Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 47 (2026), No. 7, pp. 2667–2680
DOI : 10.47974/JIOS-2322

An optimization model for software design process using soft computing techniques

Sufia Nadeem Chishti *
Ajeet Singh †
School of Computer Science and Engineering
Galgotias University
Greater Noida 203201
Uttar Pradesh
India

Abstract

There is an urge of a scheme to provide an optimization program that calculates optimal values of parameters being used in software design process. This paper is beneficial in AOP (Aspect Oriented Programming) model for software design. It is formulated to achieve optimal values of the selected parameters. The scheme involves code size and the number of inherited classes. Thus, a general model for accessing software design system is being framed by considering these significant parameters. The three significant parameters comprising of LOC (Line of codes), Number of modules and level of Inheritance are used in the model. The basic ALHN (Augmented Lagrange Hopfield Network) algorithm is modified and a supplementary Fuzzy system that calculates the value of an index (represents the parameters) are considered in the proposed algorithm. The purpose of this research work is to approach a flair that facilitate developers in edifice software design practice. To compute optimal values of the parameters ALHN algorithm along with a risk term which is calculated by using Fuzzy logic is presented. Machine Learning is the logical investigation of calculations that perform explicit assignment without being expressly modified, ML is applied to achieve optimal model.

Subject Classification: *90C26, 90C70, 68N30.*

Keywords: *Aspect oriented design, Optimization, ALHN, Fuzzy sets, Machine learning.*

---

* *E-mail:* **er.snadeem@gmail.com (Corresponding Author)**
† *E-mail:* **ajeetsinghiet@gmail.com**

1. Introduction

Over the time, software products are growing more complicated, necessitating the development of a new method that can manage the complexity more effectively. The increasing complexity of software systems over time motivates engineers to employ new technologies in software development. The only example that provides appropriate generalizations for modularizing the software concerns is Aspect-Oriented Software Development (AOSD). This is a fortunate paradigm that promotes responsible fragmentation leading to easy maintenance and reusability of the software system. Numerous techniques for facilitating the software design methods that are based upon parametric centric design are in use for around two decades. Design based on parameters selection can be useful in enhancing the modular design, which in turn leads to a reduction in maintenance costs. The old software system was constructed using procedural language which ended up being too large and complicated. The evolution of software engineering in the past inspired the researchers to improve the software design issues. The researchers make use of the software design issues' improvement, which is inspired by the evolution of software engineering in the past. Additionally, the researchers have used the object-oriented system software design process, which is easier to read and write and has a better encapsulation. However, it is not perfectly possible to reuse the implementation in Object-Oriented Programming (OOP) because of some properties of the software like logging, security check, transaction etc. intertwine several components. Crosscutting concerns lead to entangled code which is hard to code and maintain [1]. An important feature of OOP is modularization; the concept does not address the modularization problem which is an unsatisfactory future of programming paradigm.

Our Contribution: In this paper we have introduced a new scheme that can be useful to a designer to compute optimal values during the software development process. The risk term is computed with the help of Fuzzy sets and applied in the model, getting optimal levels of attributes associated with selected parameters. Machine learning (ML) is employed to train machines to transfer the data more accurately. After screening the data a few times, we are not able to extract the extorted information from the data. Machine Learning is the sensible investigation of calculations and factual models that PC frameworks utilize to perform a specific task without being explicitly altered.

The following components make up this paper: Section 2 describes the suggested model that uses the ALHN algorithm. This includes a way to find the optimal values of the parameters and select the appropriate coefficients of the model's cost function. Section 3 describes the Fuzzy sets that are used for calculating the risk term. Section 4, Discussion that gives description of Fuzzy Sets, Machine learning algorithm for classifying the sample data obtained from ALHN algorithm. Section 5, the results and the interpretation of the proposed model are provided. Section 6, Includes conclusion and future work.

2. Literature Review

The research area and software architecture and design have now become an important activity of development, and the research area is growing rapidly. Over the past few decades, techniques for optimizing software architecture have grown in numbers, with the goal of automatically finding an optimal software architecture, based on a specified set of quality attributes. However, the reported outcomes are spread across a number of research articles, multiple systems, domains, multiple quality attributes and system.

