Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

Predicting the Quality of Software by Using Cat Swarm Optimization (CSO) Algorithm

Zuhal Adil Madlool
Computer science and information technology, Al-Qadisiyah University, Al-Qadisiyah, Iraq. [e-mail : zuhal.adil@qu.edu.iq

ARTICLE INFO
Article history:
Received: 10 /1/2025
Rrevised form: 7 /3/2025
Accepted : 27 /3/2025
Available online: 30 /3/2025

Keywords:
Software Defect
Machine Learning
Prediction
Optimization.

ABSTRACT

In the early stages of software engineering, efforts were made to enhance software quality by using quality indicators that are considered vital in software development. Software testing was improved and software issues were identified via the use of predictive quality metrics. An evaluation, testing, and analysis system based on a dataset extracted from the NASA quality metrics database utilising a cat swarm optimization approach is the goal of this project, this is doing by applying two steps: first step is preprocessing of the data to process the missing value and after that apply normalization, second step using CSO algorithm to find the goal of this study is to predicting the quality of software. This research demonstrates how effective machine learning approaches are at extracting knowledge from big databases and offers useful insights into software quality prediction, when compared to all of the classification techniques used in the research, the results demonstrated that the approach considerably improved the accuracy rate to 96% in predicting quality performance.

MSC..

https://doi.org/10.29304/jqcsm.2025.17.11981

1. Introduction

Building high-quality software is an extremely challenging task due to the numerous essential components, their flexibility, and the relationships between them, as well as the management, measurement, and assurance of software quality. It is the responsibility of software engineers to provide methods, models, tools, and procedures that simplify the management of such complicated problems. Tools, techniques, process, and quality make up the four tiers of software engineering technology shown in Figure (1).[2]

∗Corresponding author

Email addresses:

Communicated by 'sub etitor'

Figure (1): Basic principles of software engineering

The main concerns of the Quality layer are the qualities and characteristics of a product or service, as well as its ability to meet explicit or implicit standards. Software engineering may also be seen as a discipline of computer science that involves gathering requirements, designing systems, and developing and maintaining software[3][4]. There are two very important statements in this definition:

1. **Engineering Discipline .**

The process involves making selective use of relevant theories, methods, and tools to solve the problem at hand, as well as identifying and resolving issues for which no such solutions exist, all while adhering to the limitations and constraints of the institution.

2. **All aspects of software production .**

Software engineers are not only responsible for the technical parts of software development, but also for managing projects and coming up with ideas, strategies, and tools to help with software production.

One of the most important defects in the software quality testing process is when the data set contains unreliable properties or missing data. This defect in the data set may lead to incorrect uses using traditional methods, this defect was process through the first step of this model.

Optimization is achieved by choosing features that include the natural colonies of most wild animals, such as the particle swarm optimization algorithm and the genetic algorithm and others. Chu et al. developed and introduced a novel swarm intelligence technique called cat swarm optimization (CSO). The cat swarm optimization (CSO) algorithm was selected for solving continuous and multi-objective optimization problems by studying how these swarms behave. The CSO algorithm uses multiple iterations to achieve its goal, which makes it easy to implement, searches globally, and converges quickly, table (1) shows comparison between main prediction techniques. Some important benefits of this study are as follows:

- The research aims to build a robust prediction model for software quality by examining the elements or aspects that effect it. Reliable predictions of software quality are essential to the world economy, and this will help with that.
- Prediction methods are used in the study to guarantee more accurate results by taking into account a large number of independent variables that might affect the dependent variables' variability.
- Based on real-world data and prediction algorithms like the CSO algorithm, this research seeks to provide a reliable method for predicting software quality.
- Because it addresses an actual issue, the research is both practical and relevant. The findings of this research have practical implications, for example in the context of the international economy.
- The research exemplifies the use of cutting-edge technologies, such deep learning, to accurately tackle complicated issues.

- Improved decision-making, for example in the engineering field, might result from accurate software quality forecasts.
- This research adds to the body of knowledge in AI and data analysis by demonstrating the feasibility of using machine learning for software quality prediction. Additional research in this field may be informed by the results of this study.

