within-group scatter matrix, capturing the variance of elements e_{ji} within each group relative to their group mean μ_i. Equation (5), C represents the value used in LDA to maximize class separability.

3.4 NSGA II for Modularity and Scalability

is a multi-objective evolutionary optimization method used in this paper to search for optimal system architecture configurations by iteratively improving a population of candidate solutions. It works by evaluating multiple competing design objectives and guiding the search toward better trade-offs through selection, crossover, and mutation mechanisms while maintaining solution diversity. To optimize complex system architectures by simultaneously improving modularity and scalability, while minimizing structural complexity and ensuring balanced complexity distribution across system modules for efficient and scalable system design.

$$c_d = \sum_{i=1}^{k} \frac{f_j^{i+1} - f_j^{i-1}}{f_j^{max} - f_j^{min}} \tag{6}$$

In Equation (6), c_d represents the crowding distance of the i solution, which is used in NSGA-II to maintain diversity among solutions in the optimal front for multi-objective optimization. Here, (i) denotes the index of the current solution, (j) is the index of the objective function, and (k) is the total count of objectives considered. f_j^{i+1} and f_j^{i-1} represent the values of the j objective function for the next and previous neighboring solutions, respectively, after sorting. f_j^{max} and f_j^{min} are the maximum and minimum values of the j objective among all solutions, used for normalization.

Algorithm 1: Complex System Engineering

```
INPUT:
  D = system data, Y = features, N = population, MaxGen
OUTPUT:
  P * = optimal Pareto architecture set
STEP 1: MIN − MAX NORMALIZATION
Y_norm = (Y − Y_min) / (Y_max − Y_min)
IF Y_max == Y_min THEN STOP
STEP 2: LDA FEATURE EXTRACTION
μi = 1/mi Σ_{eji gi} ej
IF determinant(T_x) == 0 THEN
  APPLY regularization or remove redundant features
ELSE
  continue
END IF
STEP 3: NSGA − II OPTIMIZATION
Initialize population P
FOR gen = 1 TO MaxGen DO
  Evaluate objectives:
    f = {modularity, scalability, complexity}
 Pareto sorting + selection
  IF solution A dominates B THEN
    keep A
  ELSE
    compare crowding distance
  END IF
  Apply crossover + mutation → Q
  P = best(P ∪ Q)
END FOR
Return P * (Pareto − optimal solutions)
END
```

The proposed Algorithm 1 consists of three stages: preprocessing, feature extraction, and multi-objective optimization for system architecture design. Min–max normalization is first applied to scale input features uniformly and remove bias due to differing magnitudes. LDA is then used to project data into a lower-dimensional space by maximizing class separability using class-wise statistics. Finally, NSGA-II optimizes system architectures by evolving solutions based on modularity, scalability, and structural complexity. optimal dominance and crowding distance guide selection, produce a diverse set of optimal trade-off architectures.

4. Result and discussion

The experimental analysis for the suggested Complex System Engineering approach for scalable and efficient design of system architecture is discussed in this section. For this purpose, the comparison of the proposed NSGA-II-based approach and other hybrid optimization methods like Success-History Intelligent Optimizer Differential Evolution Gaussian Transformation (SHIODEG) [16] is made based on measures such as complexity reduction, modularity improvement, and scalability increase. All the experiments are done using the Python programming language, based on the same benchmark set.

4.1 Multi-Dimensional Analysis of Structural Architectural Suitability in Large-Scale Systems

As seen in Figure 2 below, there is an interconnection between Modularity Score (0.2 - 1.0) and Scalability Score (0.0 - 1.0), with Structural Complexity Scores being superimposed on the graph based on color coding (20-90). The data set shows a cluster of data points spread evenly throughout the whole modularity-scalability score range, thus indicating that the design of systems cannot be limited to a particular range but encompasses large varieties of architectural options. High structural complexity scores (around 80-90) are commonly found in medium to high modularity ranges (0.6 - 0.9), hence indicating that high modularity does not equal low complexity.

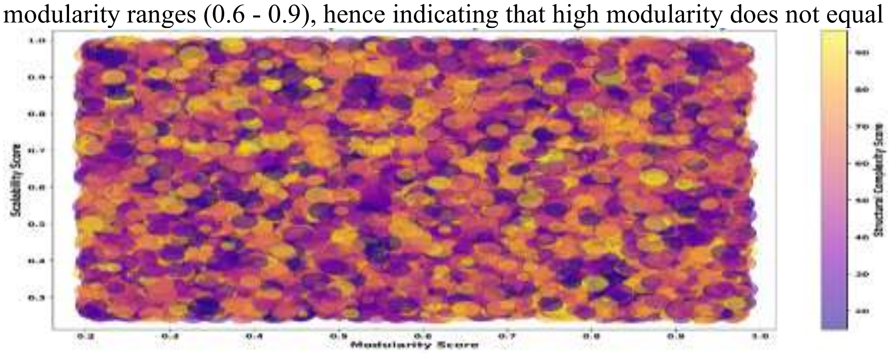


Figure 2: Joint distribution of modularity, scalability, and structural complexity

Figure 3 shows development cost (USD million, roughly between 0 and 300) for these systems, using the same categories of suitability via density distribution method. Highly suitable systems tend to fall within high costs, ranging from roughly 150 to 250 million USD. Meanwhile, low and moderately suitable systems exhibit broader cost distributions along the entire range.