The improvement of software engineering techniques has been developed in the direction of separation of concern which leads to improvement of the modularity and aspect orientation in the software development cycle. Aspect Oriented Programming (AOP) is a concept and as such, it is not bound to a certain programming language. It can be used to overcome the limitations of any language with single, hierarchical decomposition. AOP can be a solution to the issues of design, quality, and maintenance. Aspect Oriented Software Design (AOSD) is a new paradigm of software construction, which inherits and brings the current models.

Martin et al. [1] explored that more attention should be given to the approaches related to 'aspect-oriented' software model designing. In this work, designing and optimizing the model for software design process has been proposed which is based on aspect oriented architectural design by using ALHN (Augmented Lagrange Hopfield Network) Algorithm, Fuzzy sets and Machine learning.

Tramontana [2] reveals the extensions to UML for AOD extend to the design to allow modularization and composition of cross cutting and non-cross cutting concern. A non – parametric application can be seen which has integrated with software design requirement. Galester, Weyns, and Goedicke [3] made the efforts to apply optimized feature extraction techniques to realize the structure software design using approach based on use codes.

Iqbal and Allen [4] concluded the use of target-based approach. The method involves modules resurge, data mining and aggregation of the necessary information. One of them is the machine learning realizing. Recently, machine learning algorithms were used for classification applications in variety of models and machine learning techniques. Rasool and Arshad [5] discussed the review work on machine learning plays vital role in understanding of algorithms as well. Machine learning algorithms are accommodating bridging this breach of perceptive and methods. Khomh and Gueheneuc [6] presented the topic of role of machine learning predictions and applications.

Gautam, Khunteta, and Ghosh [7] discussed recent advances in software engineering have demonstrated the effectiveness of adaptive optimization frameworks that employ reinforcement learning to dynamically adjust optimization strategies based on performance feedback. Such adaptive learning mechanisms improve prediction accuracy, robustness, and scalability, indicating that intelligent optimization techniques can effectively support complex software engineering tasks and can be extended to software design optimization problems.

Approaches given by Dey [8] need reinvestigation and there is also an urge for effective and automated tools to support their approach. Kotagiri et al. [9] proposed aspect-oriented feature extraction and sentiment analysis by using deep learning models. Advance sentimental analysis work was implemented using extreme gradient boosting algorithm. Purbey and Lakhwani [10] describes optimized multilayer deep BiLSTM classifier and was applied by using aspect based sentimental analysis for product and reviews which contain multiple aspects and sentiments.

Sharma and Lakra [11] proposed hybrid metaheuristic optimization techniques have recently attracted considerable attention for solving complex optimization problems by integrating evolutionary and nature-inspired search strategies. These intelligent optimization models improve solution quality, computational efficiency, and resource utilization, thereby providing a strong foundation for designing optimization frameworks applicable to software engineering and software design processes.

The optimization model for software design process is defined in context of the improvement of the performance from the start to the end of software design process. Optimization model is based on an underlying mechanism of three components. Firstly, the function that needs to be optimized. Secondly, the set of decision variables and last the set of rules.

Constraints that are used to limit the decision variables. In the paper a new scheme is presented for an optimization utility during the software design process, described in terms of parameters. ALHN algorithm is used to calculate the parameters using a risk term calculated by a Fuzzy rule. This scheme can be helpful in designing the modular approach for designing the software designing process using parameter. In his findings the relationship between the heights of parents and their sons was main concern. He was looking in particular for a relationship between the height of parents and children. The outcome of the study was that height of a child is approximately same as his parents. Another finding by the study was that child's height is roughly the average of the height of all people. This outcome was termed as regression where the height of a child is approximately closer to the average height. In the regression theory, we attempted to determine a regression line that is near each Pixel in the graph. For instance, classic linear equation or 'Least space method' we only calculate the closer attributes in upper and down directions.

3. Methodology

The problem of software cost estimation is still a complex problem in the project management and is hard to solve because of the uncertainty of the problem and the factors that are dependent on each other. This work suggests a complete optimization model by incorporating deterministic cost models with fuzzy logic based risk assessment to yield more reliable

Figure 1

Software cost estimation framework

cost predictions. The three-stage optimization framework in the following stages have been applied. In order to opt the methodology for software cost estimation, the diagram below is to be considered as shown in Figure 1. The steps are cost equation mapping, ALHN optimization, Fuzzy Risk assessment, Iterative optimization loop and result evaluation.

1. **Cost Equation Mapping:** Software design attributes are associated with exponential cost functions.
2. **ALHN Optimization:** Parameter values are optimized using modified ALHN algorithm with risk integration.
3. **Fuzzy Risk Assessment:** Uncertainties are quantified using fuzzy logic principles.

