applied sciences

MDPI

Article

OffshoringDSS: An Automated Tool of Application Maintenance Offshoring

Hanif Ur Rahman [1,*], Mushtaq Raza [1,*], Asaad Alzayed [2], Palwasha Afsar [1], Abdullah Alharbi [3], Wael Alosaimi [3] and Umar Hayat Khan [1]

[1] Department of Computer Science, Abdul Wali Khan University Mardan, Mardan 23200, Pakistan
[2] Computer Science and Information System Department, Public Authority for Applied Education and Training, PAAET, Kuwait City 13092, Kuwait
[3] Department of Information Technology, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia
* Correspondence: hanif.maidan@gmail.com (H.U.R.); mushtaq@awkum.edu.pk (M.R.)

Abstract: The rapid spread of the internet over the last two decades has prompted more and more companies to deploy their work internationally. The offshoring strategy enables organizations to cut down costs, boost shareholder value, acquire a competitive advantage, reduce cycle time, increase workforce flexibility, generate revenue and focus on their core business. The number of worldwide software development projects has increased due to globalization. Global Software Development (GSD) projects are forecast to grow by 20% to 30% in countries like India and China. The outsourcing experts choose one of the suitable models from the available global delivery options to deliver services in the global software paradigm. However, adopting the appropriate model for application maintenance is a complicated process. In addition, the right model is selected based on various influencing factors, type of the project and client requirements. Additionally, sufficient domain expertise is necessary for the decision making of offshore outsourcing. Currently, there is no dynamic and automated tool for the decision making of application maintenance offshoring. Therefore, this study presents an Offshoring Decision Support System (OffshoringDSS), an automated and novel tool to make the offshoring decisions of application maintenance. The suggested tool is based on the Analytic Hierarchy Process (AHP) technique. The tool automatically performs all the calculations involved in the decision making and ranks the sourcing models.

Keywords: application maintenance; automated tool; decision support system; decision making; offshoring; outsourcing; model; software maintenance

check for updates

Citation: Rahman, H.U.; Raza, M.; Alzayed, A.; Afsar, P.; Alharbi, A.; Alosaimi, W.; Khan, U.H. OffshoringDSS: An Automated Tool of Application Maintenance Offshoring. *Appl. Sci.* **2022**, *12*, 10913. https://doi.org/10.3390/app122110913

Academic Editor: Antonio Sarasa Cabezuelo

Received: 12 September 2022
Accepted: 25 October 2022
Published: 27 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The rapid spread of the internet over the last two decades has prompted more and more companies to deploy their work internationally and to outsource their IT services to specialized vendors. Outsourcing is a contract that hires subcontractors to develop and maintain software. In global outsourcing, on the other hand, a company engages a distant provider. It is a contract that allows quality software to be developed across national borders. Offshoring, offshore outsourcing, and global outsourcing are terms used interchangeably. The offshoring strategy enables organizations to cut down on costs, boost shareholder value, acquire a competitive advantage, reduce cycle time, increase workforce flexibility, generate revenue, and focus on their core business [1,2].

The number of worldwide software development projects has increased due to globalization. Likewise, the number of offshore projects is likely to increase over time. Global Software Development (GSD) projects are forecast to grow by 20% to 30% in countries such as India and China. Software companies are moving to Eastern Europe and Asia because of the lower labor costs in these countries. Other benefits of using GSD include faster time to market by leveraging the time zone difference and the use of virtual teams with diverse

skills. There are many other benefits of using GSD, including reduced time to market due to time zone variations and the use of virtual teams with diverse talents [3–6].

IT outsourcing has subcategories such as software maintenance, software development, business process and infrastructure outsourcing. The four basic types of software maintenance are corrective, adaptive, perfective and preventive. Software maintenance is considered to be one of the most time-consuming and expensive phases in the software development lifecycle. The upgrading process of the existing program to fix bugs, adapt to new technologies and platforms, and adapt to a new environment is known as software maintenance. The cost of the software maintenance phase accounts for approximately 60% of total IT expenditures. Thus, an offshore outsourcing strategy is adopted to reduce the high cost of application maintenance. It is one of the common business techniques for developing and maintaining quality applications at a low price. Organizations save between 20% and 50% when outsourcing software development and offshoring maintenance [7,8].

The outsourcing experts choose one of the suitable models from the available global delivery options to deliver services in the global software paradigm. However, adopting the appropriate model is a complicated decision-making process, because the selection of the right model is based on various influencing factors, type of project and client requirements. Additionally, the process of offshore outsourcing requires knowledge and experience in the same domain.

The existing literature indicates that a minimum number of studies have concentrated on offshore outsourcing decisions. Particularly scarce research studies addressing the decision-making process of the offshoring decision of application maintenance can be found in the literature, such as [9–19]. However, we have not observed any research study that presented an automated and novel tool to make the offshoring decisions of application maintenance. Currently, IT professionals use their experience as well as frameworks, conceptual models or multi-criteria decision-making (MCDM) models to assess the projects and choose one of the available sourcing alternatives. Hence, the current study presents Offshoring Decision Support System (OffshoringDSS), an automated and novel tool to make the offshoring decisions of application maintenance. The suggested tool is able to make the sourcing decision based on the Analytic Hierarchy Process (AHP) technique that automatically performs all the calculations according to the input provided and ranks the sourcing models. To the best of my knowledge, no such automated tool exists to make sourcing decisions. The developed tool, i.e., OffshoringDSS, is freely available online at http://47.243.206.42/dmm/.

2. Study Background

The literature review shows that several researchers [20–30] have proposed different tools as well as models that are used in the GSD context. Rahman et al. [15] suggested a model based on the factors that are used to help IT experts in taking decisions in the global delivery context. The proposed model was developed using the AHP technique, which identifies and ranks the available options and helps IT experts to choose one of the delivery models. Similarly, Ikram et al. [11] performed an empirical investigation to derive a dataset, and then a model was built based on the identified dataset using machine learning techniques. Using the proposed model, the client proposal is evaluated for software maintenance in the offshore outsourcing context.

Raza and Faria [31] presented ProcessPAIR, a revolutionary tool used by developers to analyze their performance data. Additionally, it is used to automatically identify and prioritize the performance issues as well as the basic reasons in the development of software. Beecham et al. [32] proposed the Global Teaming Decision Support System, which helps the software managers to navigate the numerous suggestions in the GSD literature and the Global Teaming Model. It is an interactive application which gathers data from the development organization and customizes global teaming model approaches to fit specific businesses and the goals of a company. The suggested tool utilizes standardized procedures that are contextualized to meet the needs of specific companies. Practitioners are capable of

finding out the procedures essential to effective global software development and may go a step further by engaging with the proposed model to produce their own personalized list of recommendations.

To enable virtual software design sessions, Cataldo et al. [33] developed the CAMEL tool. The basic challenges found in the literature are addressed by the tool, such as the sharing of information, the resolution of conflict and consensus building between geographically dispersed designers, as well as providing sufficient drawing surfaces, developing understanding and maintaining focus during discussions and storing all design information for graphical representation. Hattori and Lanza [34] created a tool called Syde to help developers to restore the team awareness by disseminating information regarding the change and any dispute across developers' workstations. This method recognizes source code alterations as first-class entities to track the precise evolution of the project with several developers. Therefore, Syde offers developers detailed information about the changes. The Internet-Based Inspection System (IBIS) was developed by Lanubile et al. [35] as a web-based tool that is used to assist inspection teams that are geographically distributed. To eliminate coordination difficulties, IBIS utilizes a reengineered inspection methodology based on insights from empirical study on software inspection. Further, they discussed the use of tools during the process of inspection and presented their experience of using the IBIS tool for distributed software inspection as enabling infrastructure.

TAMRI is a planning technique developed by Lamersdorf and Münch [36] for calculating task assignments based on numerous characteristics and weighted project goals. In its implementation, it combines the distributed systems notion with Bayesian networks. Modifying the underlying Bayesian network allows the tool to be adapted to unique organizational environments. They presented and discussed the task allocation approaches, three tool application scenarios and tool implementation demos.

When a project is offshored, IT specialists pick the appropriate model among the available sourcing models to deliver the services. The appropriate model is chosen based on the projects requirements and customer preferences. Therefore, IT experts use their knowledge to evaluate and rank the sourcing options. Similarly, the experts can use an offshoring framework or an MCDM model is adopted to make the sourcing strategy. The literature, on the other hand, shows the need for an automated tool. A model that receives input from experts and automatically ranks the sourcing alternatives of application maintenance. Therefore, the present study introduces the Offshoring Decision Support System (OffshoringDSS), an innovative and automated decision-making tool. The suggested tool uses the AHP technique to make sourcing decisions and performs all calculations automatically based on the input.

The rest of the paper is organized as follows. Section 3 presents the proposed research method. Section 4 presents construction of automated tool that includes tool architecture, AHP technique for MCDM models and algorithms along with their flowcharts, and case study and discussion. While Section 5 contains limitations, and Section 6 outlines the conclusions and future work.

3. Proposed Method

The current section discusses the research method of developing an automated tool for application maintenance. Furthermore, the four steps as shown in Figure 1, are used to create the suggested tool.

The most important factors of application maintenance were derived by conducting two systematic literature reviews, as reported in [12,13]. Both literature reviews presented a list of 15 influencing factors, as given in Table 1, that will be used to assess the projects for making decisions regarding application maintenance.

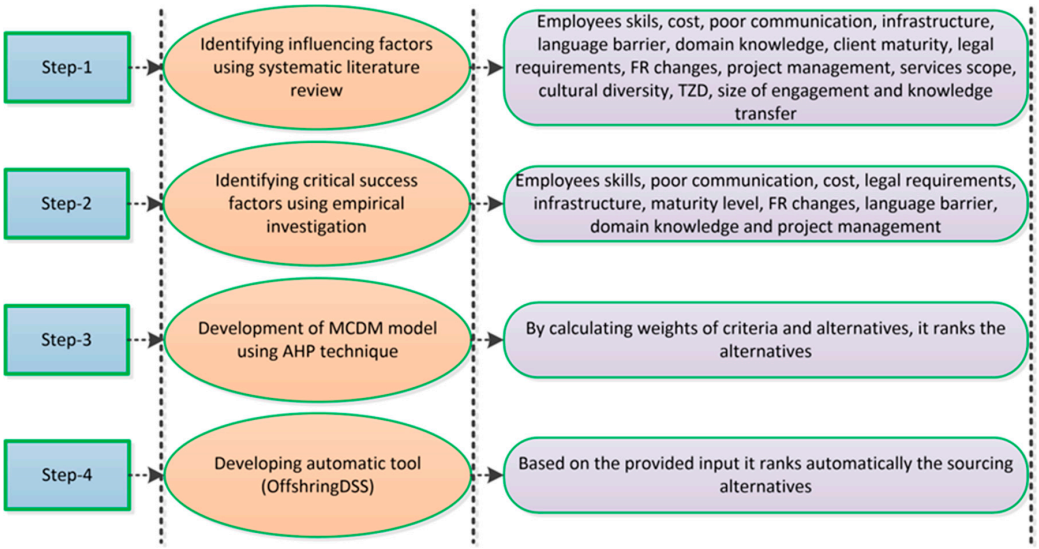


Figure 1. Proposed method of the OffshoringDSS.

Table 1. List of influencing factors.

Influencing Factors			
1. Employee skills	5. Infrastructure	9. Cultural diversity	13. Domain knowledge
2. Cost	6. Maturity level	10. Project management	14. Service scope
3. Legal requirements	7. Language barrier	11. Knowledge transfer	15. Project duration
4. Poor communication	8. Frequent requirements changes	12. Time zone difference	

The identified list of influencing factors was further reviewed and validated by 93 outsourcing experts using an empirical study, as published in [14]. Table 2 shows that among the 15 factors, 10 factors were identified as critical success factors. The critical factors will be used to evaluate the projects for sourcing decisions.

Table 2. Factors ranked as critical success factors.

Critical Success Factors		
1. Employee skills	5. Infrastructure	9. Domain knowledge
2. Poor communication	6. Maturity level	10. Project management
3. Cost	7. Frequent requirement changes	
4. Legal requirements	8. Language barrier	

Using the AHP approach, an MCDM model was created based on identified critical success factors, as published in [15]. The model gives the ranking of alternatives for the application maintenance based on the provided priorities by the decisions makers to the factors and sourcing models.

The proposed sourcing frameworks [14,15] are employed to assess projects that help the experts to choose one of the suitable alternatives. However, using the frameworks, projects are evaluated on the basis of "Yes" and "No" or "Low", "Medium" and "High"; for example: is "legal requirements" high in the project (Yes or No)? Similarly, the decision is taken whether the impact of the legal requirements is high or medium or low. However,

it is not always possible to make appropriate sourcing decisions on the basis of "Yes" or "No" and "Low", "Medium" or "High", as shown in these presented frameworks. More importantly, sufficient domain expertise is necessary to evaluate the projects based on the presented frameworks.

