Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Optimizing Software Cost Estimation: A Hybrid Approach Integrating Soft Computing Techniques for Enhanced Accuracy and Efficiency

Veerpal Kaur[1], Dr. Sunny Arora[2]

[1]Research Scholar, Department of Computer Applications, Guru Kashi University, Talwandi Sabo
[2]Professor Faculty of Computer, dept. Science & Engineering, Guru Kashi University, Talwandi Sabo
Email: sidhuveer94@gmail.com

In software engineering, software cost estimate is a necessary process directly affecting project deadlines, budget planning, and resource allocation. Conventional estimating methods could overlook the inherent complexity and uncertainty of software development, therefore causing mistakes and maybe project failures. To optimise software cost estimation and so address these challenges, this work proposes a hybrid framework incorporating fuzzy logic, artificial neural networks (ANN), and genetic algorithms (GA). Although fuzzy logic models imprecise and ambiguous information to control uncertainty, ANN is employed for its ability to learn and generalise patterns from past data. GA guarantees accurate and effective assessment of the process by strengthening already used techniques with strong optimising characteristics. Combining these techniques makes use of their respective benefits to generate a synergistic plan beyond more traditional methods. The proposed framework is investigated using industry-standard datasets exhibiting appreciable increases in prediction accuracy, computational efficiency, and flexibility to varied project situations. Comparatively, for complex and large-scale projects the hybrid approach not only reduces estimating errors but also enhances scalability and dependability. Additionally included in the paper is a case application for a practical project, therefore stressing the relevant usefulness of the framework. By addressing the limitations of present approaches, this work reduces the gap between theoretical breakthroughs in soft computing and their pragmatic use in software engineering. stressing the need of applying contemporary computing techniques to solve evolving challenges in cost estimation, the outcomes provide useful information for scholars and business practitioners. At the end of this work is explored the possibility of new machine learning models and the investigation of domain-specific customisations to increase the applicability of the framework.

Keywords: Software cost estimation, fuzzy logic, artificial neural networks, genetic algorithms, hybrid framework, project management, soft computing, optimization, uncertainty management, scalability, prediction accuracy, computational efficiency, software engineering.

1. Introduction

The success of project management depends much on software cost estimate by influencing overall project scheduling, resource planning, and budget allocation. Accurate estimation guarantees the success of projects and helps to prevent cost overruns, thereby ensuring also the Conventional estimation methods, however, can find it challenging to manage the complexity and uncertainty of modern software projects—where needs are dynamic, team dynamics change, and technology is growing rapidly. These challenges highlight the need of original concepts that can provide more accuracy and freedom. Advanced computing techniques such fuzzy logic, artificial neural networks (ANN), and genetic algorithms (GA) show promise to break free from these limitations. Combining these methods enables one to harness their combined benefits to get beyond the constraints of conventional methods. ANN excels in pattern recognition and data learning; fuzzy logic provides the means to effectively control uncertainty and imprecision; GA offers great optimising power. These techniques used together form the foundation of a hybrid approach designed to raise reliability and estimate accuracy. Although these hybrid models show significant potential, their use brings challenges including the need to customise them to specific project settings, assure scalability for demanding projects, and management of the computing cost. This work aims to tackle these challenges by means of a hybrid framework combining fuzzy logic, ANN, and GA to give superior estimate performance. The applicability of the proposed framework is evaluated using industry-standard datasets, which also demonstrate its advantages above traditional approaches. The ultimate goal is to provide practitioners and researchers with useful insights so they may apply these models correctly in real-world conditions, hence enabling enhanced project results and decision-making. By exploiting the synergy of soft computing techniques, this work intends to significantly contribute to the field of software cost estimation, so bridging the gap between theoretical developments and actual implementations. Through careful research and review, the findings of this work seek to open the road for more accurate and efficient estimating methods in the software industry, so addressing both existing and future challenges.