The following specifications were applied.

a) Parameter Selection

Three significant design parameters were identified:

m_1: Code size (lines of code), m_2: Number of modules, m_3: Number of inherited classes.

b) Cost Function

Each parameter follows an exponential cost model:

$$\text{Cost}(m_i) = a_i \times e\text{^}(b_i \times m_i) + c_i \tag{1}$$

c) Optimization Objective

$$\text{Minimize: Total Cost} = \Sigma\ \text{Cost}(m_i) + \text{Risk Term} \tag{2}$$

$$\text{Subject to: } |\text{Total Cost - Desired Value}| \rightarrow \min \tag{3}$$

d) ALHN Constraints

$$\Sigma\ (m_i \times \text{weight}_i) \leq \text{Resource Limit} \tag{4}$$

e) Fuzzy Set Classification

Parameters are categorized using fuzzy indices:

Large set: 0.6-0.9 Medium set: 0.58 Small set: 0.20

Code Size (m_1):

Baseline: 500 LOC/module, 5 modules average, $a_1 \approx 107.9$, $b_1 = 0.58 \times 100 = 58$

$$c_1 = 0.20/100 = 0.002,$$

Classification: Small (≤50), Medium (50-200), Large (>200)

Module Count (m_2):

Proportional weight: 0.29, $a_2 \approx 108.5$ and b_2 optimized for medium/small range.

Inherited Classes (m_3):

Scaling factor: k_2 = inherited classes × 20 and Weight: 0.0051, $a_3 \approx 109.4$, $b_3 = 16$.

4. Optimization Problem Formulation

Objective: Minimize overall cost.

Constraints: Equality: Desired value matching and Inequality bounds: $10 \leq m_1 \leq 200$, $40.5 \leq m_2 \leq 300$, $35 \leq m_3 \leq 200$

Risk Management: Error tolerance incorporated through fuzzy logic to accommodate calculation uncertainties and design process inaccuracies.

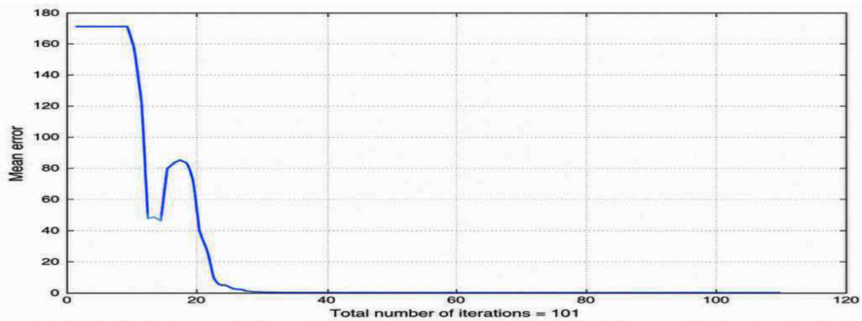

Figure 2

The Output of ALHN algorithm in terms of number iterations and error

Results: For 101 iterations as shown in Figure 2, the optimization process was performed. It started with a suboptimal solution, with a maximum error of around 175. The error seems to have converged quickly, as there was a substantial drop within the first 20 iterations. The learning curve shows a small increase in the error curve around the iteration 17; this may be due to the exploration phase of the ALHN algorithm or adaptation based on the changes caused by the fuzzy logic. After that, the model has been converged and the error was almost at near zero level when it is getting to be 25th iteration and stayed stable in all iterations from 25th to 60th, which proves the effectiveness and stability of the proposed optimization scheme.

5. ALHN Algorithm with Fuzzy Logic Implementation Summary

Algorithm Overview

The proposed ALHN (Adaptive Learning Harmony Network) algorithm integrates fuzzy logic for software design cost optimization with risk management. The algorithm operates through a multi-step process combining parameter optimization with uncertainty assessment.

