Robust Feature Extraction Techniques in Speech Recognition: A Comparative Analysis

Isra Khan[1] Ashhad Ullah[2] Rafi Ullah[3] Shah Muhammad Emad[4]

Abstract

As the world is moving towards new era known as the era of 'Artificial intelligence' where many of things will be controlled automatically through many sources such as face and thumb lock like this we can control things through sound as people are trying to do so and this thing getting hot day by day but it is not explored that much, in this paper we are exploring sound and its feature extraction techniques through which we can extract features from various types of sound and can make them applicable as this paper presents a survey on feature extraction to comparative analysis with respect to properties such as noisy data, complexity, accuracy, extraction method it will be helpful to use which data set with which type of sound. Feature extractions process has a direct relation with any of the machine learning algorithm. If feature extracted is robust, the use of underlining machine learning algorithm will increase accuracy. This paper targeted only the comparative analysis of features used in literature for sound. In future, two or more features will be combined to enhance the impact of sound recognition systems.

Keywords: Sound Recognition, Feature Extraction in Sound Recognition, Sound Detection, Robust Feature in Sound Recognition and Detection, Robust Features in speech recognition.

1 Introduction

Sound is the vibration that travels through air or any medium and these vibrations are audible when they reach an individual's ear and sound is formed by the unbroken and consistent vibrations. The first ever sound that was noted by invented by Édouard-Léon Scott de Martinville, was assembled by a gadget called a phonautograph in 1957. Phonautograph write out sound waves into a line that is drawn on paper but with these waves there are some features through which sound can be categorized in many classes or categories were extracted. Let's take an example when we hear any kind of sound our brain start processing on it and categorize that sound like we can predict that this is the voice of a female without seeing that female because we know which value range belongs to which category, but the major challenge is to extract the features and their different ways of doing it such as MFCC, RASTA, LPCC, Cepstral Analysis, LPC and many others [1]. The majority of these proposed frameworks consolidate two handling stages. The first stage studies the received sound wave and extracts parameters (features) from it. The feature extraction the extracted features and both of these stages are defined briefly below. Many set of feature extraction are proposed earlier for audio classification [2,3,4]. Largest portion has been covered by low-level signal features and then second important feature set consist of Mel-frequency cesptral coefficient (MFCC) [5] and then those remaining features come.

[1]*Karachi Institute of Economics & Technology, Karachi | khanisra@gmail.com*
[2]*Karachi Institute of Economics & Technology, Karachi | ashhadullah19@gmail.com*
[3]*Karachi Institute of Economics & Technology, Karachi | rafiafridi783@gmail.com*
[4]*Karachi Institute of Economics & Technology, Karachi | shahmuhammademad@gmail.com*

All of the features are used audio classification and are very powerful in classifying the audio class but it gradually decrease when amount of classes increase. Therefore, using which feature set with which amount of classes is an issue which can create further more issues if we select wrong feature set with respect to the problem description, result will help you with comparison done which will guide you when to use which set [6, 7].

Speech is that the most typical manner of communication between humans. Speech also carries the information related to the speaker. To recognize the speaker there are features exists in the speech signal. These extracted features will be useful in training of the model for speech recognition.

In audio processing, feature extraction is the backbone. The importance of feature extraction technique can never be ignored in speech recognition and processing systems [8]. But these features that are extracted must fulfill these criteria while doing speech recognition. These standards are [9]:

- Easy to measure extracted speech features
- Not be susceptible to mimicry
- Perfect in showing environment variation
- Stability over time

For feature extraction audio samples are collected and then converted to digital signals at a regular interval. At these voice samples noise reduction is performed so that the original audio sample can be find to perform feature extraction on it. For the speech recognition we extract the features from the digital signals which provide the acoustic properties of that specific digital dataset that is really useful for representing the speech signal.

These speech signals are slowly timed varying signals (quasi-stationary). When analyzed for a short time interval for example examined for example 5ms-100ms, the attributes seems to be relatively stationary. However if sound/vocal features are modified for a specified time interval, it reflects the different values of spoken audio features. The information of audio signal can be categorized by using short term amplitude spectrum of the audio wave form. These techniques are known as phonemes helps in the extraction of sound features of short term amplitude spectrum from audio signals called phonemes [10].

Rest of the paper is divided as follow; Section I is about the literature review or Related Work, Section II is the detail explanation of different features that can be extracted from sound, Section IV is Result section that is detail comparative analysis of different features extraction technique, Section V is the concluding the topic and Section VI is the future potential area.

2 Related Work

Authors of [11] focused on the comparative analysis of widely used feature extraction techniques related to speech recognition and in the end of the research has conclude that the PLP is extracted on the conception of logarithmically spaced filter bank, combined with the conception of human hearing system and has improved results than LPC.

According to paper [29], author has extracted MFCC feature and de-noise the audio sample and also enhanced the MFCC feature by calculating the delta energy for the coefficient.

Authors has extracted MFCC feature for the speech emotion detection discussed in detail in [30]. MFCC feature is extracted and worked very efficiently and train the model for the detecting of speech detection emotion.

Isolated speech recognition by using the MFCC and Dynamic Time Wrapping (DTW) was focused by the authors of [31]. In this research features for the isolated speech recognition were extracted by using the MFCC.

In [14], authors has identified and focused on the problem of optimizing the acoustic features set by Ant Colony Optimization for the Automatic speech recognition. Speech signal is considered as input in this research and feature extraction is performed over this signal using MFCC extraction method, total 39 coefficients are extracted in this research by using MFCC.

Comparative analysis of speech recognition has done in paper [33]. These analysis was performed on noisy conditions on the widely used feature extraction techniques named MFCC,LPCC,PLP, RASTA-PLP and HMM and has analyzed that PLP distinctly gave the maximum percentage of recognition and the grouping of LPCC, PLP and RASTA provided the output as third maximum recognition percentage.

In [34], Authors have worked on the change in detection in multi-dimensional unlabeled data in which features were extracted by using the PCA feature extraction technique.