Table 1: Comparison among main prediction techniques

Prediction Techniques	Advantage	Disadvantage
Particle Swarm Optimization	▪ PSO's global search is lightning quick, in contrast to its slowness in local search,	▪ Since PSO converges quickly, it becomes stuck in local optima very quickly. ▪ produces solutions of poor quality.
Genetic Algorithm	▪ Ability to Search on a Global Scale. ▪ Ability to adapt ▪ the concept of parallelism ▪ Ensuring durability and efficiency ▪ Flexibility ▪ No Need for Gradient Data	▪ High computational demand ▪ Sensitivity to parameters ▪ Speed of convergence ▪ The Absence of an Optimal Promise ▪ Design of Fitness Functions ▪ Difficulty in Implementation

2 Machine learning

A method that utilizes algorithms to process data to construct systems that can autonomously learn from their experiences and subsequently perform better over time to gain useful skill sets, particularly in the areas of prediction and optimization. This is especially important in many areas of emerging technologies [8]. Machine learning is a branch of science aimed at two main goals: to design computer programs that use experience and practice to enhance their performance, and to gain insight into the principles governing the operation of learning systems from a computational, information and theoretical perspective [8].

Machine learning comes in three flavors:

Supervised learning: In this method, a model is trained with known inputs/outputs and labels to predict future results using newly labeled data. This is the most frequently used method that allows specialists to teach and train algorithms to be able to produce desired results

Unsupervised learning: This approach takes in un-classified input data, processes it, analyzes, and consolidates it, but no predetermined output is produced like in supervised methods. It free-trains on your data to uncover insights and patterns. This technique follows a self-learning approach that allows machines to learn hidden patterns and methods needing no human intervention.

Unsupervised machine learning : Some of the fastest ways to structure and assess tasks based on data. These techniques can obtain the similarities and verify the go between data and simplify the model by decreasing the number of dimensions in the data, which will also reduce the number of features required. These attempts guide duplicate detection as finding similarities through images to classify. Neural networks, K-means clustering, probabilistic clustering algorithms, etc., are all examples of such approaches. The other common approaches are PCA and SVD.

Reinforcement learning: In a different approach to learning, models are constructed to make decisions one at a time, guided by performance rewards. This method rewards each successful step towards an objective in a complex environment, which makes it possible to learn how to achieve things in uncertain environments. Similar to some direct feedback based approaches, one major difference here is that there is no pre-defined data to learn from, and learning takes place in the form of trial and error, and no right answers or results are provided.

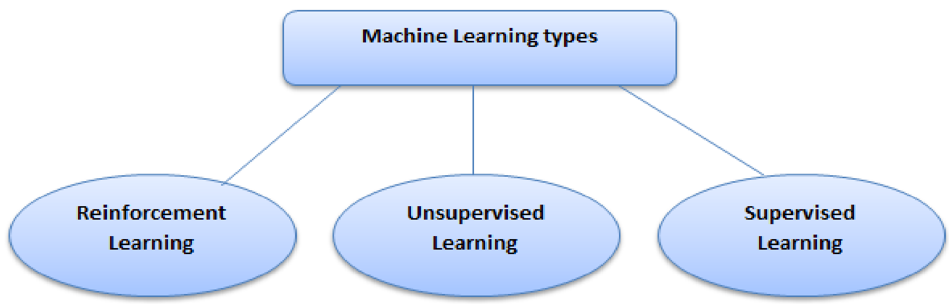


Fig. 2. Types of machine learning

2.1. Prediction

Prediction algorithms are increasingly employed to examine the structure and development of data through methods like data mining and statistics. Various techniques of content analysis utilizing specific algorithms have been formulated to meet set objectives. Prediction entails utilizing established model parameters to estimate how a system will react to a specific input. Both prediction and optimization may involve statistical and probabilistic methods. The aim of optimization is to improve results by making the best use of available resources.

2.2. Optimization

Optimization encompasses a wide range of mathematical ideas and methods used to quantitative problems in disciplines as varied as economics, biology, engineering, physics, and business. The realisation that apparently unconnected areas of study have mathematical underpinnings with quantitative problems gave rise to the concept of the field. The optimization field is able to provide a consistent framework for the analysis and resolution of several problems because of this common ground.

