2019 5th International Conference for Convergence in Technology (I2CT)
Pune, India. Mar 29-31, 2019

Intelligent Recruitment System

Chamila Maddumage[1], Dulanjaya Senevirathne[1], Isuru Gayashan[1], Tharusha Shehan[1],
Sagara Sumathipala[2]
Faculty of Information Technology, University of Moratuwa, Sri Lanka
chamilamaddumage@gmail.com, dulanjaya.sl@gmail.com, isurugy@gmail.com tharushashehan@gmail.com,
sagaras@uom.lk

***Abstract*—this paper reviews studies and developments in recruitment techniques interact with computer science. And novel approach proposed as Intelligent Recruitment System (IRS) which consist of resume classification and ranking with deep learning alongside with NLP techniques, Automatic question generation(AQG) system which measure technical proficiency of an applicant using the knowledge base of merged ontology designed using the web and local sources and soft skills measuring with question answering about different skill sets and matching the similarity of the predefined answers against the applicant given answer by using both the syntactical similarity measurement and semantic similarity measurements. The final output comes from a combination of the above input parameters. The purpose of this model is to propose a new framework based on the fuzzy inference system (FIS) and Mamdani's method. The decision-making mechanism of the IRS is based on the final total score of each parameter and maximum quota. If two or more candidates receive the same final total score, it is difficult to make a definitive decision. Therefore, a FIS approach is proposed to help decision makers choose the most qualified applicants and avoid unnecessary problems.**

Keywords-NLP, FLC, FIS, Ontology, AQG, Deep Learning, IRS

I. INTRODUCTION

Companies and recruitment agencies must handle a large amount of work each day. This process consumes skilled labor and time. So put forward the development of intelligent recruitment system. The system includes resume parser and classification module, question and answer module, soft skills measurement system and the final decision-making system.

II. OTHER'S WORK

Competency measuring is a vast area of study. The skill measuring in an interview process for a particular organization is very useful in selecting the best applicant for the job role. There are various approaches that being done in the past in particularly measuring the skill level of the employees. There are several approaches in the past where they tried to measure the skill levels using several model. Data mining, text mining, opinion mining, fuzzy logic, Neural Networks. Opinion Mining Technique.

In addition, many existing systems are done only with Natural language processing Techniques. Intelligent hiring with resume parser and ranking system [1] developed by undergraduates of Mumbai University. India with natural language processing alongside with Machine learning. Therefore, they could provide a ranking mechanism for resume data. In 2017 another thesis, which is entirely based on Natural language processing techniques, has been published by three undergraduates of KIIT University. [2]

Many AQG systems are mostly bounded with just NLP techniques that lets the system to recognize a phase and generate questions out of it. Based on this same core principle at their automation process approach and this has limited the knowledge area that this kind of a system can be influenced. Some of the early systems include Writers Workshop[3], developed at Bell Laboratories, and Editor[4], developed at Rochester Institute of Technology. Both systems focus on grammar and style. It could also be argued that these systems only aim at supporting writing to communicate as opposed to writing to learn. A writing tutoring system[5] developed at the University of Memphis, is based on the notion of voices that speak to the writer during the process of composition uses avatars to associate each voice with a face and personality.

Luther Alexander Latumakulita, Fajar Purnama, Tsuyoshi Usagawa, Sary Paturusi, Delta Ardy Prima[6] has proposed a new framework based on FIS with the Mamdani methods for "Bidik Misi" scholarship selection to provide a final decision as a combination of poverty and academic performance levels to help human evaluator to choose the most qualified applicants and avoid an unnecessary problem in the selection process. Aries Susanto, Lisa Latifah, Nuryasin, Aida Fitriyani[7] have proposed a review on Decision Support System based on Murabaha method using Fuzzy multi attribute decision making based Yarger model which is expected to help give solutions to the assessment of potential customers filling financing to management in future. In 2017 a fuzzy decision support system for choosing a marine deliver company was designed by Marina Solesvic, Yuriy Kondratenko, Galyna Kondratenko, Levgen Sidenko, vyacheslav Kharchenko, Artem Boyarchuk with the help of discrete fuzzy inference engine [8]. From the above reference in can be seen that fuzzy logic can be used to help decision making and a new approach is proposed in this paper.

III. OUR APPROACH

A. High Level Architecture

The process is comprised with four major modules such as Resume Parser, Automatic Question and Answer Generation (AQG), Competency Evaluation and Decision Making.

