An Approach for Analyzing ISO / IEC 25010 Product Quality Requirements based on Fuzzy Logic and Likert Scale for Decision Support Systems

Hasnain Iqbal
Computing and Technology
Iqra University
Islamabad, Pakistan

Muhammad Babar
Computing and Technology
Iqra University
Islamabad, Pakistan

Abstract*—Decision Support Systems (DSS) are collaborative software systems that are built to support controlling of an organization in decision making process when faced with non-routine problems in a specific application domain. It's important to measure portability, maintainability, security, reliability, functional suitability, performance efficiency, compatibility, and usability quality requirements of DSS properly. ISO / IEC 25010 which replaced ISO 9126, used for three different quality models for software products, such as: a) Quality in use model, b) Product quality model, and c) Data quality model. There is a lack of methodologies to measure and quantify these quality requirements. Fuzzy logic used to specify quality requirements of DSS, because it's an approach to computing based on degrees of truth, rather than true or false logics. Likert scale is a method in which it converts qualitative values into quantitative values to make a best statistical analysis. The measurement and quantification of quality requirements of DSS is a challenging task, because these quality requirements are in qualitative form and can't be represented in quantitative way. Although, several quality requirements methods for DSS have been proposed so far, but the research on analyzing quality requirements of DSS are still limited. In this paper, quantitative approach proposed for analyzing ISO / IEC 25010 product quality requirements based on fuzzy logic and likert scale for DSS which aims to quantify quality requirements. Moreover implemented proposed framework on a case study 'Internet Banking' and got data from 25 respondents i.e. System Analysts and Domain Experts of banking sector.

Keywords—ISO / IEC 25010; Product Quality Requirements; Fuzzy Logic; Likert Scale; Functional Requirements; Non-Functional Requirements; Internet Banking; Decision Support Systems

I. Introduction

In this paper, mainly focused for analyzing ISO / IEC 25010 product quality requirements based on the fuzzy logic and likert scale for DSS which aims to quantify the quality requirements.

ISO / IEC 25010 "Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – Systems and Software Quality Models"; which replaced ISO 9126 "Software Engineering – Product Quality", used for three different quality models for software products: 1) Quality in use model, 2) Product quality model, and 3) Data quality model [8].

DSS are collaborative software systems that are built to support the controlling of an organization in decision making process when faced with non-routine problems in a specific application domain. It's important to measure portability, maintainability, security, reliability, functional suitability, performance efficiency, compatibility, and usability quality requirements of DSS properly.

Non-Functional Requirements (NFRs) are requirements that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors of the system. These requirements must be distinguished with the major Functional Requirements (FRs) that define specific behavior or functions of the systems. There are some major quality attributes of NFRs framework that must be measurable before start working on the system like; risk analysis, configurability, modifiability, performance, efficiency, traceability, recoverability, reliability, reusability, security, availability, interfaces, design constraints, and failure management.

There is a lack of methodologies to measure and quantify these quality requirements. Fuzzy logic used to specify quality requirements of DSS, because it's an approach to computing based on the degrees of the truth, rather than the true or false logics.

As a scaling method, likert scale is a method in which it converts qualitative values into quantitative values to make a best statistical analysis. It is commonly used to measure defendant's attitudes or behaviors by asking the extent to which they agree or disagree with a particular statement.

The measurement and quantification of the quality requirements of DSS is a challenging task, because these quality requirements are in the qualitative form and can't be represented in a specific quantitative way. Although, several quality requirements methods for DSS have been proposed so far, but the research on analyzing quality requirements of DSS is limited.

In this study, our objectives to provide a quantitative approach for analyzing ISO / IEC 25010 product quality requirements based on the fuzzy logic and likert scale for DSS which quantify the quality requirements.

The rest of the paper is organized as follows. The literature review of product quality requirements of DSS discussed in Section II. Section III contained proposed framework that consists upon five steps. Section IV validated the results of respective 280 rules. Section V consists of a case study and finally Section VI concluded the paper.

II. Literature Review

In [10], discussed about DSS Life Cycle and highlighted its importance for interactive software systems while decision making process for an organization when facing non-repetitive difficulties in a specific application domain. For this, author mentioned a list of NFRs that were divided into three categories i.e. DSS development and pre-development, DSS operation, and DSS maintenance and evaluation.

In [1], highlighted NFRs importance, implementation, and its overall effects on software architecture. Author proposed an approach that provides decision support in a software development process for designing decision model in the field of NFRs. By using this, developer's productivity will be increase by reusing design decisions.

In [2], proposed a quantitative approach that based on fuzzy logic and Alpha cut approach which objectives to achieve process of prioritizing NFRs. Proposed approach divided into four different steps; first step identify FRs and NFRs, second step generates decision matrix (n x m), third step stimulate importance degree of each NFR with admiration to each FR, whereas fourth step calculates all NFRs with respect to all FRs by using fuzzy logic and Alpha cut approach.

In [13], presented a fuzzy model for software reliability prediction. Authors proposed three parameters i.e. availability, failure probability, and recoverability for combined measure of the software reliability. Proposed approach also helped to progressed intermediary stages among reliable and unreliable state of a system.

In [17], showed an approach that participates FRs, measurable NFRs, and scalable NFRs. Authors originated use of fuzzy logic and likert scale for treatment of separately quantifiable as well as scalable NFRs.

In [11], discussed capability of fuzzy logic in control fuzziness and ambiguity to come up with an efficient maintainability prediction model. Authors proposed a model that was constructed using by object-oriented metrics data as there are at-least two major important sources of information for building the prediction model, such as: historical data, and human experts.

In [15], survey paper reviewed an improvement of performance of DSS to meet the challenges and development of integrated DSS. It determines that by measuring integration, well support will be provided to decision makers, with anticipation of both better decisions and enhanced decision making processes.

In [8], discussed about measuring performance of cloud computing based applications by using ISO / IEC 25010 quality characteristics. Authors used Bautista's proposed performance measurement framework for measuring overall performance of cloud computing based applications. There were three key challenges become deceptive as a result of this case study analysis, such as: collecting, processing, and representing data.

In [3], highlighted importance of measuring software quality in use, also described that why software quality in use measurement is so much difficult especially in the e-government applications, embedded systems, and mobile based applications. Authors divided paper into two contributions: a) classification and definition of key issues and challenges while measuring software quality in use in context of ISO SQuaRE series, and b) prediction of software quality in use.

In [4], discussed about NFRs, as it's difficult to identify them for specific domains. Authors introduced model based approach that based on fuzzy logic and DSS, which helped to classify different design alternatives. Proposed approach were accomplished by building a model of the NFRs and then performing analysis on the model.

In [19], discussed about major difference between business intelligence and decision support systems or applications. Authors also highlighted software decision making difficulties while taking any decisions and focused on two basic types of software solutions that used to support software decision making, such as: DSS, and business intelligence.

In [6], highlighted that most of time project fails due to NFRs. Authors mentioned that NFRs are very vital in any software project that supports in finalizing major functionality of system. Authors declared that NFRs is very difficult to identify, so in most of cases developers ignored NFRS regardless of significant their importance in functionality of system.

In [14], discussed about importance of NFRs for an effective development and deployment of software product. Authors projected a four layered analysis approach for identification of NFRs, and some rules also proposed for each layer. Proposed approach successfully applied on two case studies i.e. online library management system, and ATM system. They identified NFRS and then validated by using a check list.

In [7], discussed regarding importance of service oriented architecture in organizations and underlined that quality should be preserved as a key issue. Authors mentioned that there is a need for development of a specific quality model for service oriented architecture based on the latest ISO / IEC 25010.

In [18], addressed about clashes among NFRs that identified individually, whereas existing approaches were fail to detention nature of clashes among those NFRs. Proposed framework categorizes and examines the clashes that based on relationships among quality attributes, functionalities and constraints.

In [16], highlighted about importance of NFRs in software architecture and its contribution to success of a software project. Authors identify different types of NFRs that based on different types of systems and application domains, and originate that there are some other NFRs which have no explained yet. Only 20.18% NFRs have definition and attributes, 26.32% NFRs have definition, whereas 53.51% NFRs were without definition and attributes.

In [12], discussed about NFRs that how much these are difficult to software engineers for many years, although since long time different methods and techniques have been proposed to improve the elicitation, documentation, and validation. Authors mentioned that by knowing more about these issues will beneficial for both parties i.e. practitioners and researchers in their daily routine work. Authors presented an empirical study which based on thirteen interviews with software architects.

In [9], highlighted regarding quality attributes, eliciting quality attributes requirements, quality attribute workshop and quality attribute workshop eight steps, and quality attribute scenarios. Paper were consists on these questions: a) what is the best time to specify quality attribute requirements, b) what is an approach that an organization uses to identify quality attributes requirements.

In [5], showed importance of online banking for development and improvement over the world and manipulating organizations, society and individuals. NFRs are as important as NFs, and NFRs should be specify in initial phase. Many of software projects fails due to not considering NFRs. NFRs such as accuracy, usability, security and performance are regularly critical to online banking system. For conducting survey, authors set a questionnaire and send to 122 online banking customers and measured results.

However in above mentioned studies, no particular method or approach has been proposed for analyzing ISO / IEC 25010 product quality requirements for DSS based on fuzzy logic and likert scale. In this study, proposed an enhanced approach for analyzing ISO / IEC 25010 product quality requirements for DSS based on fuzzy logic and likert scale. By this approach, we can classify different quality requirements of DSS from multiple views of stakeholders, that how much quality requirements are High Important, Important, Low Important, or Not Important.

III. PROPOSED FRAMEWORK

In order for analyzing, maintaining, and determining the quality requirements of DSS; proposed a framework as showed in TABLE I.

TABLE I. STEPS OF PROPOSED FRAMEWORK FOR ANALYZING QUALITY REQUIREMENTS FOR DSS

Step No.	Description
1	Compare quality requirements of DSS with ISO / IEC 25010:2012 with respect to the product quality
2	Set the values of importance of quality requirements by using likert scale
3	Use of Fuzzy Model four modules i.e. Rule Base, Fuzzification, Inference Engine, and Defuzzification for determining the quality requirements
4	Calculating quality requirements of DSS and plot values by using Mamdani Style Inference Mechanism
5	Defuzzify the fuzzified outputs by using Joint Membership Function plotting on Two-Dimensional Surface View

Fig. 1. shows overall steps of proposed framework for analyzing quality requirements of DSS.

