

Requirements Engineering: Conflict Detection Automation Using Machine Learning

Hatim Elhassan^{1,*}, Mohammed Abaker¹, Abdelzahir Abdelmaboud² and Mohammed Burhanur Rehman¹

¹Department of Computer Science, College of Applied Sciences, King Khalid University, Muhayil, 63772, Saudi Arabia

²Department of Information System, College of Science, King Khalid University, Muhayil, 63772, Saudi Arabia

*Corresponding Author: Hatim Elhassan. Email: hibrhem@kku.edu.sa

Received: 20 September 2021; Accepted: 12 November 2021

Abstract: The research community has well recognized the importance of requirement elicitation. Recent research has shown the continuous decreasing success rate of IS projects in the last five years due to the complexity of the requirement conflict refinement process. Requirement conflict is at the heart of requirement elicitation. It is also considered the prime reason for deciding the success or failure of the intended Information System (IS) project. This paper introduces the requirements conflict detection automation model based on the Mean shift clustering unsupervised machine learning model. It utilizes the advantages of Artificial Intelligence in detecting and classifying the requirement conflicts occurring in the requirement elicitation phase. An experiment of the proposed model was conducted, composed of 207 observations and 11 parameters. The results show that the correct detection accuracy for the (Conflicted Requirements, Partial Conflicted Requirements & Conflict Free Requirements). The proposed model findings provide a promising and effective detection process regarding requirements classification. The model validation process provides a performance comparison between the model output vs. the output produced by the requirement conflict verification phase, detailing the Standard Error (SE) measure of accuracy values and the detected clusters. The implications of this study could be used to promote the automatization of the requirement elicitation process. Thus, increasing the potentiality of enhancing the produced systems designs.

Keywords: Requirement's elicitation; requirements conflict detection; hierarchical clustering unsupervised machine learning; automatic conflict detection

1 Introduction

Requirement elicitation is at the heart of system and software development. Without viable requirement elicitation between the stakeholders and the requirement engineers, the requirement will not be elicited correctly. Therefore, requirement elicitation is critical and pivotal to the effective execution of software development. The massive reliance of the elicitation process is placed on human factors. Thus, including an unwanted complexity to the phase. The stakeholders and innovation community generally bring



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

distinctive viewpoints to the elicitation process. The views of communities can be adversely different, reflecting a huge variance between what is being communicated by the stakeholder and what is captured by the requirement engineer. As a result, complexity manifests in the form of conflicts that affect the outcome of the requirement process in such a negative way, and some cases leading to catastrophic events. Therefore, requirements conflicts are difficult to distinguish due to the complex nature of the requirements and source reliability verifications. It is established that requirement conflict is significant in the success or failure of the intended system or software. The existing surveys and reports on the requirements elicitation process's challenges establish the importance of unresolved requirement conflict that can lead to unsatisfying product outcomes. As reported by the Standish report, requirements conflict issues are categorized as one of the leading causes of IS project failure. Furthermore, the report estimated the worldwide cost of failure for IS project to be approximately \$500 billion per month [1].

The introductions section illustrated how requirements conflict is increasing the risk of IS project failure. Furthermore, the literature did not provide a suitable resolution to identify and detect requirements conflicts during the elicitation process. Thus, it's our motive to create a requirement conflict detections model using machine learning algorithms to automate the classification process of the requirements based on the accuracy rates. Hence, eliminating the conflicting and ambiguous requirements to build a successful IS project based on precise requirements reflects stakeholder needs. Finally, producing such a mechanism will significantly reduce the risk of the IS project failure. Thus, increasing the success rates.

The remaining part of the paper is organized as follows: Section 2 presents the related study. Section 3 illustrates the methodology strategy used to implement the proposed requirement conflict, detection model. Section 4 analysis and results Section 5 model validation. Section 6 discussion. Section 7 concludes this paper.

2 Related Study

This section focuses on problems of existing requirement conflict approaches and their usage context in terms of the domain. Moreover, this section also focuses on the literature evidence as collected to discover the extent to which due consideration is given to problem domains.

The literature is rich with proposals that depend on textually specified requirements to mark the conflicts among them. In what follows, we discuss the approaches that consider requirement conflicts as part of their supported concepts. Few of the published approaches focused on the arrangement aspects of the requirements elicitation process in cooperative information systems (CIS). However, due to the heterogeneity of the interests, conflicts and contradictions were generated among the enterprise objectives. e.g., Boulekdam et al. [2] the Jebreen et al. [3] successful attempts in reducing the number of requirement conflicts, overlooking the fact that existing requirements conflicts cannot be unambiguously well communicated. Therefore, both approaches can only be permissible in specific cases of tangible conflicts. Equally, Ramadan et al. [4] presented "A semi-automated BPMN-based framework" for detecting conflicts between security, data-minimization, and fairness requirements approach in an effort to tackle the requirement conflicts challenges. However, the BPMN model had external validity threats concerning the experimental design. Additionally, they did not provide an analytical mechanism to detect the conflict between these requirements [5–8]. Moreover, the produced report of the reviewed approaches was also classified as a set of textual requirements. Thus, the requirement conflict detection mechanism is merely a manual process and an error prone task. Consequently, Egyed et al. [9] highlighted the challenges of detecting requirement conflicts. In their analysis. The authors developed an approach to trace the dependencies among the textual and non-functional requirements. The trace dependencies process is initiated for similar test scenario patterns. Finally, the approach relies on the matrix scheme for the conflict detection process of the dependencies. Comparatively, Mairiza et al. [10] developed a

