

Analytic Hierarchy Process (AHP), Weighted Scoring Method (WSM), and Hybrid Knowledge Based System (HKBS) for Software Selection: A Comparative Study

Anil Jadhav¹, Rajendra Sonar²

Indian Institute of Technology Bombay, Powai, Mumbai-400 076, India.

¹aniljadhav@iitb.ac.in, ²rm_sonar@iitb.ac.in

Abstract: Multi criteria decision making (MCDM) methods help decision makers to make preference decision over the available alternatives. Evaluation and selection of the software packages is multi criteria decision making problem. Analytical hierarchy process (AHP) and weighted scoring method (WSM) have widely been used for evaluation and selection of the software packages. Hybrid knowledge based system (HKBS) approach for evaluation and selection of the software packages has been proposed recently. Therefore, there is need to compare HKBS, AHP and WSM. This paper studies and compares these approaches by applying for evaluation and selection of the software components. The comparison shows that HKBS approach for evaluation and selection of the software packages is comparatively better than AHP and WSM with regard to (i) computational efficiency (ii) flexibility in problem solving (iii) knowledge reuse and (iv) consistency and presentation of the evaluation results.

1. Introduction

Nowadays number of information technology (IT) products and tools entering in the market place are increasing rapidly as IT is changing very fast. Accessing applicability of such a wide array of IT products, especially software packages, to business needs of the organization is tedious and time consuming task. The several research studies on evaluation and selection of the specific software products such as ERP packages [8], CRM packages [6], data warehouse systems [9], data mining software [7], simulation software [5], knowledge management (KM) tools [12], COTS components [11], original software components [3] shows the growing importance of software evaluation and selection decision making process. Evaluation and selection of the software packages involves simultaneous consideration of multiple factors to rank the available alternatives and select the best one [9]. MCDM problem refers to making preference decision over the available alternatives that are characterized by multiple, usually conflicting, attributes [16] [15]. Therefore, evaluation and selection of the software packages can be considered as MCDM problem.

A number of approaches for evaluation and selection of the software packages have been proposed. Among them AHP and WSM have widely been used for evaluation and selection of the software packages [1]. A hybrid knowledge based system (HKBS) approach for evaluation and selection of the software packages has been proposed recently in [2] and no comparison of it with AHP and WSM was found in the literature. Therefore, aim of this paper is to study and compare AHP, WSM and HKBS approach for evaluation and selection of the software packages. Rest of the paper is organized as follows.

Section 2 introduces MCDM methods: AHP, WSM, and HKBS for evaluation and selection of the software packages. Section 3 studies and compares these approaches by applying them for evaluation and selection of the software components. Section 4 concludes the paper.

2. Multi criteria decision making methods

MCDM problem generally involves choosing one of the several alternatives based on how well those alternatives rate against a chosen set of structured and weighted criteria as shown in the decision Table 1. Consider the MCDM problem with m criteria and n alternatives. Let $C_1, C_2, \dots, C_m$ and $A_1, A_2, \dots, A_n$ denote the criteria and alternatives, respectively. The generic decision matrix for solving MCDM problem is shown in the Table 1. Each column in the table represents the criterion and each row describes the performance of an alternative. The score S_{ij} describes the performance of alternative A_i against criterion C_j . As shown in the decision table, weights $W_1, W_2, \dots, W_k$ reflects the relative importance of criteria C_j in the decision making.

Table 1 The decision table

	Criteria				
	W1	W2	W3	...	Wk
Alternatives	C1	C2	C3	...	Cm
A1	S11	S12	S13	...	S1m
A2	S21	S22	S23	...	S2m
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
An	Sn1	Sn2	Sn3	...	Snm

2.1 Analytic Hierarchy Process (AHP)

AHP was proposed by Dr. Thomas saaty in the late 1970s [14] and has been applied in wide variety of applications in various fields. This method allows consideration of both objective and subjective factors in selecting the best alternative. The methodology is based on three principles: decomposition, comparative judgments; and synthesis of priorities. The decomposition principle calls for the construction of a hierarchical network to represent a decision problem, with the top representing overall objectives (goal) and the lower levels representing criteria, sub-criteria, and alternatives. With the comparative judgments, users are required to set up a comparison matrix at each level of hierarchy by comparing pairs of criteria or sub-criteria. In general comparison takes the form: "How important is criteria C_i relative to criteria C_j ?. Questions of this type are used to establish the weights for criteria. The possible judgments used