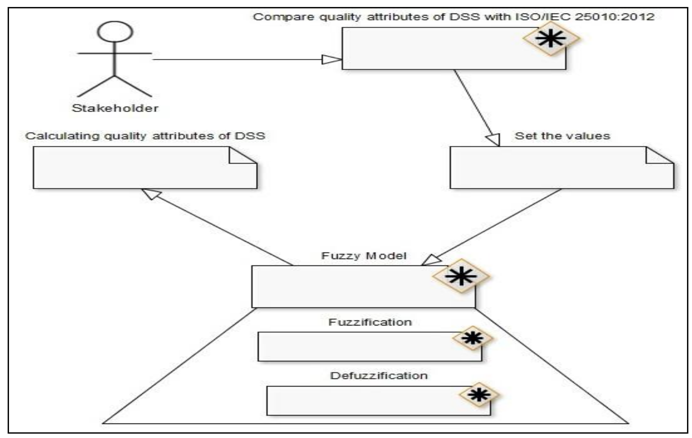

Fig. 1. Steps of Proposed Framework for Analyzing Quality Requirements of DSS

Step 1

Comparison between quality requirements of DSS with ISO / IEC 25010:2012 with respect to the product quality:

TABLE II. COMPARISON BETWEEN ISO / IEC 25010 PRODUCT QUALITY REQUIREMENTS OF DSS

Sr.	**ISO / IEC 25010: 2012 General Characteristics (Product Quality)**	**ISO / IEC 25010: 2012 Sub – Characteristics (Product Quality)**	**Quality Requirements of DSS [1]**
1	Portability	Adaptability	YES
		Installability	YES
		Replaceability	YES
2	Maintainability	Modularity	YES
		Reusability	YES
		Analyzability	YES
		Modifiability	YES
		Testability	YES
3	Security	Confidentiality	YES
		Integrity	YES
		Non-repudiation	YES
		Accountability	YES
		Authenticity	YES
4	Reliability	Maturity	YES
		Availability	YES
		Fault Tolerance	YES
		Recoverability	YES
5	Functional Suitability	Functional Completeness	YES
		Functional Correctness	YES
		Functional Appropriateness	YES
6	Performance Efficiency	Time Behavior	YES
		Resource Utilization	YES
		Capacity	YES
7	Compatibility	Co-existence	YES
		Interoperability	YES
8	Usability	Appropriateness Recognisibility	YES
		Learnability	YES
		Operability	YES
		User Error Protection	YES
		User Interface Aesthetics	YES
		Accessibility	YES

Step 2

Set the values of importance of quality requirements by using the Likert Scale. Here Likert Scale will give a value to each quality requirement of DSS as shown in TABLE III. The inputs of quality requirements of DSS are Portability (PORT), Maintainability (MAIN), Security (SEC), Reliability (REL), Functional Suitability (SUIT), Performance Efficiency (PER), Compatibility (COMP), and Usability (USA). Levels of all eight inputs nominal values are: Portability, Maintainability, Security, Reliability, Functional Suitability, Performance Efficiency, Compatibility, Usability = {High Important (I_H), Important (I), Low Important (I_L), Not Important (I_N)}.

TABLE III. LIKERT SCALE FOR NOMINAL VARIABLES WITH ACTUAL VALUES OF PORTABILITY, MAINTAINABILITY, SECURITY, RELIABILITY, FUNCTIONAL SUITABILITY, PERFORMANCE EFFICIENCY, COMPATIBILITY, AND USABILITY

Portability, Maintainability, Security, Reliability, Functional Suitability, Performance Efficiency, Compatibility, Usability	
Nominal Variables	**Actual Values**
High Important (I_H)	4
Important (I)	3
Low Important (I_L)	2
Not Important (I_N)	1

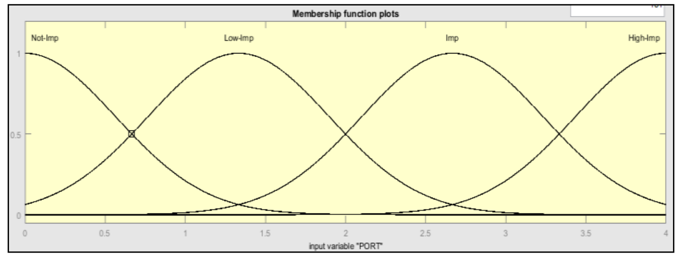

Fig. 2. Fuzzification of Input Variable: PORT (Portability)

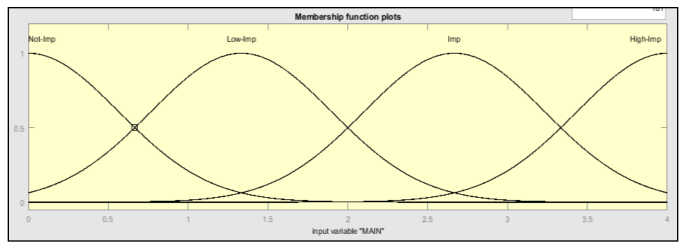

Fig. 3. Fuzzification of Input Variable: MAIN (Maintainability)

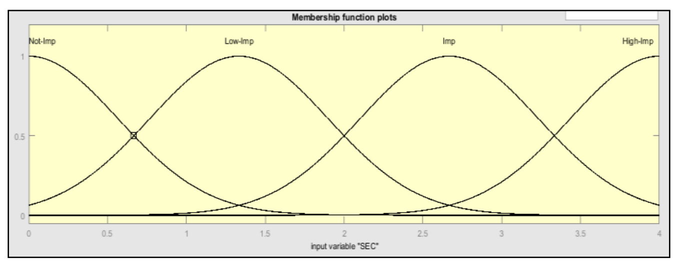

Fig. 4. Fuzzification of Input Variable: SEC (Security)

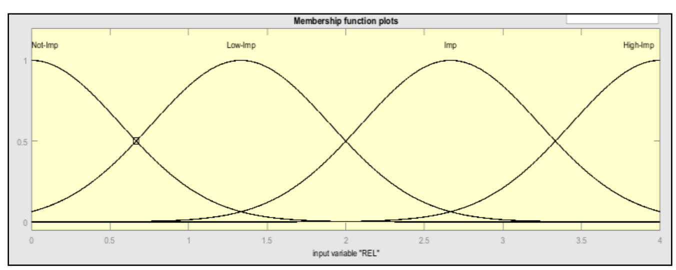

Fig. 5. Fuzzification of Input Variable: REL (Reliability)

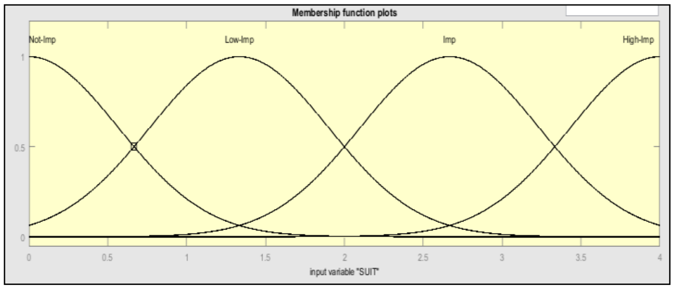

Fig. 6. Fuzzification of Input Variable: SUIT (Functional Suitability)

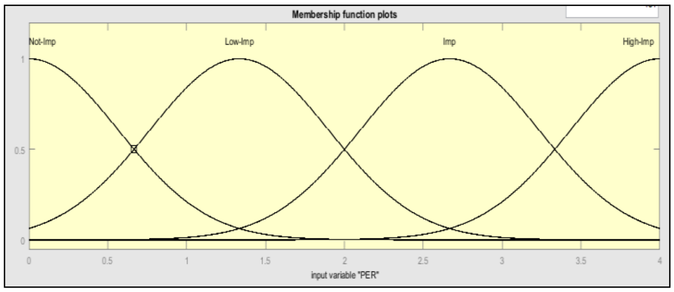

Fig. 7. Fuzzification of Input Variable: PER (Performance Efficiency)

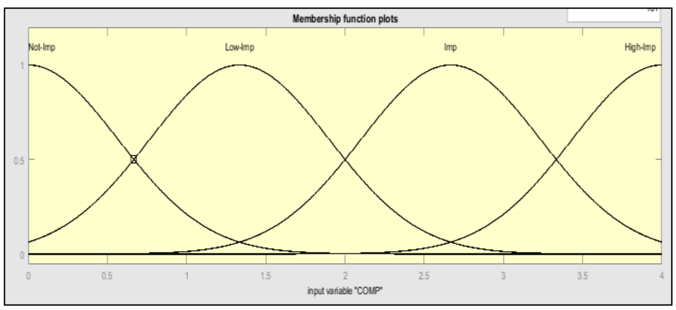

Fig. 8. Fuzzification of Input Variable: COMP (Compatibility)

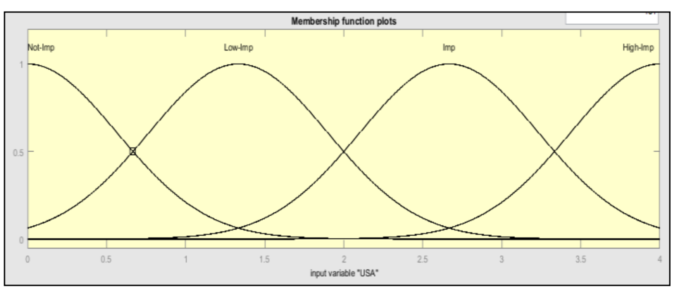

Fig. 9. Fuzzification of Input Variable: USA (Usability)

Step 3

Fuzzy Model is a greatest choice for analyzing, maintaining, and determining the quality requirements of DSS in the form of quantitative way. Here we used Fuzzy Model four modules i.e. Rule Base, Fuzzification, Inference Engine, and Defuzzification for this as shown in Fig. 10.

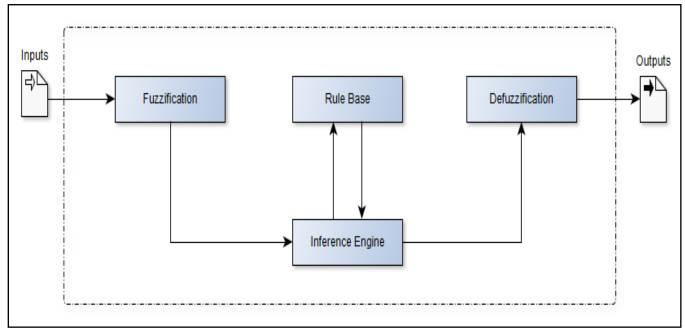

Fig. 10. Fuzzy Logic Controller Block Diagram (Fuzzy Model)

Fig. 10. describes the Fuzzy Logic Controller Fuzzy Model that converts the crisp inputs into the fuzzy values, after that these values are handled by the Inference Engine in Fuzzy Domain via Rule Base. Finally the handled output is converted from fuzzy domain to the crisp domain by the defuzzification module [11].

Here we fuzzified the inputs of quality requirements of DSS by using rule base Fuzzification and assign them Product Quality Range Values as shown in TABLE IV.

TABLE IV. LIKERT SCALE FOR NOMINAL VARIABLES WITH ACTUAL RANGE VALUES OF PRODUCT QUALITY

Nominal Variables	**Actual Range Values**
High	24 – 32
Average	17 – 23
Low	0 – 16

TABLE IV. shows likert scale for nominal variables with actual range values of product quality. Here we used Product Quality High (PQ_{High}), Product Quality Average ($PQ_{Average}$), and Product Quality Low (PQ_{Low}) as a nominal values having (24 - 32), (17 - 23), and (0 - 16) actual range values respectively.

After completing the Fuzzification, we defuzzify the fuzzified output and then plot them by using MATLAB Fuzzy Tool Box through Mamdani Style Inference Mechanism and displayed the results.