According to the authors of [35], they focused on the PCA drawbacks which are high computational cost, extensive memory utilization and low adequacy in handling expansive dimensional datasets, so author has proposed a new technique Folded-PCA. By using this new proposed technique these drawbacks can be resolve.

Drawbacks of PCA was discussed in paper [36]. These drawbacks are: computational cost, extensive memory utilization and low adequacy in handling expansive dimensional datasets, so they analyzed two variation of the PCA technique SPCA and Seg-PCA. These variations can be helpful to reduce the drawbacks of PCA.

Authors in [20] have done the survey over the feature extraction technique and conclude that the LPC is vector dimension and has high computational cost and also reduce accuracy and their window size which is not good for non-stationary speech signals such as speech signal.

In [38], Authors has proposed the new technique for the noisy speech recognition based on auditory filter modeling-based feature extraction and gives the result that LPC is less efficient in this manner in comparison with PLPaGc.

Comparative analysis for the speech recognition specific for Hindi language words, and has analyzed that LPCC gives less recognition rate for isolate, paired and hybrid words as compared to MFCC has performed in [39].

A new recognition system was proposed in [40]. This system uses the acoustic waves generated by the construction equipment, this will be very helpful to avoid external damages. Feature extraction for the recognition system was done by combining LPCC and SVM.

The RASTA feature extraction technique in combination with TANDEM was used by the authors of [41]. The authors stated that this technique is an efficient way to represent the message-information in the speech signal.

3 Feature Extraction Techniques

Various Features Extraction techniques has been observed in the literature, that is used for sound recognition and sound detection. Each one has its own advantages and disadvantages depending upon the environment I-e nature of problem. For example features extraction used in sounds related to school cafe will be having different impact on sound of vehicles. Some of the features extraction techniques that are observed during our research are:

- Mel-frequency Cepstral Coefficients (MFCC)
- Perceptual Linear Predictive (PLP)
- Relative Spectral Processing (RASTA)
- Linear Prediction Cepstral Coefficient (LPCC)
- Principle Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Wavelet
- Dynamic Time Warping (DTW)
- Combined LPC and MFCC
- Kernel based feature extraction
- Independent Component Analysis (ICA)
- Integrated phoneme subspace method
- Probabilistic Linear Discriminant Analysis (PLDA)
- Linear Prediction Coefficient (LPC)
- Discrete Wavelet Transformation (DWT)
- Wavelet Packet Decomposition (WPD)
- Gammatone Frequency Cepstral Coefficient (GFCC)
- Gaussian Mixture Model (GMM)

This paper targeted only six features extraction technique (MFCC, PLP, PCA, LPC, LPCC and RASTA) to compare on the basis of several parameters such as Impact in presence of noise I-e noisy data, Complexity (in case of features extraction and computation), Accuracy and Feature Extraction Method.

A Mel-Frequency Cepstral Cofficients

MFCC is one of the most important techniques used to extract the feature from speech signal [11] that is actually based over the human's ear scale bandwidth. It uses the low and the high

frequencies, measured in Hertz (Hz) to get the speech signal. MFCCs considered as frequency domain features that are more accurate in comparison with time domain features [11]. These signals are then divided into the audio frames to calculate the MFCC. Let each frame of audio signal contains the N samples and considers the next and the previous frames of the audio signal is separated by M samples where M<N. All audio frames are multiplied by a Hamming window. The hamming window [16] value can be calculated using this equation 1.

$$W(n) = 0.54 - 0.46\ cos(2^{\pi n}/_{N} - 1) \qquad (1)$$

Then speech signal is transformed to frequency domain from time domain by utilizing its Discrete Fourier Transform.

The melfrequency scale [17] is consider as linear frequency having spacing less than 1000 Hz and a logarithmic spacing more than 1000Hz .As a reference point ,a pitch of a 1 KHz tone ,40 dB above the threshold perceptual hearing, is defined as 1000 mels. So, to find the mels for a specific given frequency f in Hz we can use this equation 2.

$$Mel(f) = 2595*\log10(1 + f/700) \qquad (2)$$

The MFCC features correspond to the total power of the log in a critical band around the center frequencies. Finally, for the calculation of cepstral coefficients, the Inverse Discrete Fourier Transformer is applied; finally calculate the DCT of the output from the filter bank. The resultant value is the actual Mel-Frequency Cepstral Coefficient.

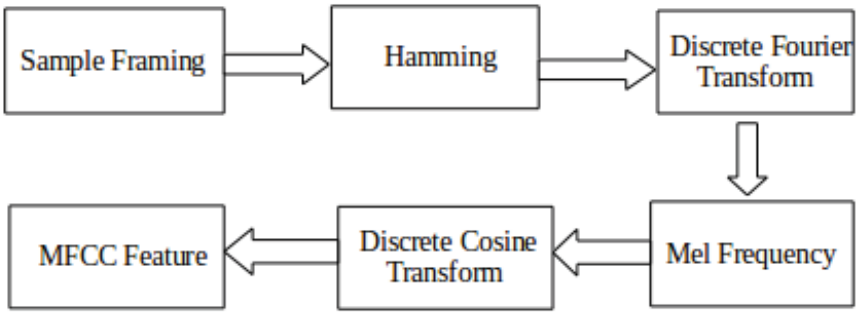

Figure 1: MFCC features Extraction Technique

B Perceptual Linear Predictive

The PLP model aims at human vocalizations based on the concept of hearing psychophysics and then more precisely in the process of extracting features. PLP increases the rate of speech recognition and also eliminates irrelevant speech information [18]. PLP technique is quite similar to LPC but differs from MFCC. PLP mainly consists of three steps. First one is for critical band analysis. Second is for equal loudness and the third one is for intensity-loudness and power-law relation. PLP carries out spectral analysis with frame of N samples with N band filters on the speech vector. For the experiments, 256 window sizes and 24 filter banks are used. The PLP filters are then produced with pre-emphasis and scale of bark. Next step is the estimation

of power spectrum with the power law [18]. Now computed PLP spectrum is forwarded to LP analysis with the frequencies. At-last LP analysis is performed along FFT and then final values are observed by calculating the inverse of FFT.