Furthermore, an MCDM model was presented [15] to solve the afore-mentioned limitations of the suggested frameworks. Although the presented MCDM model ranks the alternatives based on the weights assigned to factors by experts but the model lacks automation. It does not automatically calculate the various options according to the input provided. Furthermore, it is based on complex formulas and equations that need to be solved for decision making.

The current study intends to improve the process of decision-making concerning application maintenance by addressing the limitations of the previously proposed frameworks, as presented in [14,15]. Furthermore, the present study also introduces automation into the previously developed MCDM model [15]. Thus, a novel and automated tool for offshoring decisions using the JavaScript language is developed. The proposed automated tool is developed using the JavaScript language by implementing the AHP techniques. It would be used as a decision support system for the sourcing decision making of application maintenance. It automatically ranks the three alternatives, namely the onshore model, nearshore model and offshore model, according to the provided input. It can be adopted by both vendors and clients to make suitable sourcing decisions. Vendors may utilize the developed tool to analyze the projects in order to adopt an effective sourcing strategy that increases the likelihood of future business success. Clients may assess their projects and make the best possible selection of alternatives based on their priorities, needs and preferences.

The suggested tool consists of three major components, which are the weight calculation of criteria, the weight calculation of available alternatives, computing the consistency ratios, and ranking the sourcing alternatives. Figure 2 shows the whole working mechanism of the suggested OffshoringDSS.

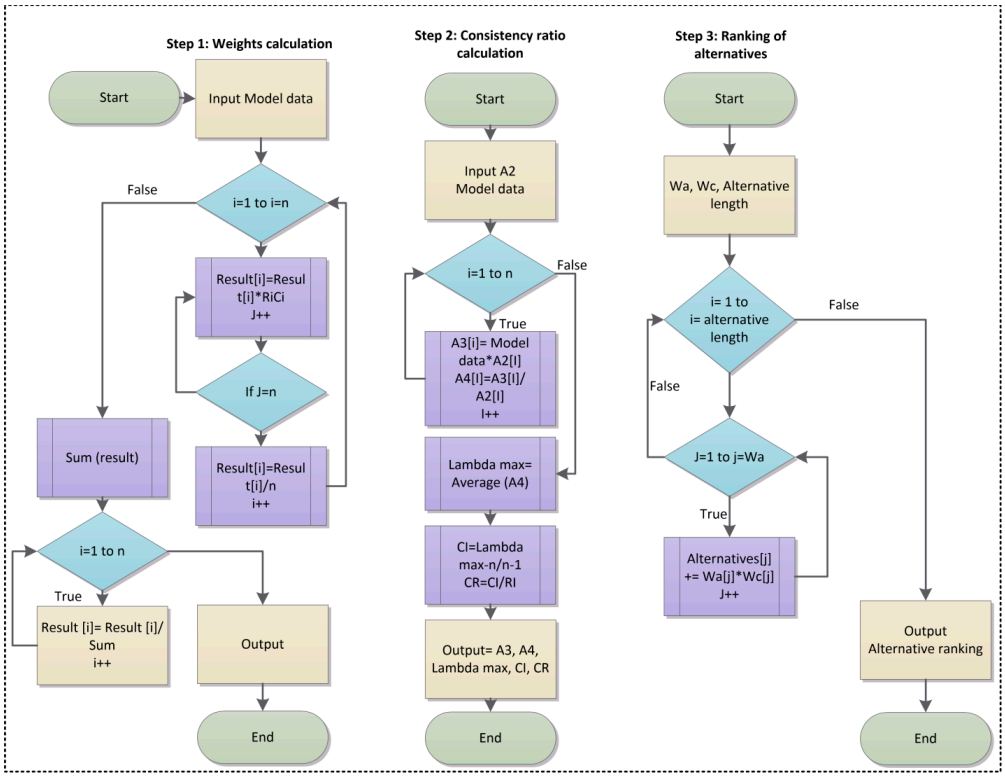


Figure 2. Working procedure of the suggested OffshoringDSS.

4. Construction of Automated Tool

This section outlines the steps required to develop the proposed tool. The steps include architecture of proposed tool, the AHP technique for the MCDM model, the designed algorithms for the tool and the respective flowcharts of the developed algorithms. The following subsections provide a detailed description of these steps.

4.1. Architecture of the Tool

The architecture of the proposed OffshoringDSS is discussed in this section. Figure 3 shows a high-level overview of the artifact that describes the architecture of the proposed tool. The figure shows that the main components are calculation of the criteria weights with respect to the goal, the alternative weights with respect to the criteria and the calculation of consistency ratio with Lambda max value. Finally, the criteria weights and the alternative weights are combined to obtain the total weights, showing the ranking of the sourcing alternatives.

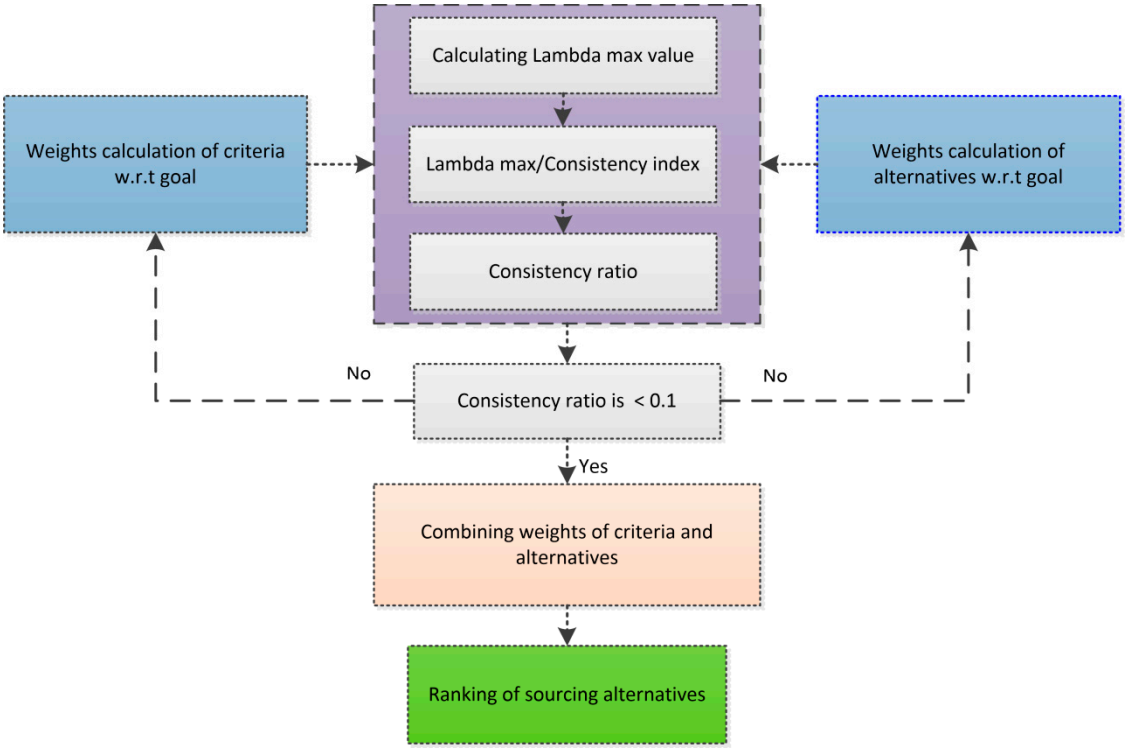


Figure 3. OffshoringDSS Architecture.

Three different algorithms are embedded in the OffshoringDSS to perform all the required calculations for the sourcing decision-making of application maintenance. Algorithm 1 is used to compute the criteria weights and alternative weights, Algorithm 2 is used to find consistency ratios that include Lambda max value, and Algorithm 3 ranks the alternatives. The tool would help in making the sourcing decisions as follows. First, values are assigned to criteria based on the purpose and nature of the project. The tool calculates the criteria weights accordingly. The criteria used by OffshoringDSS, known as critical success factors, include the employees skills, legal requirements, cost, maturity level, poor communication, infrastructure, domain knowledge, frequent requirements' changes, project management and language barrier. Second, IT specialists obtain the weights of alternatives by assigning values to alternatives with respect to criteria. The scores to alternatives with respect to each factor are assigned based on their impact or effect on the sourcing decision. More importantly, the IT expert needs prior experience and knowledge of outsourcing, sourcing decisions and global delivery to make effective sourcing decisions.

Finally, the tool ranks the available models for sourcing decisions. In addition, the tool also calculates the value of the lambda max.

4.2. AHP Technique for MCDM Model

The present study introduces the AHP-based decision support system that accepts expert input and ranks the available models accordingly. Decision-making that uses multiple criteria is known as multi-criteria decision-making. The AHP technique has various applications in fields like engineering, economics and management, etc. It mainly comprises two subcategories, namely Multi-Attribute Decision Making (MADM) and Multi-Objective Decision Making (MODM). The MADM has discrete elements and options or alternatives with limited numbers, while MODM includes continuous elements and unlimited alternatives. The two names, MADM and MCDM, are used interchangeably in the literature. MODM features continuous variables and limitless possibilities, whereas MADM has discrete elements and a limited number of options. Researchers have suggested a variety of MCDM approaches in recent years, including Elimination and Choice Expression Reality (ELECTER), Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), Step-Wise Weight Assessment Ratio Analysis (SWARA), Gray Decision Making (GDM), AHP and Analytical Network Process (ANP) [37–40].

Thomas Saaty introduced AHP in 1980, one of the most important MCDM techniques. It is considered a precise and accurate technique to rank and prioritize variables. It is easy to use and offers a flexible multi-criteria decision-making technique. A complex problem is represented using the hierarchical structure of AHP, which simplifies the problem by decomposing it into the sub-problems. AHP systematically arranges both elements, such as intangible and tangible elements, resulting in a structured and relatively simple, as well as easy, solution to multi-criteria problems. This method works in areas where risk and uncertainty are accompanied by intuition, logic and irrationality. The problem can involve significant social, economic, political and technical quantities as well as a range of objectives, criteria and options [41–43].

AHP is a popular tool for analyzing and structuring complicated decision-making problems. It is very beneficial for assessing complex multi-criteria choices that include subjective criteria. The AHP technique can assist decision makers in calculating the weight of each criterion through pairwise comparative assessments. The weighted scoring system also has significant flaws. First, the decision-makers need to select suitable scores for the factors and available alternatives. This will need tremendous effort on the part of the parties to establish consensus. Additionally, both the weights and the ratings indicate trade-offs between the performance levels of the available options [42,44,45].

In the decision making of multi-criteria problem, AHP is often used to assess the relative importance of a collection of activities. This technique makes it possible to combine intangible qualitative aspects with measurable quantitative decision criteria. The AHP method mainly consists of three principles: first, the structure of the model; second, the comparison of alternatives and criteria; and third, prioritization. The decision-maker first divides the problem into three or more hierarchical levels. As shown in Figure 4, the goal of decision is clearly expressed at the top level of hierarchy.

Similarly, criteria and sub-criteria are provided in the hierarchy at the middle level. The bottom level represents the alternatives. The decision maker uses pairwise comparisons of criteria relative to the goal to generate ratio-scaled priorities that show the relative importance of the factors. Similarly, the decision maker develops ratio-scaled priorities of alternatives with respect to each criterion. Finally, the weights are merged to obtain the ranking of the sourcing alternatives and to choose the best model. The current study aims to determine the relative importance of factors and ranks the sourcing options accordingly. Therefore, the AHP is the best method to calculate the weights of the attributes and to identify the rankings of available models [42,46–48].

