Coal City University Journal of Science

ISSN: 2734-3758(Print), 2734-3766 (Online)

A publication of faculty of Natural and Applied Sciences, Coal City University, Enugu Nigeria.

CCU Journal of Science
Vol. 4, Issue 1, May, 2025

ISSN: 2734-3758(Print), 2734-3766 (Online)
https://ccuios.com

Criteria Related Techniques for Software Evaluation

Benjamin Ubokobong Effiong[1], E.E. Ogheneovo[2] and B. O. Eke[3]

[1, 2,3]Department of Computer Science, University of Portharcourt, River State, Nigeria

ABSTRACT

The paper examines a review of techniques and criteria for software evaluation. It reveals software evaluation as a serious aspect of software engineering process that involves choosing one software product amongst other alternatives. Techniques for software evaluation are discussed such as Analytic Hierarchy Process (AHP) Software Evaluation Technique, weighed average sum and Fuzzy Based Approach Software Evaluation Technique. The strength and weakness of each technique is presented to give a broad understanding of the techniques for software evaluation. Criteria to be considered for software evaluation are usually classified in several groups. Several groups of evaluation criteria are needed for software evaluation such as vendor, hardware/software, cost/benefits, security and reliability. It is concluded that **s**oftware evaluation is important as it enables Software companies and individuals/organizations that make use of software to make sure they provide software solutions that are reliable and can meet the needs of users.

Keywords: Software evaluation, Fuzzy Based Approach, Evaluation Criteria, Software Quality, Decision-Making Techniques

Introduction

Evaluation is a task, which results in one or more reported outcomes, evaluation is an aid for planning, and therefore the outcome is an evaluation of different possible actions and finally evaluation is goal oriented. The primary goal is to check results of actions or interventions, in order to improve the quality of the actions or to choose the best action alternative. Evaluation is dependent on the current knowledge of science and the methodological standards. Evaluation as an aid for software development has been applied since the last decade, when the comprehension of the role of evaluation within Human-Computer Interaction had changed. The activities "Task analysis", "Requirement specification", "Conceptual and formal design", "Prototyping", "Implementation" are each supplemented by an activity "Evaluation" which helps to decide progression to the next step. Software can be evaluated with respect to different aspects, for example, functionality, reliability, usability, efficiency, maintainability, portability (Benjamin, 2019).

Software evaluation is a serious aspect of software engineering process, choosing one software product amongst other alternatives requires a careful line of multiple actions through a well guided criteria-based procedure. A successful evaluation is not simply picking a product based on intuition. It involves a formal process, the right mixture of evaluators, and a specific quantifiable set of evaluation criteria. The process should include how to handle differences in scoring by the evaluators. The task of choosing a software component for a specific function in order to integrate it in a software system is a typical case of multi-criteria decision making that frequently occurs in Software Engineering. Consider a decision maker with a set of components to fulfill a function in a software system, for example, creating digital signatures on files. A number of decision factors will come into play such as functional suitability, security, performance efficiency, interoperability and costs. Some of these may pose conflicts: For example, increased security may come at the price of decreased performance efficiency or increased price. The decision maker has to follow a trustworthy and repeatable procedure to choose the component that best fulfils the objectives at hand (Benjamin 2019, Onyema et al, 2025). Software can be evaluated with respect to different aspects, for example, security functionality, reliability, usability, efficiency, maintainability, portability using different techniques or methods. In earlier times evaluation of software took place at the end of the developing phase, using experimental designs and statistical analysis. Evaluation is nowadays used as a tool for information gathering within iterative design: "Explicit human-factors evaluations of early interactive systems (when they were done at all) were poorly integrated with development and therefore ineffective. They tended to be done too late for any substantial changes to the system still be feasible and, in common with other human-factors contributions to development, they were often unfavourably received. Instruments for evaluation are not primarily used for global evaluation of an accomplished product, but these instruments are applied during the development of a product. Indeed, most experts agree nowadays that the development of usable software can only be done by a systematic consideration of usability aspects within the life-cycle model. One prominent part is the evaluation of prototypes with respect to usability aspects, employing suitable evaluation techniques in order to find usability errors and weaknesses of the software at an early stage (Benjamin, 2019).

Software Evaluation Techniques

Software evaluation is multi-criteria decision making problem that refers to making preference decisions over the available alternatives (Figueira, Mousseau& Roy, 2016). At this point, the various software evaluation techniques are discussed and their strengths and weaknesses are examined.