978-1-5386-8075-9/19/$31.00 ©2019 IEEE

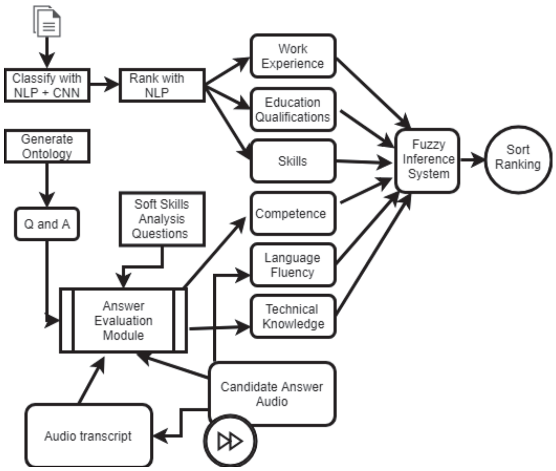

Fig 1. High Level Architecture

B. Resume Parser and Classify module

After a deep study about above all designs and technologies a novel method for resume(CV) parser module could be proposed. This module mainly divided into two sections such as resume ranking module and classify module. Designs for both modules have been described below with relevant technologie used.

1) Resume Parser

Extract all necessary information from resume with natural language processing and rank them with relevant attributes included below. Those structured data passed to final Decision-making Model with fuzzy logic.After data resume upload module and data extraction module from '.pdf' or '.doc' implemented. Then relevant segments of resume identified and extract all necessary information using NLP and convert them into structured format. In addition those resumes will be ranked by parameters mentioned below:

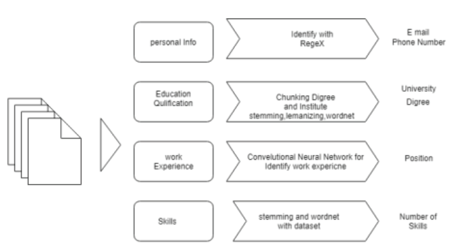

Fig. 2.Architecture of resume parser

Data segmentation implemented with concept of Tokenization alongside with chunking and POS tagging. In addition NLP concept called lemmatization, stemming are used for segmentation. In fact wordnet library is using for Data extraction. Those extracted information should be ranked properly before send to Fuzzy module according to following parameters.

TABLE I
RANKING CRITERIA

Parameter	**Program**	**Marks**
Education Qualification	Diploma	2/10
	Degree	4/10
	Msc	6/10
	2*Msc/PhD	8/10
Work Experience	Relevant field applies	Accuracy of CNN output * 10
Skills	Verified by QnA	Number of skills
Projects	Verified by QnA	Number of projects

For the ranking purpose and identifying (Classifying) work experience from which area candidate has most work experience Novel deep learning approach was used as following.

2) Classifier with CNN

There are many methods have been used for document classifying. Natural language processing is base technology for all those mythologies.Recent trend of NLP is to use with deep learning approaches. Therefore, convolution neural network, which is a type of deep neural network, used in resume classification module.
In this module, resumes are categorized into five types, which are job roles in Software Company as follows:

- ➢ Software Engineer
- ➢ Network Engineer
- ➢ Quality Assurance Engineer
- ➢ Business Analyst
- ➢ System Engineer

3) Datasets

After the implementation of this convolution neural network would be trained by 5000 resumes, which are downloaded by web site called indeed according to above categories. Architecture of CNN, shown in figure 2, is a slight variant CNN architecture from the architecture developed by Collobert et al in 2011. Which gives good performance in text classification tasks emerges with deep learning. Manually extracted work experience from 5000 resumes and wrote all labeled data to CSV file, which contain 2000 phrases per each category (total 10000 phrases).

- ➢ sequence_length –length of sentences (max length).
- ➢ num_classes – Number of classes in the output layer,five in our case.
- ➢ vocab_size –size of vocabulary. needed to define the size of embedding layer, which will shape as [vocabulary_size, embedding_size].
- ➢ embedding_size –dimensionality of embeddings.
- ➢ filter_sizes –number of words want our convolution filters to cover. We will have num_filters for each size specified here. For example, [3, 4, 5] means that we will have filters that slide over 3, 4 and 5 words respectively, for a total of 3 * num_filters filters.
- ➢ num_filters –number of filters per filter size