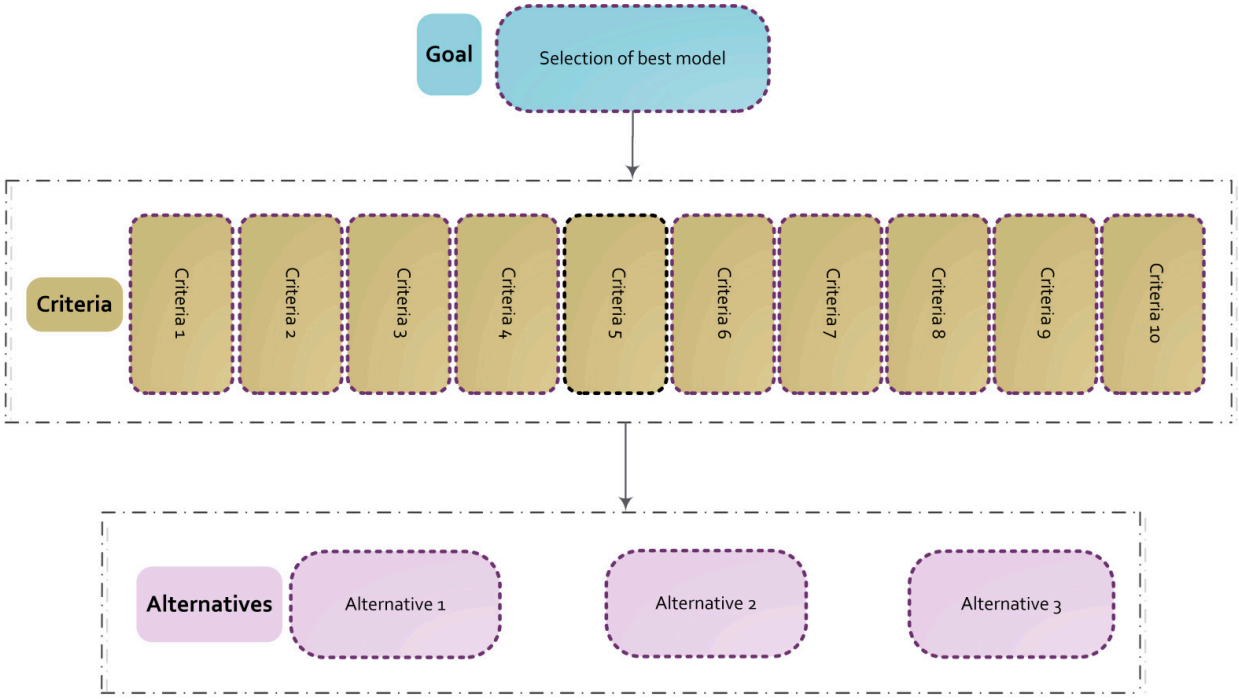


Figure 4. Hierarchical model to choose the appropriate alternative.

4.3. Developed Algorithms and Their Respective Flowcharts

The current subsection provides a detailed description of the working mechanism of the suggested tool. The OffshoringDSS performs automated operations, such as calculating the criteria weights and alternative weights and identifying the preferences of sourcing alternatives. The developed tool is based on Analytic Hierarchy Process (AHP) approach and developed using JavaScript language. The overall AHP method is represented by three algorithms: Algorithm 1 "finding weights", Algorithm 2 "finding inconsistency ratios" and Algorithm 3 "ranking of alternatives". Similarly, these algorithms are represented by their respective flowcharts, as given in Figures 5–7.

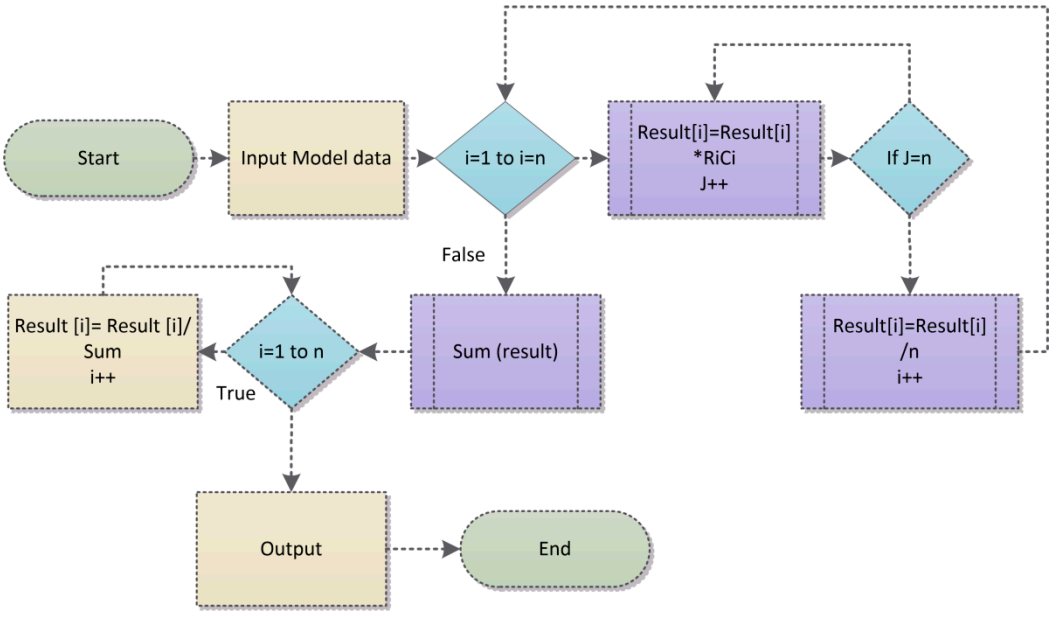


Figure 5. Flowchart of Algorithm 1 for calculating weights.

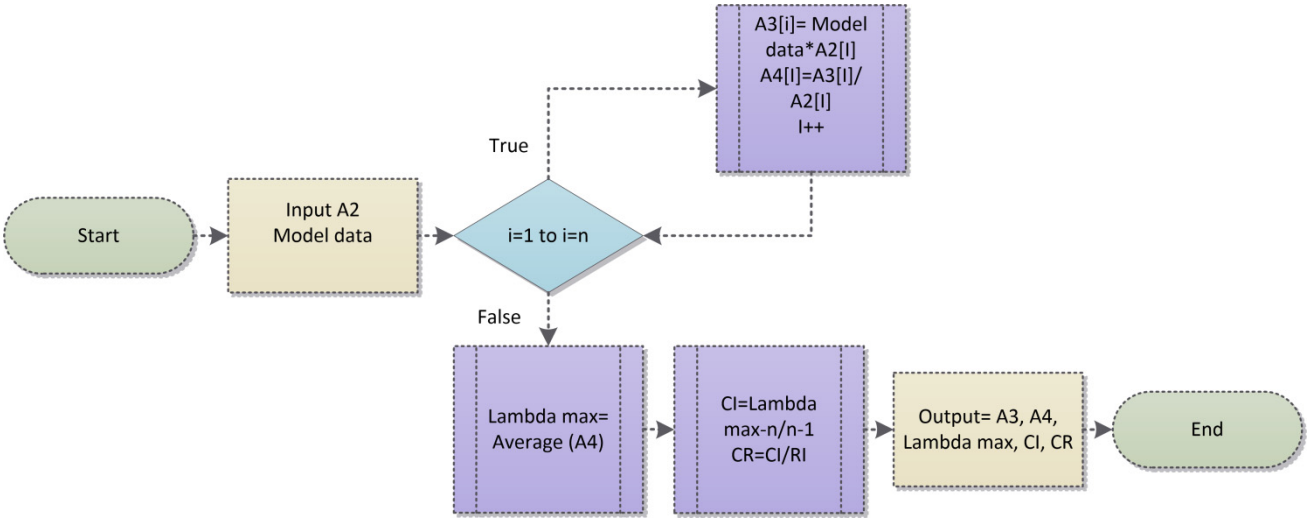


Figure 6. Flowchart of Algorithm 2 for calculating CR.

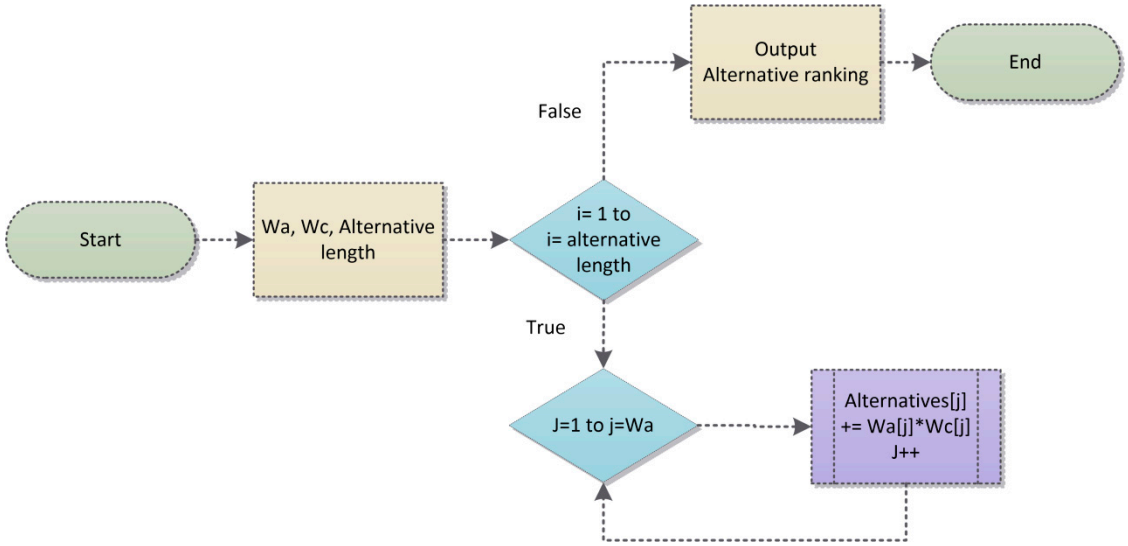


Figure 7. Flowchart of Algorithm 3 for alternatives ranking.

In the first step, input is provided to factors and alternatives by IT experts on a 1–9 points scale. Algorithm 1, as represented by its corresponding flowchart shown in Table 3 and Figure 5, calculates the weights of factors relative to goal and the weights of alternatives relative to criteria.

Table 3. Calculating weights of factors and alternatives.

Algorithm 1: Finding Weights (Model Data)
1. *Length model = find the length (model data- > column)*
2. *Define temp weight A2 =empty array*
3. *Define sum*
4. *Define weight A2 = empty array*
5. *Loop on model data- > column*
6. *Define temp A2 = 1*
7. *Define counter = 0*
8. *Loop on model data- > row*

Table 3. *Cont.*

Algorithm 1: Finding Weights (Model Data)	
9.	*Temp A2 = temp A2*model data- > score*
10.	*Loop on model data- > row ends*
11.	*Temp weight A2 [counter] = calculate_power*
12.	*(Temp A2, 1/length model)*
13.	*Sum = sum + temp weight A2 [counter]*
14.	*Counter = 0*
15.	*Loop on temp weight A2*
16.	*Weight A2 [counter] = temp weight A2/ sum*
17.	*Loop on temp weight A2 ends*
18.	*Loop on model data- > column ends*
19.	*Return weights A2*
20.	*Exit function*

The second step is about the calculation of inconsistency ratios that check the results obtained in the previous step. If the CR is less than 0.1 the output is acceptable and if it is greater than 0.1 then the expert should revise the input to factors and alternatives. Checking and calculating inconsistency ratios are achieved in the second step using Algorithm 2 and its respective flowchart, as given in Table 4 and Figure 6.

Table 4. Calculating consistency ratios.

Algorithm 2: Finding the Consistency Ratios (A2, Model Data)	
1.	*Initialize var_counter = 0*
2.	*Initialize A3 = array*
3.	*Initialize A4 = array*
4.	*Initialize Lambda_max = 0*
5.	*Loop till model_data*
6.	*A3 [var_counter] = model_data- > score *A2 [var_counter]*
7.	*A4 [var_counter] = A3 [var_counter]/A2 [var_counter]*
8.	*Var_counter = var_counter + 1*
9.	*Loop on model_data end*
10.	*Lambda_max = average(A4)*
11.	*Return (A3, A4, Lambda_max)*
12.	*Exit function*

The third step provides ranking of the available alternatives. It is the final step that ranks the alternatives by combining the weights identified in the previous two steps. Algorithm 3 and its corresponding flowchart are given in Table 5 and Figure 7. This final step shows the raking process and identifies the order of sourcing models for offshoring application maintenance.

4.4. Case Study Analysis and Discussion

This section discusses a case study conducted in the outsourcing industry to assess and validate the OffshoringDSS results. Before making an offshoring decision and selecting the best sourcing model from the available options, the project is reviewed on the basis of goals and requirements of the organization. The critical success factors used to eval-

uate projects and make the decisions of application maintenance are employees' skills, infrastructure, cost, poor communication, domain knowledge, legal requirements, project management, frequent requirements changes, language barrier and maturity level. The proposed OffshoringDSS determines the criteria weights and alternative weights based on the supplied input. Consequently, the overall weights of the alternatives are computed, and the sourcing models are ranked accordingly. The developed innovative tool is used by experts to make application maintenance outsourcing decisions. Similarly, it is able to automatically compute the weights of factors and available alternatives and accurately determine the ranking of sourcing options with less effort. The tool is freely available online at http://47.243.206.42/dmm/. Figure 8 shows the login page of the developed tool.

Table 5. Identifying the ranking of sourcing alternatives.