1. **Analytic Hierarchy Process (AHP) Software Evaluation Technique:** AHP has been widely used for evaluation of the software packages. AHP has been identified as an important approach to multi-criteria decision making problems of choice and prioritization. AHP is based on a hierarchical framework of criteria. The upper level deals with the goal of the selection process. The next level defines the major factors which are subdivided into their constituents in lower levels of hierarchy. The bottom level contains the alternatives to be analyzed. Local priorities are established for each

factor on a given level with respect to each factor on the level immediately above it. This is done by pair-wise comparison between the factors at each level. The application of AHP to the evaluation of software package has been successfully applied in many research studies (Ramírez-Nafarrate, Marín-Raventós and Montoya-Torres, 2019).

Strengths of Analytical Hierarchy Process (AHP): The strength of this process are:

i. AHP enables decision makers to structure a decision making problem into a hierarchy, helping them to understand and simplify the problem.
ii. It is flexible and powerful tool for handling both qualitative and quantitative multi-criteria problems.
iii. AHP procedures are applicable to individual and group decision making.

Weaknesses of Analytical Hierarchy Process (AHP): The weaknesses are:

i. AHP is time consuming because of the mathematical calculations and number of pair-wise comparisons that increases as the number of alternatives and criteria increases.
ii. The decision makers need to re-evaluate alternatives when the number of criteria or alternatives are changed.
iii. Ranking of alternatives depends on the alternatives considered for evaluation hence adding or deleting alternatives can lead to changes in the final rank (Koziolek, 2009).

2. **Weighted Average Sum (WAS) Software Evaluation Technique:** Another technique used for evaluation of software package is the weighted scoring method. In this method weights and rating scales are assigned to each criterion. The weight reflects the relative importance of each of the criteria while the rating scale indicates how easily each package is able to meet the specific criterion. The rating scales are then multiplied by weight factor of each criterion. Using this scheme a score is calculated for every criterion for each tool. These scores are then totaled to produce a score for each criteria category and the average is also computed. Finally, the categorical scores are compared to determine the highest (Benjamin, 2019).The overall score for an alternative, is the weighted sum of the criteria utilities aggregated across the hierarchy of objectives. This approach combines objective evidence measured in specific scales, subjective assessment represented in explicitly defined scenario-specific utility functions, and relative weights across the goal hierarchy. As such, it is a flexible model, but it requires a profound understanding of the intricacies of decision making scenarios. Furthermore, a careful distinction between the key concepts of evidence, utility, and weighting is expected from the decision maker (Rafique and Mišić, 2019).

Strengths of Weighted Average Sum Software Evaluation Technique

i. Main advantage of WAS is its ease of use.

Weaknesses of Weighted Average Sum Software Evaluation Technique:

i. Weights to the attribute are assigned arbitrary and it is very difficult to assign weight when number of criteria is high.

ii. To obtain a score using this method a common numerical scaling is required.
iii. Difficulties emerge when WAS is applied to multi-dimensional MCDM problems.

Fuzzy Based Approach Software Evaluation Technique: A fuzzy based approach for software evaluation has been used in different studies. This technique is used when performance rating and weights cannot be given precisely (Goztepe, Karaman, Catalkaya, 2015, Tran et al, 2024). In such cases the fuzzy set theory is used to model the uncertainty of human judgments and such problem is known as fuzzy multiple criteria decision making (FMCDM). In daily life people usually make decisions based on imprecise or uncertain knowledge rather than some computer algorithms that require exact data. Fundamental elements of fuzzy logic are human language rules. The fuzzy systems convert these rules to their mathematical equivalents. Outcomes of fuzzy systems are more accurate representations then logic. The real world is not precise and certain. Thus, fuzzy sets handle uncertainty by reducing it and develop precise conclusions for real life problems (Mansoor, Streitferdt and Fubl, 2015, Sucharitha et al, 2023). The general architecture and components of a fuzzy rule based inference system is shown in Figure 2.1. The main modules of a fuzzy rule based system are fuzzification, fuzzy rules, inference system, data base and defuzzification.