for pairwise comparison and their respective numerical values are described in the Table 2. Similar questions are to be answered to access the performance scores for the alternatives on the subjective (judgmental) criteria. Let A_{ij} denotes the value obtained by comparing alternative A_i to alternative A_j relative to the criterion C_i . As decision maker is assumed to be consistent in making judgments about any one pair of alternative and since all alternatives will always rank equally when compared to themselves, we have $A_{ij}=1/A_{ji}$ and $A_{ii}=1$. This means that it is only necessary to make $1/2*m*(m-1)$ comparisons to establish full set of pairwise judgments. The final stage is to calculate aggregate performance value for each alternative and ranking of alternatives according to their performance. Aggregate score is obtained using the following formula.

$$R_i = \sum W_k A_{ik}$$

Where R_i is overall score of i^{th} alternative, W_k is importance (weight) of k^{th} criterion, and A_{ik} is relative score of i^{th} alternative with respect to k^{th} criterion.

Table 2 Pair-wise comparison judgments

Judgment	Values
X is equally preferred to Y	1
X is moderately preferred over Y	3
X is strongly preferred over Y	5
X is very strongly preferred over Y	7
X is extremely preferred over Y	9
Intermediate values	2,4,6,8
Preference of Y compared to X	1/2, 1/3, 1/4, 1/5, 1/6, 1/7, 1/8, 1/9

2.2 Weighted Scoring Method (WSM)

WSM is another common approach used for evaluation and selection of the software packages [1]. Consider m alternatives $\{A_1, A_2, \dots, A_m\}$ with n deterministic criteria $\{C_1, C_2, \dots, C_n\}$. The alternatives are fully characterized by decision matrix $\{S_{ij}\}$, where S_{ij} is the score that measures how well alternative A_i performs on criterion C_j . The weights $\{W_1, W_2, \dots, W_k\}$ accounts for the relative importance of the criteria. The best alternative is the one with highest score. In WSM the final score for alternative A_i is calculated using the following formula.

$$S(A_i) = \sum W_j S_{ij}$$

Where sum is over $j=1,2,\dots, n$; W_j is relative importance of j^{th} criterion; S_{ij} is score that measures how well alternative A_i performs on criterion C_j .

2.3 Hybrid knowledge based system (HKBS)

Formal and precise description of the software packages is usually not available. A reasonable approach is to augment the available documentation with the informal knowledge derived from the literature, practices, and experience of the experts. Knowledge based system (KBS) provides a way to organize the knowledge and deliver a tool that assists decision makers in evaluation and selection of the software packages [4]. Evaluation and selection of the software packages is knowledge intensive process and KBS has potential to play

significant role in evaluation and selection of the software packages [10].

Knowledge based systems are computer based information systems which embody the knowledge of experts and manipulates the expertise to solve problem at an experts level of performance [13]. Rule based reasoning (RBR) and case based reasoning (CBR) are two fundamental and complementary reasoning methods of the KBS. KBS has four major components namely knowledge base, inference engine, user interface, and the explanation subsystem.

This paper provides only short description of HKBS. Please refer [2] for detailed description of the HKBS. HKBS employs an integrated RBR and CBR techniques for evaluation and selection of the software packages. RBR component of the HKBS stores knowledge about software evaluation criteria, its meaning, and metrics for assessment of the candidate software packages. It assists decision makers in choosing software evaluation criteria, specifying user requirements of the software package, and formulating a problem case. It also provides flexibility in changing evaluation criteria and user requirements of the software package. User requirements of the software package are collected in the form of feature and feature value. Once user requirements of the software package are captured, those are then submitted to CBR component of the HKBS. CBR is the most important component of the HKBS. It is used to determine how well each candidate software package meets user requirements of that package. Candidate software packages to be evaluated are stored as cases in case base of the system. Case base is collection of the cases described using well defined set of feature and feature values.

3. Comparison of AHP, WSM & HKBS

In this section we compare AHP, WSM and HKBS by applying these techniques for evaluation and selection of the software components. The data about the software components to be evaluated is taken from the study [3]. The reason behind using the data from this study is that the case described in the study represents real world situation and evaluation results are also available for the comparison. The study proposed a quality framework for developing and evaluating original software components. The framework is demonstrated and validated by applying it in searching for a two-way SMS messaging component to be incorporated in an online trading platform. The software components considered for evaluation are: ActiveSMS (SC1); SMSDemon (SC2); GSMActive (SC3); SMSZyneo (SC4). Table 3 provides details of these four software components. Table 4 provides details of the evaluation criteria, its importance, metrics, and user requirements of the software component.