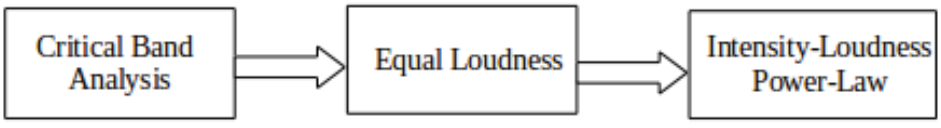

Figure 2: PLP features Extraction Technique

C Linear Prediction Coefficient

The LPC is actually works on the prediction. In samples of speech signal, we can predict the nth samples, which can be represented by summarizing the previous samples of the target signals (k). The inverse filter production should be carried out to match the formants region of the speech samples [19]. The LPC process is therefore the application of these filters in the samples [20]. The main idea of LPC is to approximate the current (n) acoustic sample s(n) with the previous samples s(p).

$$s(n) \approx a_1(n-1) + a_2(n-2) \pm \cdots \pm a_p(n-p) \quad (3)$$

Then LPC is obtained using the Levinson-Durbin recursive algorithm [20].

$$H(z) = \frac{1}{1 - \sum_{k-1}^{p} a_k z^{-k}} \quad (4)$$

H(z) reflects the propagation path of the acoustic signal. Let c (n) be the impulse response [20]:

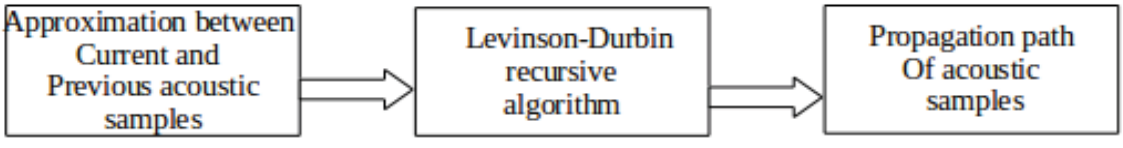

Figure 3: LPC features Extraction Technique

$$H^{Z} = lnH(z) = \sum_{n=0}^{00} c(n) z^{-k} \quad (5)$$

D Linear Prediction Cepstral Coefficient

Linear Prediction Cepstral is an enhanced version of LPC method. The representation of linear predictive coefficients is in cepstrum domain can be reflected as new coefficients known as

linear predictive cepstral coefficients [21]. The value of LPCC coefficient can be computed by using LPC equations which are as follows.

$$c_1 = a_1 \tag{6}$$

$$c_n = a_n + \sum_{k-1}^{n-1} \frac{k}{n} \; c_k a_{n-k} 1 < n \leq p \tag{7}$$

$$c_n = a_n + \sum_{k-1}^{n-1} \frac{k}{n} c_k a_{n-k} n > p \tag{8}$$

Where $C_1, C_2, \cdots C_n$ are the LPCC.

E Principle Component Analysis (PCA)

The PCA is thought a Principle part Analysis – this is often a statistical analytical tool that's used to explore kind and cluster information. PCA take an over-sized variety of correlate (interrelated) variables and rework this information into a smaller variety of unrelated variables (principal components) whereas holding largest quantity of variation, so creating it easier to work the information and build predictions. PCA could be a method of distinguishing patterns in information, and expressing the information in such some way on highlight their similarities and variations. Since a pattern in information is hard to seek out in information of high dimension, wherever the posh of graphical illustration isn't offered, PCA could be a powerful tool for analyzing information [10].

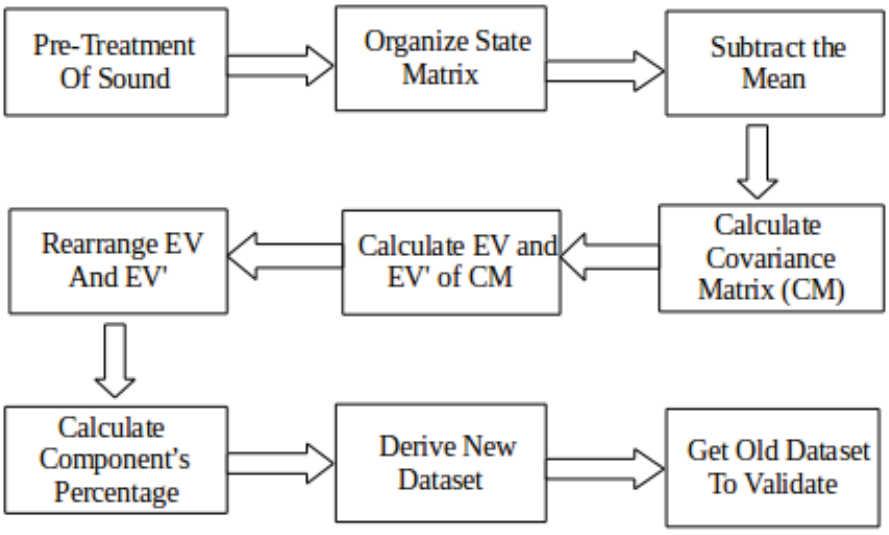

Figure 4: PCA features Extraction Technique

Where EV is Eigen Vector and EV' is Eigen Value.

F Relative Spectral Processing

The RASTA is method of extracting the relevant information from a sound or any speech signal and the main objective of this technique is to eliminate the robustness of speech recognition in noise or in the real time environments [16] and it is usually done by using time trajectories of band pass filter of logarithmic speech value, infact it is the extension of the original method by combining additive noise and convolution noise [15].

RASTA is a voice improvement based on linear filtering of the short-term power spectrum of the noisy audio signal, as shown in Figure 5. The input speech signal spectral values are compressed by a nonlinear compression rule (a = 2/3) before filtering and expanded after filtering (b = 3/2) [16].