Algorithm 3: Finding the Ranking of Sourcing Alternatives (Wa, Wc, AlternateLength)
1. *Initialize var_counter = 0*
2. *Initialize var_alternates = array*
3. *Loop till var_alternate*
4. *Loop till Wa*
5. *Var_alternates [var_counter] + = Wa [var_counter]*W c[var_counter]*
6. *Var_counter = var_counter + 1*
7. *End Loop on Wa*
8. *Return (var_alternates)*
9. *Exit function*

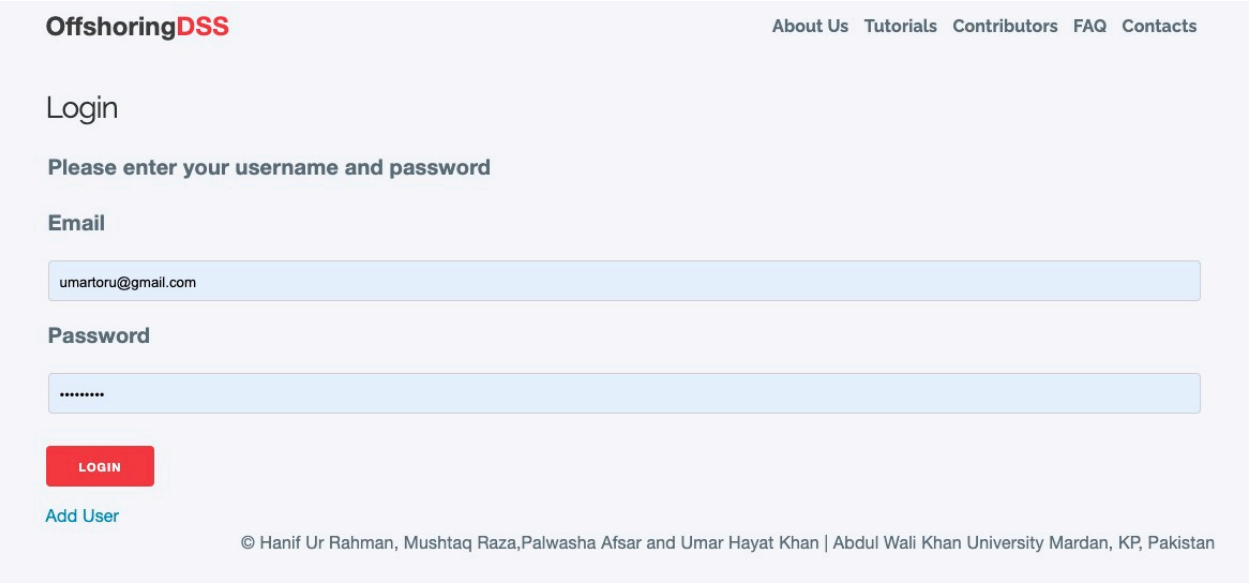


Figure 8. Login page of the proposed tool.

The input for the case study comes from an IT specialist from Vattenfall AB, Stockholm, Sweden. In addition, a strategist and consultant from Stairo Global LLC in the United States and an IT specialist from Khyber Pakhtunkhwa Information Technology Board (KPITB), Peshawar, Pakistan, validated the results generated by the developed tool. The complete decision-making process of application maintenance utilizing the developed OffshoringDSS is as follows.

4.5. Input to Criteria

The proposed automated tool is developed using the AHP approach. In order to develop the proposed tool, the first step is to determine the criteria. In the current study the adopted criteria are employees' skills, domain knowledge, legal requirements, cost, infrastructure, poor communication, project management, maturity level, frequent requirements changes and language barrier. Each component is assigned a relative value using the 9-point scale as shown in Table 1. The score is given based on its importance in relation to the objective. By comparing each pair of components, the relative significance of one attribute to the other is determined. A score of 9 shows that one criterion is extremely superior to another, while a value of 1 indicates equality. If the second factor has a higher priority than the first factor, a reciprocal is included. Thus, using the 9-point scale, the values from 1/9 to 9 are obtained [42,49,50]. An IT professional from Vattenfall AB, Stockholm, Sweden, supplied his choices for the sourcing criteria, as indicated in Table 6.

Table 6. Relative scores of criteria.

	ES	PC	Cos	LR	Inf	ML	FRC	LB	DK	PM
ES	1	4	1/3	1/4	1	6	2	1/2	6	3
PC	1/4	1	1/7	1/8	1/2	2	1/2	1/5	2	1
Cos	3	7	1	1	4	8	5	3	8	6
LR	4	8	1	1	5	9	6	3	9	7
Inf	1	2	1/4	1/5	1	4	1	1/3	4	2
ML	1/6	1/2	1/8	1/9	1/4	1	1/3	1/7	1	1/2
FRC	1/2	2	1/5	1/6	1	3	1	1/3	3	1
LB	2	5	1/3	1/3	3	7	3	1	7	4
DK	1/6	1/2	1/8	1/9	1/4	1	1/3	1/7	1	1/2
PM	1/3	1	1/6	1/7	1/2	2	1	1/4	2	1

4.6. Criteria Weight Calculation

In the second step, the weights of the criteria and sourcing alternatives are calculated. Table 7 shows the weights calculation of the criteria. The proposed tool automatically determines the weights of the factors after receiving the criteria scores. As shown in Figure 9, the factors weights are achieved by providing the specified scores into the constructed tool.

Table 7. Calculation of criteria's weights.

Criteria	Calculation	Weights of Criteria (Wc)
ES	[1*4*0.333*0.25*1*6*2*0.5*6*3]1/10 = a1	Wc1 = a1/sum
PC	(0.25*0.142*0.125*0.5*2*0.5*0.2*2*1) = a2	Wc2 = a2/sum
Cos	[3*7*1*1*4*8*5*3*8*6]1/10 = a3	Wc3 = a3/sum
LR	[4*8*1*1*5*9*6*3*9*7]1/10 = a4	Wc4 = a4/sum
Inf	[1*2*0.25*0.2*1*4*1*0.33*4*2]1/10 = a5	Wc5 = a5/sum
ML	[0.166*0.5*0.125*0.111*0.25*1*0.333*0.142*1*0.5]1/10 = a6	Wc6 = a6/sum
FRC	[0.5*0.2*0.166*1*3*1*0.333*3*1]1/10 = a7	Wc7 = a7/sum
LB	[2*5*0.33*0.33*3*7*3*1*7*4]1/10 = a8	Wc8 = a8/sum
DK	[0.166*0.5*0.125*0.111*0.25*1*0.333*1*0.5]1/10 = a9	Wc9 = a9/sum
PM	[0.333*1*0.166*0.142*0.5*2*1*0.25*2*1]1/10 = a10	Wc10 = a10/sum
	Sum = a1 + a2 + a3 + a4 + a5 + a6 + a7 + a8 + a9 + a10	

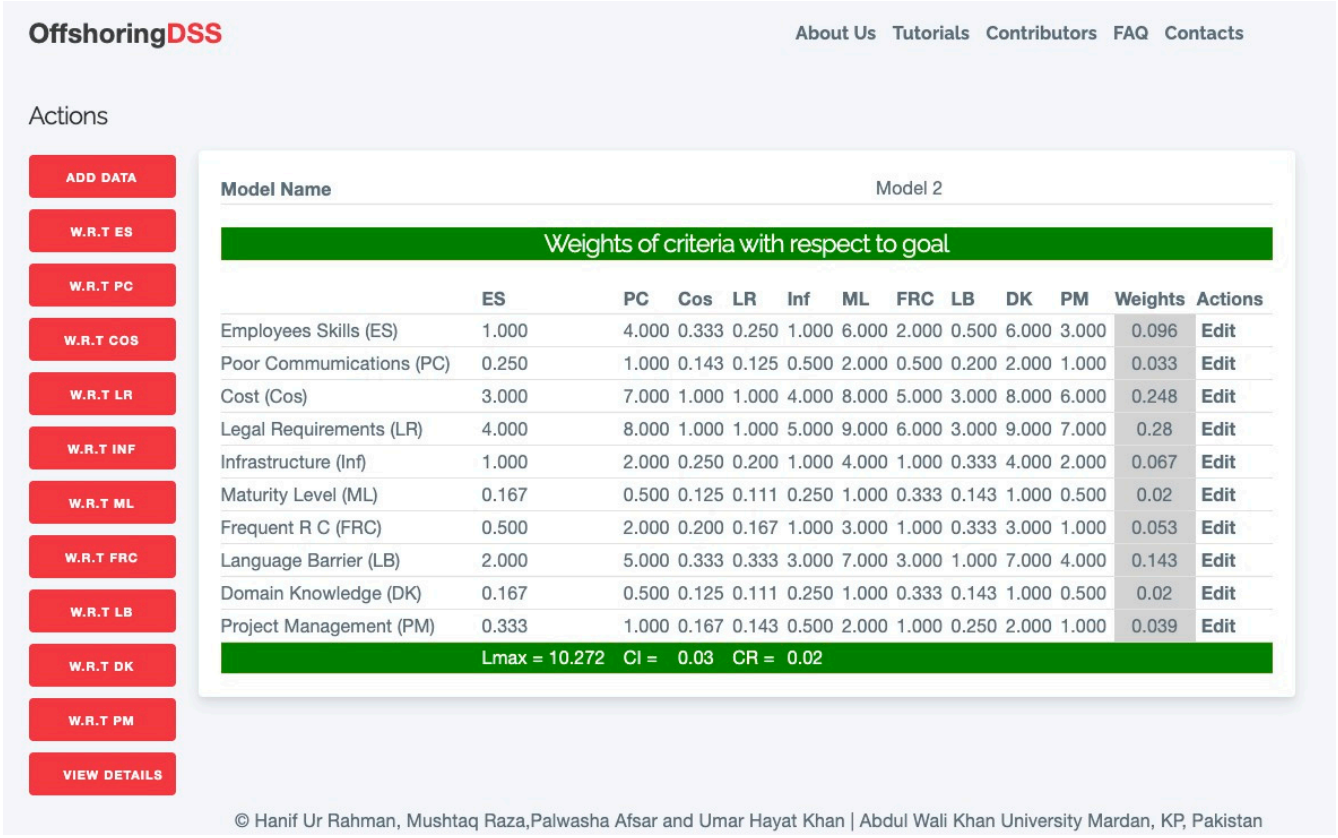


Figure 9. Weights of criteria with reference to the goal.

4.7. Calculation of Inconsistencies

Some inconsistencies may be observed as the comparisons are based on personal or subjective judgments. One of the most important advantages of the AHP is the final operation called the consistency check. It is used to calculate the consistency ratio between pairwise comparisons. This ensures that the judgments are consistent.

The upper value limit of CR is 0.1; therefore, the IT experts should revise the pairwise comparisons if a consistency ratio is greater than 0.1. It is used to evaluate the consistency of the expert and the overall hierarchy [42,51,52]. Equation (1) is used to calculate the Consistency Index (CI), whereas Equation (2) is used to identify the Consistency Ratio (CR) [53,54]. These formulas are implemented in the proposed tool to present the consistency ratios.

$$\mathrm{CI} = \frac{(Lambda\ x - 1)}{(n - 1)} \tag{1}$$

$$\mathrm{CR} = \frac{(\mathrm{CI})}{(\mathrm{RI})} \tag{2}$$

In order to find value of CI, the Lambda max value need to calculated using Equation (3).

$$Lambda\ max = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Awi}{wi}\right) \tag{3}$$

Furthermore, the step by step, calculation of Lambda max is given as follows:
→A1*A2 results A3
→A3/A2 results A4
→Average (A4) = Value of Lambda max

4.8. Alternatives' Weight with Reference to Criteria

The calculation of the alternative weights in relation to criteria is presented in this section. Only three options for providing global services are considered in this study: Nearshore Model (NM), Offshore Model (OfM) and Onshore Model (OM). The relative scores of alternatives with respect to the employees' skills are given by an IT expert as shown in Table 8. Similarly, Table 9 shows the calculation of alternatives' weights with respect to criteria.

Table 8. Models preferences with respect to the criterion (employee skills).

Models	Onshore Model	Nearshore Model	Offshore Model
Onshore model	1	1/4	1/4
Nearshore model	4	1	1
Offshore model	4	1	1

Table 9. Weights calculation of alternatives.

Models	Calculation	Weights of Alternatives (Wa)
Onshore model	[1*0.250*0.250]1/3 = a1	Wa1 = a1/sum
Nearshore model	[4*1*1]1/3 = a2	Wa2 = a2/sum
Offshore model	[4*1*1]1/3 = a3	Wa3 = a3/sum
	Sum = a1 + a2 + a3	

A weight calculation algorithm was added to the OffshoringDSS. Thus, the score given by the IT expert for each alternative is entered into the developed tool. It calculates the weights of the sourcing alternatives in relation to employees' skills, as given in Figure 10.