two-dimensional matrix catalog method for detecting requirement conflicts in non-functional concept. Thus, examining non-functional requirement conflict, specifically the (usability & security) requirements with the proposed ontology-based framework, in efforts to provide conflict free requirement among non-functional requirements [11]. Regardless of the great efforts by the authors, there were a few technical issues in the framework automation process. Much of the work on “non-functional requirement” of Mairiza has been carried out by Poort et al. [12] In their proposed framework, Poort et al proposed to categorize the requirements according to their functionality relevance. As such, creating clustering systems of similar non-functional requirement to illustrate the conflicts. However, the proposed framework was merely an abstract theory in which no analysis was conducted. Likewise, various approaches have been proposed to solve this issue [9–12]. However, these approaches were basing their detection on textual specified requirements. Moreover, studies show the unconfirmed detections are due to inexact semantics. On the other hand, Ali et al. [13] proposed a contextual goal model that weaves together the variability of both context and requirements. The provided proposition was supported and analyzed with a CASE tool to provide a systematic process guideline, and analysis for the contextual goal models. However, the evaluations stage of the model was conducted on a small scale. Nevertheless, the idea was considered a successful trial.

Furthermore, Alkubaisy et al. [14] analyzed the conflict detection between privacy and security requirements. In their approach they tried to simplify the detection by using patterns to describe the problem. However, this approach increased the cost and the complexity, due to the redundancy factor. Furthermore, Horkoff et al. [15] surveyed 264 top cited research papers and conferences related to requirement engineering field in an attempt to provide a systematic mapping reference to classify the recurring issues. Furthermore, the study evaluated the advancement of the field in area such as agents, security, scenarios, and the implemented frameworks. Moreover, the study revealed that requirements conflict detections has recorded lower rate of interests in the research area, due to the overlapping complexity of the field. Thus, new methodology has emerged to tackle the requirement conflict issues in its full content. Such as, Ribeiro et al. [16] presented a different prospective to address the requirement conflict issue, an approach involving the Six Thinking Hats method as collaborative tool, using gamifications theory. Furthermore, the results revealed a decent number of contributions. However, the study is quantified as psychological approach for enhancing the stakeholder involvement. Thus, the complexity appears to have positive relationship with stakeholder involvement. In the efforts of reducing such complexity, Chi-Lun Liu [17] proposed a model based on the previous findings of Mairiza et al. [10] an ontologies-based concept that relied on defining conflict detection rules to refine the requirement. However, these rules were very limited to specific scenarios. Dey et al. [18] followed the steps of Chi-Lun Liu [17] and Ribeiro et al. [16] by attempted to rely on the ordinary cognitive level concept. However, the presented approach was successful only in exploring the system design based on the stakeholder views. Similarly, Anand et al. [19] Frequently asked requirements model, using the Apriori algorithm to reduce the conflict ratio among the elicited requirements. Although, the approach relied on similarity count. However, it was successful in providing new insides to the elicitation process. Accordingly, Mishra et al. [20] introduced a model driven approach for implementing early requirement conflict detection at the stakeholder elicitation phase. However, the required efforts and compliance by the stakeholder, were excessive and time consuming. Furthermore, Waheed et al. [21] attempted to eliminate the knowledge vaporization occurring in distributed team members to enhance the elicitation process. Although, the case study result was very promising. However, their result is still a reflection of a single context. Moreover, Saeed et al. [22] conducted a study to analyze the effectiveness of software development methodologies. The study provided a helpful reference in how to select the suitable model, according to the project properties. Finally, Ibrahim et al. [23] presented a framework to automate the requirement elicitation process through the use of machine learning algorithm. However, the presented

model was only successful in nominating the optimal elicitation technique for the elicitation scenario. Thus, producing broader requirement.

The literature review was rich with approaches that rely on textually specified requirements, to identify requirements conflicts. Mostly, these recorded approaches can be categorized as an ontological, psychological, cognitive and several algorithmic approaches. Moreover, these studies do not explicitly address the issue of requirement conflicts detections. However, they deal with core aspect of the fields emergent behavior. These studies lack in providing a mechanism to classify and detect requirement conflicts. Thus, there is a dire need for a smart formal method to assist in detecting the requirement conflict. A fresh perspective that focuses more on modeling, reasoning and theorem proving, and a methodology to sufficiently address the requirements elicitation issues, thereby enhancing the performance and reliability of the phase. Therefore, in this paper, Conflict Detection Automation Using Machine learning, is an attempt to automates the requirement detection process and improve the industry record of successful products.

3 Methodology and Materials

This section describes the methodology strategy used to develop the requirement conflict detection machine learning model. In addition to data collection process, mean shift algorithm implementation and the model analysis and results.

3.1 The Methodology Strategy

This section describes the methodology strategy used to implement the proposed requirement conflict detection model. Moreover, the proposed model incorporates its process in four steps [Fig. 1](#). The model starts at the data preparation phase to initialize the dataset through a transformation and mapping process. Next, the dataset is examined through the implementation of mean shift clustering process. Thus, producing data clustering representation of the requirements. Next, model validation phase processes the produced clusters accuracy, using the Standard Deviation (SD) and SE to confirms the clustering representation of requirements. Finally, the requirement classification process for amending the qualified requirements and dropping the detected conflicts sets in the requirements.

3.2 Data Collection

Conflict Detection Automation Using Machine Learning Model will be implemented and tested on a requirement elicitation dataset survey collected from alive IS project. Thus, the collected dataset is a descriptive data stating the desired IS project functionality and goals (Requirement Functionality). Based on the requirements gathered from the stakeholders, we performed data formatting for the sake of transforming the training set data. The dataset consists of 207 observations and 11 parameters of McCall's quality model [24]. Which are: Usability, to define the simplicity of the software. Understandability, to define the software purpose and functions process to the stakeholders. Consistency, to define the software GUI suitability. Efficiency, to define the software ability and resources to perform task within a specific timeframe. Effectiveness, to define the stakeholder satisfaction level of the software. Accuracy, to define the accuracy ratio of the produced result. Reliability, to define how reliable the software is in performing required functions under different scenarios/conditions. Robustness, to define the software robustness against the foreseen events. Portability, to define the software transportability measures functioning in different environments. Testability, to define the software testing measures. Maintainability, to defines the software maintaining process, such as fixing software bugs and adding new features. As shown in [Tab. 1](#).