Output of each filter is given as,

$$S_i\,(\mathrm{K}) = \sum_{j=-M}^{M} W_i\,(j)\,Y_i\,(k) \qquad (9)$$

Si(k) is a clean speech estimate and Yi(k) is the noisy audio spectrum, Wi(j) is the filter weights and M is the filter order.

These values can be set to the required processing or the corresponding set of problem.

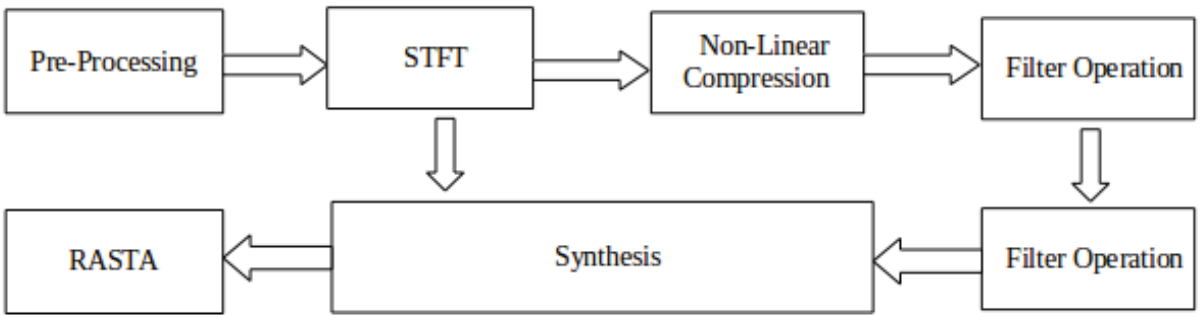

Figure 5: RASTA features Extraction Technique

4 Results

Details comparison observed during this survey are following that will help researchers and practitioner to know the insights of different feature extraction techniques used in sound recognition and detection problems.

Table 1: Comparative Analysis of Features Extraction Techniques Used in Sound Recognition and Detection

Features Extraction	Noisy Data	Complexity	Accuracy
MFCC	Poor result on noisy data [27]	Less Complex and High performance rate	92% [24]
PLP	poor result on noisy data due to spectral balance of formant	Slightly Complex	Better Performance than LPCC and MFCC [25]
PCA	Doesn't work well on noisy data as it does not reduce noise completely.	Slightly Complex and High Performance Rate	54.66% [8]
LPC	Not good for noisy data [27].	Less complex [27].	Good Accuracy, reliability and robustness [24]
LPCC	Shows poor result on highly noised data [27].	Simple and good performance [27]	Accuracy is 88% [26]
RASTA	Works good on noisy data as it enhances data by removing noisy distortions [27].	Slightly Complex	A robust technique. Low modulation frequencies are captured [24].

Table 2: Comparative Analysis of Features Extraction Techniques Used in Sound

Features Extraction	Extraction Method	Final Comments
MFCC	Dynamic method [27].	Mostly used where human ear bandwidth scale exists.
PLP	Combines the linear prediction analysis and spectral analysis [9]	Increases the recognition rate and also removes noise.
PCA	Non-Linear method [27].	Eigen vector based. Reduce Components / Dimensions of Features
LPC	A static method [29].	Used for extraction at lower rate. It can be used in sound recognition of abnormal sounds
LPCC	Use Autocorrelation analysis [27].	Used in cepstral domain.
RASTA	Non-Linear Compression [16].	Highly recommended in domain where there is noise, it will extract good features in noisy data

5 Conclusion

In this paper we have discussed some widely used feature extraction techniques in the domain of speech recognition. The motivation for doing this comparative analysis is because there are many feature extraction techniques that are available and very few of them are really helpful. This paper will guide the researchers for methods feature extraction technique and it will also help them to differentiate between different Novel and Robust features can also be extracted by combining many of the existing features to enhance the capability of sound detection and recognition systems. Developing a system that will record complete meeting conversation in a dialogue form, sentence spoken by each person against their name (if known), otherwise a separate line by some person “i”. This system will reduce time of recording meeting or writing manual points where some points may skipped or interpreted wrong.

References

[1] McKinney, Martin, and Jeroen Breebaart. "Features for audio and music classification." (2003).

[2] Davis, Steven, and Paul Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences."IEEE transactions on acoustics, speech, and signal processing 28, no. 4 (1980): 357-366.

[3] Wold, Erling, Thom Blum, Douglas Keislar, and James Wheaten. "Content-based classification, search, and retrieval of audio."IEEE multimedia 3, no. 3 (1996): 27-36.

[4] E. Scheirer and M. Slaney. Construction and evaluation of a robust multifeature speech/music discriminator. In Proc. ICASSP, pages 1331–1334, Munich, Germany, 1997.

[5] H. Hermansky and N. Malayath. Spectral basis functions from discriminant analysis. In International Conference on Spoken Language Processing, 1998

[6] M. Zhang, K. Tan, and M. H. Er. Three-dimensional sound synthesis based on head-related transfer functions. J. Audio. Eng. Soc., 146:836–844, 1998.

[7] T. Zhang and C. C. J. Kuo. Audio content analysis for online audiovisual data segmentation and classification. IEEE Transactions on speech and audio processing, 2001.

[8] Prasad, K.S., Ramaiah, G.K. and Manjunatha, M.B., 2017. Speech features extraction techniques for robust emotional speech analysis/recognition. Indian Journal of Science and Technology.

[9] Vimala, C. and Radha, V., 2014. Suitable feature extraction and speech recognition technique for isolated tamil spoken words. International Journal of Computer Science and Information Technologies (IJCSIT), 5(1), pp.378-383.

[10] Shrawankar, U. and Thakare, V.M., 2013. Techniques for feature extraction in speech recognition system: A comparative study. arXiv preprint arXiv:1305.1145.

[11] Dave, N., 2013. Feature extraction methods LPC, PLP and MFCC in speech recognition. International journal for advance research in engineering and technology, 1(6), pp.1-4.