2. Literature Review

Sno	Author(s)	Year	Title	Advantages	Limitations
1	Butt et al.	2023	Prediction-based cost estimation technique in agile development	Agile-friendly approach that adapts to changing project dynamics.	Might not work well in non-agile or hybrid methodologies.

2	Sreekanth et al.	2023	Evaluation of estimation using deep learning-modified neural network	Uses advanced neural networks to improve estimation in diverse projects.	High computational costs and training data requirements.
3	Mishra et al.	2023	Uncrewed Aerial Systems in water resource management	Provides valuable insights for environmental monitoring using UAVs.	UAS deployment in real-world scenarios can be costly and complex.
4	Hameed et al.	2023	Optimized case-based software project effort estimation using genetic algorithm	Genetic algorithms provide flexibility in adapting to project variations.	May require extensive time and resources for convergence in large datasets.
5	Dupletsa et al.	2023	GWFish: simulation software for gravitational-wave parameter estimation	Provides simulations crucial for gravitational wave research.	Simulation accuracy depends on the input data and modeling assumptions.
6	Sharma & Sadam	2023	Predictive decision impacts on software projects	Helps in optimizing resource allocation for software projects.	Predictive models may fail if the underlying assumptions are inaccurate.
7	Parmasehr et al.	2023	Challenges in construction management using BIM solutions	BIM provides solutions for better cost and quality management.	BIM adoption in construction faces integration and training barriers.
8	Singh et al.	2023	Hybrid fuzzy AHP-TOPSIS technique for software testing parameter selection	Combines AHP and TOPSIS for better decision-making in testing.	The hybrid method can be computationally intensive.
9	Rathor et al.	2023	Technological evaluation and software bug training using GA-TCN	Effective in detecting software bugs with enhanced predictive capability.	High computational cost and complexity in training models.
10	Qian et al.	2023	Communicative agents for software development	Automates repetitive tasks, improving developer productivity.	Dependence on the agent's ability to interpret complex tasks accurately.
11	White et al.	2023	ChatGPT prompt patterns for improving code quality	Provides a novel approach to integrating AI for software quality improvement.	Dependence on the AI model's understanding of complex code contexts.
12	Khuwaileh et al.	2023	Nuclear powered desalination in UAE: technology and cost estimates	Promotes sustainable desalination solutions with nuclear energy.	High initial costs and safety concerns surrounding nuclear technology.
13	Traini et al.	2023	Assessment of steady-state performance in Java software	Improves performance evaluation in Java applications.	Limited to Java applications, not generalizable to other languages.
14	Antoniou et al.	2023	Cost and material quantity prediction for metro station construction	Provides accurate forecasting for large-scale construction projects.	May not apply to smaller-scale or non-urban projects.
15	Beiranvand & Zare Chahooki	2023	Software cost estimation based on feature selection of homogeneous clusters	Reduces the dimensionality and increases the accuracy of cost estimations.	Homogeneous clustering may not always reflect the complexity of real-world projects.
16	Martini, Enrico, Michele Boldo,	2022	Cost estimation model for IoT projects using fuzzy logic and machine learning techniques	Combines fuzzy logic and machine learning to improve IoT project cost estimation.	May require a large amount of data and pre-modeling analysis.
17	Martínez-Fernández et al.	2022	Software engineering for AI-based systems: a survey	Provides a comprehensive framework for AI system development.	May be limited to large-scale AI applications, not applicable to smaller systems.
18	Abdel-Hamid & Madnick	2022	Dynamic system modeling for software cost estimation	Dynamic approach improves cost accuracy over traditional methods.	Complex modeling may require substantial data and computational resources.
19	Mahmood et al.	2022	Software effort estimation accuracy prediction using ML	Uses ML to improve estimation accuracy across diverse projects.	ML models require large datasets for training, which may not always be available.