Optimization problems often include three primary parts. An objective function is one example; it's a single numerical value that must be met, either at the maximum or the lowest. It might be the time it takes for a car to go to a given place, the expected return on a stock portfolio, the production costs or profits of a firm, or even a candidate's vote share. The second portion consists of variables, which are integers whose values may be adjusted to improve the aim.

2.3 Statement of the problem:

The challenge in a system that can predict, with little error and high accuracy, the quality of software

2.4 Contribution:

The research aims to improve software quality prediction using a global search algorithm, thereby enabling the swift discovery of high-quality solutions.

2.5 Strategy of Evaluation :

In order to evaluate the effectiveness of the proposed system update, we used the cat optimization approach to predict the quality of software components using a NASA dataset. Accuracy, error rate, and mean square error (MSE) are some of the fundamental metrics used to evaluate unsupervised performance.

3. Related Work

Liqun, Shaoqiang, and Jiangshan, 2023[1] developed an algorithm to maximise reactive power in energy storage and distribution networks that use wind and solar power. We will show related research that predict the quality of software, our work similarity of it by using the same algorithm while differ it by applying the different type of measurement tools and dataset.

Ibrahim Ahmed Saleh and Salha Raa'd Mahammead, 2018 [25] Proposing a particle swarm optimization algorithm as a tool for evaluating software quality, where a software system was used that combines the PSO algorithm with Neural Network Back Propagation, meaning the PSO algorithm was improved to get rid of its problem in the research, so the research showed results with accuracy (98%). We will conduct research using the same data set and evaluation metrics (error rate, MSE) by applying the Cat optimization algorithm to predict software quality, where we obtained an accuracy of (96%). The aforementioned work can be developed using the genetic algorithm to obtain higher accuracy with the lowest error rate.

N. Prabhakaran, R. Nedunchelian, 2023[26] provide a feature selection approach for credit card fraud detection based on oppositional swarm optimization; our work is comparable to it in that we use the same method, but we use different measurement techniques and datasets.

4. Methodology of Cat Swarm Optimization Algorithm(CSO)

Cat Swarm Optimization (CSO) is a meta-heuristic optimization method that finds optimal solutions by emulating the cooperative behaviour of a swarm of cats.

The original cat swarm optimization approach had a single goal and was implemented continuously [4, 5]. The movements of cats when they slept and traced were a source of inspiration. It would seem that cats are generally rather lazy and satisfied to do nothing much at all. However, when they're sleeping, they become acutely aware of everything around them and of themselves. For this reason, they maintain a keen awareness of their surroundings and react swiftly to anything that catches their sight. Together, these two main feline characteristics provide the CSO algorithm's foundation.

Searching and tracing are the two modes that comprise the CSO algorithm. No two sets of answers are ever identical with respect to flag, fitness value, and location. The location definition is affected by each of the M dimensions of the search space, and each of these dimensions has its own velocity. The fitness value indicates that the solution set, which includes cats, is generally doing well. As a last step, the flag assigns each cat a specific task: searching or tracing. That is why there must be a certain number of cats in the initial cycle of the algorithm. The best cat from each iteration is stored in memory and represents the solution.

4.1. Proposed Technical System

1- Reading the dataset.