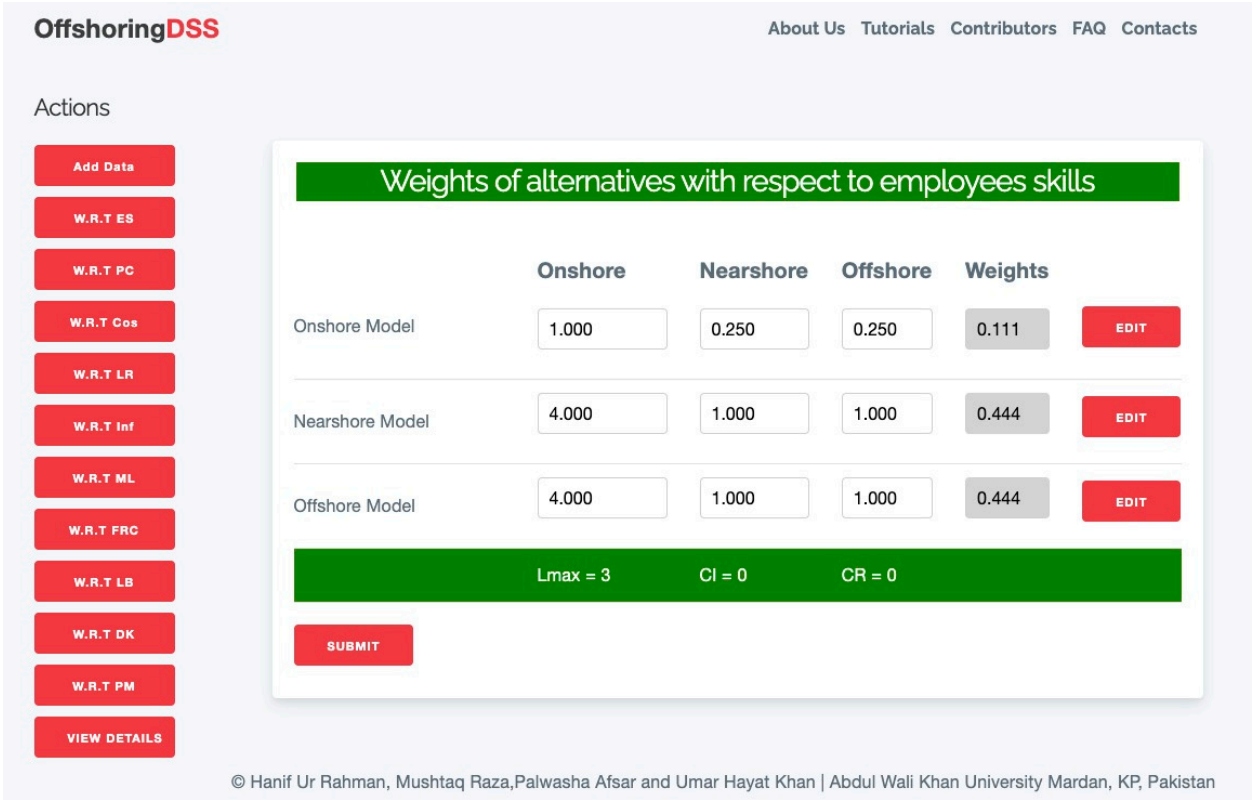


Figure 10. Weights of alternatives.

Besides the weights, it also provides the consistency ratios. The results of the tool indicate that both the offshore model and nearshore model received equal importance of 44%, while the received preference of onshore model is just 11%. Similarly, Figures A1–A9, as given in Appendix A, show the assigned scores for sourcing alternatives in relation to the remaining criteria, their corresponding calculated weights using remaining criteria and their corresponding calculated weights using the developed OffshoringDSS, along with the consistency ratios.

4.9. Ranking of Sourcing Alternatives

The current section discusses the ranking process of sourcing alternatives as well as the ranking of alternatives using the developed tool. The preference of each alternative is achieved by summing the total weights (Wci*Wai). Wci means the weight of criteria in relation to the goal as shown in Section 4.6, whereas Wai means the weights of alternatives in relation to the criteria as given in Section 4.8. Similarly, Table 10 shows the process of identifying the preferences of sourcing alternatives. Algorithm 3 was added to the OffshoringDSS to automatically rank the sourcing alternatives.

Table 10. Identifying ranking of the given alternatives.

	Alternatives		
Criteria	**Onshore Model**	**Nearshore Model**	**Offshore Model**
Criteria weights (Wc)	Wt (Wa) Total (Wc*Wa)	Wt (Wa) Total (Wc*Wa)	Wt (Wa) Total (Wc*Wa)
ES = wc1	wa1 wa1*wc1	Wa1 wa1*wc1	wa1 wa1*wc1
PC = wc2	Wa2 wa2*wc2	wa2 wa2*wc2	wa2 wa2*wc2
Cos = wc3	wa3 wa3*wc3	wa3 wa3*wc3	wa3 wa3*wc3
LR = wc4	Wa4 wa4*wc4	wa4 wa4*wc4	wa4 wa4*wc4
Infra = wc5	Wa5 a5*wc5	wa5 wa5*wc5	wa5 wa5*wc5
ML = wc6	Wa6 a6*wc6	wa6 wa6*wc6	wa6 wa6*wc6
FRC = wc7	wa7 a7*wc7	wa7 wa7*wc7	wa7 wa7*wc7
LB = wc8	wa8 a8*wc8	wa8 wa8*wc8	wa8 wa8*wc8
DK = wc9	wa9 a9*wc9	wa9 wa9*wc9	wa9 wa9*wc9
PM = wc10	wa10 10*wc10	wa10 wa10*wc10	wa10 wa10*wc10
Ranking of Alternatives	Sum = Preference of Onshore model	Sum = Preference of Nearshore model	Sum = Preference of Offshore model

Figure 11 shows the ranking of three sourcing models. The received preferences of the onshore model and offshore model are 34.8% and 35.7%, respectively, while the offshore model gained only 29%. The nearshore model appears to be the most appropriate model for the project, whereas the onshore model comes second. The obtained result was shared back with the same offshoring expert who supplied the input for this case study and also with other outsourcing specialists. They noted that OffshoringDSS is simple, easy to use and produces accurate results.

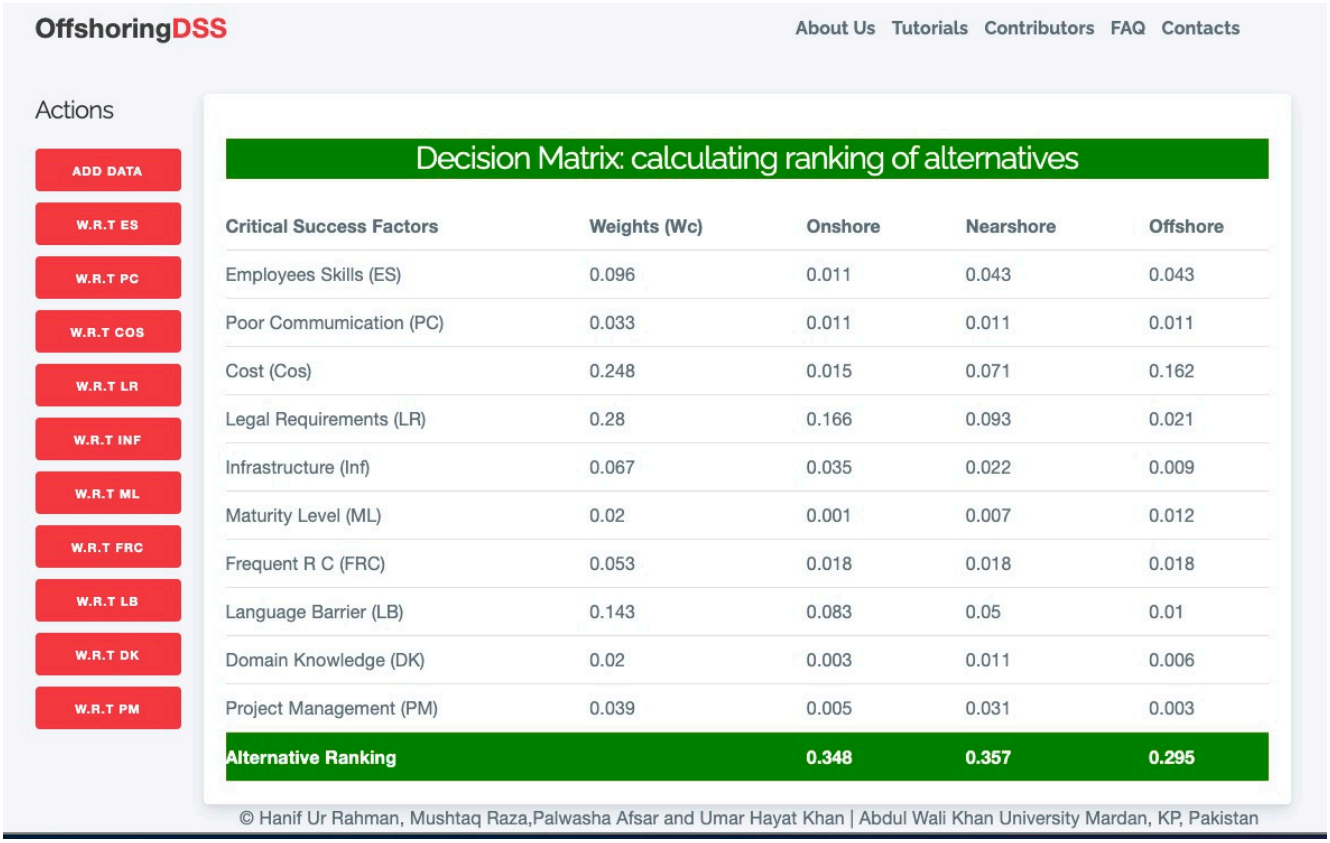


Figure 11. Ranking of sourcing alternatives.

5. Limitations of the Proposed Tool

To our knowledge, there is no such automated and novel tool to make the application maintenance offshoring decisions. The OffshoringDSS is a novel tool that helps decision makers to make effective decisions of application maintenance offshoring. However, the automated tool has several limitations that may hamper its wide application in the outsourcing industry.

To make sourcing decisions, IT professionals need to evaluate factors and alternatives. Based on the type of project and client's requirements, the influencing factors are quantified. Therefore, decision makers need to be experts in offshore outsourcing and GSD to make effective sourcing decisions.

The current version of the tool is based on 10 influencing factors. These factors were identified by conducting two systematic literature reviews [12,13]. Furthermore, these factors were validated by an empirical study [14]. Therefore, the developed tool considers no more than 10 factors to make sourcing decisions.

It is an automated tool capable of ranking the onshore model, nearshore model and offshore model. The ranking is based on the supplied ratings of variables and alternatives. The created tool, however, can only rank three different sourcing options. Although other multi-criteria decision-making strategies, such as the analytic network process, are available, the current version of the tool relies solely on the AHP technique to make offshore decisions.

6. Conclusions and Future Research

The purpose of this study is to examine the application maintenance outsourcing decision-making process. OffshoringDSS, a novel and automated tool, was introduced for reaching application maintenance offshoring decisions. A set of 10 influencing factors were used in the development of tool, namely employee skills, cost, legal requirements, poor communication, infrastructure, project management, maturity level, domain knowledge, frequent requirements changes and language barrier. The AHP technique was implemented

using the JavaScript language to create the tool. The tool was validated in the outsourcing industry by conducting a case study. An IT expert from Vattenfall AB, Stockholm, Sweden, provided input for the decision making. In addition, the result obtained with the developed tool was validated by a strategist and consultant at Stairo Global LLC in the United States and an IT expert at KPITB, Peshawar, Pakistan. The developed tool is used to make the offshoring decisions based on the influencing factors, project nature and preferences of the clients. It can be adopted by vendors as well as clients for making suitable sourcing decisions. It can be used as a decision support system that ranks three alternatives, i.e., the onshore model, nearshore model and offshore model and shows the appropriate model for application maintenance. Vendors may utilize the developed tool to analyze the projects in order to adopt an effective sourcing strategy that increases the likelihood of future venture success. Clients may assess their projects based on their priorities and make the best possible selection based on their needs and preferences. Therefore, the suggested tool also assists clients in making effective and appropriate sourcing decisions. The current tool is based on the AHP technique, and it is therefore planned to incorporate the ANP technique in the future. The developed tool only ranks onshore, nearshore and offshore models based on a set of 10 factors. Therefore, by implementing the ANP technique, the current automated model will be further improved, as well as the developed tool will be enabled to rank more than three alternatives.

Author Contributions: Conceptualization, H.U.R., and M.R.; methodology, H.U.R.; software, U.H.K., M.R.; validation, H.U.R., A.A. (Asaad Alzayed) and W.A.; formal analysis, H.U.R.; investigation, H.U.R.; resources, A.A. (Asaad Alzayed), W.A.; data curation, H.U.R.; writing—original draft preparation, H.U.R.; writing—review and editing, H.U.R., M.R.; visualization, H.U.R.; supervision, M.R. and P.A.; project administration, M.R., P.A.; A.A. Abdullah Alharbi; funding acquisition, A.A. (Abdullah Alharbi) and W.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Taif University Researchers Supporting Project number (TURSP-2020/231), Taif University, Taif, Saudi Arabia.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We deeply acknowledge Taif University for Supporting this research through Taif University Researchers Supporting Project number (TURSP-2020/231), Taif University, Taif, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Using the proposed tool (OffshoringDSS), Figures A1–A9 show the weight calculation of sourcing alternatives with respect to the criteria.