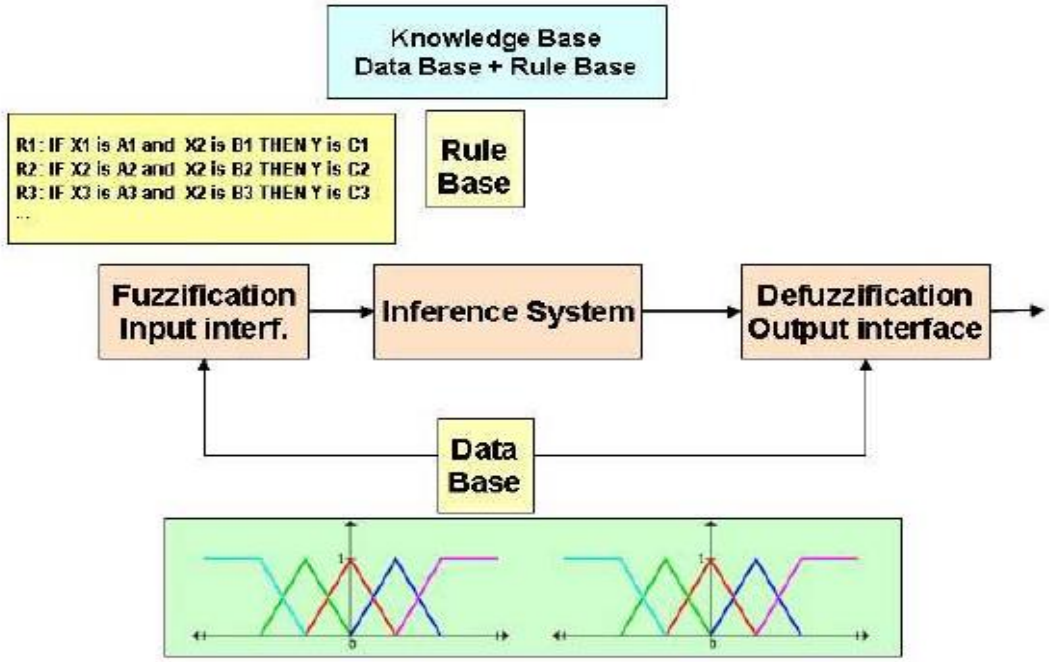

Figure 2.1: Fuzzy based approach model (Source: Goztepe, Karaman and Catalkaya, 2015)

Fuzzy rules consist of IF-THEN statements. Proposed model consisted of number of rules, and they make a group which forms the basis of ERP evaluation. The following fuzzy rules have been taken with the combination of linguistic variable values for ERP evaluation process.

Defuzzification: Fuzzy outputs need to be converted into a scalar output quantity in a fuzzy ruled model. The nature of the action can be evaluated by the system by this way. The

converting process of the fuzzy output is called defuzzification. Whole fuzzy outputs of the system are aggregated with a union operator before an output is defuzzified. Standard defuzzification methods are bisector, centroid, mean value of maximum values, smallest value of maximum values and largest value of maximum (Mansoor et al., 2015, Karpagam, Christy, and Edeh, 2024).

Strengths of Fuzzy Based Approach Software Evaluation Technique:

i. The decision makers can use linguistic terms to evaluate alternatives easily and intuitively.
ii. It improves decision making procedure by accommodating the vagueness and ambiguity occurred during human decision making.

Weakness of Fuzzy Based Approach Software Evaluation Technique:

i. Difficult to compute fuzzy appropriateness index values and ranking values for all alternatives.

Software Evaluation Criteria

Criteria to be considered for software evaluation are usually classified in several groups. Quality characteristics of the software package such as functionality, reliability, usability, efficiency, maintainability, and portability have been used as evaluation criteria group in several studies. The following are used as criteria for evaluation of software A and B in Benjamin (2019):

(1) Vendor
(2) System requirements (hardware and software)
(3) Cost and benefits of the software packages

Vendor: *Vendor assessment* is an evaluation and approval process that businesses can use to determine if prospective vendors and suppliers can meet their organizational standards and obligations once under contract. The end goal is to secure a low-risk, best-in-class vendor and supplier portfolio. Vendors and suppliers both furnish services or goods, but there is a distinction: The term *vendor* applies to business-to-business (B2B) and business-to-consumer (B2C) sales relationships, while *supplier* applies only to B2B relationships **(**Goepel, 2018**).**