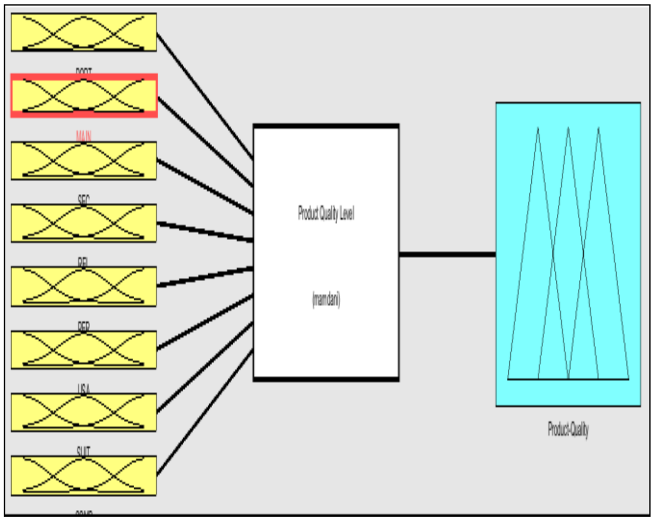

Fig. 11. Product-Quality Model Diagram by Mamdani Style Inference Mechanism

Fig. 11. describes the product-quality model diagram by using Mamdani Style Inference Mechanism. Here inputs parameters for Mamdani Style Inference Mechanism are PORT, MAIN, SEC, REL, SUIT, PER, COMP, and USA. Whereas Product-Quality is the output of Product-Quality Model.

The proposed framework integrates the quality requirements of DSS that analyzes PORT, MAIN, SEC, REL, SUIT, PER, COMP, and USA for determining Product-Quality level based on the following rule base Fuzzification. Here all 280 possible sets of inputs (rules) are consider, as per given Mathematical Combination Formula:

$$C(n, r) = n! / (r!(n-r)!)$$
$$C(8, 4) = 8! / (4!(8-4)!)$$
$$C(n, r) = 70$$

In the given formula, 'C' used for Total Combinations, 'n' used for 'Total Number of Objects / Parameters', whereas 'r' used for 'Total Number of Elements' without any repetition. In this study, total eight objects / parameters used i.e. PORT, MAIN, SEC, REL, SUIT, PER, COMP, and USA. Whereas 4 total numbers of elements i.e. I_N, I_L, I, and I_H have been used.

As in this study, we used four elements for each parameter (total 8 parameters), so that for calculating total inputs multiply possible combinations (70 Combinations) with 4, such as:

$$70 \times 4 = 280 \text{ Total Inputs}$$

These rules are classified as Product Quality High (PQ_{High}), Product Quality Average ($PQ_{Average}$), and Product Quality Low (PQ_{Low}) as given below in TABLE V (all 280 rules given in Annexure A):

TABLE V. ANALYZING PORTABILITY, MAINTAINABILITY, SECURITY, RELIABILITY, FUNCTIONAL SUITABILITY, PERFORMANCE EFFICIENCY, COMPATIBILITY, AND USABILITY QUALITY REQUIREMENTS OF DSS FOR DETERMINING PRODUCT QUALITY LEVEL

Sr.	(Quality Requirements Inputs)								Product Quality Level
	PORT	MAIN	SEC	REL	SUIT	PER	COMP	USA	
1	I_N	I_N	I_N	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
28	I_L	I_N	I_H	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
56	I	I_N	I_N	I_N	I_H	I_N	I_N	I_N	PQ_{Low}
84	I_H	I_N	I_N	I_N	I_N	I_N	I_H	I_N	PQ_{Low}
112	I	I	I_N	I_N	I_L	I_N	I_N	I_N	PQ_{Low}
140	I_H	I_H	I_N	I_N	I_N	I_N	I_N	I	PQ_{Low}
168	I	I	I	I_N	I_N	I_N	I_H	I_N	$PQ_{Average}$
196	I_L	I_L	I_L	I_L	I_N	I_N	I_N	I_L	PQ_{Low}
224	I_L	I_L	I_L	I_L	I_L	I_L	I_N	I_N	PQ_{Low}
252	I_L	I_L	I_L	I_L	I_L	I_L	I	I_N	PQ_{Low}
256	I_L	I_L	I_L	I_L	I_L	I_L	I_N	I_H	$PQ_{Average}$
260	I	I	I	I	I	I	I_N	I_L	$PQ_{Average}$
276	I	I	I	I	I	I	I	I_H	PQ_{High}
280	I_H	I_H	I_H	I_H	I_H	I_H	I_H	I_H	PQ_{High}

TABLE V. shows respective rules (all 280 rules given in Annexure A) for analyzing PORT, MAIN, SEC, REL, SUIT, PER, COMP, and USA quality requirements of DSS for determining Product-Quality Level, that were classified as Product Quality High (PQ_{High}), Product Quality Average ($PQ_{Average}$), and Product Quality Low (PQ_{Low}).

Step 4

All 280 possible rules were implanted and then created a rule base. In this model, Mamdani Style Inference Mechanism has been castoff. Output variable Product-Quality is observed by using the MATLAB Fuzzy Tool Box for a particular 280 sets of inputs. For respective given set of input parameters i.e. [PORT, MAIN, SEC, REL, SUIT, PER, COMP, USA] as [3.25, 3, 3.75, 3.25, 3, 3.50, 3, 3.25] and then Rule Viewer helps to realize the output Product-Quality level generated i.e 24.6 corresponding to this assumed set of input variables which is shown at Fig. 12.

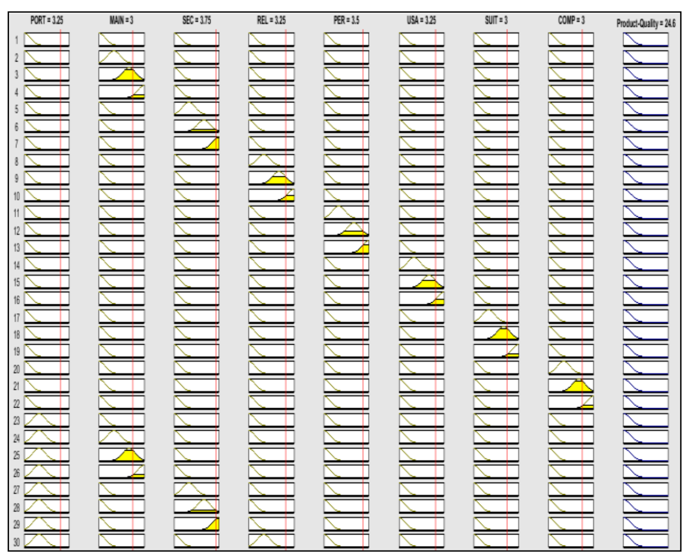

Fig. 12. Rule Viewer for Product-Quality Model

Step 5

After creating all possible 280 rules by using MATLAB Fuzzy Tool Box through Mamdani Style Inference Mechanism, we defuzzify the fuzzified outputs by using the Joint Membership Function by plotting on Two-Dimensional Surface View, as shown in Fig. 13. to 26 accordingly:

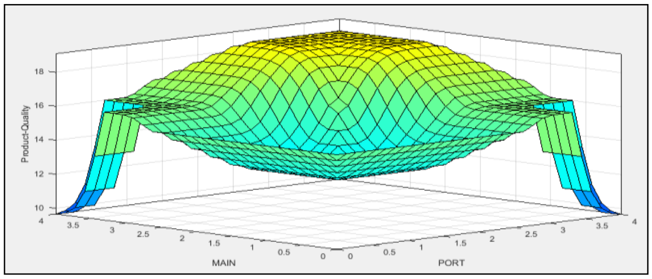

Fig. 13. Two-Dimensional Surface View with PORT (input) on X-axis, MAIN (input) on Y-axis, Product-Quality (output) on Z-axis

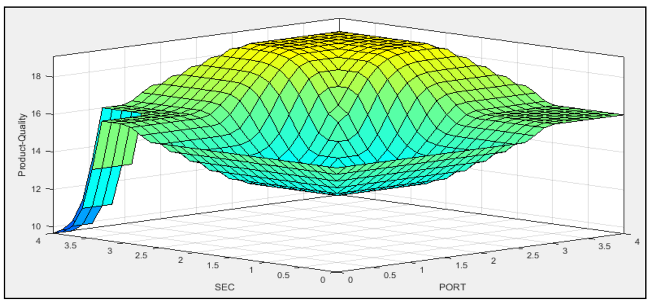

Fig. 14. Two-Dimensional Surface View with PORT (input) on X-axis, SEC (input) on Y-axis, Product-Quality (output) on Z-axis

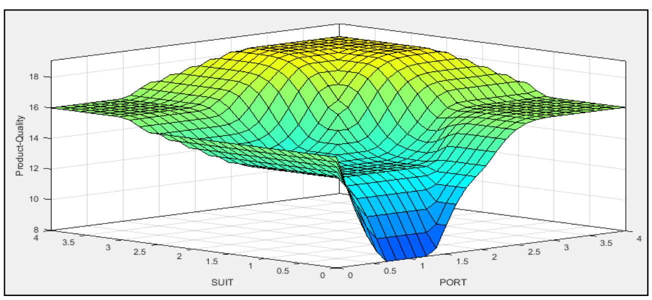

Fig. 15. Two-Dimensional Surface View with PORT (input) on X-axis, SUIT (input) on Y-axis, and Product-Quality (output) on Z-axis

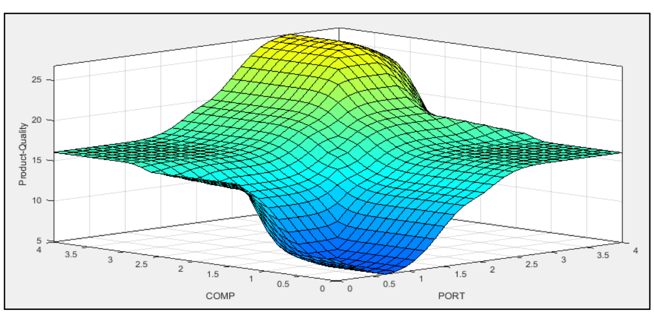

Fig. 16. Two-Dimensional Surface View with PORT (input) on X-axis, COMP (input) on Y-axis, and Product-Quality (output) on Z-axis

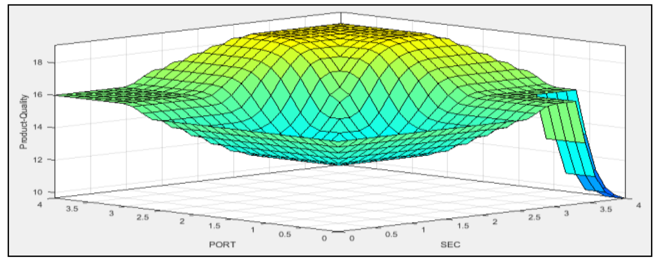

Fig. 17. Two-Dimensional Surface View with SEC (input) on X-axis, PORT (input) on Y-axis, and Product-Quality (output) on Z-axis

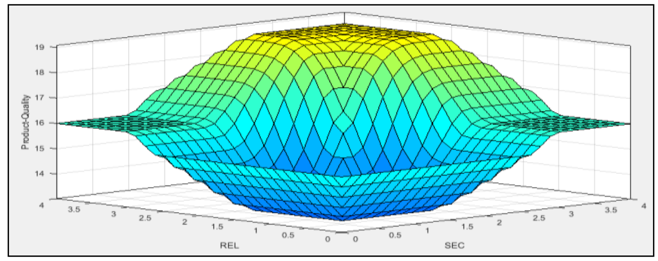

Fig. 18. Two-Dimensional Surface View with SEC (input) on X-axis, REL (input) on Y-axis, and Product-Quality (output) on Z-axis