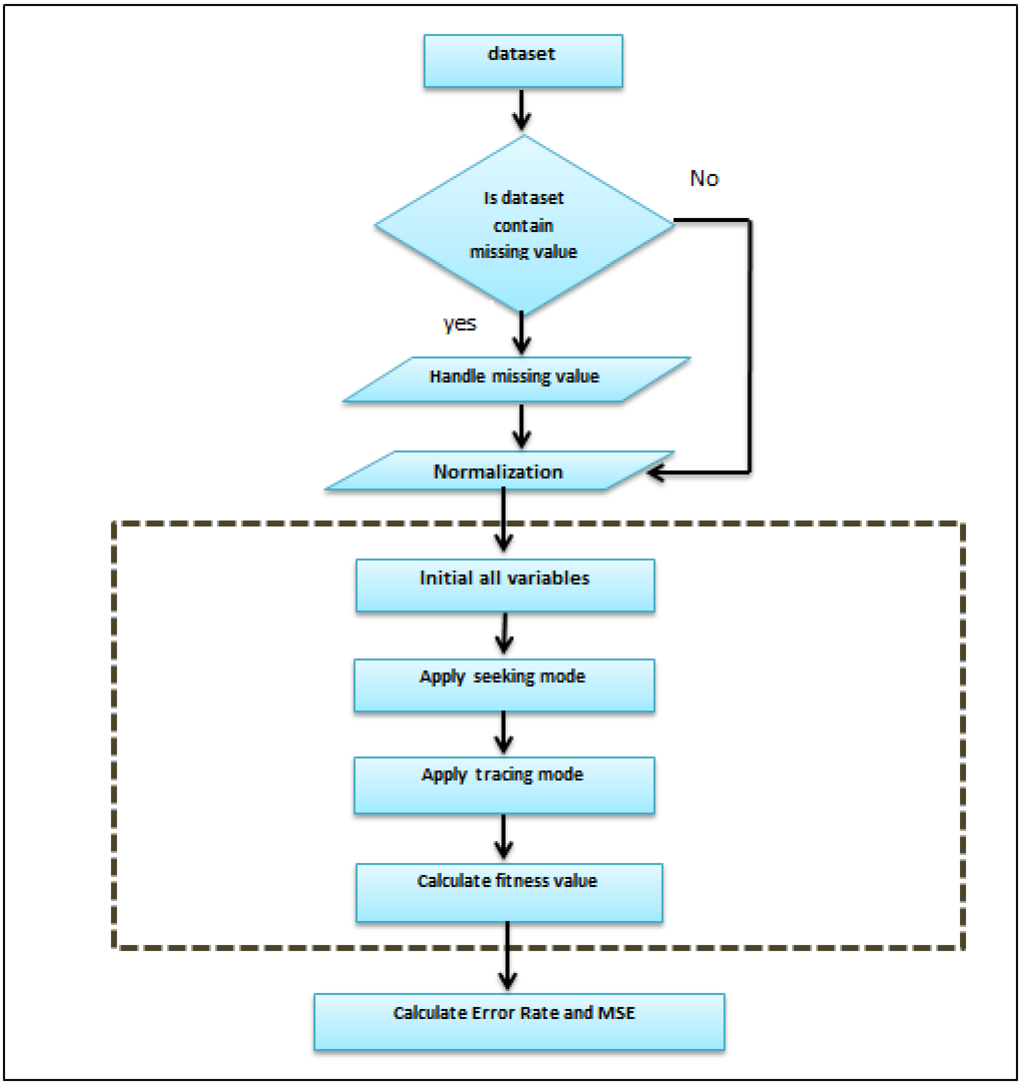
2- Handling missing values.

3- Normalizing the data.

4- Implementing the Developed Whale Optimization Algorithm (DWOA) to achieve the best fitness function (Accuracy).

5- Calculating the error rate and Mean Squared Error.

The graphic below depicts the technological stages of the proposed system:



Figure(3): The suggested system

4.2. General Structure of the Algorithms

In order to find the optimal solutions, the algorithm does the following:

(1) State the top and lower bounds of the range of feasible answers.

(2) In a space with M dimensions, scatter N sets of solutions, or cats, with velocities that are both unpredictable and limited to a predetermined maximum.

(3) Use MR to randomly allocate certain cats to the search group and others to the trace group. A value between zero and one is chosen for the mixture's MR ratio. Suppose, for the sake of argument, that we have ten cats and an MR of 0.2. At random, we may put two of them in the tracing mode and eight in the hunting mode.

(4) Apply the fitness scores for each cat to the domain-specific fitness function. The next step is to choose and remember the most worthy feline.

(5) Following this, the felines enter a period when they seek or trace prey.

(6) After each iteration of the cats' investigating or tracking mode, randomly assign each cat to a new mode based on their MR.

(7) Stop the program if the termination condition is true; otherwise, repeat Steps 4-6.

4.3. Seeking Mode

This mode is based on cats' sitting behaviours and consists of four parameters: searching memory pool (SMP), seeking range of the selected dimension (SRD), counts of dimension to change (CDC), and self-position considering (SPC). Through trial and error, the user determines and fine-tunes each of these parameters.