20	Rhmann et al.	2022	Software effort estimation using ensemble of hybrid search-based algorithms	Hybrid approach enhances accuracy by combining multiple algorithms.	May lead to high computational overhead due to the hybrid nature.

Table 1 Literature Review

Cost Estimation Models

Cost estimation models are mathematical tools or frameworks used to predict the financial resources required for a software project based on various input parameters. These models are essential for planning, budgeting, resource allocation, and decision-making throughout the project lifecycle. Here are some common types of cost estimation models:

1. Expert Judgment: This unofficial method depends on the knowledge and experience of people or groups with software development experience. Experts make estimates based on judgement and intuition by drawing on their knowledge of previous projects, industry standards, and best practices.

2. Analogous Estimation: Also known as top-down estimation, this approach uses historical data from similar past projects as a basis for estimating the costs of the current project. It assumes that the cost of the new project will be similar to those of past projects with comparable characteristics.

3. Parametric Estimation Models: Utilising statistical methods, parametric models calculate project costs by utilising past data and project specifications. By establishing links between project characteristics (such size, complexity, and productivity) and expenses, these models enable more precise estimations. Function Point Analysis and COCOMO (Constructive Cost Model) are two examples.

4. Algorithmic Estimation Models: Algorithmic models employ mathematical algorithms and formulas to estimate project costs based on input parameters such as lines of code, function points, or other size measures. These models often incorporate factors such as labor rates, overhead costs, and productivity rates to calculate estimates. COCOMO II and SLIM (Software Life Cycle Management) are examples of algorithmic models.

5. Machine Learning Models: With advancements in data science, machine learning techniques are increasingly being used for cost estimation. These models analyze large datasets of past projects to identify patterns and relationships between project attributes and costs, allowing for more accurate predictions.

6. Delphi Method: With this method, standardised surveys or rounds of debate are used to gather opinions from a panel of experts. Until an agreement is obtained on the expected project expenses, the replies are combined and improved iteratively.

7. Monte Carlo Simulation: “Monte Carlo simulation” involves generating multiple random scenarios based on input parameters and running simulations to estimate project costs. By simulating a range of possible outcomes, this approach provides a probabilistic estimate of project costs along with the associated uncertainty.

8. Each of these cost estimation models has its advantages, limitations, and suitability for different types of projects. Organizations often use a combination of these models and

techniques to derive more accurate and reliable cost estimates for software development projects.

A model might be dynamic or static. One variable is considered to be the primary factor in determining both cost and time in a static model. There is no fundamental variable and every variable in a dynamic model is reliant on every other variable.A model might be dynamic or static. One variable is considered to be the primary factor in determining both cost and time in a static model. There is no fundamental variable and every variable in a dynamic model is reliant on every other variable.

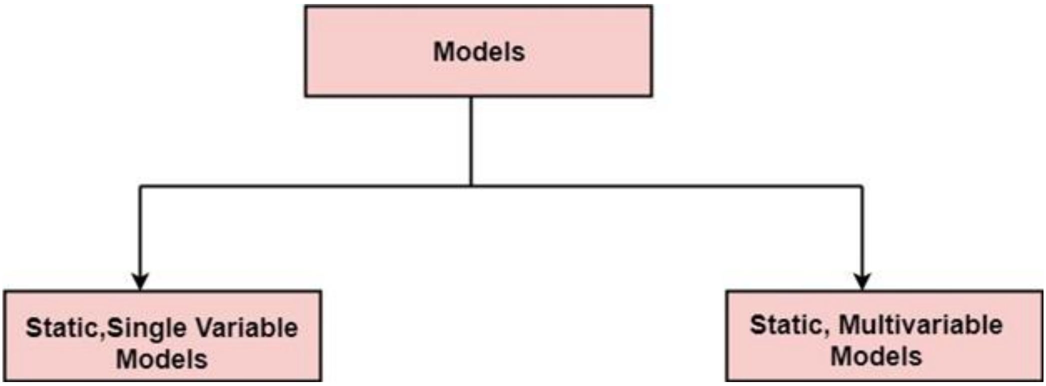

Fig: 1 Cost Estimation Models

Static, Single Variable Models: When a model makes use of single variables to calculate desired values such as cost, time, efforts, etc. is said to be a single variable model. The most common equation is:

$C=aL^b$

Where C = Costs, and L= size a and b are constants

A software production estimation model known as the SEL model was developed by the Software Engineering Laboratory. One example of a static, single-variable model is this one.