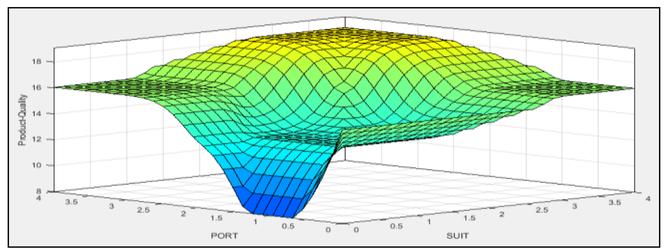

Fig. 19. Two-Dimensional Surface View with SUIT (input) on X-axis, PORT (input) on Y-axis, and Product-Quality (output) on Z-axis

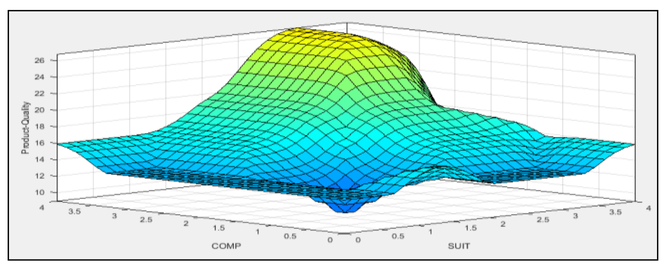

Fig. 20. Two-Dimensional Surface View with SUIT (input) on X-axis, COMP (input) on Y-axis, and Product-Quality (output) on Z-axis

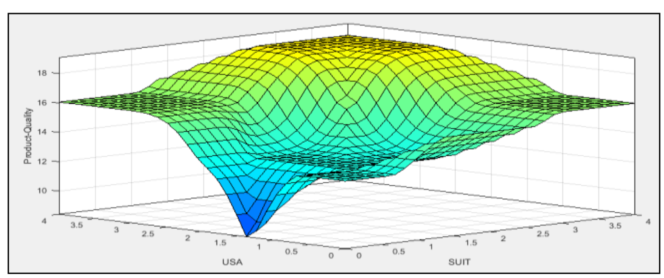

Fig. 21. Two-Dimensional Surface View with SUIT (input) on X-axis, USA (input) on Y-axis, and Product-Quality (output) on Z-axis

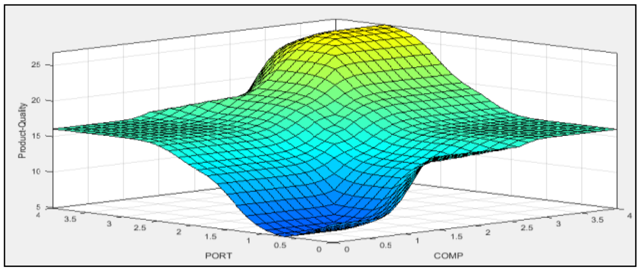

Fig. 22. Two-Dimensional Surface View with COMP (input) on X-axis, PORT (input) on Y-axis, and Product-Quality (output) on Z-axis

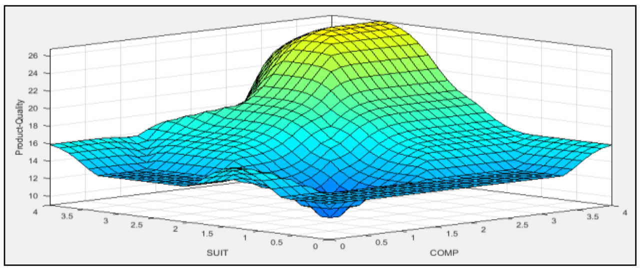

Fig. 23. Two-Dimensional Surface View with COMP (input) on X-axis, SUIT (input) on Y-axis, and Product-Quality (output) on Z-axis

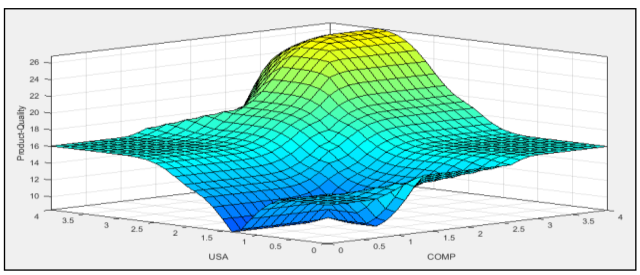

Fig. 24. Two-Dimensional Surface View with COMP (input) on X-axis, USA (input) on Y-axis, and Product-Quality (output) on Z-axis

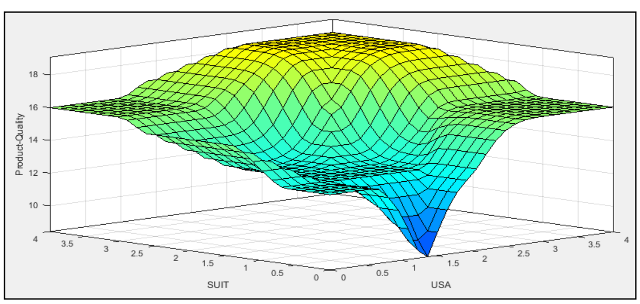

Fig. 25. Two-Dimensional Surface View with USA (input) on X-axis, SUIT (input) on Y-axis, and Product-Quality (output) on Z-axis

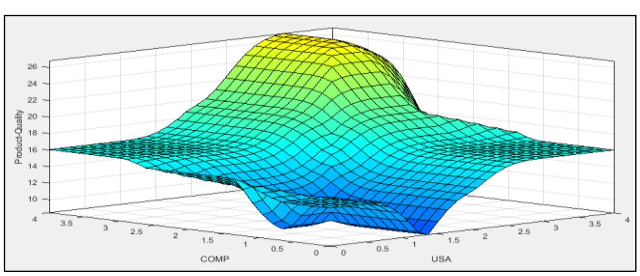

Fig. 26. Two-Dimensional Surface View with USA (input) on X-axis, COMP (input) on Y-axis, and Product-Quality (output) on Z-axis

IV. RESULTS

After doing experiments on respective rules (as given in Annexure A), got following results as given on TABLE VI. regarding eight parameters i.e. PORT, MAIN, SEC, REL, SUIT, PER, COMP, and USA with respect to Product-Quality Level and Membership Grade of Product-Quality:

TABLE VI. PRODUCT QUALITY LEVEL AND MEMBERSHIP GRADE OF PRODUCT QUALITY FOR GIVEN RULES

Rule No.	PORT	MAIN	SEC	REL	SUIT	PER	COMP	USA	PQ Level	Membership Grade of Product-Quality
1	I_N	I_N	I_N	I_N	I_N	I_N	I_N	I_N	Low (8)	Min 1, Max 16
28	I_L	I_N	I_H	I_N	I_N	I_N	I_N	I_N	Low (12)	Min 1, Max 16
56	I	I_N	I_N	I_N	I_H	I_N	I_N	I_N	Low (13)	Min 1, Max 16
84	I_H	I_N	I_N	I_N	I_N	I_N	I_H	I_N	Avg (14)	Min 17, Max 23
112	I	I	I_N	I_N	I_L	I_N	I_N	I_N	Avg (13)	Min 17, Max 23
140	I_H	I_H	I_N	I_N	I_N	I_N	I_N	I	Avg (16)	Min 17, Max 23
168	I	I	I	I_N	I_N	I_N	I_H	I_N	Avg (17)	Min 17, Max 23
196	I_L	I_L	I_L	I_L	I_N	I_N	I_N	I_L	Low (12)	Min 1, Max 16
224	I_L	I_L	I_L	I_L	I_L	I_L	I_N	I_N	Low (14)	Min 1, Max 16
252	I_L	I_L	I_L	I_L	I_L	I_L	I	I_N	Low (16)	Min 1, Max 16
256	I_L	I_L	I_L	I_L	I_L	I_L	I_N	I_H	Avg (17)	Min 17, Max 23
260	I	I	I	I	I	I	I_N	I_L	Avg (21)	Min 17, Max 23
276	I	I	I	I	I	I	I	I_H	High (25)	Min 24, Max 32
280	I_H	I_H	I_H	I_H	I_H	I_H	I_H	I_H	High (32)	Min 24, Max 32

V. CASE STUDY

Internet banking is a major innovation in the field of banking. Earlier banking was in a very traditional manner, and there were no such innovations. Internet Banking is actually a facility under which the customers can perform the basic banking transactions electronically, round the clock throughout the world.

A system of banking in which customers can view their account details, pay bills, and transfer money through personal computers or from other devices by means of the internet. Normally internet banking provides account information, bill payments, online shopping payments, ticket booking, recharging prepaid phone, fund transfer, insurances services, investments services, credit cards facilities, and general customer services.

Data Collection Procedure

Useful data was collected from system analysts and domain experts of banking sector. For this, a set of questionnaire was given to respondents. This questionnaire was divided into two parts. First part of the questionnaire covers overall importance of product quality requirements for internet banking. whereas second part of questionnaire comprises how much each product quality requirement is important for internet banking by using scale from 1 to 4, such as: Not Important = 1, Low Important = 2, Important = 3, and High Important = 4.

25 respondents provided essential information regarding product quality requirements of internet banking, and further mentioned that which product quality requirement is high important for internet banking.

Data Analysis

While gathering the information, it was perceived that 72% of males and 28% of females were respondents. In which 44% of respondents were belongs to the age of above 45 years, 28% of respondents were belongs to the age of 35 - 44 years, 20% respondents were belongs to the age of 26 - 34 years, and 08% of respondents were belongs to the age of 18 - 25 years.

Information gathered analyzed and total weightage score were computed from observation of analysis. It was discovered that in the online banking, respondents were more worried about the security followed by performance efficiency, usability, reliability, portability, compatibility, maintainability, and functional suitability of internet banking services as per graph indicated in Fig. 27.

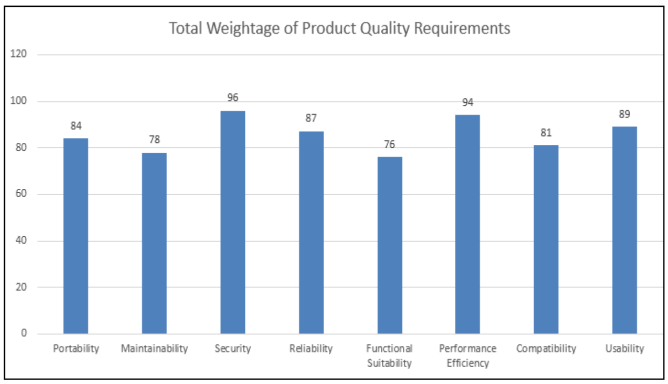

Fig. 27. Total Weightage of Product Quality Requirements

During that survey it was observed that security is the most important product quality requirement having 96 total weightage for internet banking from all other product quality requirements, so that is the reason Security is at first place. Here security consists of confidentiality, integrity, non-repudiation, accountability, and authenticity. Performance efficiency is also very important product quality requirement having 94 total weightage for internet banking, so that's why performance efficiency is at second place. Performance efficiency comprises of time behaviour, resource utilisation, and capacity. To have the disturbance free operations, respondents were concerned about usability having 89 total weightage of the internet banking that is at third place. Usability comprises on appropriateness recognisibility, learnability, operability, user error protection, user interface aesthetics, and accessibility. Reliability is at fourth place having 87 total weightage that consists of maturity, availability, fault tolerance, and recoverability. Portability is at fifth place having 84 total weightage that comprises of adaptability, installability and replaceability. Compatibility is at sixth place having 81 total weightage that consists of co-existence, and interoperability. Maintainability is at seventh place having 78 total weightage, as it comprises of modularity, reusability, analysability, modifiability, and testability. Functional suitability is at eighth place having 76 total weightage that consists of functional completeness, functional correctness, and functional appropriateness.