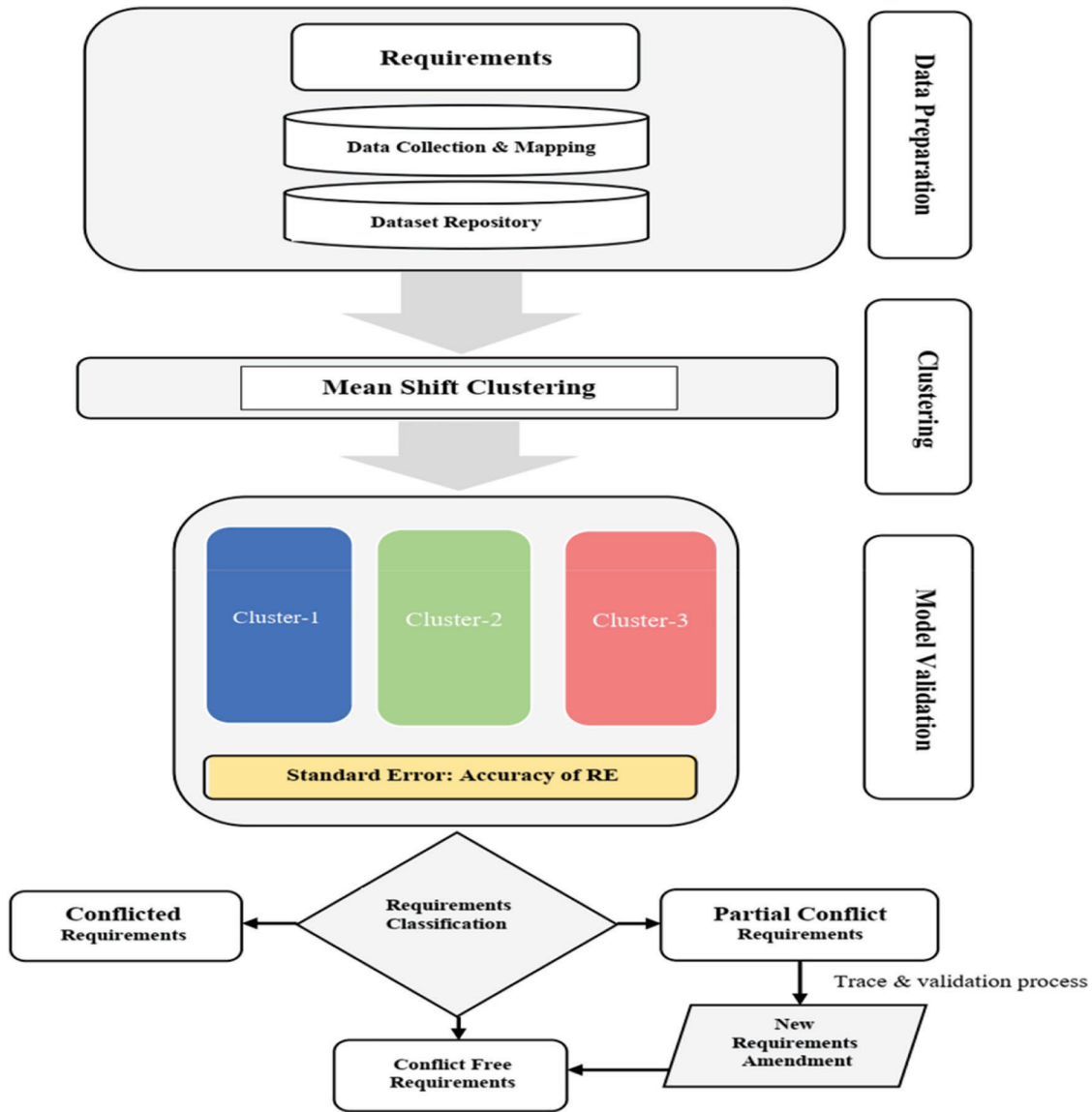


Figure 1: Methodological flowchart used in this study

Table 1: Functional requirement mapping and transformation process

S. No	Parameters	Qualifications
1	A	Usability
2	B	Understandability
3	C	Consistency
4	D	Efficiency
5	E	Effectiveness
6	F	Accuracy
7	G	Reliability

(Continued)

Table 1 (continued)		
S. No	Parameters	Qualifications
8	H	Robustness
9	I	Portability
10	J	Testability
11	K	Maintainability

Tab. 1 Summarize the mapping and transformation process of the collected functional requirement dataset.

The mapping process will be conducted on the surveyed requirements. Thus, transforming the records and observation points to numerical weights, corresponding to the 11 parameters of McCall's quality model [24]. Therefore, producing a full functional dataset. Representing the requirements spectrums of all the stakeholders.

3.3 Conflict Detection Automation Using Machine Learning

Fig. 2 illustrates the model process in context of the requirement elicitation phase. The model starts by transforming and preparing the collected requirements datasets. Furthermore, the unsupervised machine learning process is deployed using the mean shift algorithm. Additionally, requirement conflict verification to validate the clustering result, using the SD and SE. Next is the classification process in which the requirement accuracy rate determines the elimination of the ambiguous conflicting requirement. Thus, producing a conflict free requirement. The model was developed and evaluated using Python and the Scikit-learn (sklearn) module.

3.4 Mean Shift Algorithm

Mean shift is a type of clustering algorithm in contrast of unsupervised learning. It operates by shifting the data point towards the mode, to create a cluster representing data points with a similar feature. The algorithm iterates every data point closer to the centroid of the cluster till it assigns every data point to a group.