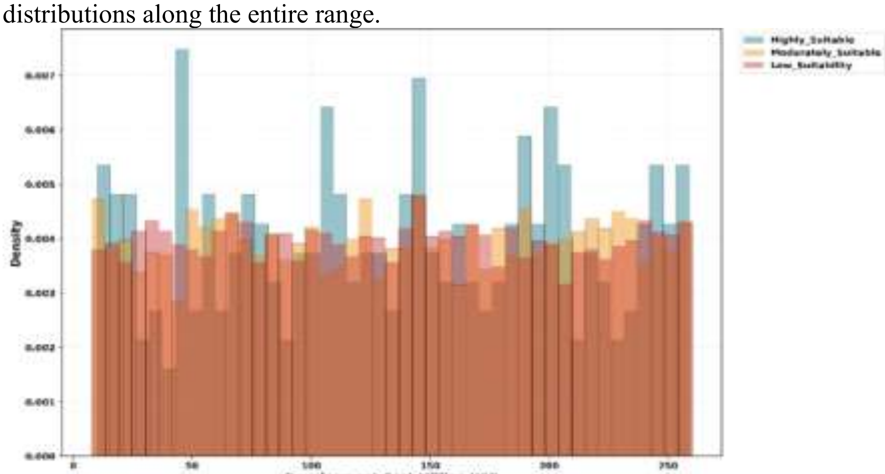
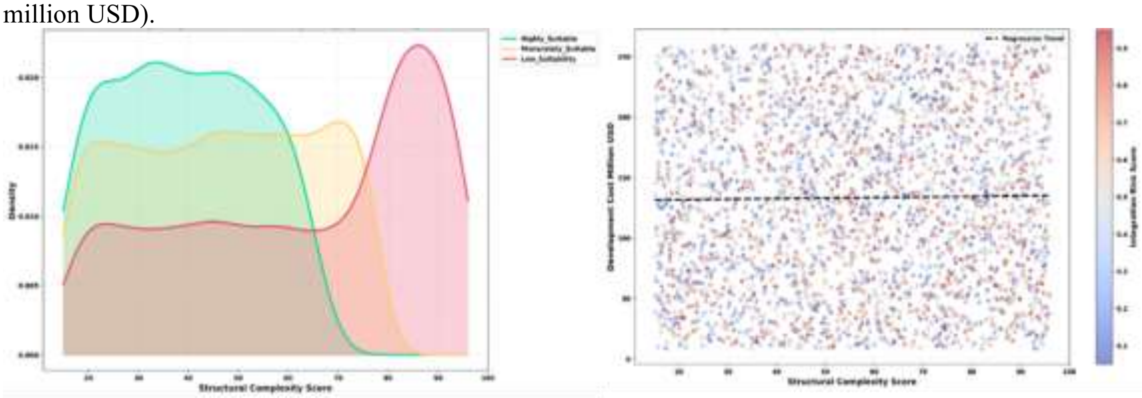


Figure 3: Development cost variation across suitability classes.

The graph in figure 4(a) below highlights the distribution of the Structural Complexity Score (0–100) among systems in the three suitability categories. It is evident from the graph that low suitability systems have a tendency of having high complexity (70–90), hence overloading of architecture, while those of moderate suitability have moderate complexity (40–70) due to transitional efficiency. The high suitability systems tend to cluster around lower scores of complexities (20–50). Thus, improved design leads to reduced structural complexity and efficiency in organization. Figure 4(b) below represents the correlation between the Structural Complexity Score and Development Cost (0-300 million USD).



(a) (b)

Figure 4: Analysis of (a) Density distribution of structural complexity across suitability levels; (b) correlation analysis between complexity and development cost.

4.2 Metrics explanation

Optimization Cost: Refers to the computing effort required to arrive at the optimal solution through evaluation of function calls and algorithmic complexity in the optimization process.
Structural Complexity: It is a metric for determining the degree of dependence and interconnections between system components.
Convergence Iterations: Refers to the number of generations or iterations necessary for the optimization algorithm to achieve optimality or stability.
Execution Time (s): Refers to the total time needed for the optimization algorithm to execute its optimization operation.
Scalability Score: Determines how efficient the system architecture becomes in handling system growth.