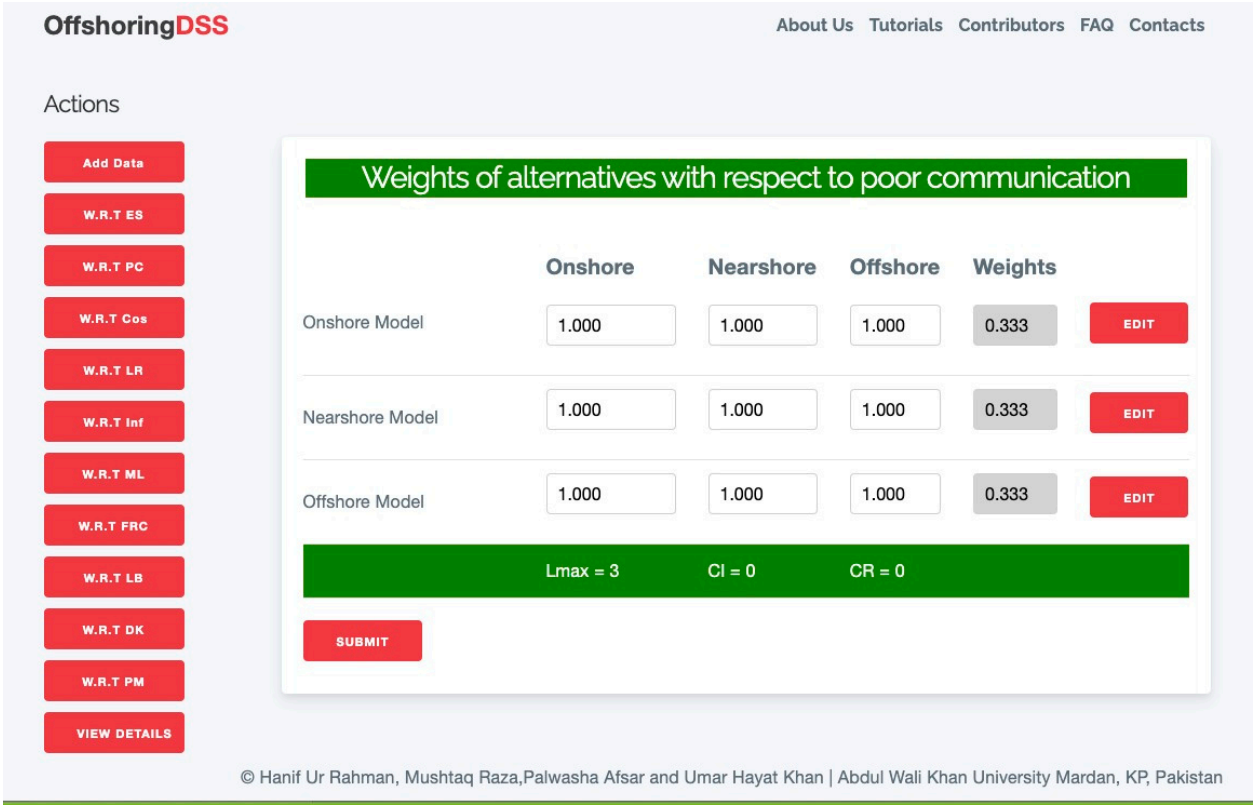


Figure A1. Alternatives weights with reference to PC and CR.

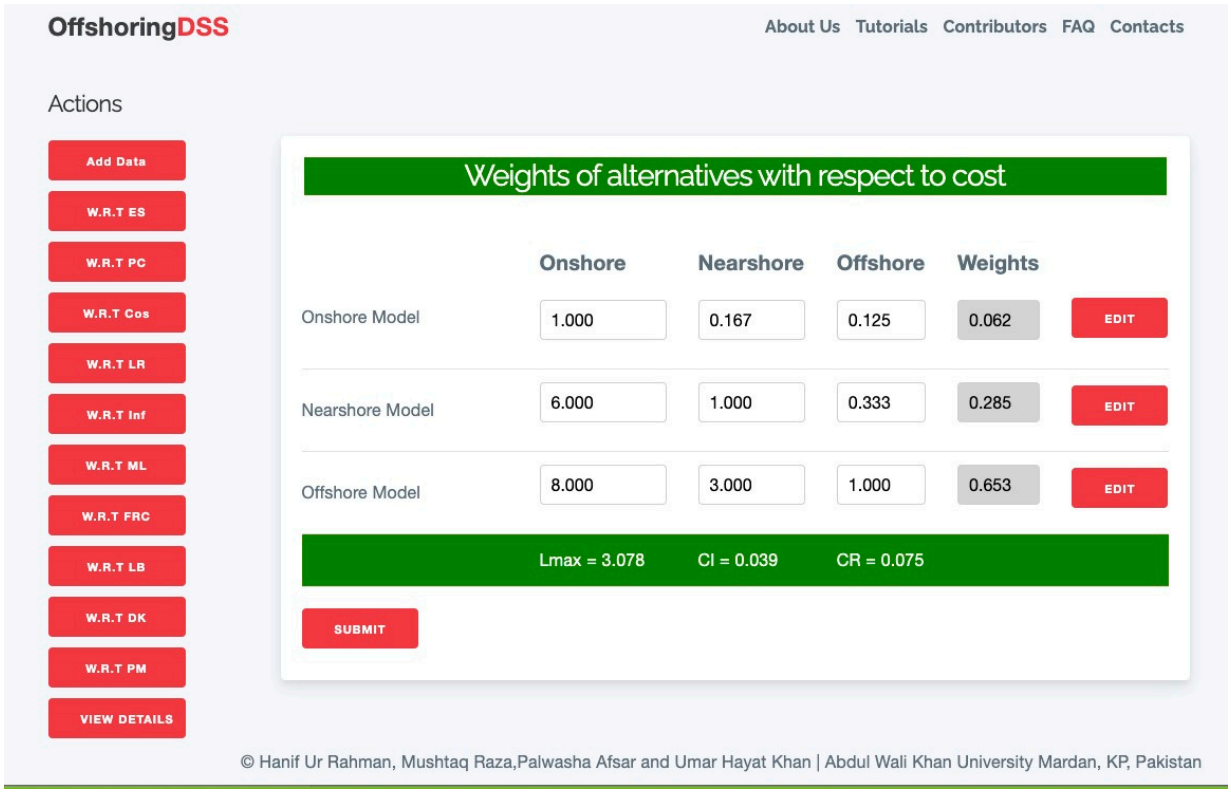


Figure A2. Alternatives weights with reference to cost and CR.

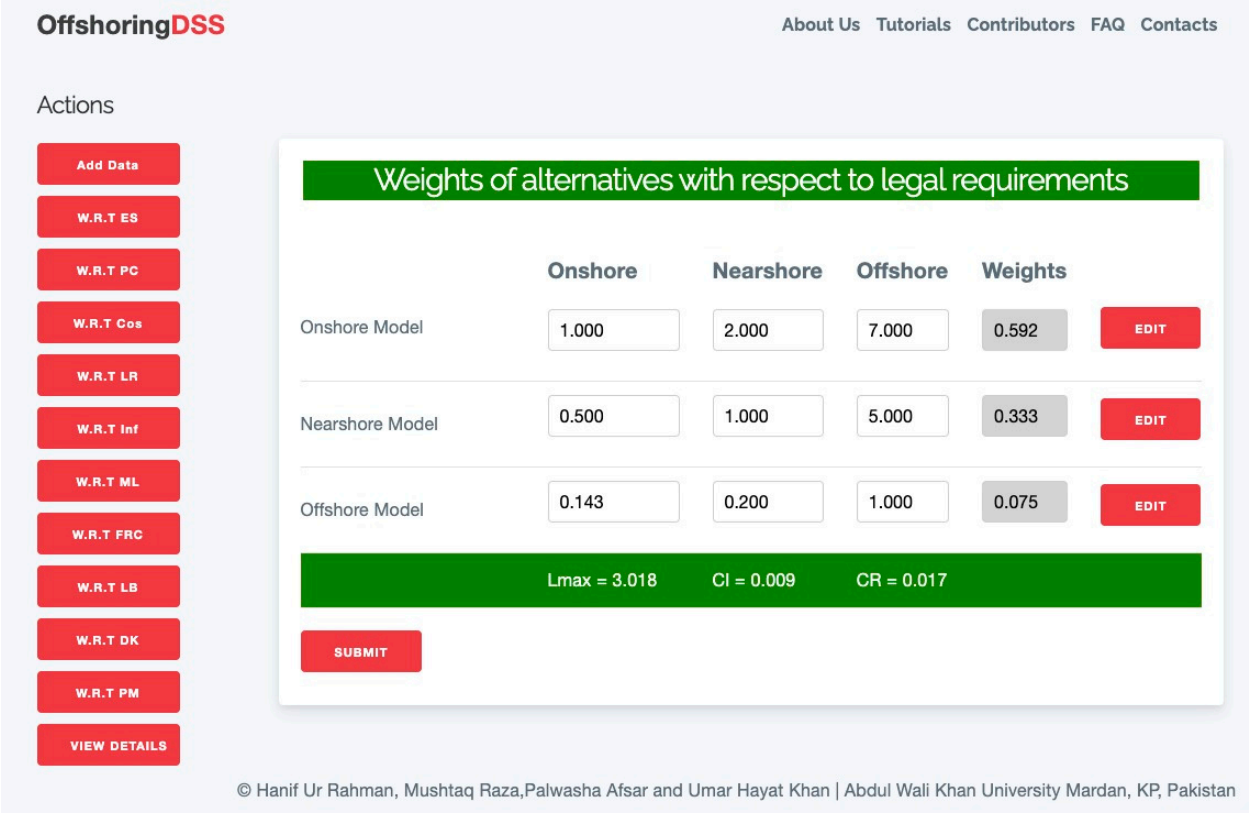


Figure A3. Alternatives weights with reference to LR and CR.

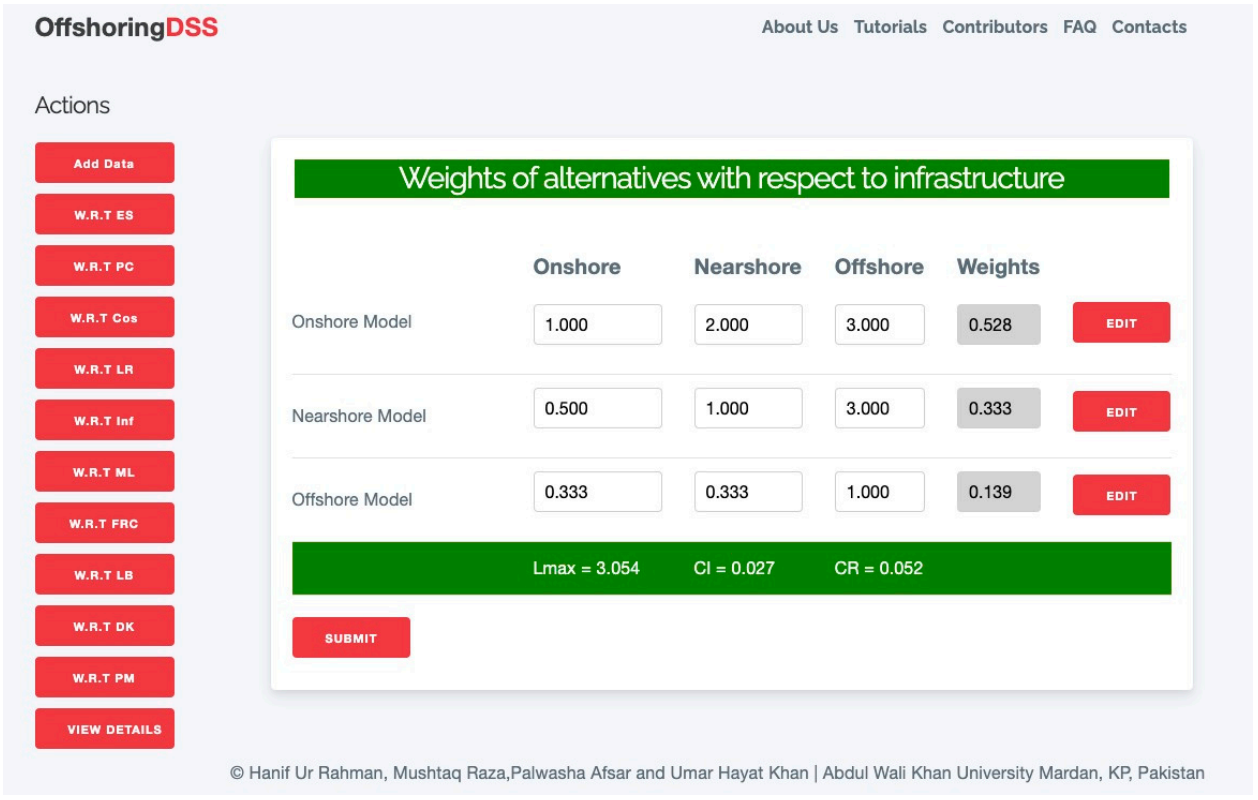


Figure A4. Alternatives weights with reference to Inf and CR.