Table 3 Details of the software components

Evaluation criteria	SC1	SC2	SC3	SC4
User Satisfaction	5	2	5	1
Service Satisfaction	9/12	5/12	8/12	6/12
Access Control	Provided	provided	Provided	Provided
Error Prone	0/day	1/day	0/day	0/day

Correctness	1	1	1	0.85
Throughput	60/min	8/min	120/min	8/min
Capacity	8	1	16	1
Upgradeability	5	4	5	4
Backward compatibility	Provided	provided	Provided	Provided

(Source: Andreou & Tziakouris, 2007)

Table 4 Details of evaluation criteria

Criteria	Sub-criteria	Weight (%)	Metrics	User Requirements
Functionality	User Satisfaction	20	Level of satisfaction on the scale of 5	5
	Service Satisfaction	20	Functions Ratio	12
	Access Control	5	Provided or Not	Provided
Reliability	Error Prone	10	Number of errors/crashes per unit of time	0
	Correctness	10	Ratio of successful SMS sending	1
Efficiency	Throughput	15	Number of requests per unit of time	50
	Capacity	10	Number of GSM modems supported	5
Maintainability	Upgradeability	5	Level of satisfaction on the scale of 5	5
	Backward compatibility	5	Provided or Not	Provided

(Source: Andreou & Tziakouris, 2007)

3.1 Software component selection using AHP

The first stage in AHP is formulating decision hierarchy. The decision hierarchy for selection of the software components is depicted in the Figure 1. The highest level of the hierarchy represents goal, second level represents criteria, third level represents sub-criteria, and fourth level represents software components to be evaluated.

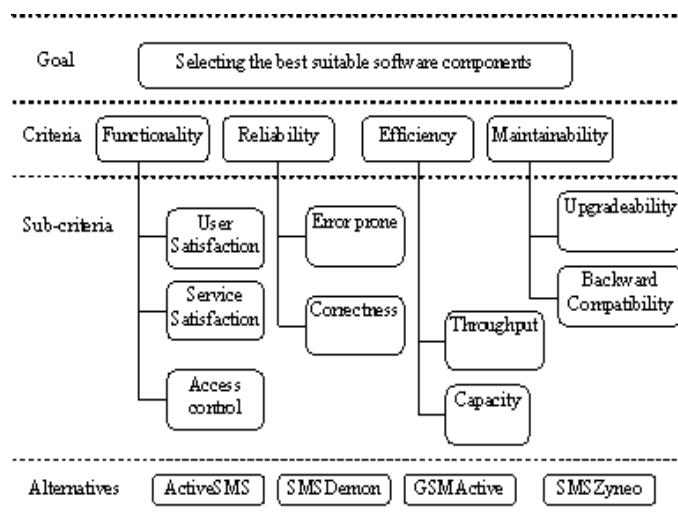


Figure 1 Decision hierarchy for component selection

As importance (weight) of each evaluation criterion is given, the second stage in AHP is obtaining pairwise comparison matrix and normalized matrix by comparing each alternative over the others with regard to each evaluation criterion. The pairwise comparison matrix and normalized matrix with respect to user satisfaction and service satisfaction criterion is shown in the Table 5 to Table 8 respectively. Similarly normalized score is obtained for each alternative with regard to each evaluation criterion.

Table 5 Pair-wise comparison matrix with respect to user satisfaction

	SC1	SC2	SC3	SC4
SC1	1	5	1	8
SC2	1/5	1	1/5	3
SC3	1	5	1	8
SC4	1/8	1/3	1/8	1

Table 6 Normalized alternative score with respect to user satisfaction

	SC1	SC2	SC3	SC4	Average
SC1	0.43	0.44	0.43	0.40	0.43
SC2	0.09	0.09	0.09	0.15	0.10
SC3	0.43	0.44	0.43	0.40	0.43
SC4	0.05	0.03	0.05	0.05	0.05

Table 7 Pair-wise comparison matrix with respect to service satisfaction

	SC1	SC2	SC3	SC4
SC1	1	4	2	3
SC2	1/4	1	1/3	1/2
SC3	1/2	3	1	2
SC4	1/3	2	1/2	1

Table 8 Normalized alternative score with respect to service satisfaction

	SC1	SC2	SC3	SC4	Average
SC1	0.48	0.40	0.52	0.46	0.47
SC2	0.12	0.10	0.09	0.08	0.10
SC3	0.24	0.30	0.26	0.31	0.28
SC4	0.16	0.20	0.13	0.15	0.16

The third stage in AHP is identifying preferred alternative by calculating aggregate score of each alternative. Aggregate score is calculated by multiplying normalized score by weight (importance) of that criterion, and sum the result for all criteria. The preferred alternative will have the highest score. The calculation of aggregate score for each alternative using AHP is shown in the Table 9.