The mean shift algorithm function that defines the clustering of the requirement dataset is expressed as follows in Eq. (1).

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \quad (1)$$

where x = a point, n = number of points, K = selected kernel, d = dimension of the space, and h = window radius or bandwidth.

This model relies on the use of RBF kernel to compute the data point closeness in fashion similar to gaussian distribution. RBF kernel expressed as follows in Eq. (2).

$$K(X_1, X_2) = \exp\left(-\frac{\|X_1 - X_2\|^2}{2\sigma^2}\right) \quad (2)$$

where σ = the variance and our hyperparameter

$\|X_1 - X_2\|$ = the Euclidean (L_2 -norm) Distance between two points X_1 and X_2 .

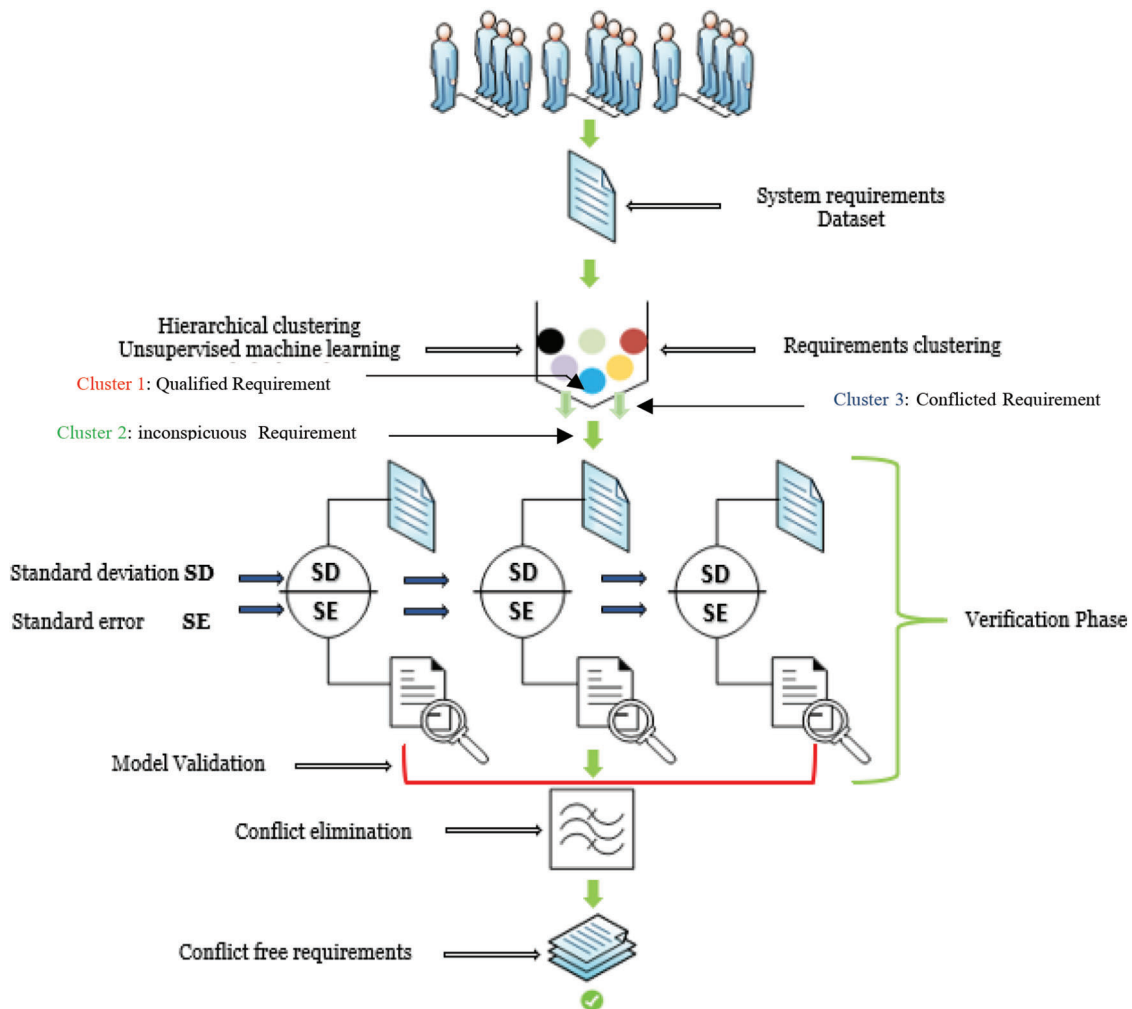


Figure 2: Conflict detection automation using machine learning

4 Analysis and Results

In this section, we analyze the Conflict Detection Automation Using Machine Learning Model. The model starts by initializing the mean shift algorithm. Furthermore, we start the fitting phase based on the data formatting (training sets), beside populating cluster centers and machine-chosen labels. To better visualize the cluster analysis process, a 3D presentation of the produced result is presented in Fig. 3.

The clustering output Fig. 3 was obtained just after a cluster analysis was conducted through the python programming language simulation. Whereas different clusters are represented by different colors, to illustrates the cluster recognition for the surveyed requirements datasets. Results of the cluster recognition corresponding to the analysis of the model, illustrate that our model was successful in detecting three centroid points representing three category level of requirements. Moreover, different colors in the image represent different identified clusters as shown in Tab. 2.

Tab. 2 Summarize the three clusters centroid points classification by the model.