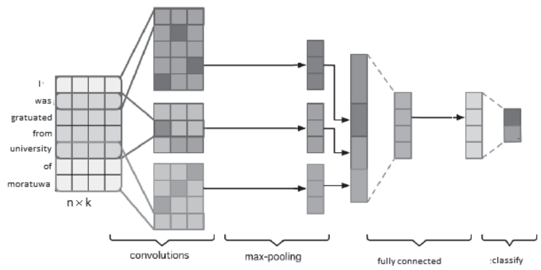

Fig. 3. CNN Architecture

Results of this classifier was sent to fuzzy module in json format as following

[{ "label": "SE",
"Phrase": "I worked as a software engineer in epic lanka”
"accuracy": "0.72" }]

C. AQG Approach

The ontology will be generated automatically considering the five user roles of an IT company which are SE/ QA/ BA/ SysEng/ NetworkEng, that will be given at the beginning of the process and with the structured data set that is arranged in the format of python dictionary. Which then can be turned to a newly generated ontology with separate branches to each knowledge area. The reason behind this implementation is to use the web RDF data from each website (Like W3school, Tutorials Point, etc.) to automatically generate an updated ontology each time this process is started. This way, even when a new subject area is introduced to the field of IT this generated ontology will be up to date with all those knowledge components. here the python library called pronto is used to manipulate the ontology knowledgebase.

After the ontology is generated by the system the questions and answers will be generated at the next step. Here questions are generated considering the grammatical structure of a general conversational questions in English. Which forms with "wh word + Auxiliary verb + Subject + Object" basic structure. Relationships between classes and subclasses are considered here when generating the Q&A set. After the Q&A set is generated those will be passed to another NLP modal which converts to a human understandable level and filters which aren't in a satisfactory level.

A weight to each Q&A set will be added using ginger-py library concept and with several NLP techniques for further use in the Q&A presentation stage. These each weight consists of a representation to the question's degree of difficulty. Using this and with the user role and user experience level the presentation Q&A set will be filtered out and asked by the candidate.

Then the candidate given answers will be considered with the degree of confidence towards each question area. The next question to be asked by the candidate will be determined by this level. Then the next Q&A set will be related to the best answered knowledge area of the candidate. Likewise, the system will be traversing from one knowledge are branch of the ontology to create next generated Q&A's to the candidate. For example, if the candidate answered a question related to OOP concepts with a best result the next generated Q&A set will be more related to that knowledge area and the system will be traversing downwards from that OOP parent class to its branches to generate those next Q&A set.

Then this same procedure will be continuing till the interview is finished and the results are passed to the fuzzy modal. Answered Q&A set results will be passed with the weight to each Q&A set.

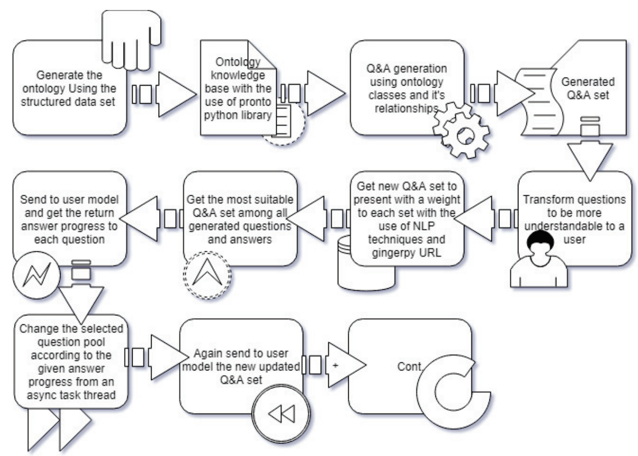

Fig. 4. AQG approach proposed model

The given result for each Q&A set and the weight to it is then passed to the fuzzy model in order to get the final result to rank out the candidates accordingly. Parallel to this module another algorithm test is given right after the technical Q&A session to check the coding capacity of the candidate. The result of this section is also passed to the fuzzy model to help with result.