Table 9 Aggregate score of software component using AHP

Component	Criteria	Weight	Normalized Score	Score
SC1	User Satisfaction	20	0.43	8.60
	Service Satisfaction	20	0.47	9.40
	Access Control	5	0.25	1.25
	Error Prone	10	0.31	3.10
	Correctness	10	0.29	2.90
	Throughput	15	0.43	6.45
	Capacity	10	0.43	4.30
	Upgradeability	5	0.38	1.90
	Backward	5	0.25	1.25

	compatibility			
	Total score			39.15
SC2	User Satisfaction	20	0.1	2.00
	Service Satisfaction	20	0.1	2.00
	Access Control	5	0.25	1.25
	Error Prone	10	0.06	0.60
	Correctness	10	0.29	2.90
	Throughput	15	0.07	1.05
	Capacity	10	0.07	0.70
	Upgradeability	5	0.13	0.65
	Backward compatibility	5	0.25	1.25
	Total score			12.40
SC3	User Satisfaction	20	0.43	8.60
	Service Satisfaction	20	0.28	5.60
	Access Control	5	0.25	1.25
	Error Prone	10	0.31	3.10
	Correctness	10	0.29	2.90
	Throughput	15	0.43	6.45
	Capacity	10	0.43	4.30
	Upgradeability	5	0.38	1.90
	Backward compatibility	5	0.25	1.25
	Total score			35.35
SC4	User Satisfaction	20	0.05	1.00
	Service Satisfaction	20	0.16	3.20
	Access Control	5	0.25	1.25
	Error Prone	10	0.31	3.10
	Correctness	10	0.14	1.40
	Throughput	15	0.07	1.05
	Capacity	10	0.07	0.70
	Upgradeability	5	0.13	0.65
	Backward compatibility	5	0.25	1.25
	Total score			13.60

3.2 Software component selection using WSM

Weighted scoring method works only with the numeric data. Therefore, rating of each alternative with regard to each evaluation criterion must be done before calculating the final score. In case of the component selection, except user satisfaction and upgradeability criteria, direct rating is not given for any other criteria. Therefore, all alternatives with regard to each evaluation criterion have first rated by considering user requirements of the software component. The rating and aggregate score for each alternative calculated using WSM is shown in the Table 10.

Table 10 Aggregate score of software component using WSM

Component	Criteria	Weight	Rating	Score
SC1	User Satisfaction	20	5	100
	Service Satisfaction	20	4	80
	Access Control	5	1	5
	Error Prone	10	5	50
	Correctness	10	5	50
	Throughput	15	5	75
	Capacity	10	5	50
	Upgradeability	5	5	25
	Backward	5	1	5

	compatibility			
	Total score			440
SC2	User Satisfaction	20	2	40
	Service Satisfaction	20	2	40
	Access Control	5	1	5
	Error Prone	10	3	30
	Correctness	10	5	50
	Throughput	15	1	15
	Capacity	10	1	10
	Upgradeability	5	4	20
	Backward compatibility	5	1	5
	Total score			215
SC3	User Satisfaction	20	5	100
	Service Satisfaction	20	3	60
	Access Control	5	1	5
	Error Prone	10	5	50
	Correctness	10	5	50
	Throughput	15	5	75
	Capacity	10	5	50
	Upgradeability	5	5	25
	Backward compatibility	5	1	5
	Total score			420
SC4	User Satisfaction	20	1	20
	Service Satisfaction	20	3	60
	Access Control	5	1	5
	Error Prone	10	5	50
	Correctness	10	4	40
	Throughput	15	1	15
	Capacity	10	1	10
	Upgradeability	5	4	20
	Backward compatibility	5	1	5
	Total score			225