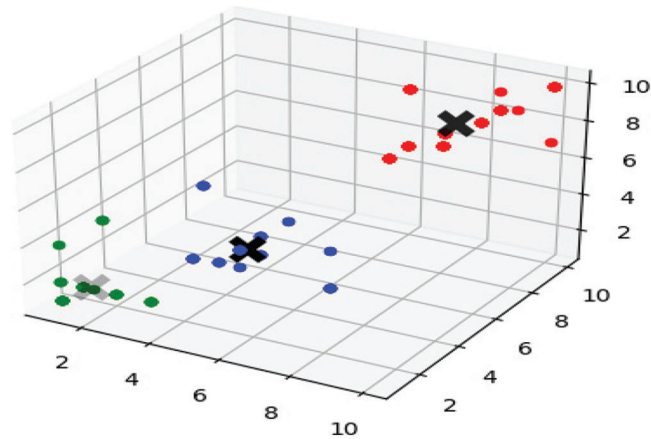


Figure 3: Model clustering output

Table 2: Cluster centroid points

Cluster	Centroid points
Cluster 1	in red color was successful in qualifying (134) observations.
Cluster 2	in green color was successful in qualifying (43) observations.
Cluster 3	in blue color was successful in qualifying (29) observations.

Extracting the following cluster analysis of the processed dataset has shown all three different centroid likelihood regions are correctly identified, based on the reviewing of the survey scale distribution. Subsequently, the clusters distributions were a representation of a range of views depending on the weight of the given parameters of McCall's quality model [24], as shown in Tab. 3.

Table 3: Requirements variant distribution

Cluster	Distribution
Cluster 1	is a representation of the stakeholders who appears to have a positive conclusion of the proposed system
Cluster 2	is a representation of the stakeholders who appears to have an average conclusion of the proposed system
Cluster 3	is a representation of the stakeholders who appears to have a negative (variations) conclusion of the proposed system

Tab. 3 Summarize the requirements variant distribution of the stakeholder views.

According to our hypothesis, in the proposed model and based on the comparative analysis of the produced results and dataset, we categorized these clusters, as shown in Tab. 4.

Tab. 4 Summarize the requirements categorization of the elicited functional requirements as per the research hypothesis.

In next section, we will verify the following results to illustrate the clustering recognition efficiency and ability to detect the requirement conflicts.

Table 4: Requirements categorization

Cluster	Categorization
Cluster 1	is a representation of the qualified functional requirements
Cluster 2	is a representation of the inconspicuous requirements, which needs more inspections and validation from the source to either be classified as a new amendment to the system or marked as disqualified requirement
Cluster 3	is a representation of the conflicted requirements

5 Model Validation

The requirement conflict validation is the third phase of proposed model. We explicitly placed this phase to be considered as a verification process, to create an assertive judgement of the model output for the sake of generalizing the results, as can be seen in Fig. 2.

In this phase will put into practice the use of the SD and SE. The SD function that measure of dispersion of requirement dataset from its mean is expressed as follows in Eq. (3).

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (c_i - \bar{c})^2}{N - 1}} \tag{3}$$

where σ = populations SD, N = the size of the population, c_i = each value from the population, \bar{c} = the population mean.

Where $\{c_1, c_2, \dots, c_n\}$ are the observed values of the sample items, \bar{c} is the mean value of these observations, and N is the number of observations in the sample and σ is the SD, which will be calculated for the all of the parameters as shown in Tabs. 5–7.

Table 5: Cluster – 1 SD dataset summary

		Requirement functionality										
		A	B	C	D	E	F	G	H	I	J	K
Summary Statistics	Count	134	134	134	134	134	134	134	134	134	134	134
	Mean	8.298	8.044	8.619	7.880	8.216	8.186	8.500	8.313	8.276	7.805	8.425
	Std	1.117	1.142	1.135	1.274	1.099	1.202	1.148	1.400	1.064	1.427	0.878
	Min	7.000	6.000	7.000	5.000	7.000	6.000	6.000	6.000	6.000	6.000	7.000
	25%	7.000	8.000	7.000	7.000	7.000	8.000	7.250	7.000	8.000	7.000	8.000
	50%	9.000	8.000	9.000	8.000	8.000	8.000	8.500	9.000	8.000	8.000	8.000
	75%	9.000	9.000	10.000	9.000	9.000	9.000	10.000	10.000	9.000	9.000	9.000
	Max	10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000

Tab. 5 Summarize the results of the cluster-1 statistical summary report produced by the SD function in the model validation phase. The data shows that cluster-1 represents 65% of the observations. Furthermore, cluster-1 scored mean average of 8.232 and SD average of 1.171. Thus, cluster-1 scored better on average and the data will be more spread out from the mean.

Table 6: Cluster – 2 SD dataset summary

		Requirement functionality										
		A	B	C	D	E	F	G	H	I	J	K
Summary Statistics	Count	43	43	43	43	43	43	43	43	43	43	43
	Mean	1.465	1.697	1.372	1.767	1.441	1.488	2.279	1.255	2.139	1.581	1.790
	Std	0.735	0.599	0.900	0.570	0.733	0.505	1.053	0.492	1.081	1.179	0.773
	Min	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	25%	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	50%	1.000	2.000	1.000	2.000	1.000	1.000	3.000	1.000	2.000	1.000	2.000
	75%	2.000	2.000	1.000	2.000	2.000	2.000	3.000	1.000	3.000	1.000	2.000
	Max	3.000	3.000	4.000	3.000	3.000	2.000	5.000	3.000	5.000	5.000	4.000

Tab. 6 Summarize the results of the cluster-2 statistical summary report produced by the SD function in the model validation phase. The data shows that cluster-2 represents 20.8% of the observations. Furthermore, cluster-2 scored mean average of 1.661 and SD average of 0.783. Thus, cluster-2 scored less than the average mean of cluster-1 and the data will be less spread out from the mean.