For example, giving cats the choice to choose their searching memory quantity from 1 to 5 would be a good example of SMP in action. In other words, we'll choose one cat at random from a pool of five and go on to the next. The new positions will be randomly assigned using CDC and SRD as additional criteria. The number of dimensions that may need upgrading might be anything from zero to one, according to the CDC. When CDC is set to 0.2, the fifth dimension stays the same, but four out of five dimensions need to be modified for each cat. The SRD stands for the mutative ratio, number of mutations, and changes for the CDC-selected dimensions. Last but not least, SPC is a Boolean value that specifies whether the present cat position will be added to the set of potential locations for the future iteration. If the SPC flag is set to true, we should construct (SMP-1) candidates for every cat instead of SMP ones, since this number includes the current position. Make sure you complete the following when in search mode:

(1) For SMP, you may make as many copies of Catk's present position as you need.

(2) For each copy, choose at random the number of CDC dimensions you want to change. Furthermore, the current values replace the previous ones in the following equation, thus you may add or subtract SRD values at random:

$$Xjd\ \text{new} = (1 + rand * SRD) * Xjd\ \text{old} \qquad (1)$$

where the current location is represented by Xjd*old* and the future position by Xjd*new*, where j is the cat number, d is the dimension, and rand is an integer from 0 to 1.

(3) get the fitness value (FS) for every available position.

(4) Before deciding where to send the cat next, weigh the chances of each possible spot; as stated in equation (2), the spots with higher FS are more likely to be selected. However, if they are identical, then assign a chance of 1 to each potential candidate point.

$$Pi = \frac{|FSi - FSb|}{FS\text{max} - FS\text{min}} \qquad (2)$$

For any integers i between zero and j, To minimise, set FSb=FSmax; otherwise, set FSb=FSmin.

4.4. Tracing Mode

The idea behind this setup is to mimic the way cats track prey. We start by giving a cat positional dimensions and speeds that are completely up to the user. However, for the subsequent steps, the velocity parameters need to be modified. This is how a cat walks when it walks in this manner:

(1) Using equation (3), update the velocities (Vk, d) of all dimensions.

(2) Outside the range of feasible values is a velocity that is equal to or higher than the maximum value.

$$Vk,\mathrm{d} = Vk,\mathrm{d} + r1c1(X\ \mathrm{best},\mathrm{d} - Xk,\mathrm{d}) \qquad (3)$$

Where c1is constant, a real integer r1 between 0 and 1, and the coordinates of the best-valued cat in dimension d ($X_{best,d}$) and the worst-valued cat ($X_{k,d}$) are used here.

(3) Change Catk's location based on this equation:

$$Xk,\mathrm{d} = Xk,\mathrm{d} + Vk,\mathrm{d} \qquad (4)$$

Algorithm 1. (CSO) Cat Swarm Optimization algorithm

```
Input data, No. of maximum and Population.
Initialization the Cat population Xi with values from 1 to n.
// Seeking mode
1: If SPC is True Then
      j=SMP
    Else
      j=SMP-1
2: Iterates j times, replicating the current position of cat Ck.
3: The amount of CDC determines whether the value of SRD is decreased or increased.
4: All candidate points should have their Fitness value (FV) computed.
5: Equation 2 is used to convert FV to selection probability in cases when all of the FV are not
equal.
6:    If the objective is minimization, then
7:         FS_b = FS_max
8:    else
9:         FS_b = FS_min
10:   Endif
11: Endif
//The objective of the fitness function determines whether it should maximise or minimise.
12: Simply replace the current position of cat Ck with one of the randomly selected candidate
spots.
// Mode for Tracing
10: Calculate new velocities for each cat using equations 3, 4.
13:  If Vk,d>Vmax  then
14:       Vk,d=Vmax
15:   else
```

4.5. Valuation stage

We implemented suggested strategy in Python and validated it using metrics such as accuracy, error rate, and mean square error to ensure its validity. Accuracy is directly proportional to the likelihood of obtaining the optimal outcome, defined here as the difference between anticipated and actual values. See Table (2) for an evaluation of the suggested method's predictive power with respect to software quality. The mean square error is the sum of all the values used in the computation divided by the total number of values. It is the squared error that results. The accuracy is used to compute the error rate. The results of the tests performed on the proposed technique for software quality prediction are shown in Figure (4).