3.3 Software component selection using HKBS

HKBS is an integration of rule based and case based reasoning components. The rule based component of the HKBS assists decision makers to: (1) select criteria which he/she wants to consider for evaluation of the software components, (2) capture users needs of the software component through simple or knowledge driven sequence of form, (3) formulating a problem case. An example of how system assists decision makers in selecting evaluation criteria and specifying user requirements of the software component is shown in the Figure 2 and Figure 3 respectively. Once user requirements of the software component are captured those are then submitted to the CBR component of the HKBS. The CBR component of HKBS is used to (1) retrieve software components from case base of the system (2) compare user requirements of the software components with the description of each retrieved software component (3) rank software components in descending order of the similarity score. Similarity score indicates how well each component meets user requirements of that component. Case schema is collection of case features and it is heart of the CBR system. Each case feature is linked to similarity measure, a function, which is used to calculate individual feature level similarity between problem case and the solution cases. In this study a problem case is nothing but user requirements of the software

component and solution cases are nothing but software components to be evaluated. The similarity knowledge which is stored in the form of case schema is used to determine the fit between software component and user requirements of that component. Assessing similarity in CBR at the case (global) level involves combining the individual feature (local) level similarities. Formula used to calculate case level similarity is as follows:

$$\frac{\sum_{i=1}^n W_i * sim(qv, cv)}{\sum_{i=1}^n W_i}$$

Where W_i is relative importance (weight) of the feature in similarity assessment process, and $sim(qv, cv)$ is local similarity between query value and case value of the feature. The functions used to calculate local similarity depends on type of the feature. The result of evaluation of the software components produced by HKBS is shown in the Figure 4. Functional and quality criteria column indicates how well each component meets functional and quality requirements of the component respectively. The case matching column indicates how well each software component meets overall (functional & quality requirements) requirements of the software component. From the result it can be easily observed that ActiveSMS component seems better option than the others.

Figure 2 Form for selecting evaluation criteria

Figure 3 Form for specifying user needs of the software component

ComponentID	ComponentName	Quality Criteria	Functional Criteria	Case Matching	Criteria Matching
1	ActiveSMS	99.73	87.73	94.33%	94.33%
3	GSMActive	99.74	83.64	92.49%	92.49%
4	SMSzyneo	85.17	30.45	60.55%	60.55%
2	SMSDemon	69.87	37.61	55.35%	55.35%

Figure 4 Result of evaluation of the software components

3.4 Comparison of AHP, WSM and HKBS

The ranking of evaluation of the software components obtained using HKBS is similar to the ranking obtained using AHP and WSM. Therefore, it can be concluded that HKBS produce not only correct results but also can be used as a tool for evaluation and selection of the software components/packages.

The comparison of AHP, WSM and HKBS is summarized in the Table 11. The comparison and application of AHP, WSM, and HKBS for evaluation and selection of the software components shows that HKBS approach is comparatively better than AHP and WSM with regard to the following aspects.

Computational efficiency:

- HKBS works well with both qualitative as well as quantitative parameters.
- HKBS is comparatively easy to use when
 - number of evaluation criteria or a number of alternatives to be evaluated are large in number
 - requirements changes
 - number of alternatives to be evaluated changes
 - evaluation criteria changes

Knowledge reuse:

- HKBS retains knowledge about software evaluation criteria and similarity knowledge for determining the fit between software component and user requirements of that component. Therefore, it can be reused later for evaluation of the same or other software components with different requirements of the same or different organization.

Consistency and presentation of the Results:

- Resulting score of AHP and WSM represents the relative ranking of the alternatives whereas HKBS produce results not only representing ranking of the alternatives but also indicates how well each alternative meet user requirements of the software component (Refer Figure 2).
- In case of AHP and WSM aggregate score of each alternative may not remain same even though requirements are same because aggregate score depends on expert's own judgment which may not remain consistent for all the time. Whereas HKBS produce same

results unless user requirements of the software component changes.

- Adding an alternative may cause rank reversal (reversal in ranking) problem in AHP which never occurs in HKBS.

Flexibility in problem solving:

- HKBS assists decision makers not only in choosing evaluation criteria but also for specifying and changing user requirements of the software component.