ALHN Algorithm Parameters

Initial Configuration

- **Parameters:** N = 3 (m_1, m_2, m_3) and **Desired Output:** 6000
  - **Risk Factor:** 0.5 and **Learning Parameters:** $\sigma = 1000$, $\alpha_1 = 0.01$, $\alpha_2 = 0.004$, $\beta = 0.001$
  - **Convergence Settings:** Max iterations = 1000, Accuracy = 0.0001

Algorithm Steps

1 **Initialization:** Set parameters and variables

2 Initial Value Calculation:

m_i = (desired_output × m_i_max) / Σ (parameters)

$\mu_i = \log[(m_i - m_{i}\_min)/(m_{i}\_max - m_i)] / (2\sigma)$

1 **Iterative Optimization:**

- Calculate derivatives: $de\_dm_i = b_i + 2c_i \times m_i$
- Update parameters using adaptive learning rates
- Apply risk factor: $\mu_i = \mu_i(1\text{-risk}) - \alpha_1 \times de\_dm_i$

Fuzzy Logic Integration

Fuzzy Rule Base: Five key rules govern uncertainty assessment:

- **Rule 1-3:** Small parameter combinations → Low uncertainty.
- **Rule 4:** Medium-Large-Large → High uncertainty.
- **Rule 5:** All Large parameters → Very high uncertainty.

Figure 3 shows Fuzzy Logic Designer interface with outputs. Description of fuzzy design interface is given below.

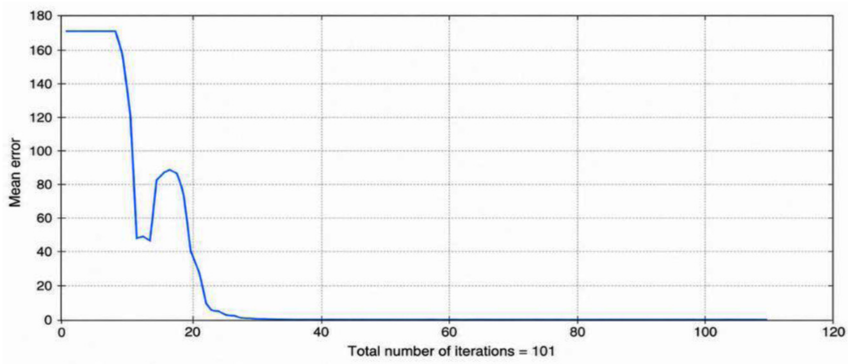

Figure 3

The Output of ALHN algorithm in terms of number iterations and total cost

Main Components:

Left side (Yellow boxes): Three input variables (M1, M2, M3) representing your software design parameters.

Center (White box): Fuzzy inference system processing unit.

Right side (Blue box): Output variable (Uncertainty).

Interface Elements:

FIS Variables panel: Lists input/output variables with their properties.

Current Variable panel: Shows details for selected variables including:

Name field, Type selection, Range settings

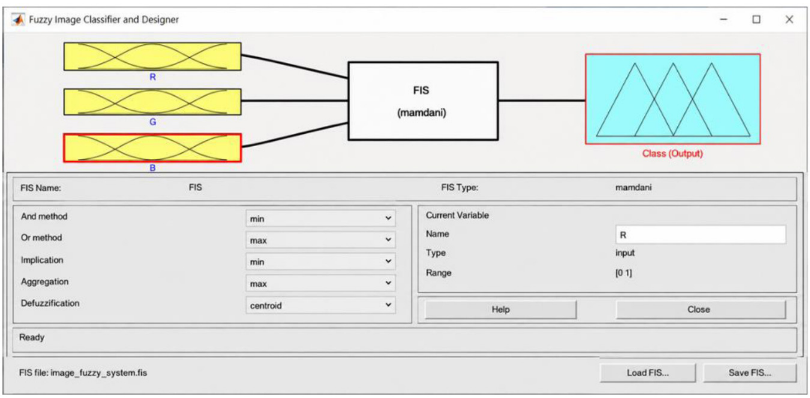

Figure 4

The outputs obtained for the fuzzy system Interface Layout

Rule Viewer interface from MATLAB's Fuzzy Logic Toolbox is shown in Figure 4 above.

Input Variables (Columns 1-3):

- **M1** (Code Size): Range 0-10, current value ≈ 5.6
- **M2** (Modules): Range 0-334, current value ≈ 4.334
- **M3** (Classes): Range 0-236, current value ≈ 0.236

Output Variable (Column 4):

- **Uncertainty:** Range 0-100, resulting value ≈ 3.368

Current Input Values:

- **M1** = 5.6 (relatively small code size)
- **M2** = 4.334 (small number of modules)
- **M3** = 0.236 (minimal inherited classes)

Result:

- **Uncertainty** = 3.368 (very low uncertainty level)

This outcome aligns with your fuzzy rules where small values for all parameters result in low uncertainty. The red vertical lines show the current input values, and the yellow highlighting indicates which membership functions are active. Table 1 shows parameters with category, range and membership function.