After completing survey, got data from respondents that were based on following inputs as shown in TABLE VII. regarding eight parameters i.e. PORT, MAIN, SEC, REL, SUIT, PER, COMP, and USA with respect to Product-Quality Level and Membership Grade of Product-Quality:

TABLE VII. PRODUCT QUALITY REQUIREMENTS INPUTS FROM SYSTEM ANALYSTS AND DOMAIN EXPERTS OF INTERNET BANKING WITH RESPECT TO THE PRODUCT QUALITY LEVEL AND MEMBERSHIP GRADE OF PRODUCT QUALITY

PORT	MAIN	SEC	REL	SUIT	PER	COMP	USA	PQ Level	Membership Grade of Product-Quality
I_H	I_H	I_H	I_H	I_H	I_H	I_H	I_H	High (32)	Min 24, Max 32
I_H	I_H	I_H	I_H	I_H	I_H	I_H	I_H	High (32)	Min 24, Max 32
I	I_L	I_H	I	I_L	I	I_L	I	Avg (22)	Min 17, Max 23
I_H	I_H	I_H	I_H	I_H	I_H	I_H	I_H	High (32)	Min 24, Max 32
I_H	I_H	I_H	I_H	I_H	I_H	I_H	I_H	High (32)	Min 24, Max 32
I	I	I	I	I_L	I_H	I	I	High (24)	Min 24, Max 32
I	I_L	I_H	I	I_L	I	I_L	I	Avg (22)	Min 17, Max 23
I_H	I_H	I_H	I_H	I_H	I_H	I_H	I_H	High (32)	Min 24, Max 32
I	I	I_H	I	I_L	I_H	I	I	High (25)	Min 24, Max 32
I	I_L	I_H	I	I_L	I	I_L	I	Avg (22)	Min 17, Max 23
I_H	I_H	I_H	I_H	I_H	I_H	I_H	I_H	High (32)	Min 24, Max 32
I	I	I_H	I_H	I	I_H	I_H	I_H	High (29)	Min 24, Max 32
I	I_L	I	I	I_L	I_H	I	I	Avg (23)	Min 17, Max 23
I	I_L	I_H	I	I_L	I	I_L	I	Avg (22)	Min 17, Max 23
I	I	I_H	I	I	I_H	I	I_H	High (27)	Min 24, Max 32
I_H	I_H	I_H	I_H	I_H	I_H	I_H	I_H	High (32)	Min 24, Max 32
I	I	I_H	I	I	I_H	I	I_H	High (27)	Min 24, Max 32
I	I_L	I_H	I	I_L	I	I_L	I	Avg (22)	Min 17, Max 23
I	I	I	I	I	I_H	I	I	High (25)	Min 24, Max 32
I	I	I_H	I_H	I	I_H	I_H	I_H	High (29)	Min 24, Max 32
I_H	I_H	I_H	I_H	I_H	I_H	I_H	I_H	High (32)	Min 24, Max 32
I	I_L	I_H	I	I_L	I	I_L	I	Avg (22)	Min 17, Max 23
I	I_H	I_H	I_H	I_H	I_H	I_H	I_H	High (31)	Min 24, Max 32
I	I	I	I	I	I_H	I	I	High (25)	Min 24, Max 32
I_H	I_H	I_H	I_H	I_H	I_H	I_H	I_H	High (32)	Min 24, Max 32

For this model, Mamdani Style Inference Mechanism has been castoff. Output variable Product-Quality is observed by using the MATLAB Fuzzy Tool Box for a particular 25 sets of inputs received from respondents. For respective given set of input parameters i.e. [PORT, MAIN, SEC, REL, SUIT, PER, COMP, USA] as [3.36, 3.12, 3.84, 3.48, 3.04, 3.76, 3.24, 3.56] and then Rule Viewer helps to realize the output Product-Quality level generated i.e 26.2 corresponding to this assumed set of input variables which is shown below at Fig. 28.

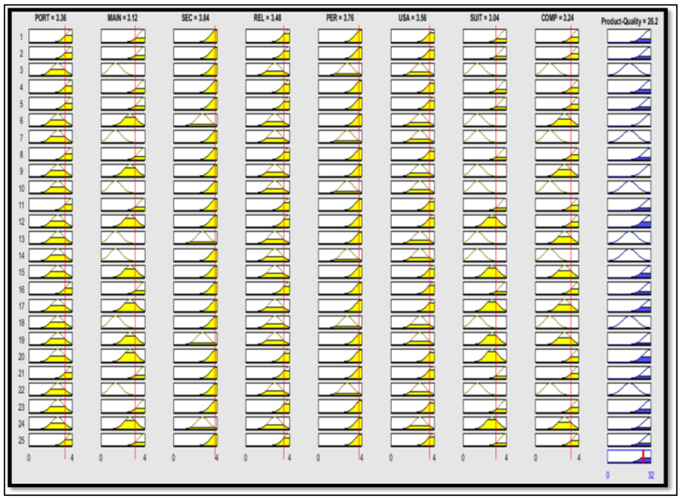

Fig. 28. Rule Viewer for Product-Quality Model with respect to the Respondents Inputs

After creating all 25 rules by using MATLAB Fuzzy Tool Box through Mamdani Style Inference Mechanism, defuzzify the fuzzified outputs by using the Joint Membership Function by plotting on Two-Dimensional Surface View, as shown in Fig. 29 to 38 accordingly:

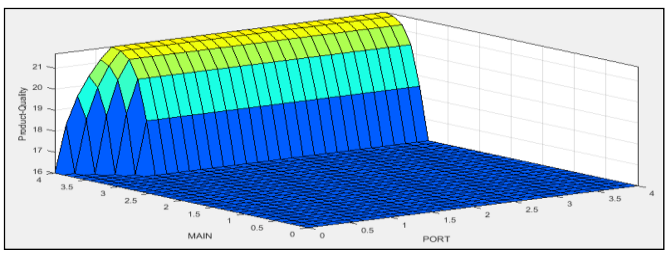

Fig. 29. Two-Dimensional Surface View with PORT (input) on X-axis, MAIN (input) on Y-axis, and Product-Quality (output) on Z-axis with respect to the Respondents Inputs

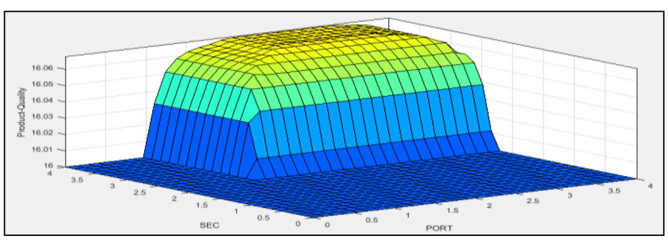

Fig. 30. Two-Dimensional Surface View with PORT (input) on X-axis, SEC (input) on Y-axis, and Product-Quality (output) on Z-axis with respect to the Respondents Inputs

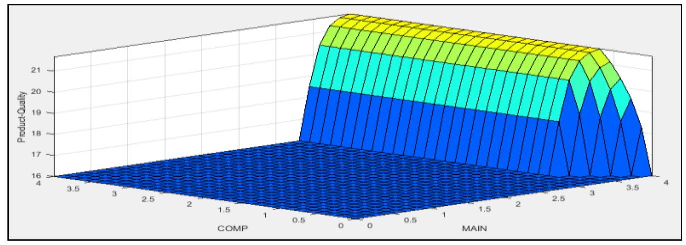

Fig. 31. Two-Dimensional Surface View with MAIN (input) on X-axis, COMP (input) on Y-axis, and Product-Quality (output) on Z-axis with respect to the Respondents Inputs

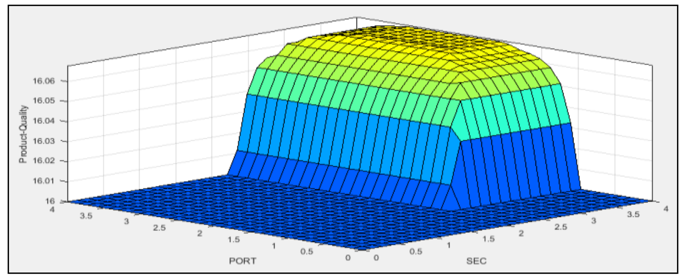

Fig. 32. Two-Dimensional Surface View with SEC (input) on X-axis, PORT (input) on Y-axis, and Product-Quality (output) on Z-axis with respect to the Respondents Inputs

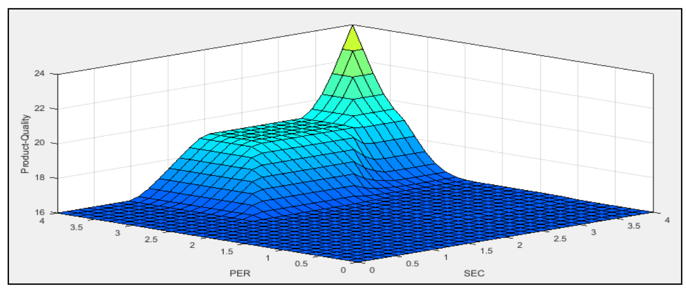

Fig. 33. Two-Dimensional Surface View with SEC (input) on X-axis, PER (input) on Y-axis, and Product-Quality (output) on Z-axis with respect to the Respondents Inputs

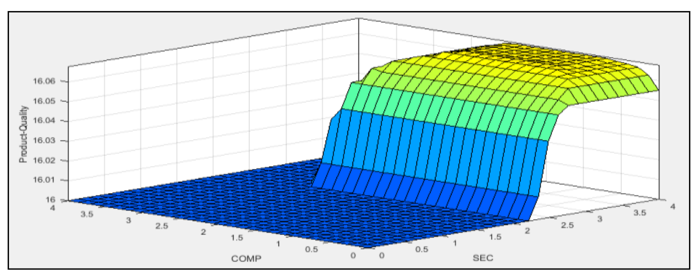

Fig. 34. Two-Dimensional Surface View with SEC (input) on X-axis, COMP (input) on Y-axis, and Product-Quality (output) on Z-axis with respect to the Respondents Inputs

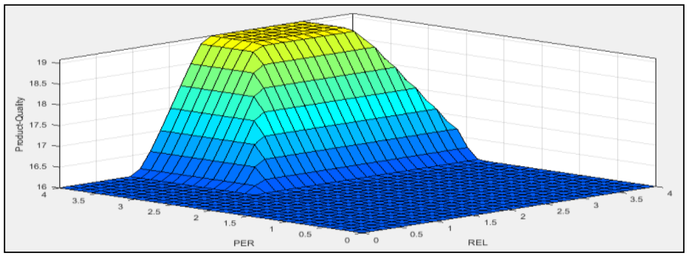

Fig. 35. Two-Dimensional Surface View with REL (input) on X-axis, PER (input) on Y-axis, and Product-Quality (output) on Z-axis with respect to the Respondents Inputs