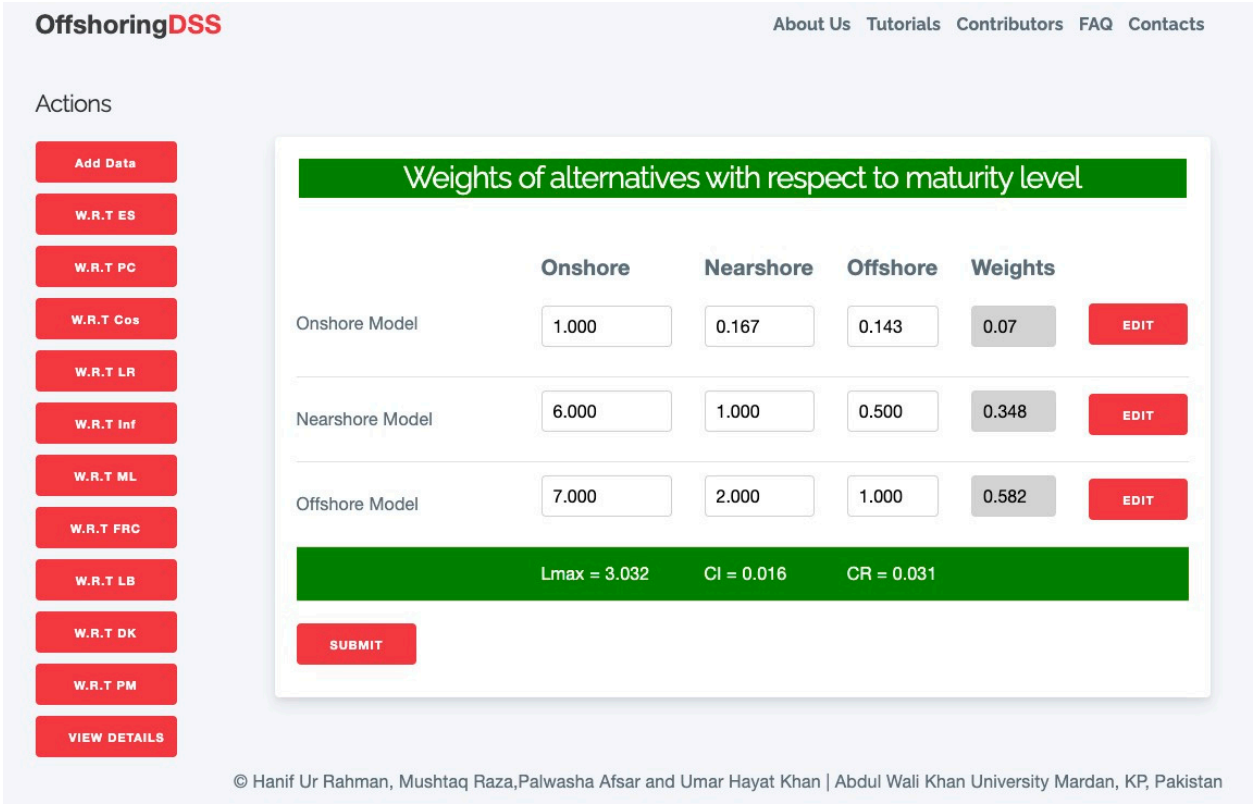


Figure A5. Alternatives weights with reference to ML and CR.



Figure A6. Alternatives weights with reference to FRC and CR.

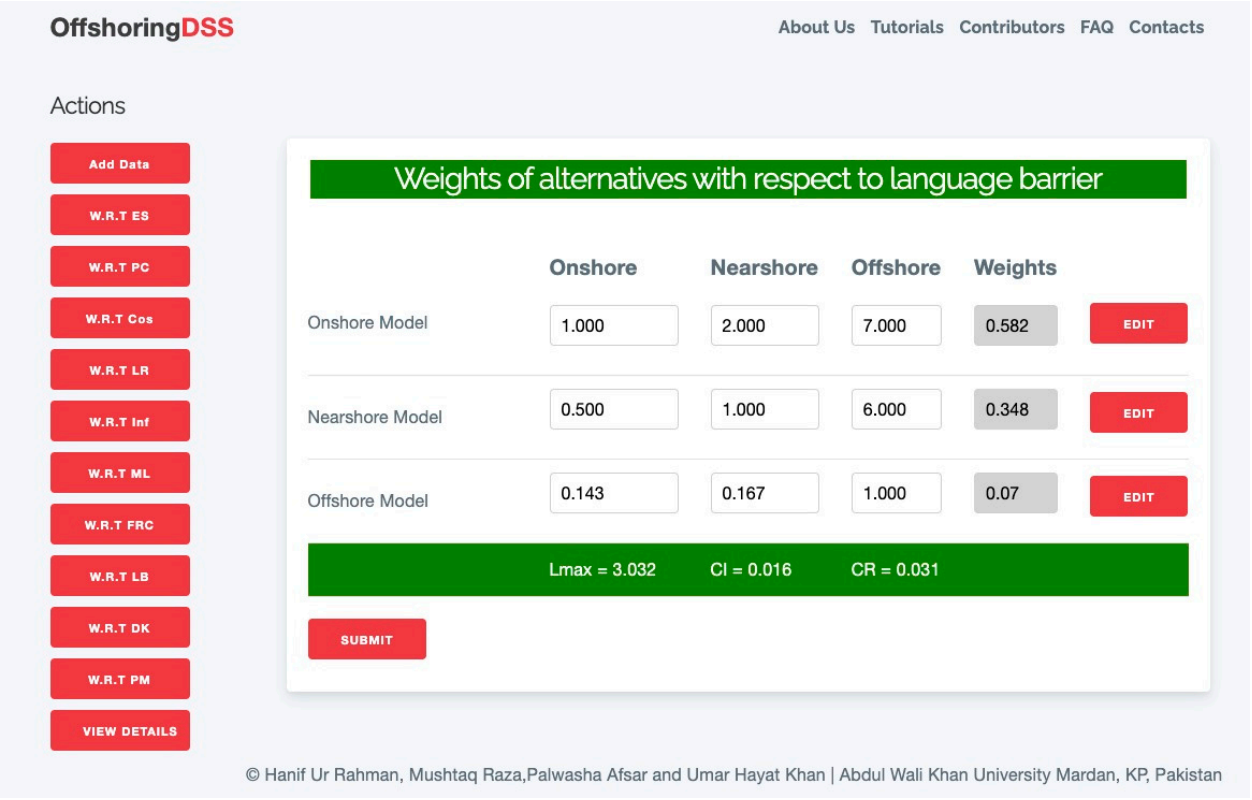


Figure A7. Alternatives weights with reference to LB and CR.

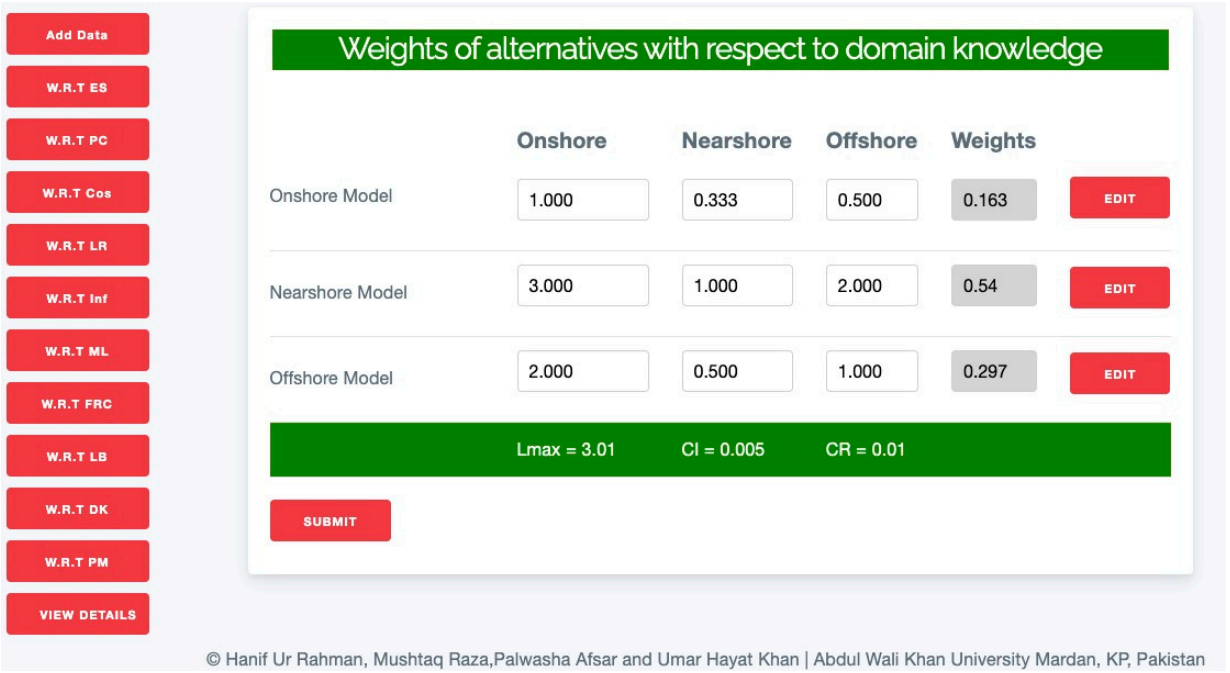


Figure A8. Alternatives weights with reference to DM and CR.

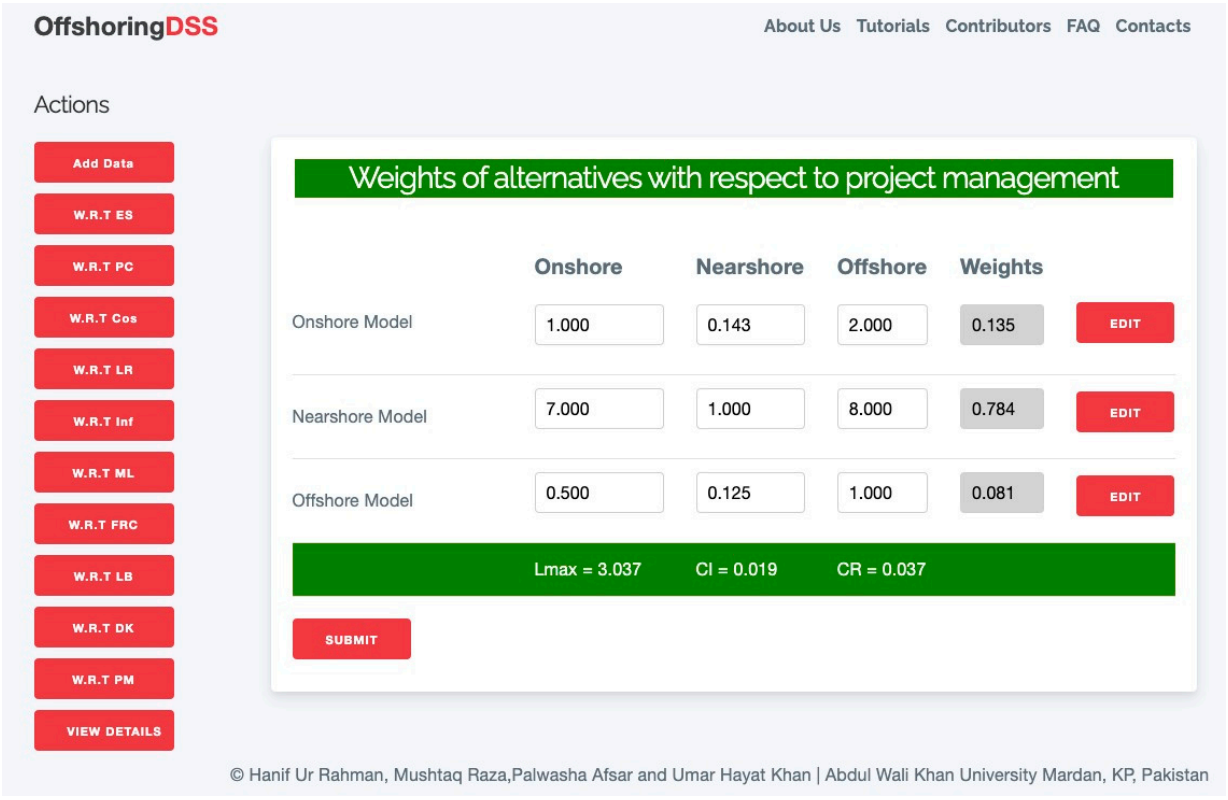


Figure A9. Alternatives weights with reference to PM and CR.