D. Answer Evaluation Module

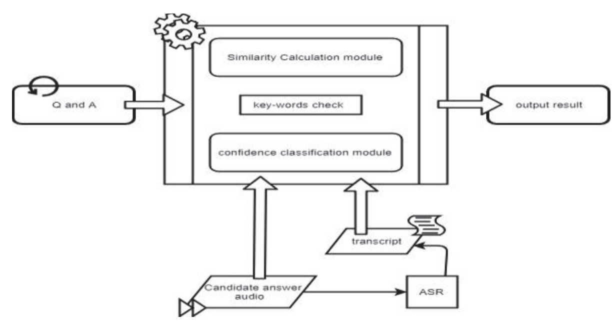

Fig. 5. High-level architecture of AEM

Generated Q and as need to be evaluated against the user’s answer. This mechanism should involve a potion which calculates the similarity between the two answers semantics as well as the quality of the answer. The reason behind having a quality factor for the answer measuring is that the same question could have multiple answers and the same answer could be expressed in several other ways, both which we could not evaluate through the ontology Q and A generation architecture. We propose several approaches for this matter. Checking the answers against extremely larger knowledge base which could be the internet. This approach is much confusing and very complicated when trying to implement. Another approach would be measuring the confidence the candidate is having when giving the answer. This would be a reasonable approach we can rely on when evaluating the answer. Because if the candidate knows the answer, the confidence in the answer given by the candidate is much higher. The confidence of the answer could be measured with respect to many aspects like eye contact, voice patterns, and hand gestures. In this paper only voice patterns classification would be considered.

1) Similarity Calculation

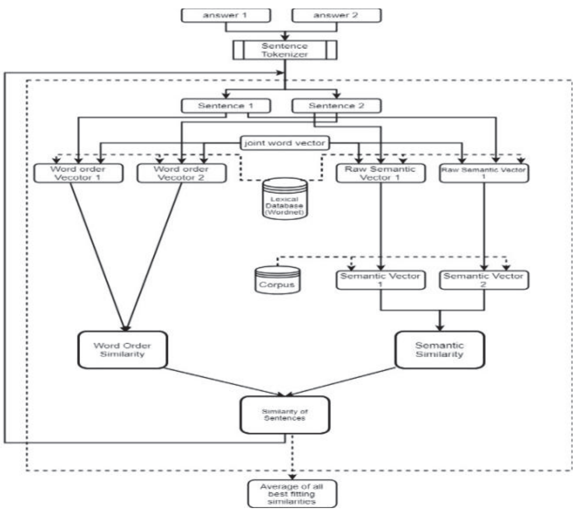

Fig. 6. Similarity calculation model

Similarity calculation module for the two answers was done using a semantic network and a corpus base architecture. The architecture is based on calculating the sentence similarities first and then calculate a similarity score for the entire answer. Word similarity would be calculated using the word net path length and depth measuring in the ontology. This word similarity would be a parameter for calculating the sentence semantic similarity. When calculating the sentence semantic similarity the information content is also considered. This was measure by the probability of occurrence in a larger corpus. More probable to occur means the information content is less. Then the word order similarity need to be calculated. This is calculated because the sentence that are having the same bag of word, which actually gives a different meaning, would get a best similarity. Finally, the overall sentence similarity is calculated using a weight value for each component.

The value need to be more than 0.5 as more potion need to be given for semantic similarity. The best similarity score getting sentence pairs sentence similarity would be accumulated with weight values given for each sentence. These weight values could be obtained from the Q and A generation module for each Q and A pairs.

2) Confidence Classification

The module is created using an RNN which was developed in order to classify the candidates answer is confident or not. Three labels were used for the classification. Confident, Moderately Confident, Not Confident. The data sets were obtained from Ted Talks and Columbia-SRI-Colorado (CSC) corpus. The labeling is done by a 10 members group of university students.

3) Preprocessing

Only some necessary talking sessions of the TED talk speech were taken and the labeling is done for each clipped version. Then the opening and ending silences were precisely clipped. Word transcript for each clip is arranged accordingly. And the labels as well.

4) Feature Extraction

MFCC features were extracted from the audio clips using the PyAudio Analysis python library and because the audio clips were different from the duration a normalization method needed to be implemented to get all the feature vectors into the same size. Paddings were used as the option to get them all in the same size. Then again some acoustic and prosodic features like energy, fundamental frequency computed from the Cepstrum snd various contour features were extracted using openSMILE software suite. The feature set is a preconfigured feature set in openSMILE which is largly known as INTERSPEECH 2009 Emotion Challenge feature set. In addition to that transcripts for the relevant clips were also used as input features/. These transcripts were encoded with GloVe vectors.