[12] Barchiesi, D., Giannoulis, D., Stowell, D. and Plumbley, M.D., 2015. Acoustic scene classification: Classifying environments from the sounds they produce. IEEE Signal Processing Magazine, 32(3), pp.16-34.

[13] Desai, N., Dhameliya, K. and Desai, V., 2013. Feature extraction and classification techniques for speech recognition: A review. International Journal of Emerging Technology and Advanced Engineering, 3(12), pp.367-371.

[14] Këpuska, V.Z. and Elharati, H.A., 2015. Robust speech recognition system using conventional and hybrid features of mfcc, lpcc, plp, rasta-plp and hidden markov model classifier in noisy conditions. Journal of Computer and Communications, 3(06), p.1.

[15] H. Hermansky and N. Morgan, "RASTA processing of speech," in IEEE Transactions on Speech and Audio Processing, vol. 2, no. 4, pp. 578-589, Oct. 1994.

[16] Kurzekar, P.K., Deshmukh, R.R., Waghmare, V.B. and Shrishrimal, P.P., 2014. A comparative study of feature extraction techniques for speech recognition system. International Journal of Innovative Research in Science, Engineering and Technology, 3(12), pp.18006-18016.

[17] Tiwari, V., 2010. MFCC and its applications in speaker recognition. International journal on emerging technologies, 1(1), pp.19-22.

[18] Chandra, E., and K. Manikandan M. Sivasankar. "A proportional study on feature extraction method in automatic speech recognition system." (2014).

[19] Narang, S. and Gupta, M.D., 2015. Speech Feature Extraction Techniques: A Review. International Journal of Computer Science and Mobile Computing, 4(3), pp.107-114.

[20] Yang, S., Cao, J. and Wang, J., 2015, July. Acoustics recognition of construction equipments based on LPCC features and SVM. In Control Conference (CCC), 2015 34th Chinese (pp. 3987-3991). IEEE.

[21] Kaur, K. and Jain, N., 2015. Feature Extraction and Classification for Automatic Speaker Recognition System-A Review. International Journal of Advanced Research in Computer Science and Software Engineering, 5.

[22] Gupta, D., Bansal, P. and Choudhary, K., 2018. The state of the art of feature extraction techniques in speech recognition. In Speech and Language Processing for Human-Machine Communications (pp. 195-207). Springer, Singapore.

[23] Chaudhary, G., Srivastava, S. and Bhardwaj, S., 2017. Feature Extraction Methods for Speaker Recognition: A Review. International Journal of Pattern Recognition and Artificial Intelligence, 31(12), p.1750041.

[24] Gupta, H. and Gupta, D., 2016, January. LPC and LPCC method of feature extraction in Speech Recognition System. In Cloud System and Big Data Engineering (Confluence), 2016 6th International Conference (pp. 498-502). IEEE.

[25] Khara, S., Singh, S. and Vir, D., 2018, April. A Comparative Study of the Techniques for Feature Extraction and Classification in Stuttering. In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT) (pp. 887-893). IEEE.

[26] Kianisarkaleh, A. and Ghassemian, H., 2016. Spatial-spectral locality preserving projection for hyperspectral image classification with limited training samples. International journal of remote sensing, 37(21), pp.5045-5059.

[27] Singh, P.P. and Rani, P., 2014. An approach to extract feature using mfcc. IOSR Journal of Engineering, 4(8), pp.21-25.

[28] Kaur, J. and Sharma, A., 2014. Emotion detection independent of user using mfcc feature extraction. International Journal of Advanced Research in Computer Science and Software Engineering, 4(6).

[29] Dhingra, S.D., Nijhawan, G. and Pandit, P., 2013. Isolated speech recognition using MFCC and DTW. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 2(8), pp.4085-4092.

[30] Poonkuzhali, C., Karthiprakash, R., Valarmathy, S. and Kalamani, M., 2013. An approach to feature selection algorithm based on ant colony optimization for automatic speech recognition. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 2(11), pp.5671-5678.

[31] Kuncheva, L.I. and Faithfull, W.J., 2014. PCA feature extraction for change detection in multidimensional unlabeled data. IEEE transactions on neural networks and learning systems, 25(1), pp.69-80.

[32] Zabalza, J., Ren, J., Yang, M., Zhang, Y., Wang, J., Marshall, S. and Han, J., 2014. Novel Folded-PCA for improved feature extraction and data reduction with hyperspectral imaging and SAR in remote sensing. ISPRS Journal of Photogrammetry and Remote Sensing, 93, pp.112-122.

[33] Ren, J., Zabalza, J., Marshall, S. and Zheng, J., 2014. Effective feature extraction and data reduction in remote sensing using hyperspectral imaging [applications corner]. IEEE Signal Processing Magazine, 31(4), pp.149-154.

[34] Hibare, R. and Vibhute, A., 2014. Feature extraction techniques in speech processing: a survey. International Journal of Computer Applications, 107(5).

[35] Zouhir, Y. and Ouni, K., 2014. A bio-inspired feature extraction for robust speech recognition. SpringerPlus, 3(1), p.651.

[36] Gulzar, T., Singh, A. and Sharma, S., 2014. Comparative analysis of LPCC, MFCC and BFCC for the recognition of Hindi words using artificial neural networks. International Journal of Computer Applications, 101(12), pp.22-27.

[37] Hermansky, H. and Fousek, P., 2005. Multi-resolution RASTA filtering for TANDEM-based ASR (No. REP_WORK). IDIAP.

Improving Requirement Prioritization process in Product line using Artificial Intelligence technique

Wasi Haider[1] Yaser Hafeez[2] Sadia Ali[3]

M Azeem Abbas[4] M Numan Rafi[5] Abdul Salam[6]

Abstract

Product families emerged a new and useful development technique in the field of software development. In Software Product Line (SPL) there are some core assets and some variants so using these assets anyone can build their desired product in very short time and effort. While working in product family's environment we must keep an eye on the requirement prioritization and ranking because that requirement are very important because these requirement lay the foundation of the core and variants assets which are the building blocks of SPL. So there are some major issues which we face are the more human interaction, ambiguous requirements and wrong or no requirement ranking. In this paper we proposed a framework for the ranking of stakeholders' requirements for the SPL's variant and core assets using the case base reasoning CBR if available in previous use or assign them new ranking according to their requirement and their assign ranking for software product line. We evaluated our framework by empirical study. The results prove that the considerable improvement for different parameters is achieved by our framework as compared to conventional approaches of requirement prioritization.