Benefits of Vendor Assessment: Although the vendor assessment process can be challenging, the benefits include finding low-risk sources for high-quality goods and services, as well as the development of mutually beneficial, long-term business relationships (Kumar & Singh, 2020).Additional benefits of vendor assessment include the following:

a) **Risk Mitigation:** By carefully vetting vendors and suppliers, you can lower the regulatory, contract, and security risks of working with entities outside your company.
b) **Lower Regulatory Compliance Risk:** Confirm compliance with the laws, regulations, and standards that apply to your business. If your vendor is in another country or you sell to another country, check those countries' legal requirements, too. For example, if you do business in the European Union, the General Data Protection Regulation (GDPR) applies.

c) **Scope of Service and Contract Compliance:** Conduct a legal review of contract terms, nondisclosure agreements (NDAs), or partnership agreements to ensure that you've set favorable conditions and mitigated any risks.

d) **Decrease Security and Cyber Risk:** The threat of cyber risk grows as we increasingly move our work to the cloud. Therefore, it's crucial to protect all your customer and company data. Your selection process should focus on assessing a third-party vendor or supplier to store your data. First, identify and evaluate the type of data that vendors and suppliers may need to access, and decide whether they need to access all, some, or none of it to do their work. It's crucial to ensure the vendor takes the proper measures to encrypt and protect your data. Once you identify potential partners, have them complete a questionnaire that thoroughly covers security management system details.

2. Hardware/Software Evaluation Criteria: Selecting hardware and software for implementing information system in an organization is a serious and time-consuming process that passes through several phases **(**Zhu, Ali, & Chu, 2021).The main steps of the selection process are listed below:

1. **Requirement Analysis: –** System configuration requirements are clearly identified and a decision to acquire the system is taken in this step.
2. **Preparation of Tender Specifications: –** After studying the feasibility and deciding upon the configuration, tender documents are prepared for the benefit of vendors to clarify the details of various specifications.

Hardware Selection Factors to Consider

1. Hardware must support current software as well as software planned for procurement over the next planning interval (1 year, 18 months, 3 years).
2. Hardware must be compatible with existing or planned networks.
3. Hardware must be upgradeable and expandable to meet the needs of the next planning interval.
4. Hardware warranties must be of an appropriate length.
5. Hardware maintenance must be performed by (local/remote vendor, in-house personnel).
6. Whenever feasible, hardware standards will dictate procurement of like brands and configurations to simplify installation and support.
7. Routine assessments of installed infrastructure will feed an upgrade/replace decision process.

Software Selection Evaluation Criteria

Typically, software evaluation criteria include the following things

End-user requirements: When evaluating the software end-user's requirements are considered to be the first priority. The software must be designed in a way to facilitate the users of all skill levels. Other than that, the software should clearly tackle the problem that it

is designed to do. For example, if a software targets the audience that suffer from disabilities then it should have features to entertain such an audience. Like having built-in letter templates and supporting voice typing are must-haves for the software (Agh, &Dombi, 2019).

Functionality: Another key aspect of software evaluation is checking the functionality of the software. The software must perform the function that is required by the user. Having other related features in the software also improves the evaluation score of the software. For example, having the feature to create graphs and charts in a spreadsheet application improves the quality by a lot (Nahid and Susmita, 2019).

Performance: The performance of the software also plays a huge role in the software evaluation. The performance of the software is compared with the existing software models and the results are stored in the form of benchmarks for future use. The software needs to be perfectly optimized in order to deliver maximum performance which makes the optimization a key feature for the app.

User-Friendly: The software with a user-friendly interface is always better. The ease of use and built-in help panels add more value to the app in terms of user-friendliness.

3. Cost-benefits Software Evaluation Criteria: Cost–benefit analysis (CBA), sometimes also called benefit–cost analysis, is a systematic approach to estimating the strengths and weaknesses of alternatives. It is used to determine options which provide the best approach to achieving benefits while preserving savings in, for example, transactions, activities, and functional business requirements. A CBA may be used to compare completed or potential courses of action, and to estimate or evaluate the value against the cost of a decision, project, or policy. It is commonly used to evaluate business or policy decisions (particularly public policy), commercial transactions, and project investments. For example, the U.S. Securities and Exchange Commission must conduct cost-benefit analyses before instituting regulations or deregulations **(**Yadav & Mehta, 2020***).***

Cost–benefit analysis is often used by organizations to appraise the desirability of a given policy. It is an analysis of the expected balance of benefits and costs, including an account of any alternatives and the *status quo*. CBA helps predict whether the benefits of a policy outweigh its costs (and by how much), relative to other alternatives. This allows the ranking of alternative policies in terms of a cost–benefit ratio. Generally, accurate cost–benefit analysis identifies choices which increase welfare from a utilitarian perspective. Assuming an accurate CBA, changing the *status quo* by implementing the alternative with the lowest cost–benefit ratio can improve efficiency. Although CBA can offer an informed estimate of the best alternative, a perfect appraisal of all present and future costs and benefits is difficult; perfection, in economic efficiency and social welfare, is not guaranteed. The value of a cost–benefit analysis depends on the accuracy of the individual cost and benefit estimates **(**Sousa and Rocha, 2018).