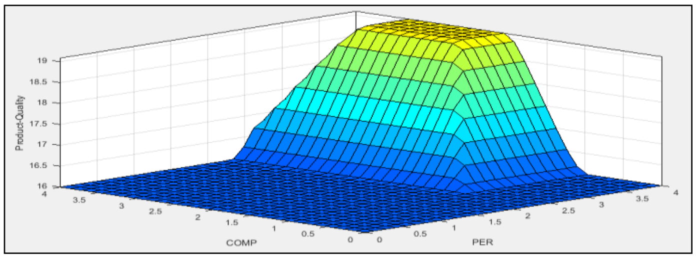

Fig. 36. Two-Dimensional Surface View with PER (input) on X-axis, COMP (input) on Y-axis, and Product-Quality (output) on Z-axis with respect to the Respondents Inputs

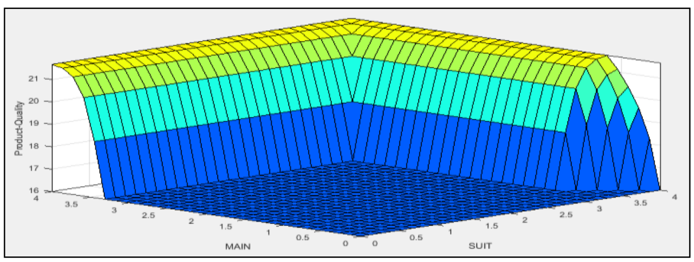

Fig. 37. Two-Dimensional Surface View with SUIT (input) on X-axis, MAIN (input) on Y-axis, and Product-Quality (output) on Z-axis with respect to the Respondents Inputs

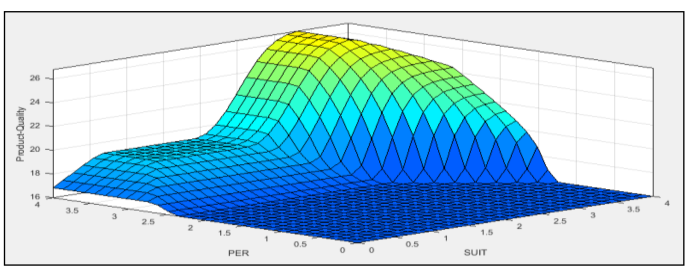

Fig. 38. Two-Dimensional Surface View with SUIT (input) on X-axis, PER (input) on Y-axis, and Product-Quality (output) on Z-axis with respect to the Respondents Inputs

VI. Conclusion

The measurement and quantification of quality requirements of DSS is a challenging task, because these quality requirements are in the qualitative form and can't be represented in a specific quantitative way. Although, several quality requirements methods for DSS have been proposed so far, but the research on analyzing quality requirements of DSS were limited.

Since last decades, researchers are focusing on quality requirements, because most of DSS only unsuccessful due to the inattention of quality requirements. As stakeholders requires a best quality DSS software, so we can't neglect quality requirements because of its primary importance into systems.

In this paper, a quantitative approach proposed for analyzing ISO / IEC 25010 product quality requirements based on fuzzy logic and likert scale for DSS which aims to quantify the quality requirements.

Proposed model has validated combined measure of Product-Quality based on PORT, MAIN, SEC, REL, SUIT, PER, COMP, and USA. After that, implemented proposed framework on a case study 'Internet Banking'. Got data from 25 system analysts and domain experts of banking sector and during that survey, we learned that respondents were more worried about the security followed by performance efficiency, usability, reliability, portability, compatibility, maintainability, and functional suitability.

In future, we are planning to develop a specific tool that will internment and manuscript these requirements.

References

[1] Busch, Axel. "Automated Decision Support for Recurring Design Decisions Considering Non-Functional Requirements." Software Engineering & Management. 2015.

[2] Dabbagh, Mohammad, and Sai Peck Lee. "An Approach for Prioritizing NFRs According to Their Relationship with FRs." Lecture Notes on Software Engineering 3.1 (2015): 1.

[3] Atoum, Issa, and Chih How Bong. "Measuring Software Quality in Use: State-of-the-Art and Research Challenges." arXiv preprint arXiv: 1503 - 06934 (2015).

[4] Saadatmand, Mehrdad, and Sahar Tahvili. "A Fuzzy Decision Support Approach for Model-Based Tradeoff Analysis of Non-Functional Requirements." Information Technology-New Generations (ITNG), 2015 12th International Conference on. IEEE, 2015.

[5] Kaur, Harmeet, and Gurvinder N. Verma. "A Case Study upon Non-functional Requirements of Online Banking System". International Journal of Computer Applications Technology and Research Volume 4 – Issue 4, 220 - 225, (2015): 2319 –8656.

[6] Zaib, Amna, Maha Chauhan, and Mehreen Sirshar. "A Review Analysis on Non-Functional Requirements and Causes of Failure of Projects." International Journal of Computer and Communication System Engineering (IJCCSE), Vol. 2 (3), (2015): 490 -497.

[7] Franca, Joyce MS, and Michel S. Soares. "SOAQM: Quality Model for SOA Applications based on ISO 25010." Proceedings of the 17th International Conference on Enterprise Information Systems (ICEIS-2015), 60-70: 978 – 989 – 758 – 097 – 0.

[8] Ravanello, Anderson, et al. "Performance measurement for cloud computing applications using ISO 25010 standard characteristics." Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2014 Joint Conference of the International Workshop on. IEEE, 2014.

[9] Wojcik, Rob. "Eliciting and Specifying Quality Attribute Requirements." Carnedie - Mellon Univ Pittsburgh PA Software Engineering Institute (SEI), 2013.

[10] Saleh, Kassem. "Development of Non-functional Requirements for Decision Support Systems." Proceedings of World Academy of Science, Engineering and Technology. No. 65. World Academy of Science, Engineering and Technology, 2012.

[11] Al-Jamimi, Hamdi A., and Moataz Ahmed. "Prediction of software maintainability using fuzzy logic." Software Engineering and Service

Science (ICSESS), 2012 IEEE 3rd International Conference on. IEEE, 2012.

[12] Ameller, David, et al. "How do software architects consider non-functional requirements: An exploratory study." 2012 20th IEEE International Requirements Engineering Conference (RE). IEEE, 2012.

[13] Kumar, Ravinder, Kiran Khatter, and Arvind Kalia. "Measuring software reliability: a fuzzy model." ACM SIGSOFT Software Engineering Notes 36.6 (2011): 1 - 6.

[14] Rao, A. Ananda, and Merugu Gopichand. "Four layered approach to non-functional requirements analysis." International Journal of Computer Science Issues 8.6 (2011).

[15] Liu, Shaofeng, et al. "Integration of decision support systems to improve decision support performance." Knowledge and Information Systems 22.3 (2010): 261 - 286.

[16] Mairiza, Dewi, Didar Zowghi, and Nurie Nurmuliani. "An investigation into the notion of non-functional requirements." Proceedings of the 2010 ACM Symposium on Applied Computing. ACM, 2010.

[17] Malik, Nasir Mahmood, et al. "Measurable & scalable NFRs using fuzzy logic and Likert scale." arXiv preprint arXiv: 0906 - 5393 (2009).

[18] Sadana, Vishal, and Xiaoqing Frank Liu. "Analysis of conflicts among non-functional requirements using integrated analysis of functional and non-functional requirements." Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International. Vol. 1. IEEE, 2007.

[19] Skrobackova, Marketa, and Hana Kopackova. "Decision support systems or business intelligence: what can help in decision making?." Scientific papers of the University of Pardubice. Series D, Faculty of Economics and Administration. 10 (2006) (2006).

ANNEXURE

Analyzing PORT, MAIN, SEC, REL, SUIT, PER, COMP, USA Quality Requirements of DSS for Determining Product-Quality Level (Annexure A)

TABLE VIII. ANALYZING PORTABILITY, MAINTAINABILITY, SECURITY, RELIABILITY, FUNCTIONAL SUITABILITY, PERFORMANCE EFFICIENCY, COMPATIBILITY, AND USABILITY QUALITY REQUIREMENTS OF DSS FOR DETERMINING PRODUCT QUALITY LEVEL (ANNEXURE A)

Sr.	**(Quality Requirements Inputs)**								**PQ Level**
	PORT	**MAIN**	**SEC**	**REL**	**SUIT**	**PER**	**COMP**	**USA**	
1	I_N	I_N	I_N	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
2	I_N	I_L	I_N	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
3	I_N	I	I_N	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
4	I_N	I_H	I_N	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
5	I_N	I_N	I_L	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
6	I_N	I_N	I	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
7	I_N	I_N	I_H	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
8	I_N	I_N	I_N	I_L	I_N	I_N	I_N	I_N	PQ_{Low}
9	I_N	I_N	I_N	I	I_N	I_N	I_N	I_N	PQ_{Low}
10	I_N	I_N	I_N	I_H	I_N	I_N	I_N	I_N	PQ_{Low}
11	I_N	I_N	I_N	I_N	I_L	I_N	I_N	I_N	PQ_{Low}
12	I_N	I_N	I_N	I_N	I	I_N	I_N	I_N	PQ_{Low}
13	I_N	I_N	I_N	I_N	I_H	I_N	I_N	I_N	PQ_{Low}
14	I_N	I_N	I_N	I_N	I_N	I_L	I_N	I_N	PQ_{Low}
15	I_N	I_N	I_N	I_N	I_N	I	I_N	I_N	PQ_{Low}
16	I_N	I_N	I_N	I_N	I_N	I_H	I_N	I_N	PQ_{Low}
17	I_N	I_N	I_N	I_N	I_N	I_N	I_L	I_N	PQ_{Low}
18	I_N	I_N	I_N	I_N	I_N	I_N	I	I_N	PQ_{Low}
19	I_N	I_N	I_N	I_N	I_N	I_N	I_H	I_N	PQ_{Low}
20	I_N	I_N	I_N	I_N	I_N	I_N	I_N	I_L	PQ_{Low}
21	I_N	I_N	I_N	I_N	I_N	I_N	I_N	I	PQ_{Low}
22	I_N	I_N	I_N	I_N	I_N	I_N	I_N	I_H	PQ_{Low}
23	I_L	I_L	I_N	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
24	I_L	I	I_N	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
25	I_L	I_H	I_N	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
26	I_L	I_N	I_L	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
27	I_L	I_N	I	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
28	I_L	I_N	I_H	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
29	I_L	I_N	I_N	I_L	I_N	I_N	I_N	I_N	PQ_{Low}
30	I_L	I_N	I_N	I	I_N	I_N	I_N	I_N	PQ_{Low}
31	I_L	I_N	I_N	I_H	I_N	I_N	I_N	I_N	PQ_{Low}
32	I_L	I_N	I_N	I_N	I_L	I_N	I_N	I_N	PQ_{Low}
33	I_L	I_N	I_N	I_N	I	I_N	I_N	I_N	PQ_{Low}
34	I_L	I_N	I_N	I_N	I_H	I_N	I_N	I_N	PQ_{Low}
35	I_L	I_N	I_N	I_N	I_N	I_L	I_N	I_N	PQ_{Low}
36	I_L	I_N	I_N	I_N	I_N	I	I_N	I_N	PQ_{Low}
37	I_L	I_N	I_N	I_N	I_N	I_H	I_N	I_N	PQ_{Low}
38	I_L	I_N	I_N	I_N	I_N	I_N	I_L	I_N	PQ_{Low}
39	I_L	I_N	I_N	I_N	I_N	I_N	I	I_N	PQ_{Low}
40	I_L	I_N	I_N	I_N	I_N	I_N	I_H	I_N	PQ_{Low}
41	I_L	I_N	I_N	I_N	I_N	I_N	I_N	I_L	PQ_{Low}
42	I_L	I_N	I_N	I_N	I_N	I_N	I_N	I	PQ_{Low}
43	I_L	I_N	I_N	I_N	I_N	I_N	I_N	I_H	PQ_{Low}
44	I	I_N	I_N	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
45	I	I_L	I_N	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
46	I	I	I_N	I_N	I_N	I_N	I_N	I_N	PQ_{Low}