Keyword: Software Product Line (SPL); Requirement Prioritization (RP); Case Base Reasoning (CBR); Artificial intelligence (AI)

1 Introduction

Software product family is a interrelated software systems, sharing a common and managed collection of features to accomplish the wants of a suitable market section [1]. The main goal of SPL is reuse in an effort to enhance the quality and production while reducing cost as well as time to market. SPL engineering has become an efficient and minimizes time-to-develop approach for providing a common model for developing product families. The central concept at the back of SPL is to provide a stage with common and distinct components of a software system identified in order to build a consistent line of products [2]. Software product variants are often develop from an early product development. These product variants are generally share some common but they are also different from each other due to upcoming change request to fulfill the specific demand and requirement of the end user [3]. As a number of features and the number of products increase, it is significance re-engineering product variants into a SPL

[1] *PMAS-Arid Agriculture University, Rawalpindi | wasihaider734@gmail.com*
[2] *PMAS-Arid Agriculture University, Rawalpindi | yasir@uaar.edu.pk*
[3] *PMAS-Arid Agriculture University, Rawalpindi | sadiaalief@gmail.com*
[4] *PMAS-Arid Agriculture University, Rawalpindi | azeem.abbas@uaar.edu.pk*
[5] *PMAS-Arid Agriculture University, Rawalpindi | miannuman10437@gmail.com*
[6] *PMAS-Arid Agriculture University, Rawalpindi | abdulsalam2301@gmail.com*

for systematic reuse. The first step of SPLE is to extract a feature model. This further suggested recognizing the common and variant features. Manual reverse engineering of feature model for the available software variants is time and effort taking [4].

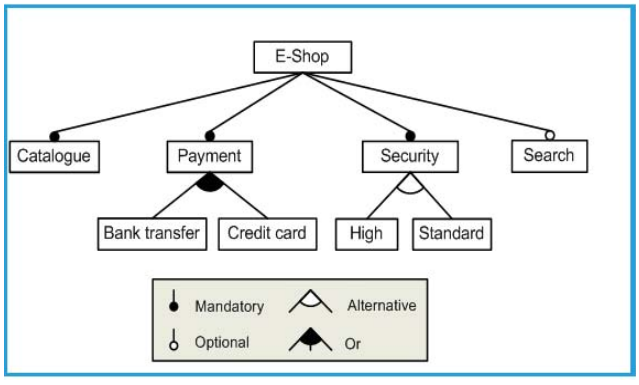

Figure 1: SPL Feature Model

When developing software, Requirements Engineering is field of defining, documenting and maintaining software requirements, mostly described in natural language [5]. This information motivated some proposals to use Natural Language Processing (NLP) to minimize the uncertainty, identify omitted information, and even enhance traceability with remaining stage of process [6].

Requirement prioritization (RP) is a main part in the requirement engineering phase. RP plays a vital part in the RE process, particularly, regarding vital tasks like requirements negotiation and software release [7]. Outstanding RP is necessary to any well-run project. It ensures that project concentrate on the main parts first, and that everybody perceived and conforms about what the project's most important parts are. There are many techniques, which are helpful for specification and prioritization of requirement according to stakeholder's time, cost, nature of the project etc. When developer used any requirement prioritization technique and find out the priority or ranking of requirements in any system, they save the ranking of the requirement with it all information and stored it in database for reuse purpose in future. For knowledge management and reuse of previous knowledge, researcher adapted AI technique called case based reasoning (CBR). CBR retrieve previous solutions for current problem solving base on expert knowledge intelligently in different scenarios [8].

In this paper, we have presented a comprehensive framework for the requirements ranking in which we extract the commonalities Cs and variabilities Vs of the software product line from the requirement document using J48 Decision algorithm. It initiates the rules for the calculation of the target variable. With the assistance of J48 classification algorithm [9] the significant distribution of the data is easily understandable. After finding the Cs and Vs apply the CBR and find the previous ranking if available then assign them else assign their ranking and find out the sorted prioritized requirement list.

The rest of this paper is structured as follows: Section 2 present literature review. In Section 3, we present our framework. In Section 4, we present evaluation and discussion.

2 Literature Review

The growing complication and cost of software-intensive systems has led developers to find the alternatives of reusing software parts in development of systems. One approach to increasing re-usability is to develop a SPL. Existing research has paying attention on techniques that create a configuration of an SPL in a single step. First, they present a formal model of multi-step SPL. Second, present the solutions to these SPL configuration problems can be automatically derived with a constraint. In future work, they plan to investigate Real-time configuration process monitoring [10].

The analysis of the requirements artifacts (SRS document, use case models) is a time taking process when performed manually. There is also required for creating consistent and complete collection of NFRs from user-specific individual projects in SPL. Therefore, they [11] propose a method to create Domain NFRs from Product NFRs using model driven approach.

It is essential for an organization to boost value creation for a given investment. The principle RE activities are to add business value that is considered for in terms of return on investment of a software product. This [12] paper provides insight into the release planning processes used in the software industry to create software product value. It presents to what degree the significant stakeholders' viewpoints are spoken to in the basic decision-making process.

SPLE strengthened high-level constructive software reuse by exploiting commonality and managing variability in a product family. To overcome the complexity of the modeling, it is divided into two views a feature tree and a dependency view [13].

In the development of a SPL, any project requires to grow core assets according to the change in environment, market, and technology. In order to successfully grow core assets, it is critical for the project to get ready and use a standardized strategy for prioritizing requirements. In paper [14], authors examine the evolution of foundation assets. Tacit knowledge for prioritizing requirements was extracted. Such knowledge was made explicit and clear to develop a way for prioritizing.