5) Model

The architecture used was a LSTM cell based RNN. The reason for choosing this architecture is that they recognize temporal patterns in audio data even if the data does not have the same number of time-steps We divided each clip into time-steps of 0.2s, and clipped the longer samples at 60s(because over 90% of the selected clips were under 60s duration. Thus, the no of time-steps for each input was capped at a maximum of 300 time-steps. 300 is a very large no of time-steps, therefore a special type of RNN cell called the LSTM cell is used. The dataset we used is a small one so there seems to have a data over fitting problem in the model, therefore needed to implement a dropout with a probability of remaining of 0.9. Thus, the final model was a single-level, single-directional LSTM with dropout. Total of 384 features in openSMILE were extracted in each time-step . lexical features couldn't be introduced in each time-step because the lexical features doesn't respond well in evenly distributed time-steps.

6) Overall Answer Evaluation

Finally in the overall answer evaluation process we got a weighted average of each potion text similarity, confidence in the answer, and key words occurrences. The weights are differ with respect to the type of the question.

TABLE II
TEXT SIMULATION

	Text Sim	Confidence	Keyword
Technical	0.7	0.2	0.1
HR Questions	0.5	0.45	.05

E. *Sort Ranking of Candidates using fuzzy inference system Approach*

The proposed fuzzy model has five fuzzy inference system based analysis blocks and behaves like FLC with two scalable inputs and one output. The output from a previous block will be an input to the next block until reaches a final output as a combination of all input parameters as follows.

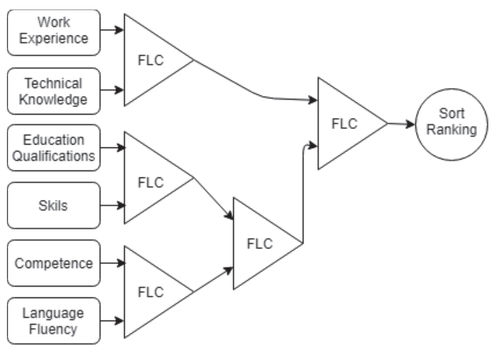

Fig. 7. High level architecture of FIS

Five steps are performed for each FLC namely fuzzification, application of the fuzzy operator in the antecedent, Implication from the antecedent to the consequent, Aggregation of the consequents across the rules and defuzzification as shown in the figure 8.

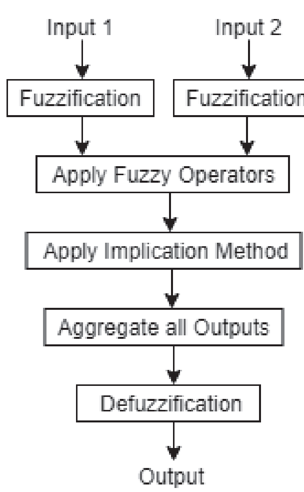

Fig. 8. Operations of FLC

1) *Fuzzification*

Fuzzification is the first step in the process of fuzzy reasoning. This involves a domain conversion in which crisp inputs are converted to fuzzy inputs by using membership functions.

Before defining the membership function, we must first allocate the appropriate number of clusters for each input data set. Developing a membership function based on the number of defined linguistics for each input data set will become a benchmark. The elbow method is a method of setting the optimal number of clusters. The first step is to define the number of clusters K to train, and then calculate the cost function for each cluster K starting from K = 2. Repeat this step and increase K by 1 until all K's cost function is calculated. In the case of the K value, a sharp drop before the cost reaches the platform means that the K value is the optimal K value for the number of clusters. [9].

2) *Apply Fuzzy Operator*

After fuzzification of given inputs, it can be seen that how satisfied each part of each rule's antecedent is. If there is more than one part in antecedent of a given rule, the fuzzy operator is applied to get one number that represents the antecedent of the rule. After that this number is applied to the output function. The input to the fuzzy operator is two or more membership values that come from fuzzified input variables. The output is a single truth value.

3) *Apply Implication Method*

Before doing the implication, weight of the rules must be determined. Each rule has a weight (a number between 0 and 1). This weight applies to the number given by the antecedent. In general, this weight is 1 and therefore has no effect on the implied process. Sometimes we may want to weigh a rule against other values by changing the weight value to something other than 1.

After each rule is assigned an appropriate weight, the implication method is implemented. The result (Consequent) is a fuzzy set represented by a membership function that appropriately trades off its linguistic features. The consequences are reconstructed using functions associated with the antecedent. The input to the implication process is a single number that is given by the antecedent, and the outputs are fuzzy sets. The implication is implemented for each and every rule.

4) *Aggregate all Outputs*

Since the decision is based on testing all the rules in the FIS, the rules must be combined in some way to make a decision. Aggregation is the process of grouping the fuzzy sets that represent the output of each rule into a single fuzzy set. For each output variable, aggregation is performed only before the fifth and final steps. The input to the aggregation process is a list of truncated output functions returned by the implied process for each rule. The output of the aggregation is a fuzzy set of each output variable.