4.3 Performance evaluation

Both the proposed and the existing models are trained on the structural complexity design dataset. Table 2 and Figure 5 present a comparison between the new method developed based on NSGA-II and the current SHIODEG [16] method for optimization in complex systems based on different criteria. It is seen from the table that the NSGA-II algorithm demonstrates superiority over the SHIODEG method based on such performance measures as optimization cost (0.65 vs. 0.78), structure complexity (0.60 vs. 0.74), convergence rate (350 vs. 420 iterations), execution time (10.2 s vs. 12.8 s), and scalability (0.89 vs. 0.81).

Table 2: Comparison of existing and proposed models trained on the structural complexity design dataset

Metrics	Optimization Cost	Structural Complexity Index	Convergence Iterations	Execution Time (s)	Scalability Score
SHIODEG [16]	0.78	0.74	420	12.8	0.81
NSGA II [proposed]	0.65	0.60	350	10.2	0.89

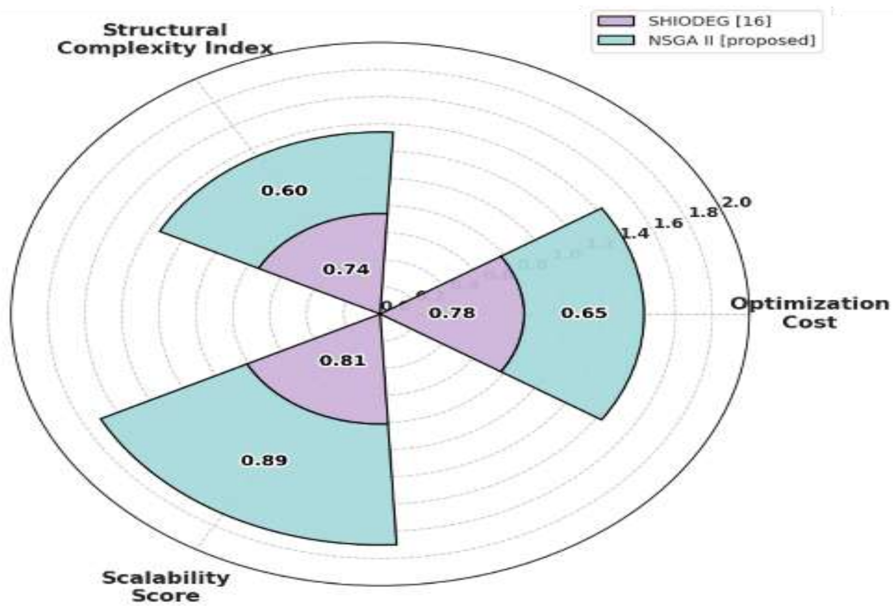


Figure 5: Representation of the existing and proposed models trained on the structural complexity design dataset

4.4. Discussion

Current approaches such as SHIODEG [16] for complex system architecture design are limited by reliance on single-objective optimization, inadequate handling of strong inter-component dependencies, and the lack of an integrated method for representing and analyzing system structure. Also, they are unable to satisfactorily handle the trade-off between modularity, scalability, and complexity in engineering systems. The method proposed here handles the above shortcomings through the use of a dependency graph to represent the system and the use of consistent measures of evaluating interactions. Structural complexity measure is used to determine the behavior of the system. An optimal solution is achieved by using NSGA-II where the trade-off between modularity, scalability, and complexity is handled. This leads to improved design efficiency and scalability than those achieved using other methods.

5. Conclusion

Research has proposed a Complex System Engineering approach for designing an effective system architecture through modeling of product structure as dependency networks with the incorporation of Min-Max normalization, LDA feature extraction, and NSGA-II for multi-objective optimization. The proposed model has successfully captured complexity, improved modularity, and scalability by developing an optimal architectural solution. Experimental result demonstrates an optimization cost of 0.65, Structural Complexity Index of 0.60, Convergence Iterations of 350, Execution Time of 10.2s, and a Scalability Score of 0.89, which is implemented in Python. Nonetheless, there exist some limitations within the proposed model, such as network dependency and computation costs incurred during optimization of evolutionary algorithms in very large-scale scenarios. Besides, the model only applies the validation aspect to the gas turbine engine example. In the future, further investigation would consider the application of adaptive optimization, the application of DL feature extraction, and scalability across various scenarios.