Table 2. Result from CSO Algorithm

No.	(Accuracy)	Error	MSE

Iteration		rate	
50	89	7.0032	0.007
100	90.2401	7.9654	0.00667
150	91.5337	7.1002	0.00587
200	93.7432	6.5684	0.00492
250	93.9128	5.5642	0.00440
300	94.4321	4.8321	0.00398
350	94.3201	3.5590	0.00322
400	95.9478	2.6441	0.00267
450	96.3538	1.6789	0.00167
500	96.3538	1.6789	0.00167

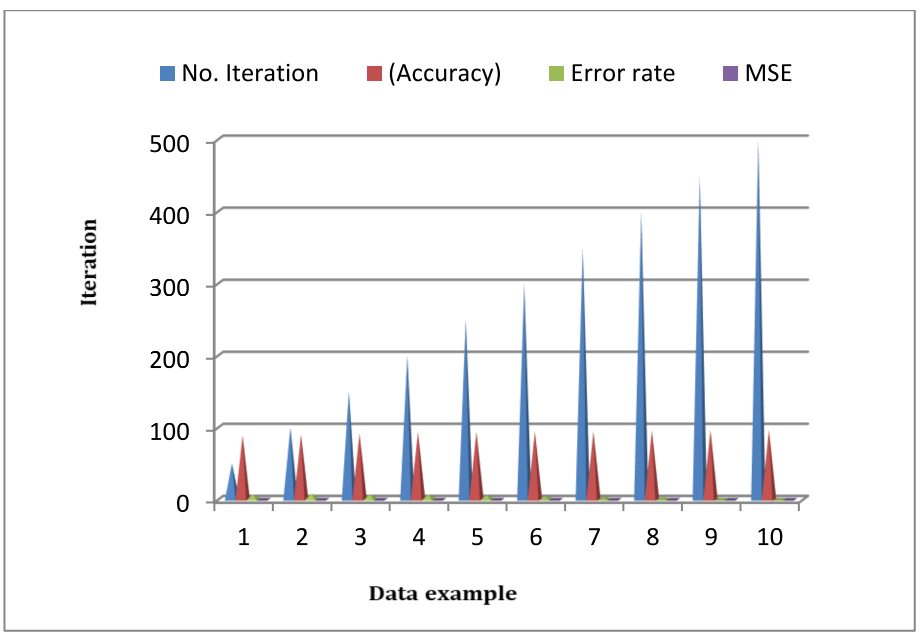


Fig. 4. Results of CSO Algorithm

5. Conclusion and Future work

These findings allowed us to get a specific level of accuracy for the previously published CAT optimization technique, which is used to forecast the quality of software. It is evident that the algorithm offers distinct benefits in terms of raising the forecast accuracy of software quality. As feature work, it aims to be able to use other optimization algorithms to achieve results with higher accuracy and lower error rates than those achieved in this original paper, or develop CSO by genetic algorithm.