Multivariable Static Models: These models, which derive from technique (1), are dependent on a number of factors that characterize different facets of the software development environment. A number of factors are required in various models to represent the software development process, and an equation was chosen to integrate these variables and provide an estimate of both time and cost. We refer to these models as multivariable models.

3. Methodology

Software cost estimation is a complex task due to the uncertainty and complexity of software projects, the lack of historical data, inexperienced project managers, outdated estimation methods, human factors, dynamic nature of projects, variability in team skills and resource availability, and inadequate risk identification and contingency planning. To address these challenges, a comprehensive approach involving improved requirement analysis, hybrid estimation methods, effective risk management, and regular reviews and updates of estimates is needed.

Objectives:

1. Examine Factors Influencing Software Cost Estimation: Analyze the key factors that impact software cost estimation, such as project scope, complexity, team expertise, resource availability, and technological advancements. Understanding these factors is crucial for improving estimation accuracy.

2. Develop a Hybrid Model Using Soft Computing Techniques: Create a hybrid model that integrates multiple soft computing techniques (such as neural networks, fuzzy logic, genetic algorithms, etc.) to enhance the accuracy and reliability of software cost estimates. This model will leverage the strengths of various techniques to provide a more comprehensive estimation approach.

Factors influencing software cost estimation

Estimating software costs accurately is influenced by a multitude of factors, each playing a critical role in determining the overall budget and resources required for a project. Key factors include:

1. Project Scope and Requirements:

- Clarity and Stability: Well-defined and stable requirements lead to more accurate estimates, while unclear or evolving requirements introduce uncertainty and potential for cost overruns.

- Complexity: Projects with complex functionalities, intricate integrations, and high customization needs are more challenging to estimate.

2. Size of the Project:

- Project Scale: Larger projects with more features, modules, and dependencies generally require more resources and time, affecting cost estimates.

3. Team Expertise and Experience:

- Skill Level: The proficiency and experience of the development team influence productivity and efficiency, impacting cost.

- Past Experience: Teams with experience in similar projects can leverage their knowledge to provide more accurate estimates.

4. Technology and Tools:

- Technology Stack: The choice of programming languages, frameworks, and tools can affect development speed and cost.

- Tool Availability: Availability and familiarity with development and project management tools can streamline processes and reduce costs.

5. Development Environment:

- Infrastructure: The quality and availability of hardware, software, and network infrastructure play a role in cost estimation.
- Support Systems: Access to robust support and maintenance systems can mitigate unforeseen issues and cost escalations.

6. Project Management and Processes:

- Methodologies: Agile, waterfall, and hybrid methodologies have different impacts on cost estimation accuracy.
- Processes: Efficient project management processes and practices enhance predictability and control over costs.

7. Risk and Uncertainty:

- Risk Management: Identifying and planning for potential risks can prevent unexpected cost increases.
- Uncertainty: High levels of uncertainty in project requirements, technology, or environment can lead to conservative estimates to buffer against potential overruns.

8. Resource Availability and Costs:

- Human Resources: The availability, cost, and turnover rates of skilled personnel influence project costs.
- Material Resources: Costs of software licenses, development tools, and other materials are significant factors.

9. External Factors:

- Market Conditions: Economic conditions, industry trends, and market demand can impact resource costs and availability.
- Regulatory and Compliance Requirements: Adhering to industry regulations and compliance standards can add to the project cost.