Table 7: Cluster – 3 SD dataset summary

		Requirement functionality										
		A	B	C	D	E	F	G	H	I	J	K
Summary Statistics	Count	29	29	29	29	29	29	29	29	29	29	29
	Mean	5.034	3.379	3.724	4.482	3.827	4.000	2.862	4.000	4.482	3.620	5.068
	Std	1.238	0.727	1.578	0.687	0.468	0.925	1.407	0.654	1.429	1.347	1.771
	Min	3.000	2.000	2.000	3.000	3.000	3.000	1.000	3.000	3.000	1.000	2.000
	25%	5.000	3.000	2.000	4.000	4.000	3.000	2.000	4.000	4.000	3.000	4.000
	50%	5.000	4.000	4.000	5.000	4.000	4.000	3.000	4.000	4.000	4.000	5.000
	75%	5.000	4.000	4.000	5.000	4.000	4.000	4.000	4.000	5.000	5.000	6.000
	Max	7.000	4.000	7.000	5.000	5.000	6.000	6.000	5.000	8.000	5.000	9.000

Tab. 7 Summarize the results of the cluster-3 statistical summary report produced by the SD function in the model validation phase. The data shows that cluster-3 represents 14% of the observations. Furthermore, cluster-3 scored mean average of 4.043 and SD average of 1.111. Thus, cluster-3 scored an average mean between cluster-1 and cluster-2.

Next, in the model validation is the SE, a statistical term that measures the accuracy. It will be calculated for the three clusters to sample the distribution of the observed values to illustrate the correlations between the scattering of individual data point around the population mean. Furthermore, lower SE values are considered a true mean representation of data. Contrarily, higher SE values represent irregularities of data. The SE function that measures the accuracy of requirement dataset from its mean is expressed as follows in Eq. (4).

$$SE_c = \frac{\sigma}{\sqrt{n}} \tag{4}$$

where σ is the SD of the population and n is the size (number of observations) of the sample.

This statistical information shown in [Tab. 8](#) supports the validation analysis and allocation of the given cluster’s hypothesis.

Table 8: Clusters SE accuracy rates

		Clusters accuracy rate		
		Cluster -1	Cluster -2	Cluster - 3
Requirement Functionality	A	0.096	0.112	0.230
	B	0.098	0.091	0.135
	C	0.098	0.137	0.293
	D	0.110	0.087	0.127
	E	0.094	0.111	0.086
	F	0.103	0.077	0.171
	G	0.099	0.160	0.261
	H	0.120	0.075	0.121
	I	0.091	0.164	0.265
	J	0.123	0.179	0.250
	K	0.075	0.117	0.328

[Tab. 8](#) Summarize the clusters SE variation

The SE is a representation of the spread of each of the data points are shown in the [Tab. 8](#). Which contains the SE values of the three clusters produced by the proposed conflict detections model. Furthermore, [Fig. 4](#) illustrates the variations of errors values rate of cluster-1, cluster-2 and cluster-3.

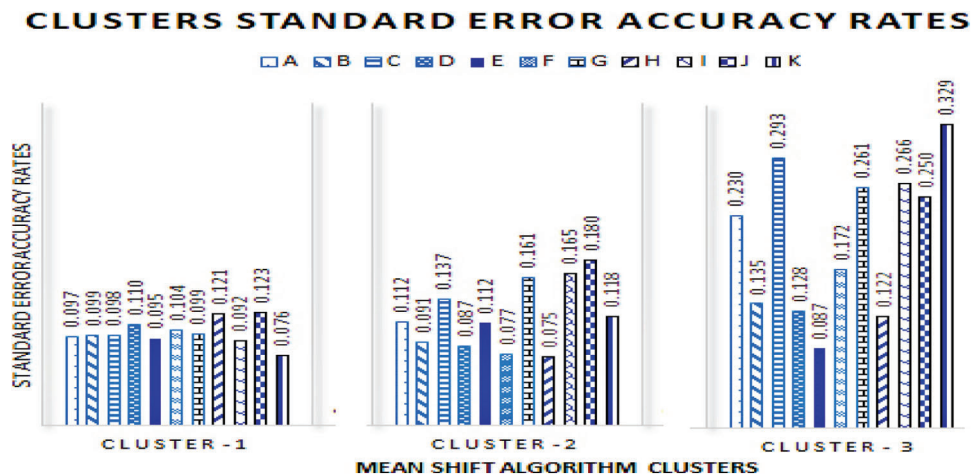


Figure 4: Cluster accuracy rate variation

The results findings of the requirements conflict validation phase, indicate the support of the proposed model results. Cluster-1, a representation of the qualified functional requirements is shown clearly in Fig. 4 exhibits the lowest SE values, as a projection of a true mean and a representation of higher measures of accuracy. Moreover, Cluster-2 which exhibits a medium SE accuracy values is a representation of these requirement elicitation observed values, that includes slightly fewer inconspicuous inclinations. Hence, calling for more inspections and validation process from the source, in order to determine the accuracy status of the requirement elicitation requests. Thus, either be classified as New requirement amendment to the system or to be labeled as a requirement conflict. Lastly, Cluster -3 which exhibits the highest SE accuracy values representing a higher variation range of requirement elicitation observed conflict values. Therefore, the finding of this model validation phase supports the results of the proposed requirement conflict detection model as expected. Thus, the given number of statistical analysis confirmed our findings was appreciable.

6 Discussion

Our research has highlighted the requirement conflict issues occurring in the requirement elicitation process. It is said to be the leading cause of the requirement conflicts due to improper elicitation process practice. This apparent lack of correlation is one of the prime reasons for increasing the ratio of unsuccessful IS projects and postponing the due date. Thus, resulting in huge losses. According to the survey, the cost of this failure is evaluated annually worldwide to be “approximately \$500 billion per month [1]. The illustrated issue is preventing the achievement of the shared understanding state in the requirement elicitation process. On the contrary, the produced requirement data was incomplete and inappropriate, producing inadvisable, defective, and conflicted requirements. Moreover, it also increased the budget cost and delayed the delivery of the IS projects due to the incremental, iterative process overhead.