References

1. Ikram, A.; Riaz, H.; Khan, A.S. Eliciting theory of software maintenance outsourcing process: A systematic literature review. *Int. J. Comput.Sci. Netw. Secur.* **2018**, *18*, 132–143.
2. Babar, M.A.; Verner, J.M.; Nguyen, P.T. Establishing and maintaining trust in software outsourcing relationships: An empirical investigation. *J. Syst. Softw.* **2007**, *80*, 1438–1449. [CrossRef]
3. Lacity, M.C.; Khan, S.A.; Yan, A. Review of the Empirical Business Services Sourcing Literature: An Update and Future Directions. In *Outsourcing and Offshoring Business Services*; Palgrave Macmillan: Cham, Switzerland, 2017; pp. 499–651. [CrossRef]
4. Khan, J.A.; Khan, S.U.R.; Iqbal, J.; Rehman, I.U. Empirical Investigation About the Factors Affecting the Cost Estimation in Global Software Development Context. *IEEE Access* **2021**, *9*, 22274–22294. [CrossRef]
5. Conchúir, E.Ó.; Ågerfalk, P.J.; Olsson, H.H.; Fitzgerald, B. Global software development: Where are the benefits. *Commun. ACM* **2009**, *52*, 127–131. [CrossRef]
6. Khan, A.A.; Basri, S.; Dominc, P. A Proposed Framework for Communication Risks During RCM in GSD. *Procedia-Soc. Behav. Sci.* **2014**, *129*, 496–503. [CrossRef]
7. Ogheneovo, E.E. Software Maintenance and Evolution: The Implication for Software Development. *West Afr. J. Ind. Acad. Res.* **2013**, *7*, 81–92.
8. Anwar, S. Software Maintenance Prediction: An Architecture Perspective. Ph.D. Thesis, Department of Computer Science, FAST National University of Computer & Emerging Sciences, Islamabad, Pakistan, August 2010.
9. Ikram, A.; Jalil, M.A.; Ngah, A.B.; Khan, A.S.; Mahmood, Y. An Empirical Investigation of Vendor Readiness to Assess Offshore Software Maintenance Outsourcing Project. *IJCSNS* **2022**, *22*, 229.
10. Ikram, A.; Jalil, M.A.; Ngah, A.B.; Khan, A.S. Critical Factors in Selection of Offshore Software Maintenance Outsourcing Vendor: A Systematic Literature Review. *J. Theor. Appl. Inf. Technol.* **2020**, *98*, 3815–3824.
11. Ikram, A.; Jalil, M.A.; Ngah, A.B.; Khan, A.S.; Iqbal, T. Offshore Software Maintenance Outsourcing: Predicting Client's Proposal using Supervised Learning. *Int. J. Adv. Trends Comput. Sci. Eng.* **2021**, *10*, 106–113.
12. Rahman, H.U.; Raza, M.; Afsar, P.; Khan, H.U.; Nazir, S. Analyzing Factors That Influence Offshore Outsourcing Decision of Application Maintenance. *IEEE Access* **2020**, *8*, 183913–183926. [CrossRef]
13. Rahman, H.U.; Raza, M.; Afsar, P.; Khan, M.; Iqbal, N.; Khan, H.U. Making the Sourcing Decision of Software Maintenance and Information Technology. *IEEE Access* **2021**, *9*, 11492–11510. [CrossRef]
14. Rahman, H.U.; Raza, M.; Afsar, P.; Khan, H.U. Empirical Investigation of Influencing Factors Regarding Offshore Outsourcing Decision of Application Maintenance. *IEEE Access* **2021**, *9*, 58589–58608. [CrossRef]

15. Rahman, H.U.; Raza, M.; Afsar, P.; Alharbi, A.; Ahmad, S.; Alyami, H. Multi-Criteria Decision Making Model for Application Maintenance Offshoring Using Analytic Hierarchy Process. *Appl. Sci.* **2021**, *11*, 8550. [CrossRef]
16. Huen, W.H. An Enterprise Perspective of Software Offshoring. Frontiers in Education. In Proceedings of the 36th Annual Conference, San Diego, CA, USA, 27–31 October 2006; pp. 17–22. [CrossRef]
17. Smith, M.A.; Mitra, S.; Narasimhan, S. Offshore outsourcing of software development and maintenance: A framework for issues. *Inf. Manag.* **1996**, *31*, 165–175. [CrossRef]
18. Erickson, J.; Ranganathan, C. Project Management Capabilities: Key to Application Development Offshore Outsourcing. In Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06), Kauai, Hawaii, 4–7 January 2006; Volume 8, p. 199b. [CrossRef]
19. Mishra, D.; Mahanty, B. Study of maintenance project manpower dynamics in Indian software outsourcing industry. *J. Glob. Oper. Strat. Sourc.* **2019**, *12*, 62–81. [CrossRef]
20. Raza, M.; Faria, J.P. A model for analyzing performance problems and root causes in the personal software process. *J. Softw. Evol. Process* **2016**, *28*, 254–271. [CrossRef]
21. Yaseen, M.; Baseer, S.; Ali, S.; Khan, S.U. Abdullah Requirement implementation model (RIM) in the context of global software development. In Proceedings of the International Conference on Information and Communication Technologies (ICICT), Karachi, Pakistan, 12–13 December 2015; pp. 1–6. [CrossRef]
22. Khan, J.A.; Khan, S.U.R.; Khan, T.A.; Khan, I.U.R. An Amplified COCOMO-II Based Cost Estimation Model in Global Software Development Context. *IEEE Access* **2021**, *9*, 88602–88620. [CrossRef]
23. Prikladnicki, R.; Audy, J.L.N.; Evaristo, R. A Reference Model for Global Software Development: Findings from a Case Study. In Proceedings of the IEEE International Conference on Global Software Engineering, Florianopolis, Brazil, 16–19 October 2006; pp. 18–28. [CrossRef]
24. Lamersdorf, A.; Münch, J.; Rombach, D. A Decision Model for Supporting Task Allocation Processes in Global Software Development. In Proceedings of the International Conference on Product-Focused Software Process Improvement, Oulu, Finland, 15–17 June 2009; Springer: Berlin/Heidelberg, Germany; pp. 332–346. [CrossRef]
25. Ali, S.; Khan, S.U. Software outsourcing partnership model: An evaluation framework for vendor organizations. *J. Syst. Softw.* **2016**, *117*, 402–425. [CrossRef]
26. Lee, J.-N.; Huynh, M.Q.; Hirschheim, R. An integrative model of trust on IT outsourcing: Examining a bilateral perspective. *Inf. Syst. Front.* **2008**, *10*, 145–163. [CrossRef]
27. Lee, J.N.; Huynh, M. An Integrative Model of Trust on IT Outsourcing: From the Service Receiver's Perspective. In Proceedings of the Pacific Asia Conference on Information Systems, PACIS 2005, Bangkok, Thailand, 7–10 July 2005; Available online: http://aisel.aisnet.org/pacis2005/66 (accessed on 1 September 2022).
28. Flemming, R.; Low, G. Information Systems Outsourcing Relationship Model. *Australas. J. Inf. Syst.* **2007**, *14*, 95–112. [CrossRef]
29. Al Khoiry, I.; Gernowo, R.; Surarso, B. Fuzzy-AHP MOORA approach for vendor selection applications. *Regist. J. Ilm. Teknol. Sist. Inf.* **2021**, *8*, 24–37. [CrossRef]
30. Dedrick, J.; Carmel, E.; Kraemer, K.L. A dynamic model of offshore software development. *J. Inf. Technol.* **2011**, *26*, 1–15. [CrossRef]
31. Raza, M.; Faria, J.P. ProcessPAIR: A tool for automated performance analysis and improvement recommendation in software development. In Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering, Singapore, 3–7 September 2016; pp. 798–803. [CrossRef]
32. Beecham, S.; Noll, J.; Lero, R.I.; Dhungana, D.; Osterreich, A.S. A decision support system for global software development. In Proceedings of the IEEE Sixth International Conference on Global Software Engineering Workshop, Helsinki, Finland, 15–18 August 2011; pp. 48–53. [CrossRef]
33. Cataldo, M.; Shelton, C.; Choi, Y.; Huang, Y.Y.; Ramesh, V.; Saini, D.; Wang, L.Y. Camel: A tool for collaborative distributed software design. In Proceedings of the Fourth IEEE International Conference on Global Software Engineering, Limerick, Ireland, 13–16 July 2009.
34. Hattori, L.; Lanza, M. Syde: A tool for collaborative software development. In Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering, Cape Town, South Africa, 1–8 May 2010; Volume 2, pp. 235–238. [CrossRef]
35. Lanubile, F.; Mallardo, T.; Calefato, F. Tool support for geographically dispersed inspection teams. *Softw. Process. Improv. Pr.* **2003**, *8*, 217–231. [CrossRef]
36. Lamersdorf, A.; Munch, J. TAMRI: A Tool for Supporting Task Distribution in Global Software Development Projects. In Proceedings of the Fourth IEEE International Conference on Global Software Engineering, Limerick, Ireland, 13–16 July 2009. [CrossRef]
37. Velasquez, M.; Hester, P.T. An analysis of multi-criteria decision making methods. *Int. J. Oper. Res.* **2013**, *10*, 56–66.
38. Vaidya, O.S.; Kumar, S. Analytic hierarchy process: An overview of applications. *Eur. J. Oper. Res.* **2006**, *169*, 1–29. [CrossRef]
39. Tam, M.C.; Tummala, V. An application of the AHP in vendor selection of a telecommunications system. *Omega* **2001**, *29*, 171–182. [CrossRef]
40. Turón, A.; Aguarón, J.; Escobar, M.T.; Jiménez, J.M.M. A Decision Support System and Visualisation Tools for AHP-GDM. *Int. J. Decis. Support Syst. Technol. (IJDSST)* **2019**, *11*, 1–19. [CrossRef]

41. Saaty, T.L. Decision making—The Analytic Hierarchy and Network Processes (AHP/ANP). *J. Syst. Sci. Syst. Eng.* **2004**, *13*, 1–35. [CrossRef]
42. Dağdeviren, M.; Yavuz, S.; Kılınç, N. Weapon selection using the AHP and TOPSIS methods under fuzzy environment. *Expert Syst. Appl.* **2009**, *36*, 8143–8151. [CrossRef]
43. Herrera, A.O.; González, R.C.; Abu-Muhor, E.B. Multi-criteria Decision Model for Assessing Health Service Information Technology Network Support Using the Analytic Hierarchy Process. *Comput. Y Sist.* **2008**, *12*, 173–182.
44. Palcic, I.; Lalic, B. Analytical Hierarchy Process as a tool for selecting and evaluating projects. *Int. J. Simul. Model. (IJSIMM)* **2009**, *8*, 16–26. [CrossRef]
45. Saaty, T.L. Decision making with the analytic hierarchy process. *Int. J. Serv. Sci.* **2008**, *1*, 83–98. [CrossRef]
46. Albayrak, E.; Erensal, Y.C. Using analytic hierarchy process (AHP) to improve human performance: An application of multiple criteria decision making problem. *J. Intell. Manuf.* **2004**, *15*, 491–503. [CrossRef]
47. Mian, S.A.; Dai, C.X. Decision-Making over the Project Life Cycle: An Analytical Hierarchy Approach. *Proj. Manag. J.* **1999**, *30*, 40–52. [CrossRef]
48. Song, B.; Kang, S. A Method of Assigning Weights Using a Ranking and Nonhierarchy Comparison. *Adv. Decis. Sci.* **2016**, *2016*, 8963214. [CrossRef]
49. Mahdi, I.; Alreshaid, K. Decision support system for selecting the proper project delivery method using analytical hierarchy process (AHP). *Int. J. Proj. Manag.* **2005**, *23*, 564–572. [CrossRef]
50. Marcikic, A.; Radovanov, B. A decision model for outsourcing business activities based on the analytic hierarchy process. In Proceedings of the International Symposium Engineering Management and Competitiveness, Zrenjanin, Serbia, 24–25 June 2011; pp. 24–25.
51. Ho, W. Integrated analytic hierarchy process and its applications—A literature review. *Eur. J. Oper. Res.* **2008**, *186*, 211–228. [CrossRef]
52. Saaty, T.L. How to Make a Decision: The Analytic Hierarchy Process. *Eur. J. Oper. Res.* **1990**, *48*, 2–26. [CrossRef]
53. Saaty, T.L. Rank from comparisons and from ratings in the analytic hierarchy/network processes. *Eur. J. Oper. Res.* **2006**, *168*, 557–570. [CrossRef]
54. Sudaryono; Rahardja, U. Masaeni Decision Support System for Ranking of Students in Learning Management System (LMS) Activities using Analytical Hierarchy Process (AHP) Method. *J. Phys. Conf. Ser.* **2020**, *1477*, 022022. [CrossRef]