10. Historical Data and Past Performance:

- Benchmarking: Historical data from previous projects provides a reference point for new estimates.
- Lessons Learned: Applying insights from past successes and failures can improve the accuracy of future estimates.

By understanding and considering these factors, project managers and estimators can enhance the accuracy of their software cost predictions, leading to better planning, resource allocation, and overall project success.

Hybrid PSO-Based Deep Learning Model

Incorporating Particle Swarm Optimization (PSO) into deep learning models can significantly enhance their performance in cost and effort estimation. PSO is a nature-inspired optimization algorithm that excels at identifying optimal solutions in high-dimensional spaces. When applied to deep learning, PSO can optimize hyperparameters such as learning rates, the number of layers, and neurons, which are critical for achieving high accuracy and reducing overfitting.

A hybrid PSO-based deep learning model can play a significant role in cost estimation by addressing the following aspects:

1. Improved Parameter Optimization: PSO efficiently tunes hyperparameters and model weights, ensuring the deep learning model converges to the most optimal solution for cost and effort prediction.

2. Enhanced Accuracy: By identifying the best model configurations, PSO reduces prediction errors in cost estimation models, especially for complex, real-world datasets.

3. Adaptability to Diverse Data: The hybrid model can be trained on a wide range of datasets, accounting for variations in project size, type, and domain. PSO further enhances adaptability by enabling the model to find optimal solutions under different constraints.

4. Reduced Computational Costs: PSO accelerates the training process by guiding the deep learning model toward convergence more efficiently, reducing the time required to fine-tune parameters manually.

This hybrid approach not only enhances prediction accuracy but also ensures the cost estimation model is robust, scalable, and efficient, addressing the limitations of conventional and standalone deep learning methods. It proves invaluable in project planning, budgeting, and resource allocation, ultimately driving better decision-making and cost control.

Equation for PSO and ACO

$$\mathrm{E_{PSO}} = 2.023 \times (\mathrm{KLOC})^{0.898}$$

$$\mathrm{E_{ACO}} = 2.01 \times (\mathrm{KLOC})^{0.898}$$

Table 2 Computed Effort Value by Basic Cost Estimation And PSO

Project No.	KLOC	Dataset Effort	Traditional cost estimation model Effort	PSO Effort	ACO effort
1	90.97	116.27	271.58	114.98	113.62
2	46.22	96.52	134.36	63.69	63.15
3	47.16	79.69	135.27	63.90	63.51
4	55.45	91.36	159.94	73.71	72.56
5	31.95	40.53	89.12	45.04	44.06
6	67.76	99.23	200.42	89.18	88.43
7	13.36	19.58	35.24	20.81	19.89
8	11.47	10.84	28.44	17.29	16.57
9	22.47	28.60	60.67	32.10	32.33
10	3.47	7.59	8.03	6.07	5.88
11	4.82	9.70	11.32	8.01	8.04
12	8.63	8.28	21.06	13.52	13.61

13	2.33	5.07	5.36	4.33	4.78
14	5.77	8.47	13.74	9.46	8.98
15	79.03	99.63	235.61	102.00	101.01
16	9.78	16.10	26.77	16.36	15.74
17	13.31	24.27	34.26	20.44	19.77
18	101.16	139.13	304.86	127.57	126.20