The proposed Conflict Detection Automation Using Machine Learning Model comes as a systematic approach to address the illustrated issue. Moreover, this model helps redefine the elicitation process by producing conflict-free requirements that support the ongoing system/software design that corroborate the stakeholder inclination. The proposed model shown in Fig. 2 begins at the level of the desired system requirement survey. However, this descriptive information stating the desired system functionality will be transformed, weighted, and mapped based on McCall’s quality model [24]. To produce a full functional dataset representing the requirements spectrums of all the stakeholders and suitability for the machine learning training and test process. Thus, we implemented the illustrated experiment, which consisted of 207 observations and 11 parameters that define the desired system. The clustering algorithm shifts the data point towards the mode to create a cluster representing data points with similar features. The algorithm iterates every data point closer to the centroid of the cluster till it assigns every data point to a group.

As a result, the model successfully selected three centroid points represented in black, as shown in Fig. 3. The produced clusters were said to be representing a wide range of views based on their given weights. Based on a comparative analysis of the results and dataset, cluster-3 was labeled as a conflicting requirement. Further analysis was required to prove the following hypothesis. Thus, we commence the model validation phase. To do this entails the implementation of the SD and SE to measure the accuracy of the three clusters sample of distribution for the observed values around the population mean. Significantly, the experiment carried out confirmed our initial findings. Whereas Cluster-3 exhibited the highest SE values, representing the highest variation range in the functional requirement dataset. Thus, this cluster truly represents the conflicted requirement on the intended IS project. This result has further strengthened our confidence in the clustering classification convictions that the model hypothesis suggested. Of course, given the statistical analysis produced in the model validation phase. Moreover,

Cluster-1, with the lowest SE values, captured the qualified functional requirements response, correlated favorably well with the given hypothesis, and further supported the idea.

Furthermore, Cluster-2 produced slightly discordant SE values. However, Cluster-2 showed an apparent different variation of views among the stakeholder of the project. These inconspicuous inclinations variations in Cluster-2 call for more inspections and validation process from the source to determine the accuracy status of the requirement elicitation requests, to either be classified as New requirement amendment to the project, or to be labeled as a requirement conflict in an effort of building IS project that represents all the views. Nevertheless, the result was a confirmation of the model output. Finally, Cluster-3 depicted the highest SE values, exhibiting the wider range of incompatibility standards among a particular set of stakeholders. Therefore, no pattern can be concluded from such cacophony data. Thus, this cluster group will be labeled as ambiguous/conflicted requirements and excluded from the system design process. The values produced by the model have been found to be typical of a conducted experiment. Therefore, the conflict detection automation model was successful in classifying the requirements, (Conflict Free Requirement/Cluster-1, Partial Conflict Requirement/Cluster-2 & Conflicted Requirement/Cluster-3) through clustering algorithm as shown in Fig. 3. Moreover, the result finding was validated through statistical accuracy test as shown in Tab. 8. Moreover, the benefits of conflict detection automation far outweigh the disadvantages concerning previous methods. Thus, the proposed model is a clear improvement on previous methods. Furthermore, the results have been very promising and satisfactory in terms of the IS project under study. As confirmed by the stakeholders, considerable progress has been made regarding implementing a satisfactory IS project. We believe this solution will aid the requirement engineers and system analysts to further advance in the practice of requirement elicitation. However, it is plausible that two limitations may have influenced the results obtained. The first is the size of the IS projects under study. The second is the difficulty of collecting data from the sources. Thus, Further data collection would be needed to generalize the result of the model based on the model application on different IS project spectrums and environments.

7 Conclusion and Future Work

This study aimed to develop a Conflict Detection Automation Using Machine Learning Model. Hence, the motive of the presented paper is to increase the IS projects success ratio, reduce the risk of failure, and enhance the elicitation process by taking advantage of the machine learning powerful abilities and data analysis to automate the conventional requirement conflict detection process. Thus, enhancing the produced results. The model Fig. 2 initializes the collected functional requirements by a transformation and mapping process. Next, it investigated the effectiveness of implementing the mean shift algorithm to detect the requirement conflicts occurring in the requirement elicitation process. The model was successful in detecting three centroid points, as shown in Fig. 3.

Furthermore, the detected clusters represent requirements views variations among the stakeholders. As follows, Cluster-1: Functional requirements due to the majority of votes. Cluster-2: Ambiguous requirements. Cluster-3: Conflicted requirements.

The model validation phase has tested the following hypotheses by applying the SD and SE to calculate the sample distribution of the observed values of the model produced three clusters. Thus, illustrating the correlations between the scattering of individual data points around the population mean. Hence, producing the accuracy measurement values for all three clusters. Fig. 4 highlighted the statistical report of the result of the cluster and confirmed the given hypotheses. Although the proposed model produced a promising result, given the small sample size of the experiment, caution must be exercised. In our view, these results represent an excellent initial step toward automatizing the requirement elicitation process with the help of the advance machine language computing technology. Finally, additional experimental

tests are needed to generalize the application of the proposed model. It's recommended to increase the data collection sample size and expand the implementation on different IS projects environments. We hope that further tests will confirm our findings. Future studies should aim at examining the effectiveness of decision tree algorithm in detecting the requirement conflict in the course of IS projects.

Acknowledgement: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through General Research Project under grant number (project/ Design and Implementation of Intelligent System for Monitoring and Forecasting Rock Falls to Enhance Traffic Safety /number 110 /2019).