5) *Defuzzification*

The input to the defuzzification process is a fuzzy set (aggregated output fuzzy set) and the output is a single number. Although ambiguity facilitates rule evaluation in intermediate steps, the final expected output of each variable is usually a single number. However, the set of fuzzy sets contains a series of output values and therefore must be defuzzified to resolve a single output value in the set. Center of Gravity method is used for defuzzification.

After defuzzification is done the process of ranking of the candidates can be performed. According to the given outputs of all candidates it can be seen that the recruitment process is more accurate and helpful for recruiters.

IV. RESULT AND DISCUSSION

Five positions can be ranked separately and here we discuss only ranking of software engineers. With 96 test data, we identified final output as shown in table III.

TABLE III
FINAL OUTPUT OF SOFTWARE ENGINEER POSITION, CLASSICAL RANKING AND FUZZY RANKING

Name	Classical value	FIS approached Value	Classical Rank	Fuzzy Rank
Achira	89.2023166	52.4327581668	2	1
Lasitha	90.4457854	50.4951110426	1	2
Sahan	88.54	50.4515916064	3	3
Nilantha	88.54	50.0726402444	3	4
Yasas	85.5548	49.987443564	5	5
Tharindu	85.1458	49.9305851932	6	6

According to the table III, it can be seen that in classical ranking human evaluators cannot take a clear decision when two candidates have same manual score. Although both candidates get the same classical ranking, their fuzzy ranking is different. These cases show that the proposed framework for the FIS approach can overcome this problem when it is difficult for human evaluators to make a definitive decision.

V. CONCLUSIONS

As for the further works, we are proposing an auto generated ontology using web data, which is always up to date. Here the system will have authentication and permissions to download RDF data of each specific web sites in order to download those. After the data, gathering the RDF dataset will be read and gathered in a structured manner to be faded in new ontology creation step. In addition, this generated ontology will be then merged with the system generated ontology which is constructed with primary considerable know areas of the organization. In addition to this, we can use optimization techniques to define membership functions.

REFERENCES

[1] Sayed Zainul Abideen Mohd Sadiq, Juneja Afzal Ayub, Gunduka Rakesh Narsayya, Momin Adnan Ayyas,Prof. Khan Tabrez Mohd. Tahir, "Intelligent Hiring with Resume Parser and Ranking using Natural Language Processing and Machine Learning," International Journal of Innovative Research in Computer and Communication Engineering, vol. 4, no. 4, p. 8, April 2016.

[2] Satyaki Sanyal,Souvik Hazra,Soumyashree Adhikary, "Resume Parser With Natural Language Processing," Satyaki Sanyal, KIIT University, India, 2017.

[3] Jeff Anderson (2005). Mechanically Inclined:Building “Grammar, Usage, and Style into Writer's Workshop” Stenhouse Publishers Portland, ME. K. Elissa, “Title of paper if known,” unpublished.

[4] Liane Acker, James Lester, Art Souther and Bruce Porter, Eds. (1991). “Generating Coherent Explanations to Answer Students' Questions. Intelligent Tutoring Systems: Evolutions in Design”, Psychology Press, New York.

[5] Arthur C. Graesser and Natalie K. Person (1994). “Question asking during tutoring” American Educational Research Journal, 31(1): 104-137.

[6] L.A Latumakulita, F. Purnama, T. Usagawa, S. Paturusi, D.A Prima, “Indonesia Scholarship Selection Framework Using Fuzzy Inferences System Approach”, 2016 International Conference on Information, Communication Technology and System (ICTS), pp. 107 –113.

[7] A. Susanto, L. Latifah, Nuryasin, A.Fitriyani “Decision Support Systems Design on Sharia Financing using Yager’s Fuzzy Decision Model”, 5th International Conference on Cyber and IT Service Management (CITSM) 2017.

[8] M.Solwsvik, Y.Kondratenko, G.Kondratenko, I.Sidenko, V,Kharchenko, A.Boyarchuk, “Fuzzy Decision Support Systems in Marine Practice”, IEEE International Conference on Fuzzy Systems 2017.

[9] P. Bholowalia and A. Kumar, “EBK-Mean: A Clustering Technique based on Elbow Method and K-Mean in WSN”, International Journal of Computer Application (0975-8887), Vol. 105 – No. 9, November 2014.