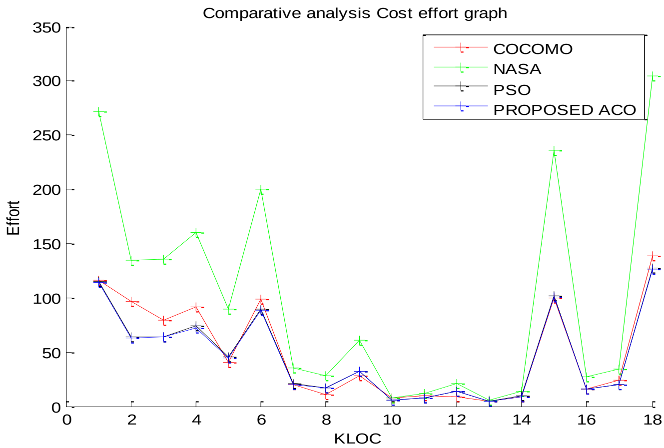

Fig 1 Comparative Analysis of Effort Graph

Simulation for Hybrid ACO/PSO-based approach

Following these steps in a Python-based simulation will help us to show training and testing accuracy curves alongside the evaluation metrics (accuracy, recall, precision, and F1-score) for both the proposed hybrid model (ACO/PSO-based Deep Learning) and the conventional model (e.g., traditional deep learning model).

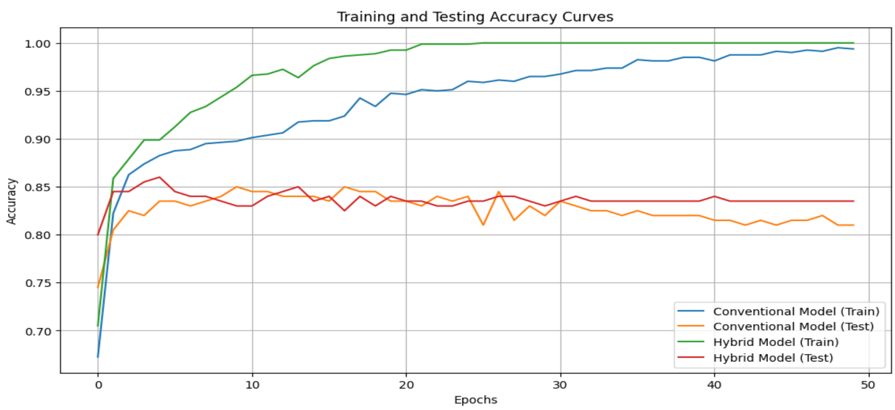

Fig 2 Training and Testing Accuracy Curves

4. Conclusion:

Accurate software cost estimation is crucial for the success of modern software projects, as it directly impacts budgeting, scheduling, and resource allocation. This paper introduces a hybrid framework combining fuzzy logic, artificial neural networks (ANN), and genetic algorithms (GA) to address the limitations of traditional estimation techniques. By leveraging the strengths of these soft computing methods, the proposed approach enhances prediction accuracy, scalability, and flexibility in various project scenarios.

The results from applying this framework to industry-standard datasets demonstrate its ability to reduce estimation errors and adapt to complex project environments. Moreover, the integration of optimization algorithms like PSO and ACO further refines the estimation process, ensuring efficient resource utilization and decision-making.

The study bridges the gap between theoretical advancements in soft computing and their practical implementation in software engineering. Future research could explore domain-specific customizations and the integration of emerging machine learning models to further extend the applicability and robustness of this hybrid approach. Ultimately, this work emphasizes the need for innovative computational techniques to address the evolving challenges in software cost estimation, paving the way for improved project outcomes in the software development industry.