47	I	I_H	I_N	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
48	I	I_N	I_L	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
49	I	I_N	I	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
50	I	I_N	I_H	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
51	I	I_N	I_N	I_L	I_N	I_N	I_N	I_N	PQ_{Low}
52	I	I_N	I_N	I	I_N	I_N	I_N	I_N	PQ_{Low}
53	I	I_N	I_N	I_H	I_N	I_N	I_N	I_N	PQ_{Low}
54	I	I_N	I_N	I_N	I_L	I_N	I_N	I_N	PQ_{Low}
55	I	I_N	I_N	I_N	I	I_N	I_N	I_N	PQ_{Low}
56	I	I_N	I_N	I_N	I_H	I_N	I_N	I_N	PQ_{Low}
57	I	I_N	I_N	I_N	I_N	I_L	I_N	I_N	PQ_{Low}
58	I	I_N	I_N	I_N	I_N	I	I_N	I_N	PQ_{Low}
59	I	I_N	I_N	I_N	I_N	I_H	I_N	I_N	PQ_{Low}
60	I	I_N	I_N	I_N	I_N	I_N	I_L	I_N	PQ_{Low}
61	I	I_N	I_N	I_N	I_N	I_N	I	I_N	PQ_{Low}
62	I	I_N	I_N	I_N	I_N	I_N	I_H	I_N	PQ_{Low}
63	I	I_N	I_N	I_N	I_N	I_N	I_N	I_L	PQ_{Low}
64	I	I_N	I_N	I_N	I_N	I_N	I_N	I	PQ_{Low}
65	I	I_N	I_N	I_N	I_N	I_N	I_N	I_H	PQ_{Low}
66	I_H	I_N	I_N	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
67	I_H	I_L	I_N	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
68	I_H	I	I_N	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
69	I_H	I_H	I_N	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
70	I_H	I_N	I_L	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
71	I_H	I_N	I	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
72	I_H	I_N	I_H	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
73	I_H	I_N	I_N	I_L	I_N	I_N	I_N	I_N	PQ_{Low}
74	I_H	I_N	I_N	I	I_N	I_N	I_N	I_N	PQ_{Low}
75	I_H	I_N	I_N	I_H	I_N	I_N	I_N	I_N	PQ_{Low}
76	I_H	I_N	I_N	I_N	I_L	I_N	I_N	I_N	PQ_{Low}
77	I_H	I_N	I_N	I_N	I	I_N	I_N	I_N	PQ_{Low}
78	I_H	I_N	I_N	I_N	I_H	I_N	I_N	I_N	PQ_{Low}
79	I_H	I_N	I_N	I_N	I_N	I_L	I_N	I_N	PQ_{Low}
80	I_H	I_N	I_N	I_N	I_N	I	I_N	I_N	PQ_{Low}
81	I_H	I_N	I_N	I_N	I_N	I_H	I_N	I_N	PQ_{Low}
82	I_H	I_N	I_N	I_N	I_N	I_N	I_L	I_N	PQ_{Low}
83	I_H	I_N	I_N	I_N	I_N	I_N	I	I_N	PQ_{Low}
84	I_H	I_N	I_N	I_N	I_N	I_N	I_H	I_N	PQ_{Low}
85	I_H	I_N	I_N	I_N	I_N	I_N	I_N	I_L	PQ_{Low}
86	I_H	I_N	I_N	I_N	I_N	I_N	I_N	I	PQ_{Low}
87	I_H	I_N	I_N	I_N	I_N	I_N	I_N	I_H	PQ_{Low}
88	I_L	I_L	I_L	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
89	I_L	I_L	I	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
90	I_L	I_L	I_H	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
91	I_L	I_L	I_N	I_L	I_N	I_N	I_N	I_N	PQ_{Low}
92	I_L	I_L	I_N	I	I_N	I_N	I_N	I_N	PQ_{Low}
93	I_L	I_L	I_N	I_H	I_N	I_N	I_N	I_N	PQ_{Low}
94	I_L	I_L	I_N	I_N	I_L	I_N	I_N	I_N	PQ_{Low}
95	I_L	I_L	I_N	I_N	I	I_N	I_N	I_N	PQ_{Low}
96	I_L	I_L	I_N	I_N	I_H	I_N	I_N	I_N	PQ_{Low}
97	I_L	I_L	I_N	I_N	I_N	I_L	I_N	I_N	PQ_{Low}
98	I_L	I_L	I_N	I_N	I_N	I	I_N	I_N	PQ_{Low}
99	I_L	I_L	I_N	I_N	I_N	I_H	I_N	I_N	PQ_{Low}
100	I_L	I_L	I_N	I_N	I_N	I_N	I_L	I_N	PQ_{Low}
101	I_L	I_L	I_N	I_N	I_N	I_N	I	I_N	PQ_{Low}
102	I_L	I_L	I_N	I_N	I_N	I_N	I_H	I_N	PQ_{Low}
103	I_L	I_L	I_N	I_N	I_N	I_N	I_N	I_L	PQ_{Low}
104	I_L	I_L	I_N	I_N	I_N	I_N	I_N	I	PQ_{Low}
105	I_L	I_L	I_N	I_N	I_N	I_N	I_N	I_H	PQ_{Low}
106	I	I	I_L	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
107	I	I	I	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
108	I	I	I_H	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
109	I	I	I_N	I_L	I_N	I_N	I_N	I_N	PQ_{Low}
110	I	I	I_N	I	I_N	I_N	I_N	I_N	PQ_{Low}
111	I	I	I_N	I_H	I_N	I_N	I_N	I_N	PQ_{Low}
112	I	I	I_N	I_N	I_L	I_N	I_N	I_N	PQ_{Low}
113	I	I	I_N	I_N	I	I_N	I_N	I_N	PQ_{Low}
114	I	I	I_N	I_N	I_H	I_N	I_N	I_N	PQ_{Low}

115	I	I	I_N	I_N	I_N	I_L	I_N	I_N	PQ_{Low}
116	I	I	I_N	I_N	I_N	I	I_N	I_N	PQ_{Low}
117	I	I	I_N	I_N	I_N	I_H	I_N	I_N	PQ_{Low}
118	I	I	I_N	I_N	I_N	I_N	I_L	I_N	PQ_{Low}
119	I	I	I_N	I_N	I_N	I_N	I	I_N	PQ_{Low}
120	I	I	I_N	I_N	I_N	I_N	I_H	I_N	PQ_{Low}
121	I	I	I_N	I_N	I_N	I_N	I_N	I_L	PQ_{Low}
122	I	I	I_N	I_N	I_N	I_N	I_N	I	PQ_{Low}
123	I	I	I_N	I_N	I_N	I_N	I_N	I_H	PQ_{Low}
124	I_H	I_H	I_L	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
125	I_H	I_H	I	I_N	I_N	I_N	I_N	I_N	PQ_{Low}
126	I_H	I_H	I_H	I_N	I_N	I_N	I_N	I_N	$PQ_{Average}$
127	I_H	I_H	I_N	I_L	I_N	I_N	I_N	I_N	PQ_{Low}
128	I_H	I_H	I_N	I	I_N	I_N	I_N	I_N	PQ_{Low}
129	I_H	I_H	I_N	I_H	I_N	I_N	I_N	I_N	$PQ_{Average}$
130	I_H	I_H	I_N	I_N	I_L	I_N	I_N	I_N	PQ_{Low}
131	I_H	I_H	I_N	I_N	I	I_N	I_N	I_N	PQ_{Low}
132	I_H	I_H	I_N	I_N	I_H	I_N	I_N	I_N	PQ_{Low}
133	I_H	I_H	I_N	I_N	I_N	I_L	I_N	I_N	PQ_{Low}
134	I_H	I_H	I_N	I_N	I_N	I	I_N	I_N	PQ_{Low}
135	I_H	I_H	I_N	I_N	I_N	I_H	I_N	I_N	$PQ_{Average}$
136	I_H	I_H	I_N	I_N	I_N	I_N	I_L	I_N	PQ_{Low}
137	I_H	I_H	I_N	I_N	I_N	I_N	I	I_N	PQ_{Low}
138	I_H	I_H	I_N	I_N	I_N	I_N	I_H	I_N	$PQ_{Average}$
139	I_H	I_H	I_N	I_N	I_N	I_N	I_N	I_L	PQ_{Low}
140	I_H	I_H	I_N	I_N	I_N	I_N	I_N	I	PQ_{Low}
141	I_H	I_H	I_N	I_N	I_N	I_N	I_N	I_H	$PQ_{Average}$
142	I_L	I_L	I_L	I_L	I_N	I_N	I_N	I_N	PQ_{Low}
143	I_N	I_N	I_N	I	I_N	I_N	I_N	I_N	PQ_{Low}
144	I_N	I_N	I_N	I_H	I_N	I_N	I_N	I_N	PQ_{Low}
145	I_N	I_N	I_N	I_N	I_L	I_N	I_N	I_N	PQ_{Low}
146	I_N	I_N	I_N	I_N	I	I_N	I_N	I_N	PQ_{Low}
147	I_N	I_N	I_N	I_N	I_H	I_N	I_N	I_N	PQ_{Low}
148	I_N	I_N	I_N	I_N	I_N	I_L	I_N	I_N	PQ_{Low}
149	I_N	I_N	I_N	I_N	I_N	I	I_N	I_N	PQ_{Low}
150	I_N	I_N	I_N	I_N	I_N	I_H	I_N	I_N	PQ_{Low}
151	I_N	I_N	I_N	I_N	I_N	I_N	I_L	I_N	PQ_{Low}
152	I_N	I_N	I_N	I_N	I_N	I_N	I	I_N	PQ_{Low}
153	I_N	I_N	I_N	I_N	I_N	I_N	I_H	I_N	PQ_{Low}
154	I_N	I_N	I_N	I_N	I_N	I_N	I_N	I_L	PQ_{Low}
155	I_N	I_N	I_N	I_N	I_N	I_N	I_N	I	PQ_{Low}
156	I_N	I_N	I_N	I_N	I_N	I_N	I_N	I_H	PQ_{Low}
157	I	I	I	I_L	I_N	I_N	I_N	I_N	PQ_{Low}
158	I	I	I	I	I_N	I_N	I_N	I_N	PQ_{Low}
159	I	I	I	I_H	I_N	I_N	I_N	I_N	$PQ_{Average}$
160	I	I	I	I_N	I_L	I_N	I_N	I_N	PQ_{Low}
161	I	I	I	I_N	I	I_N	I_N	I_N	PQ_{Low}
162	I	I	I	I_N	I_H	I_N	I_N	I_N	$PQ_{Average}$
163	I	I	I	I_N	I_N	I_L	I_N	I_N	PQ_{Low}
164	I	I	I	I_N	I_N	I	I_N	I_N	PQ_{Low}
165	I	I	I	I_N	I_N	I_H	I_N	I_N	$PQ_{Average}$
166	I	I	I	I_N	I_N	I_N	I_L	I_N	PQ_{Low}
167	I	I	I	I_N	I_N	I_N	I	I_N	PQ_{Low}
168	I	I	I	I_N	I_N	I_N	I_H	I_N	$PQ_{Average}$
169	I	I	I	I_N	I_N	I_N	I_N	I_L	PQ_{Low}
170	I	I	I	I_N	I_N	I_N	I_N	I	PQ_{Low}
171	I	I	I	I_N	I_N	I_N	I_N	I_H	$PQ_{Average}$
172	I_H	I_H	I_H	I_L	I_N	I_N	I_N	I_N	$PQ_{Average}$
173	I_H	I_H	I_H	I	I_N	I_N	I_N	I_N	$PQ_{Average}$
174	I_H	I_H	I_H	I_H	I_N	I_N	I_N	I_N	$PQ_{Average}$
175	I_H	I_H	I_H	I_N	I_L	I_N	I_N	I_N	$PQ_{Average}$
176	I_H	I_H	I_H	I_N	I	I_N	I_N	I_N	$PQ_{Average}$
177	I_H	I_H	I_H	I_N	I_H	I_N	I_N	I_N	$PQ_{Average}$
178	I_H	I_H	I_H	I_N	I_N	I_L	I_N	I_N	$PQ_{Average}$
179	I_H	I_H	I_H	I_N	I_N	I	I_N	I_N	$PQ_{Average}$
180	I_H	I_H	I_H	I_N	I_N	I_H	I_N	I_N	$PQ_{Average}$
181	I_H	I_H	I_H	I_N	I_N	I_N	I_L	I_N	$PQ_{Average}$
182	I_H	I_H	I_H	I_N	I_N	I_N	I	I_N	$PQ_{Average}$