2.5.4 Security Evaluation Criteria

One of the target of this work is to investigate evaluation methods related to software security, which is a critical extra-functional property directly and significantly impacting different development stages (i.e., requirements, design, implementation and testing). Security in software can be defined as engineeringsoftware so that it continues to function correctly undermalicious attack. The high diversity of components and complexity of current systems makes it almost impossible to identify, assess and address all possible attacks and aspects of security vulnerabilities (Ransome&Misra, 2022). An extensive body-of-knowledge already exists on softwaresecurity and many studies have already investigated security assessment as well. For example, the U.S. National Institute of Information Standards and Technology (NIST)developed a performance measurement guide for information security and in particular how an organization through the use of metrics can identify the adequacy of controls, policies and procedures. The approach is mostly focused on the level of technical security controls in the organization rather than thetechnical security level of specific products (National Institute of Standards and Technology, 2021).

2.5.5 Software Reliability Evaluation Criteria

Software Reliability can be defined as the probability of failure-free software operation for a specified period of time in a specified environment. Software Reliability is a very important factor that affects system reliability. It is quite different from hardware reliability because it shows the design perfection, rather than manufacturing perfection. The high complexity of most software is the major contributing factor of Software Reliability problems. Various approaches can be used to improve the reliability of software. Software has no shape, color, material, mass. It cannot be seen or touched, but it has a physical existence and is crucial to system functionality. Software can also have small unnoticeable errors or drifts that can culminate into a disaster. On February 25, 1991, during the Golf War, the chopping error that missed 0.000000095 second in precision in every 10th of a second, accumulating for 100 hours, made the Patriot missile fail to intercept a scud missile. 28 lives were lost. While any system with a high degree of complexity, including software, will be hard to reach a certain level of reliability, system developers tend to push complexity into the software layer, with the rapid growth of system size and ease of doing so by upgrading the software. For instance, large next-generation aircraft will have over one million source lines of software on-board; next-generation air traffic control systems will contain between one and two million lines; the upcoming international Space Station will have over two million lines on-board and over ten million lines of ground support software; several major life-critical defense systems will have over five million source lines of software (Ransome and Misra, 2022).

Conclusion

Software evaluation is important as it enables Software companies and individuals/organizations that make use of software to make sure they provide software solutions that are reliable and can meet the needs of users. This cannot be achieved without an effective decision support system for software evaluation. The need to evaluate the reliability of software system has become clearer in this era of intensive software application

in different sectors of life. Many papers provide a preferred list of evaluation criteria for evaluation of specific software package; however, a lack of common list is apparent. Software evaluation criteria may differ from one evaluator to another. The exact meaning of a criterion is open to the evaluator's own interpretation. Sometimes the terminology used by author(s) for a criterion in one literature is different than another author(s) for the same criterion. This may lead to ambiguity and gives unclear picture to the evaluator. There is need to develop a framework including a software selection methodology, an evaluation technique, an evaluation criterion, and a system to support evaluation and selection of any software package. A common evaluation technique than can be adopted by software evaluators to determine the effectiveness of the software is weighted sum evaluation technique. In this approach values or weights are assigned to the criteria used in determining the effectiveness of the software and the total weight is used to determine which is more effective than the other.

Recommendations

The following recommendations are offered based on the study:

1. The criteria utilized in making decision on the effectiveness of a software should be expanded to include more aspects so as to improve the reliability information.
2. Proper testing and debugging should be done before software systems are published or placed in the market.
3. More criteria categories should be added to boost software evaluation and selection.
4. Trial versions of software are important for users to assess the effectiveness of the software.

REFERENCES

Agh, T., &Dombi, J. (2019). "Multi-criteria decision analysis (MCDA)-based decision support framework for enterprise software evaluation." *European Journal of Operational Research, 276*(2), 678-689.