Table 1

Parameter Classification

Parameter	Category	Range	Membership/Degree
m_1 (Code Size)	Minimal	10-49	1.0
	Small	50-199	0.8
	Medium	200-399	0.58
	Large	400-499	0.21
m_2 (Modules×20)	Minimal	≤50	1.0
	Small	50-99	0.82

	Medium	100-199	0.63
	Large	200-299	0.17
m_3 (Classes×30)	Minimal	<50	0.0
	Small	50-99	0.75
	Medium	100-199	0.59
	Large	200-299	0.19

Results: Multi-Objective Optimization

Iteration 1: Parameter Values: $m_1 = 0.19$, $m_2 = 0.3929$, $m_3 = 0.4171$

Objective 1 (Risk Minimization): Cost calculation: $0.02m_1^2 + 0.045m_2^2 + 0.03m_3^2 + 0.80127m_1 + 0.70167m_2 + 0.712m_3$

Result: 0.73779

Objective 2 (Cost Minimization):

Coefficient of variation approach: cv = 15.67

Result: 0.195335811

Iteration 2

Parameter Values: $m_1 = 0.33$, $m_2 = 0.33$, $m_3 = 0.33$

Results:

Objective 1: 0.741276

Objective 2: 0.175106

The ALHN-fuzzy hybrid approach demonstrates effective balance between cost optimization and risk management, with iterative improvement showing convergence toward optimal parameter values while maintaining acceptable uncertainty levels.

4. Discussion

The novel scheme proposed in this study optimizes three parameters, namely number of modules, level of inheritance and lines of code. ALHN (Augmented Lagrange Hopfield Network) is used to get the parameters to their lowest possible value. A novel technique based on fuzzy sets and ALHN is introduced. The new scheme can help a designer to calculate the optimal values of the parameters that significantly affect the software development process, designer can determine the ideal values for the factors that have good impact on the software development process with the aid of this new scheme. This is a big change in the software development process.

Soft Computing

Through machine learning we can simulate human like thinking which aids in software development and software process and decision-making system.

Table 2 depicts the scheme, we are using one of the soft computing techniques, Linear Regression equations, to predict the cost in the proposed model.

Table 2

Data as used in Machine Learning Algorithms

S. No	Desired Value	Risk	m1	m2	m3	Cost
1	200	*1.0260	141.8100	40	35	3309.66
2	235	0.9198	182.8489	40	35	3489.17
3	260	0.9198	141.8100	40	35	3849.17
4	275	0.9198	399.9900	40	20	9654.43
5	300	0.9198	225.5700	40	67	4947.40
6	475	0.9198	303.31	40	194.75	8229.18
7	575	0.9198	440	40	200	9654.44

Figure 5 shows in detail a heat map of data used in the model (risk, m1, m1, m2, cost, and desired value), and Figure 6 shows a distortion map of output values.