References

1. G. S. Rao, C. V. P. Krishna, and K. R. Rao. (2014). Multi Objective Particle Swarm Optimization for Software Cost Estimation, ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India- Vol I pp 125-132 2014.
2. F. Pagin. (2014). An UML-based approach to software development cost estimation, Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement
3. Cheng, B., & Misra, S. (2014). Hybrid models combining fuzzy logic and neural networks for software cost estimation. Journal of Systems and Software, 91, 1–12.
4. Bardsiri, A. K., & Hashemi, S. (2014). Using particle swarm optimization for effort parameter optimization. Software Quality Journal, 22(3), 353–367.
5. F. azzahraAmazal, A. Abran. (2015). Analogy-based software development effort estimation: a systematic mapping and review, Info. Softw. Technol. 58, 206–230
6. López-Martín, C. (2015). Predictive accuracy comparison between neural networks and statistical regression for development effort of software projects. Appl. Soft Comput. 27, 434–449
7. Vaibhav Jain, Mohsin Sheikh. (2016). Quantitative Estimation of Cost Drivers for Intermediate COCOMO towards Traditional and Cloud Based Software Development, Conference: The Ninth Annual ACM India Conference 2016.
8. H. K. Sharma, R. Tomar, J. C. Patni, and A. Dumka. (2016). E-COCOMO: An effort estimation model for cleanroom software development approach, 2016 2nd International Conference on Next Generation Computing Technologies (NGCT) no. October, pp. 131–136, 2016.
9. Rohit Kumar Sachan∗, Ayush Nigam, Avinash Singh, Sharad Singh, Manjeet Choudhary, Avinash Tiwari and Dharmender Singh Kushwaha. (2016). Optimizing Basic COCOMO Model using Simplified Genetic Algorithm Motilal Nehru National Institute of Technology Allahabad, Allahabad 211 004, India Twelfth International Multi-Conference on Information Processing
10. Ajay kaushik, Hieshkumar. (2016). Performance Evaluation between GA versus PSO

International Research Journal of Engineering and Technology (IRJET) Volume: 03 Issue: 06| June-2016
11. Idri, A., Hosni, M., Abran, A. (2016). Systematic literature review of ensemble effort estimation. J. Syst. Softw. 118, 151–175
12. Nanassif, A. B., Azzeh, M., Capretz, L. F., Ho, D. (2016). Neural network models for software development effort estimation: a comparative study. Neural Comput. Appl. 27(8), 2369–2381
13. D. Banerjee, V. Kukreja, A. Gupta, V. Singh and T. Pal Singh Brar, "Combining CNN and SVM for Accurate Identification of Ridge Gourd Leaf Diseases," 2023 3rd Asian Conference on Innovation in Technology (ASIANCON), Ravet IN, India, 2023, pp. 1-5, doi: 10.1109/ASIANCON58793.2023.10269834.
14. Kumar, Y., Brar, T.P.S., Kaur, C. et al. A Comprehensive Study of Deep Learning Methods for Kidney Tumor, Cyst, and Stone Diagnostics and Detection Using CT Images. Arch Computat Methods Eng 31, 4163–4188 (2024). https://doi.org/10.1007/s11831-024-10112-8
15. D. Banerjee, V. Kukreja, A. Gupta, V. Singh and T. P. S. Brar, "CNN and SVM-based Model for Effective Watermelon Disease Classification," 2023 3rd Asian Conference on Innovation in Technology (ASIANCON), Ravet IN, India, 2023, pp. 1-6, doi: 10.1109/ASIANCON58793.2023.10270512.
16. Banga, A., & Bhatia, P. K. (2024, April). Accuracy enhancement of Component based selection model using Hybrid Soft computing. In 2024 Sixth International Conference on Computational Intelligence and Communication Technologies (CCICT) (pp. 149-156). IEEE.
17. Fahimi, F., Yaseen, Z. M., & El-shafie, A. (2017). Application of soft computing based hybrid models in hydrological variables modeling: a comprehensive review. Theoretical and applied climatology, 128, 875-903.
18. Sedano, J., Berzosa, A., Villar, J. R., Corchado, E., & de la Cal, E. (2011). Optimising operational costs using soft computing techniques. Integrated Computer-Aided Engineering, 18(4), 313-325.
19. Bonissone, P. P., Chen, Y. T., Goebel, K., & Khedkar, P. S. (1999). Hybrid soft computing systems: industrial and commercial applications. Proceedings of the IEEE, 87(9), 1641-1667.
20. Jaiswal, A., Raikwal, J., &Raikwal, P. (2024). Optimized Cost Estimation in Software Project Planning using Fuzzy Logic and Genetic Algorithm.