Reusing of software varies from operational, ad-hoc and short-term to strategic, planned and long-term. They [15] present and compare two different requirements-led approaches. The first deals with requirements reuse and re-usability in context of product line engineering and second in context of CBR. To assist large-scale development they proposed a Feature-Similarity model.

Requirements assurance seeks to maximize confidence in the quality of requirements through audit and review. Authors of [16] present a method that applies well-established text-mining and statistical methods to minimize this effort and increase traceability matrix assurance. The method is new, that it utilizes both requirements similarity and dissimilarity.

Prioritizing requirements focus on stakeholders' feedback brings a noteworthy cost because of time elapsed in a large number of human interactions. A Semi-Automated Framework has been presented in paper [17]. It predicts appropriate stakeholders' ratings to reduce human

interactions. Future work of this research is to cluster requirements.

A prioritization method called Case-Based Ranking (CB Rank), presented in [18] which integrate project's stakeholder's desires with requirements ordering approximations calculated through AI techniques.

3 Methodology

In this segment, we present our proposed framework for ranking of stakeholders' requirements using the case base reasoning CBR if available in previous use or assign them new ranking according to their requirement and their assign ranking for software product line.

A Proposed Approach

Our framework gives an inclusive model for the requirement ranking of software product line using the CBR. Our proposed framework consists of the following layers which are:

1) Description Layer:

In this first layer we performed profiling of the system, it include two main steps first is requirement elicitation which is the process of extracting the information from stakeholders. We also get the initial ranking from the stakeholders against each requirement. As the outcome of this layer we get the requirement document along with requirement initial ranking.

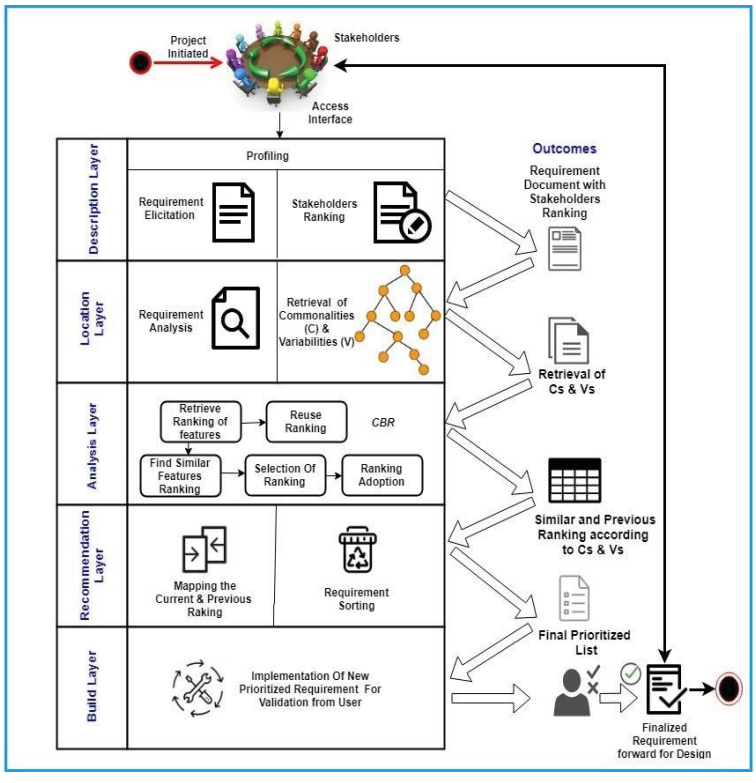

Figure 2: Proposed Framework

2) Location Layer:

In this second layer, we find the commonalities and variabilities of product line from the document using the J48 classification algorithm. It generates a binary tree. This approach is helpful in classification problem. Using this technique, a tree is constructed to model the classification process. [19]

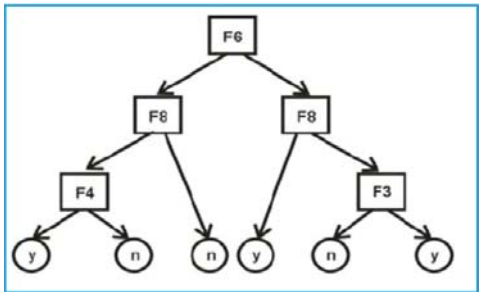

Figure 3: J48 Working

3) Analysis Layer:

In analysis Layer we apply the CBR, it is an AI technique that work on expert knowledge and previous experiences with less time, effort and cost. It works on the concept of reuse the solution of previous cases like new case and stores the cases in the database for later use. [8]

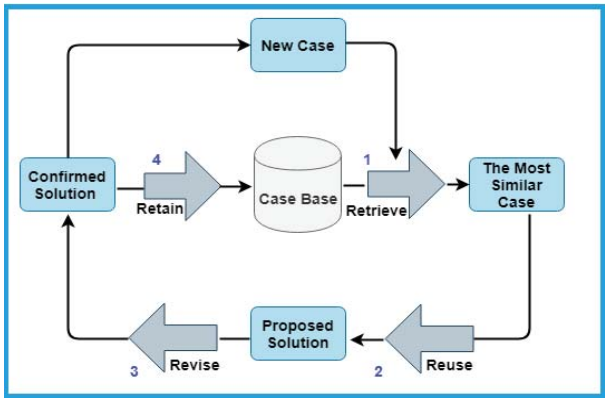

Figure 4: Case-Based Ranking (CBR)

4) Recommendation Layer:

In this layer we map the ranking of the stakeholder's requirements and the ranking find out from the CBR if we found the better result against the applied query we adopt the best available ranking and then apply the sorting on that list and we get the sorted prioritized list as the outcome of this layer.