Benjamin, E. U. (2019).*An improved decision support system for software evaluation using weighted sum technique*. International Journal of Trend in Scientific Research and Development, 3(2), 1416-1419. IJTSRD

Figueira, J. R. ,Mousseau, V. , & Roy, B. (2016). ELECTRE Methods. In S. Greco, M. Ehrgott, & J. R. Figueira (Eds.), *Multiple Criteria Decision Analysis: State of the Art Surveys* (pp. 155–185) (Eds.). New York, NY: Springer New York

Goepel, K. D. (2018).*Implementation of an online software tool for the Analytic Hierarchy Process (AHP-OS)*. International Journal of the Analytic Hierarchy Process, 10(3), 469-487. IJ-AHP

Goztepe, K., Karaman, M., Catalkaya, H. (2015). Fuzzy rule-based approach for Enterprise resource planning (ERP). Software evaluation journal of naval science and engineering2015, vol.11, no.1, pp.34-52

Koziolek, H. (2009). Performance Evaluation of Component-based Software Systems: A Survey. ABB Corporate Research, Industrial Software Systems, Wallstadter Str. 59, 68526 Ladenburg, Germany

Kumar, R., & Singh, S. P. (2020).*Multi-criteria decision-making techniques for software evaluation: A comparative study*. Expert Systems with Applications, 161, 113-134.IJ-AHP

Karpagam, V; Christy, S; and Edeh, MO. (2024). Deep Learning-Based Water Quality Index Classification Using Stacked Ensemble Variational Mode Decomposition. Environmental Research Communication, 6 (2024) 065005. https://doi.org/10.1088/2515-7620/ad549e

Mansoor, A., Streitferdt, D. and Fubl, F. (2015) Fuzzy based evaluation of software quality using quality models and goal models. International Journal of Advanced Computer Science and applications. Vol.6, No. 9. pp. 265-273

Nahid, A. &Susmita, K. (2019) Review Paper on Various Software Testing Techniques & Strategies Global J. Computer Science and Technology: Software & Data Engineering Volume 19 Issue 2 pp. 975-990.

National Institute of Standards and Technology. (2021). *Secure software development framework (SSDF) version 1.1* (NIST Special Publication 800-218). Retrieved from https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-218.pdfNIST Publications

Onyema EM; Suguna S.K; Sundaravadivazhagan B; Rutvij, H.J; Ugwuja, NE; Chinecherem ED, Kumari K.S; A secure routing protocol for improving the energy efficiency in wireless sensor network applications for industrial manufacturing, Next Energy, 7, 2025, 100219, https://doi.org/10.1016/j.nxener.2024.100219

Rafique, S., &Mišić, V. B. (2019). "A framework for evaluating cloud-based software services using decision support systems." *Information and Software Technology, 113*, 115-127.

Ramírez-Nafarrate, A., Marín-Raventós, G., & Montoya-Torres, J. R. (2019). "A decision support system for software evaluation based on Analytical Hierarchy Process (AHP)." *Journal of Decision Systems, 28*(2), 87-102.

Ransome, J. and Misra, A. (2013). Core Software Security: Security at the Source. *CRC Press. pp. 140–3.* ISBN 978-1-4665-6095-6.

Sousa, M. J., & Rocha, Á. (2018). "Software evaluation model using decision support techniques: An application in project management tools." *Procedia Computer Science, 138*, 497-504.

Sucharitha M., R Thomas, B. Jyothi, EM Onyema, G Bekele, "Design and Fabrication of Metallic-Conductive Polymer-Based Hybrid Film Interconnections for Stretchable Electronic Devices", International Journal of Polymer Science, vol. 2023, Article ID 1392115, 8 pages, 2023. https://doi.org/10.1155/2023/1392115

Tran, T.A., Onyema, E.M., & Abougreen, A.N. (Eds.). (2024). Leveraging the Potential of Artificial Intelligence in the Real World: Smart Cities and Healthcare (1st ed.). CRC Press. (Taylor and Francis/Routledge) https://Doi.Org/10.1201/9781032667508 ISBN 9781032667485

Yadav, D., & Mehta, M. (2020). "A decision support framework for ERP software evaluation using Delphi and AHP methods." *Computers in Industry, 123*, 103326.

Zhu, L., Ali, M. H., & Chu, F. (2021).*AI-based decision support system for enterprise software selection: An integrated fuzzy-AHP and TOPSIS approach*. Journal of Intelligent & Fuzzy Systems, 40(5), 1071-1085.