183	I_H	I_H	I_H	I_N	I_N	I_N	I_H	I_N	$PQ_{Average}$
184	I_H	I_H	I_H	I_N	I_N	I_N	I_N	I_L	$PQ_{Average}$
185	I_H	I_H	I_H	I_N	I_N	I_N	I_N	I	$PQ_{Average}$
186	I_H	I_H	I_H	I_N	I_N	I_N	I_N	I_H	$PQ_{Average}$
187	I_L	I_L	I_L	I_L	I_L	I_N	I_N	I_N	PQ_{Low}
188	I_L	I_L	I_L	I_L	I	I_N	I_N	I_N	PQ_{Low}
189	I_L	I_L	I_L	I_L	I_H	I_N	I_N	I_N	PQ_{Low}
190	I_L	I_L	I_L	I_L	I_N	I_L	I_N	I_N	PQ_{Low}
191	I_L	I_L	I_L	I_L	I_N	I	I_N	I_N	PQ_{Low}
192	I_L	I_L	I_L	I_L	I_N	I_H	I_N	I_N	PQ_{Low}
193	I_L	I_L	I_L	I_L	I_N	I_N	I_L	I_N	PQ_{Low}
194	I_L	I_L	I_L	I_L	I_N	I_N	I	I_N	PQ_{Low}
195	I_L	I_L	I_L	I_L	I_N	I_N	I_H	I_N	PQ_{Low}
196	I_L	I_L	I_L	I_L	I_N	I_N	I_N	I_L	PQ_{Low}
197	I_L	I_L	I_L	I_L	I_N	I_N	I_N	I	PQ_{Low}
198	I_L	I_L	I_L	I_L	I_N	I_N	I_N	I_H	PQ_{Low}
199	I	I	I	I	I_L	I_N	I_N	I_N	$PQ_{Average}$
200	I	I	I	I	I	I_N	I_N	I_N	$PQ_{Average}$
201	I	I	I	I	I_H	I_N	I_N	I_N	$PQ_{Average}$
202	I	I	I	I	I_N	I_L	I_N	I_N	$PQ_{Average}$
203	I	I	I	I	I_N	I	I_N	I_N	$PQ_{Average}$
204	I	I	I	I	I_N	I_H	I_N	I_N	$PQ_{Average}$
205	I	I	I	I	I_N	I_N	I_L	I_N	$PQ_{Average}$
206	I	I	I	I	I_N	I_N	I	I_N	$PQ_{Average}$
207	I	I	I	I	I_N	I_N	I_H	I_N	$PQ_{Average}$
208	I	I	I	I	I_N	I_N	I_N	I_L	$PQ_{Average}$
209	I	I	I	I	I_N	I_N	I_N	I	$PQ_{Average}$
210	I	I	I	I	I_N	I_N	I_N	I_H	$PQ_{Average}$
211	I_H	I_H	I_H	I_H	I_L	I_N	I_N	I_N	$PQ_{Average}$
212	I_H	I_H	I_H	I_H	I	I_N	I_N	I_N	$PQ_{Average}$
213	I_H	I_H	I_H	I_H	I_H	I_N	I_N	I_N	$PQ_{Average}$
214	I_H	I_H	I_H	I_H	I_N	I_L	I_N	I_N	$PQ_{Average}$
215	I_H	I_H	I_H	I_H	I_N	I	I_N	I_N	$PQ_{Average}$
216	I_H	I_H	I_H	I_H	I_N	I_H	I_N	I_N	$PQ_{Average}$
217	I_H	I_H	I_H	I_H	I_N	I_N	I_L	I_N	$PQ_{Average}$
218	I_H	I_H	I_H	I_H	I_N	I_N	I	I_N	$PQ_{Average}$
219	I_H	I_H	I_H	I_H	I_N	I_N	I_H	I_N	$PQ_{Average}$
220	I_H	I_H	I_H	I_H	I_N	I_N	I_N	I_L	$PQ_{Average}$
221	I_H	I_H	I_H	I_H	I_N	I_N	I_N	I	$PQ_{Average}$
222	I_H	I_H	I_H	I_H	I_N	I_N	I_N	I_H	$PQ_{Average}$
223	I_H	I_H	I_H	I_H	I_N	I_N	I_N	I_N	$PQ_{Average}$
224	I_L	I_L	I_L	I_L	I_L	I_L	I_N	I_N	PQ_{Low}
225	I_L	I_L	I_L	I_L	I_L	I	I_N	I_N	PQ_{Low}
226	I_L	I_L	I_L	I_L	I_L	I_H	I_N	I_N	PQ_{Low}
227	I_L	I_L	I_L	I_L	I_L	I_N	I_L	I_N	PQ_{Low}
228	I_L	I_L	I_L	I_L	I_L	I_N	I	I_N	PQ_{Low}
229	I_L	I_L	I_L	I_L	I_L	I_N	I_H	I_N	PQ_{Low}
230	I_L	I_L	I_L	I_L	I_L	I_N	I_N	I_L	PQ_{Low}
231	I_L	I_L	I_L	I_L	I_L	I_N	I_N	I	PQ_{Low}
232	I_L	I_L	I_L	I_L	I_L	I_N	I_N	I_H	$PQ_{Average}$
233	I	I	I	I	I	I_L	I_N	I_N	$PQ_{Average}$
234	I	I	I	I	I	I	I_N	I_N	$PQ_{Average}$
235	I	I	I	I	I	I_H	I_N	I_N	$PQ_{Average}$
236	I	I	I	I	I	I_N	I_L	I_N	$PQ_{Average}$
237	I	I	I	I	I	I_N	I	I_N	$PQ_{Average}$
238	I	I	I	I	I	I_N	I_H	I_N	$PQ_{Average}$
239	I	I	I	I	I	I_N	I_N	I_L	$PQ_{Average}$
240	I	I	I	I	I	I_N	I_N	I	$PQ_{Average}$
241	I	I	I	I	I	I_N	I_N	I_H	$PQ_{Average}$
242	I_H	I_H	I_H	I_H	I_H	I_L	I_N	I_N	PQ_{High}
243	I_H	I_H	I_H	I_H	I_H	I	I_N	I_N	PQ_{High}
244	I_H	I_H	I_H	I_H	I_H	I_H	I_N	I_N	PQ_{High}
245	I_H	I_H	I_H	I_H	I_H	I_N	I_L	I_N	PQ_{High}
246	I_H	I_H	I_H	I_H	I_H	I_N	I	I_N	PQ_{High}
247	I_H	I_H	I_H	I_H	I_H	I_N	I_H	I_N	PQ_{High}
248	I_H	I_H	I_H	I_H	I_H	I_N	I_N	I_L	PQ_{High}
249	I_H	I_H	I_H	I_H	I_H	I_N	I_N	I	PQ_{High}
250	I_H	I_H	I_H	I_H	I_H	I_N	I_N	I_H	PQ_{High}

251	I_L	I_L	I_L	I_L	I_L	I_L	I_L	I_N	PQ_{Low}
252	I_L	I_L	I_L	I_L	I_L	I_L	I	I_N	PQ_{Low}
253	I_L	I_L	I_L	I_L	I_L	I_L	I_H	I_N	$PQ_{Average}$
254	I_L	I_L	I_L	I_L	I_L	I_L	I_N	I_L	PQ_{Low}
255	I_L	I_L	I_L	I_L	I_L	I_L	I_N	I	PQ_{Low}
256	I_L	I_L	I_L	I_L	I_L	I_L	I_N	I_H	$PQ_{Average}$
257	I	I	I	I	I	I	I_L	I_N	$PQ_{Average}$
258	I	I	I	I	I	I	I	I_N	$PQ_{Average}$
259	I	I	I	I	I	I	I_H	I_N	$PQ_{Average}$
260	I	I	I	I	I	I	I_N	I_L	$PQ_{Average}$
261	I	I	I	I	I	I	I_N	I	$PQ_{Average}$
262	I	I	I	I	I	I	I_N	I_H	$PQ_{Average}$
263	I_H	I_H	I_H	I_H	I_H	I_H	I_L	I_N	PQ_{High}
264	I_H	I_H	I_H	I_H	I_H	I_H	I	I_N	PQ_{High}
265	I_H	I_H	I_H	I_H	I_H	I_H	I_H	I_N	PQ_{High}
266	I_H	I_H	I_H	I_H	I_H	I_H	I_N	I_L	PQ_{High}
267	I_H	I_H	I_H	I_H	I_H	I_H	I_N	I	PQ_{High}
268	I_H	I_H	I_H	I_H	I_H	I_H	I_N	I_H	PQ_{High}
269	I_L	I_L	I_L	I_L	I_L	I_L	I_L	I_N	PQ_{Low}
270	I_L	I_L	I_L	I_L	I_L	I_L	I_L	I_L	PQ_{Low}
271	I_L	I_L	I_L	I_L	I_L	I_L	I_L	I	$PQ_{Average}$
272	I_L	I_L	I_L	I_L	I_L	I_L	I_L	I_H	$PQ_{Average}$
273	I	I	I	I	I	I	I	I_N	$PQ_{Average}$
274	I	I	I	I	I	I	I	I_L	$PQ_{Average}$
275	I	I	I	I	I	I	I	I	PQ_{High}
276	I	I	I	I	I	I	I	I_H	PQ_{High}
277	I_H	I_H	I_H	I_H	I_H	I_H	I_H	I_N	PQ_{High}
278	I_H	I_H	I_H	I_H	I_H	I_H	I_H	I_L	PQ_{High}
279	I_H	I_H	I_H	I_H	I_H	I_H	I_H	I	PQ_{High}
280	I_H	I_H	I_H	I_H	I_H	I_H	I_H	I_H	PQ_{High}