References

1. Kim, S., 2025. Prediction-focused machine learning for performance adaptation of aero gas turbines through steady-state and transient simulation. Applied Thermal Engineering, 267, p.125732. https://doi.org/10.1016/j.applthermaleng.2025.125732
2. Irshad, R.R., Hussain, S., Hussain, I., Nasir, J.A., Zeb, A., Alalayah, K.M., Alattab, A.A., Yousif, A. and Alwayle, I.M., 2023. IoT-enabled secure and scalable cloud architecture for multi-user systems: A hybrid post-quantum cryptographic and blockchain-based approach toward a trustworthy cloud computing. IEEE Access, 11, pp.105479-105498.https://doi.org/10.1109/ACCESS.2023.3318755
3. Ashraf, W.M., Alshehri, A.S., Bin Jumah, A., Uddin, G.M., Akhtar, M. and Dua, V., 2025. Domain-enforced and operator-in-the-loop neural simulation platform for techno-enviro-economic performance enhancement of gas

turbine system. Computers & Chemical Engineering, 206 ,p.109529.https://doi.org/10.1016/j.compchemeng.2025.109529

4. Alsarayreh, M., Mohamed, O. and Matar, M., 2022. Modeling a practical dual-fuel gas turbine power generation system using dynamic neural network and deep learning. Sustainability, 14(2), p.870. https://doi.org/10.3390/su14020870
5. Boretti, A. and Huang, A., 2024. AI-driven DfAM of aeronautical hydrogen gas turbine combustors. International Journal of Hydrogen Energy, 77, pp.851-862. https://doi.org/10.1016/j.ijhydene.2024.05.437
6. Wang, Y., Wu, K., Ge, L., Li, H., Ortega-Fernández, I. and Bielsa, D., 2026. An adaptive, standalone, and scalable solar combined heat and power system with AI-enhanced universal energy management. Applied Thermal Engineering, 289, p.129725. https://doi.org/10.1016/j.applthermaleng.2026.129725
7. Deon, B., Cotta, K.P., Silva, R.F.V., Batista, C.B., Justino, G.T., Freitas, G.C., Cordeiro, A.M., Barbosa, A.S., Loucao Jr, F.L., Simioni, T. and Morais, A.M., 2022. Digital twin and machine learning for decision support in thermal power plant with combustion engines. Knowledge-Based Systems, 253, p.109578.https://doi.org/10.1016/j.knosys.2022.109578
8. Shahjalal, A., Kalawsky, R., Joannou, D., and Ozdemir, F., 2025. Benchmarking machine learning models against full-scale acoustic absorption measurements in gas turbine liners. Fuel Communications, 26, p.100152. https://doi.org/10.1016/j.jfueco.2025.100152
9. Lopez, V.E.P. and Thomas, L.D., 2022. Metric for Structural Complexity Assessment of Space Systems Modeled Using the System Modeling Language. Aerospace, 9(10), p.612. https://doi.org/10.3390/aerospace9100612
10. Wang, S., Jin, Y., and Cai, M., 2023. Enhancing the robustness of networks against multiple damage models using a multifactorial evolutionary algorithm. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 53(7), pp.4176-4188.https://doi.org/10.1109/TSMC.2023.3241621
11. Fu, J. and Wang, Y., 2024. An improved continuous-encoding-based multi-objective evolutionary algorithm for community detection in complex networks. IEEE Transactions on Artificial Intelligence, 5(11), pp.5815-5827. https://doi.org/10.1109/TAI.2024.3442153
12. Zheng, S., Wang, C., Zio, E., and Liu, J., 2024. Fault detection in complex mechatronic systems by a hierarchical graph convolution attention network based on causal paths. Reliability Engineering & System Safety, 243, p.109872. https://doi.org/10.1016/j.ress.2023.109872
13. Pati, S., Osorio, J.D., Panwar, M., and Hovsapian, R., 2025. Scalability analysis of heavy-duty gas turbines using data-driven machine learning. Next Energy, 7, p.100275. https://doi.org/10.1016/j.nxener.2025.100275
14. Li, Y., Chen, J., Hu, Z., Zhang, H., Lu, J. and Kiritsis, D., 2022. Co-simulation of complex engineered systems enabled by a cognitive twin architecture. International Journal of Production Research, 60(24), pp.7588-7609. https://doi.org/10.1080/00207543.2021.1971318
15. Ashraf, W.M., Keshavarzzadeh, A.H., Alshehri, A.S., bin Jumah, A., Debnath, R., and Dua, V., 2026. Domain-informed operation excellence of the gas turbine system with machine learning. Energy Conversion and Management, 348, p.120789. https://doi.org/10.1016/j.enconman.2025.120789
16. Alawadi, S., Fakhouri, H.N., Alkhabbas, F., Kebande, V.R., Awaysheh, F.M., and Cheddad, A., 2026. SHIODEG: a hybrid success-history intelligent optimization algorithm for engineering design problems: S. Alawadi et al. The Journal of Supercomputing, 82(5), p.282. https://doi.org/10.1007/s11227-026-08398-5