- Addition or deletion of the software components in HKBS is easy as it uses case base for storing details of the component to be evaluated.
-

Table 11 Comparison of HKBS, AHP and WSM

Evaluation Techniques Parameters	AHP	WSM	HKBS
Support for qualitative parameters	Yes	No	Yes
Support for quantitative parameters	Yes	Yes	Yes
If the number of alternatives to be evaluated increases	Pairwise comparisons also increases and needs to be done again to calculate final score	Rating of each alternative with regard to each evaluation criterion must be done before calculating final score	Any number of alternatives can be added or removed with no extra efforts required to calculate similarity score
If the number of evaluation criteria changes	Pairwise comparisons needs to be done again to calculate final score	No extra efforts are required to calculate final score	No extra efforts are required to calculate similarity score
If user requirements changes	Pairwise comparisons needs to be done again to calculate final score	Rating of each alternative with regard to each evaluation changes and needs to be done before calculating final score	Provides flexibility to change requirements and calculate similarity score accordingly with no extra efforts required
Support for Knowledge/experience reuse	No	No	Yes
Support to specify and change user requirements	No	No	Yes
Rank reversal (reversal in ranking) problem	Yes	No	No
Support to indicate how well each software component meet user requirements of that component	No	No	Yes

4. Conclusion

This paper described AHP, WSM and HKBS for evaluation and selection of the software components. The comparison of AHP, WSM and HKBS for software selection has been done by applying these techniques for evaluation and selection of the software components. The result (ranking of the software components) produced by HKBS is similar to the result obtained using AHP and WSM. Therefore, we can conclude that HKBS not only produce the correct results but also can be used as a tool for evaluation and selection of the software components. The comparison and application of AHP, WSM, and HKBS for evaluation and selection of the software components shows that HKBS approach is comparatively better than AHP and WSM with regard to the following aspects: (i) computational efficiency (ii) knowledge reuse (iii) flexibility in problem solving (iv) consistency and presentations of the results.

References

- [1] A. S. Jadhav, R. M. Sonar, Evaluating and selecting software packages: A review, Information and software technology 51 (2009), 555-563.
- [2] A. S. Jadhav, R. M. Sonar, A hybrid system for selection of the software packages, Proceeding of first international conference on emerging trends in engineering and technology ICETET-08, IEEE Xplore, 337-342
- [3] A. S. Andreou, M. Tziakouris, A quality framework for developing and evaluating original software components, Information and software technology 49, 2007, pp. 122 -141.
- [4] S. Bandini, F. Paoli, S. Manzoni, P. Mereghetti, A support system to COTS-based software development for business services, SEKE'02 ACM 2001, pp. 307-314.
- [5] J. K. Cochran, H Chen, Fuzzy multi-criteria selection of object-oriented simulation software for production system analysis, Computers and operations research 32, 2005, pp. 153-168.
- [6] E. Colombo, C. Francalanci, Selecting CRM packages based on architectural, functional, and cost requirements: empirical validation of a hierarchical ranking model, Requirements Eng. 9, 2004, pp. 186-203.
- [7] K. Collier, B. Carey, D. Sautter, C. Marjanierni, A methodology for evaluating and selecting data mining software, Proceedings of 32nd Hawaii International conference on system sciences-1999, pp. 1-11.
- [8] X. B. Illa , X Franch, J. A. Pastor, Formalizing ERP selection criteria, Proceedings of tenth international workshop on software specifications and design, IEEE, 2000.
- [9] H.-Y. Lin, P.-Y. Hsu, G.-J. Sheen, A fuzzy-based decision making procedure for data warehouse system selection, Expert systems with applications, 2006.

- [10] A. Mohamed, T. Wanyama, G. Ruhe, A. Eberlein, B. Far, COTS evaluation supported by knowledge bases, Springer-Verlag, LSO 2004, LNCS 3096, pp. 43-54.
- [11] D. Morera, COTS evaluation using desmet methodology & Analytic Hierarchy Process (AHP), Springer-Verlag, PROFES 2002, LNCS 2559, pp. 485-493.
- [12] E. W. T. Ngai, E. W. C. Chan, Evaluation of knowledge management tools using AHP, Expert system with applications, 2005 pp. 1-11.
- [13] W. B. Rauch-Hindin, A guide to commercial artificial intelligence. Englewood Cliffs, NJ: Prentice Hall, 1988
- [14] T. L. Saaty, The Analytic Hierarchy Process, McGraw Hill, 1980.
- [15] E. Triantaphyllou, Multi-Criteria Decision making Methods: A Comparative Study, Springer, 2000
- [16] K. Yoon, C. Hwang, Multiple Attribute Decision-Making: an Introduction, 1995, Sage Publisher.