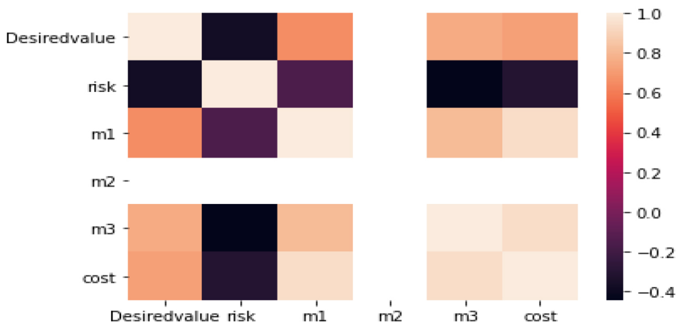

Figure 5

Heat map of data used in the model

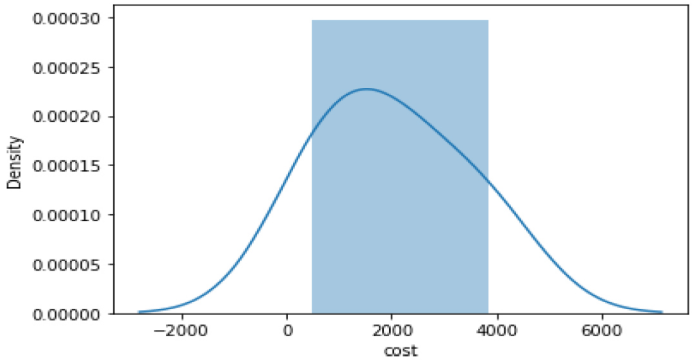

Figure 6

Distortion map of output values

5. Conclusion

We have presented a new scheme that optimizes three parameters viz. Lines of codes, number of modules and level of inheritance. The minimal value of the parameters is achieved using ALHN (Augmented Lagrange Hopfield Network). The scheme is based on ALHN and Fuzzy sets have been presented. The fuzzy set theory can be used to mathematically model subjectivity and ambiguity, providing a way to link binary logic and human reasoning. The idea behind linear regression is to find the minimum vertical distance between the fitted regression line and the pixel.

So, we attempted to get this optimal line by minimizing the distance between all pixels and the optimal line and the distance between all the pixels and the distance to the optimal line. The residual value is calculated by finding the difference between Y coordinates and the optimal value.

Approaching a skill that aids developers in the practice of building software design is the aim of this research project. Therefore, based on the above three important parameters, a generic model of software design systems has been defined. The best values of the parameters are provided by the ALHN algorithm, and a risk term computed using fuzzy logic. The logical study to create the best model is using machine learning.

References

[1] A. Martín, G. Rossi, A. Cechich, and S. Gordillo, "Engineering accessible Web applications: An aspect-oriented approach," *World Wide*

Web, vol. 13, no. 4, pp. 419–440 (Dec. 2010), doi: 10.1007/s11280-010-0091-3.

[2] E. Tramontana, "Automatically characterising components with concerns and reducing tangling," in *Proc. IEEE 37th Annu. Comput. Softw. Appl. Conf. Workshops (COMPSACW)*, Kyoto, Japan, pp. 499–504 (Jul. 2013), doi: 10.1109/COMPSACW.2013.114.

[3] M. Galster, D. Weyns, and M. Goedicke, "Variability and complexity in software design—Towards quality through modeling and testing," *ACM SIGSOFT Softw. Eng. Notes*, vol. 42, no. 4, pp. 35–37 (Jul. 2017).

[4] S. Iqbal and G. Allen, "Designing aspects with AODL," *Int. J. Softw. Eng.,* vol. 4, no. 2, pp. 3–18 (2011).

[5] G. Rasool and M. Arshad, "A review of code smell mining techniques," *J. Softw.: Evol. Process,* vol. 27, no. 11, pp. 867–895 (Nov. 2015), doi: 10.1002/smr.1737.

[6] F. Khomh and Y.-G. Guéhéneuc, "Design patterns impact on software quality: Where are the theories?," in *Proc. 25th IEEE Int. Conf. Softw. Anal., Evol. Reengineering (SANER)*, Campobasso, Italy, pp. 544–548 (Mar. 2018), doi: 10.1109/SANER.2018.8330254.

[7] S. Gautam, A. Khunteta, and D. Ghosh, "Adaptive optimization framework for multimodal software defect prediction using reinforcement learning," *J. Inf. Optim. Sci.,* vol. 46, no. 7, pp. 2351–2362 (2025), doi: 10.47974/JIOS-2162.

[8] A. Dey, "Machine learning algorithms: A review," *Int. J. Comput. Sci. Inf. Technol.,* vol. 7, no. 3, pp. 1174–1179 (2016).

[9] S. Kotagiri, A. M. Sowjanya, B. Anilkumar, and N. Lakshmi Devi, "Aspect-oriented extraction and sentiment analysis using optimized hybrid deep learning approaches," *Multimed. Tools Appl.*, vol. 83, pp. 88613–88644 (2024), doi: 10.1007/s11042-024-18964-9.

[10] L. B. Purbey and K. Lakhwani, "Aspect-based sentimental analysis using optimized multi-layered deep BiLSTM classifier," *Knowl.-Based Syst.,* vol. 324, Art. no. 113832 (2025), doi: 10.1016/j.knosys.2025.113832.

[11] Y. Sharma and S. Lakra, "Advancing cloud load balancing: An energy-aware model using a hybrid genetic and nature-inspired algorithm," *J. Inf. Optim. Sci.,* vol. 47, no. 3, pp. 1145–1162 (2026), doi: 10.47974/JIOS-2113.

Received December, 2025