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] J. Johnson, K. D. Boucher, K. Connors and J. Robinson, "Collaborating on project success," *Software Magazine*, vol. 7, no. 2, pp. 15, 2001.
- [2] B. Chaouki and N. E. Zarour, "A novel negotiation approach for requirements engineering in a cooperative context," *Multiagent and Grid Systems*, vol. 15, no. 3, pp. 197–218, 2019.
- [3] J. Issam and A. Al-Qerem A, "Empirical study of analysts' practices in packaged software implementation at small software enterprises," *Int. Arab Journal of Information Technology (IAJIT)*, vol. 14, no. 4A, pp. 543–551, 2017.
- [4] Q. Ramadan, D. Strüber, M. Salnitri, J. Jürjens, V. Riediger *et al.*, "A semi-automated BPMN-based framework for detecting conflicts between security, data-minimization, and fairness requirements," *Software and Systems Modeling*, vol. 19, no. 5, pp. 1191–1227, 2020.
- [5] K. Beckers, S. Faßbender, M. Heisel and R. Meis, "A problem-based approach for computer-aided privacy threat identification," *Springer Annual Privacy Forum*, vol. 8319, pp. 1–16, 2012.
- [6] M. Deng, K. Wuyts, R. Scandariato, B. Preneel and W. Joosen, "A privacy threat analysis framework: Supporting the elicitation and fulfillment of privacy requirements," *Requirements Engineering*, vol. 16, no. 1, pp. 3–32, 2011.
- [7] C. Kalloniatis, E. Kavakli and S. Gritzalis, "Addressing privacy requirements in system design: The PriS method," *Requirements Engineering*, vol. 13, no. 3, pp. 241–255, 2008.
- [8] H. Mouratidis, C. Kalloniatis, S. Islam, M. P. Huget and S. Gritzalis, "Aligning security and privacy to support the development of secure information systems," *Journal of Universal Computer Science*, vol. 18, no. 12, pp. 1608–1627, 2012.
- [9] A. Egyed and P. Grunbacher, "Identifying requirements conflicts and cooperation: How quality attributes and automated traceability can help," *IEEE Software*, vol. 21, no. 6, pp. 50–58, 2004.
- [10] D. Mairiza and D. Zowghim, "An ontological framework to manage the relative conflicts between security and usability requirements," in *Third Int. Workshop on Managing Requirements Knowledge (MARK)*, Sydney, Australia, pp. 1–6, 2010.
- [11] D. Mairiza, D. Zowghi and N. Nurmuliani, "Towards a catalogue of conflicts among non-functional requirement," in *Proc. the Fifth Int. Conf. on Evaluation of Novel Approaches to Software Engineering SciTePress*, Sydney, Australia, pp. 20–29, 2010.
- [12] E. R. Poort and P. H. N. de With, "Resolving requirement conflicts through non-functional decomposition," in *Proc. Fourth Working IEEE/IFIP Conf. on Software Architecture (WICSA)*, Oslo, Norway, pp. 145–154, 2004.
- [13] A. Raian, F. Dalpiaz and P. Giorgini, "Reasoning with contextual requirements: Detecting inconsistency and conflicts," *Information and Software Technology*, vol. 55, no. 1, pp. 35–57, 2013.
- [14] D. Alkubaisy, K. Cox and H. Mouratidis, "Towards detecting and mitigating conflicts for privacy and security requirements," in *Proc. 13th Int. Conf. on Research Challenges in Information Science (RCIS)*, Brussels, Belgium, pp. 1–6, 2019.

- [15] J. Horkoff, F. B. Aydemir, E. Cardoso, T. Li, A. Maté *et al.*, “Goal-oriented requirements engineering: An extended systematic mapping study,” *Requirements Engineering*, vol. 24, no. 2, pp. 133–160, 2019.
- [16] C. Ribeiro, C. Farinha, J. Pereira and M. M. da Silva, “Gamifying requirement elicitation: Practical implications and outcomes in improving stakeholders collaboration,” *Entertainment Computing*, vol. 5, no. 4, pp. 335–345, 2014.
- [17] L. Chi-Lun, “Computer Standards & Interfaces CDNFRE: Conflict detector in nonfunctional requirement evolution based on ontologies,” *Computer Standards & Interfaces*, vol. 47, pp. 62–76, 2016.
- [18] D. Sangeeta and S. W. Lee, “Requirements elicitation for adaptive sociotechnical systems using repertory grid,” *Information and Software Technology*, vol. 87, pp. 160–179, 2017.
- [19] V. Ananda and M. Dinakaran, “Handling stakeholder conflict by agile requirement prioritization using apriori technique,” *Computers & Electrical Engineering*, vol. 61, pp. 126–136, 2017.
- [20] D. Mishraa, S. Aydin, A. Mishra and S. Ostrovska, “Computer standards & interfaces knowledge management in requirement elicitation: Situational methods view,” *Computer Standards & Interfaces*, vol. 56, no. 1, pp. 49–61, 2018.
- [21] S. Waheed, B. Hamid, N. Jhanjhi, M. Humayun and N. A. Malik, “Improving knowledge sharing in distributed software development,” *Int. Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, no. 6, pp. 434–443, 2019.
- [22] S. Saeed, N. Jhanjhi, M. Naqvi and M. Humayun, “Analysis of software development methodologies,” *Int. Journal of Computing and Digital Systems*, vol. 8, no. 5, pp. 446–460, 2019.
- [23] H. M. E. Ibrahim, N. Ahmad, M. B. Rehman, I. Ahmad and R. Khan, “Implementing and automating elicitation technique selection using machine learning,” in *Proc. Int. Conf. on Computational Intelligence and Knowledge Economy (ICCIKE)*, Dubai, United Arab Emirates, pp. 564–569, 2019.
- [24] J. McCall, P. A. Richards and G. Walters, “Factors in software quality: Concept and definitions of software quality,” in *Computer Science, US Rome Air Development Center Reports, US Department of Commerce*. Vol. 3. USA, 1977.