References

1- Liqun, S., Shaoqiang, Z., Jiangshan, L. Reactive power optimization of wind/solar power storage and distribution network based on multi-objective sand cat swarm algorithm. J. Nanjing Univ. Inf. Sci. Technol, (2023). doi:https://doi.org/10.13878/j.cnki.jnuist.20230621001.
2- Kaizheng, J. & Liping, L. Optimization of fault diagnosis for stacked denoising auto encoder' s engine by improved sand cat swarm optimization algorithm. J. Mach. Des. 40, 56–62 (2023).
3- IEEE Std 610.12., 1990, IEEE Standard Glossary of Software Engineering Terminology.
4- Broy, M., (2011), " The Future of Software Engineering", S. Nanz (ed.), SpringerVerlag Berlin Heidelberg. ISBN 978-3-642-15186-6.
5- Pressman, R., 2010, "Software Engineering: A Practitioner's Approach", 7th Edition, McGraw-Hill, New York, USA, ISBN 978-0-07-337597-7
6- Chu S. C. and Tsai P. W., Computational intelligence based on the behavior of cats, International Journal of Innovative Computing, Information and Control. (2007) 3, no. 1, 163–173.
7- Chu S. C., Tsai P. W., and Pan J. S., Cat swarm optimization, Proceedings of the Pacific Rim International Conference on Artificial Intelligence, August 2006, Guilin, China, Springer, 854–858.
8- P. Mehta et al., "A high-bias, low-variance introduction to Machine Learning for physicists," Physics Reports,vol. 810, pp. 1–124, May 2019, doi: https://doi.org/10.1016/j.physrep.2019.03.001.
9- M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," Science, vol. 349,no. 6245, pp. 255–260, Jul. 2020, doi: https://doi.org/10.1126/science.aaa8415.
10- J. E. van Engelen and H. H. Hoos, "A Survey on semi-supervised Learning," Machine Learning, vol. 109, Nov.2019, doi: https://doi.org/10.1007/s10994-019-05855-6.
11- Y. Matsuo et al., "Deep learning, reinforcement learning, and world models," Neural Networks, Apr. 2022, doi:https://doi.org/10.1016/j.neunet.2022.03.037.
12- S. Sun, Z. Cao, H. Zhu, and J. Zhao, "A Survey of Optimization Methods From a Machine Learning Perspective," IEEE Transactions on Cybernetics, vol. 50, no. 8, pp. 3668–3681, Aug. 2020, doi: https://doi.org/10.1109/tcyb.2019.2950779.
13- J. R. R. A. Martins and A. Ning, Engineering Design Optimization. Cambridge University Press, 2021. Accessed: Jul. 31, 2024. [Online]. Available: https://books.google.iq/books?hl=en&lr=&id=dBVEEAAAQBAJ&oi=fnd&pg=PR13&dq=%5B13%5D%09Martins.
14- M. Prosperi et al., "Causal inference and counterfactual prediction in machine learning for actionable healthcare," Nature Machine Intelligence, vol. 2, no. 7, pp. 369–375, Jul. 2020, doi: https://doi.org/10.1038/s42256-020-0197-y.
15- W. Ben Chaabene, M. Flah, and M. L. Nehdi, "Machine learning prediction of mechanical properties of concrete: Critical review," Construction and Building Materials, vol. 260, p. 119889, Nov. 2020, doi: https://doi.org/10.1016/j.conbuildmat.2020.119889.
16- Sommerville, I. , 2011 , "Software Engineering" , 9th Edition , Addison Wesley, ISBN-13: 978-0-13-703515-1
17- S. Kim, H. Zhang, R. Wu and L. Gong, 2011 "Dealing with noise in defect prediction", Proceeding of the 33rd International Conference on Software engineering - ICSE '11, pp. 481-490.
18- M. M. Kabir, M. Shahjahan and K. Murase, , (2012) "A new hybrid ant colony optimization algorithm for feature selection", Expert Systems with Applications, vol. 39, no. 3, pp. 3747-3763
19- S. C. Yusta, (2009), "Different metaheuristic strategies to solve the feature selection problem", Pattern Recognition Letters, vol. 30, no. 5, pp. 525-534.
20- Pressman, Roger S.,(2010)," Software Engineering: A Practitioner's Approach", 7th edition , McGraw-Hill, ISBN 978–0–07–337597–7, ISBN 0–07–337597–7.
21- Mohd. Ehmer Khan. 2011, "Different Approaches to White Box Testing Technique for Finding Errors", International Journal of Software Engineering and Its Applications.
22- Grosan, C., Abraham, A.,(2011), " Intelligence Systems", Springer, Verlag Berlin Heidelberg, ISBN 978-3-642-21003-7.
23- Chang R, Lai L. , Wang J. and Kouh J. , 2007, "Intrusion Detection by Back propagation Neural Networks with Sample-Query and Attribute-Query", Department of Engineering Science and Ocean Engineering, National Taiwan University, International Journal of Computational Intelligence Research. Vol.3, No. 1 .
24- Gaitonde,V.,Karnik,S.,(2012),"Minimizing burr size in drilling using artificial neural network(ANN)-Particle swarm optimization(PSO) approach", Springer Science+ Business, J IntellManuf , Vol.23,pp.1783-1793, doi: 10.1007/s10845-010-0481-5
25- Ibrahim Ahmed Saleh, Salha Raa'd Mahammead," Apply Particle Swarm Optimization Algorithm to Measure the Software Quality", Al-Rafidain Journal of Computer Science and Mathematics, Volume (12), Issue (2018) 1.
26- N. Prabhakaran, R. Nedunchelian," Oppositional Cat Swarm Optimization-Based Feature Selection Approach for Credit Card Fraud Detection", Computational Intelligence and Neuroscience(2023), https://doi.org/10.1155/2023/2693022.