5) Build Layer:

At this last layer we send the prioritized list to the stakeholders if they accept it and approved it then we forward it to the prototype and design of the product

4 Results and Discussion

For practical implementation of our proposed work in real world context we developed an intelligent requirements prioritization recommendation (IRPR) tool using steps of proposed work. Therefore, to evaluate IRPR we performed an empirical study. For this matter of fact, we technologies development organization which work on different projects both nationally and globally, but company not allow us to disclose any information about company. From large bulk of projects pool we selected two projects (P) i.e. LMS system (P-A) and card swipe machine (P-B).

For the elicitation and prioritization of projects user requirements before implementations uses different applications. Hence, the traditional tools/techniques (TT) they adopted increase the challenges that mention in literature review section i.e. more human interaction, ambiguous requirements etc. To resolve these issues company agreed to use IRPR tool to attain higher user satisfaction and product quality. Consequently, for IRPR implementation we conducted experiment and divided participants of company employees i.e. 21 in total for experiment into two groups' i.e. experimental treatment (ET) and non-experimental treatment (NET). The participants of ET used to develop both P-A and P-B using IRPR whereas NET participants adopted TT for implementation both projects. While participants consist of project manager (PM), requirement engineers (RE), requirement analysis (RA), developers (D) and stakeholders (S). The overall working of IRPR prototype show in figure 5-9 to illustrate the interfaces of IRPR.

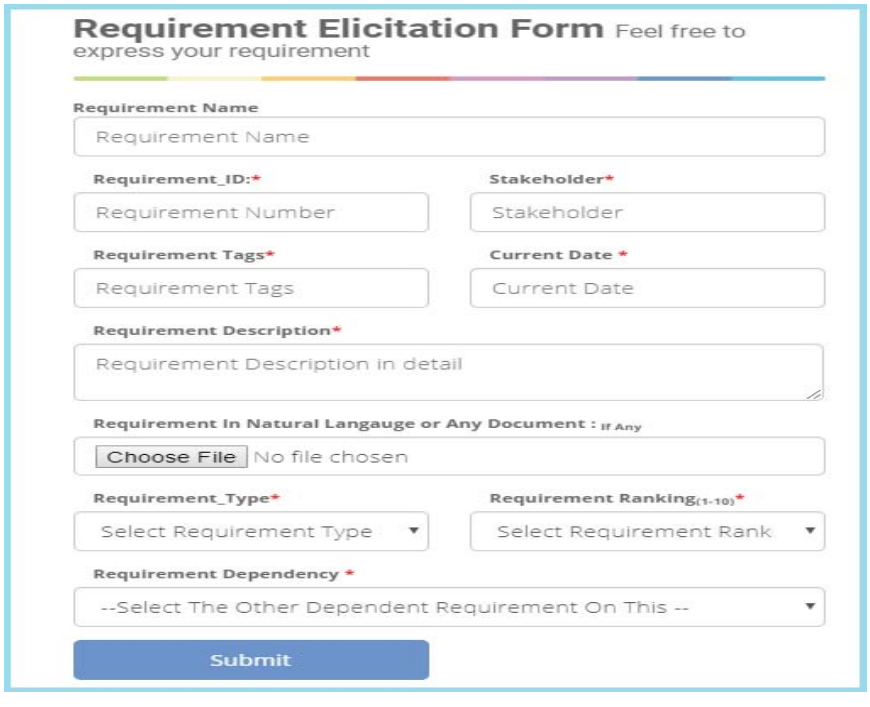

Figure 5: Requirement Elicitation & Stakeholders Ranking

When the project initiated the working of IRPR started; therefore, S connected to PM and the form open for elicitation of requirements as show in figure 5 screen shot of form interface. In the form user enter their requirements with ranking and profile of all users maintained in the database for future use. After the evaluation of profiling RE and RA analysis the requirements because these projects are SPL based. Therefore, then using j48 algorithm retrieve commonalities and variabilities in the form of decision tree for the CBR mapping to extract previous ranking as depicted in figure 6.

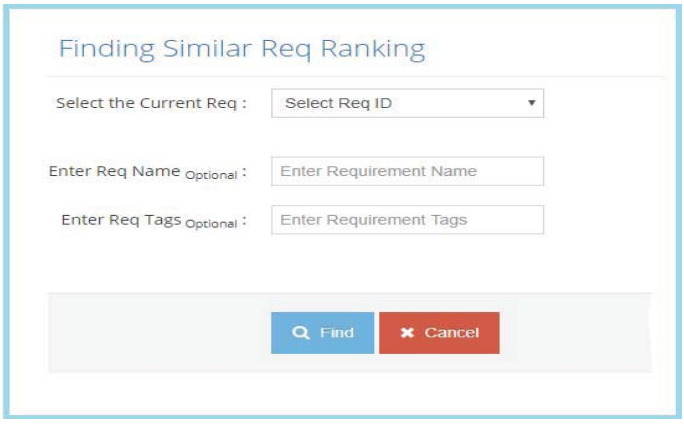

Figure 6: Finding Similar Ranking Query (CBR)

In CBR when we apply a query for finding similar ranking we will get the list (shown in figure 7) of the previous cases which are similar to the current case with the ranking. We will accept and adopt the case which is high rank amongst them.

Available Similar Rankings

Requirement ID	Requirement Name	Stakeholder Name	Type	Discription	Ranking (1-10)	Tags	Previous Used Project	Recommended	Selection Or Rejection
Requirement 4	Login	Haider	Functional Requirement	The login screen allows registered users to login to the site to access all of the features that their account gives them access to	8	LOGIN , FORM, LOGIN SCREEN	Student Portal	YES	✔ ✖
Requirement 7	Login Form	David	Functional Requirement	logon is the procedure used to get access to an operating system or application, usually in a remote computer. Almost always a logon requires that the user have (1) a user ID and (2) a password.	6	LOGIN , FORM , EMAIL, PASSWOR SECURITY CHECK	APP News Paper	NO	✔ ✖

Figure 7: Selection and adoption of similar ranking

When we adopt some cases from CBR and mapped the current and the previous ranking we will get the prioritized list of the requirement with the ranking from 1-10 in a unsorted order shown in Figure 8 below.

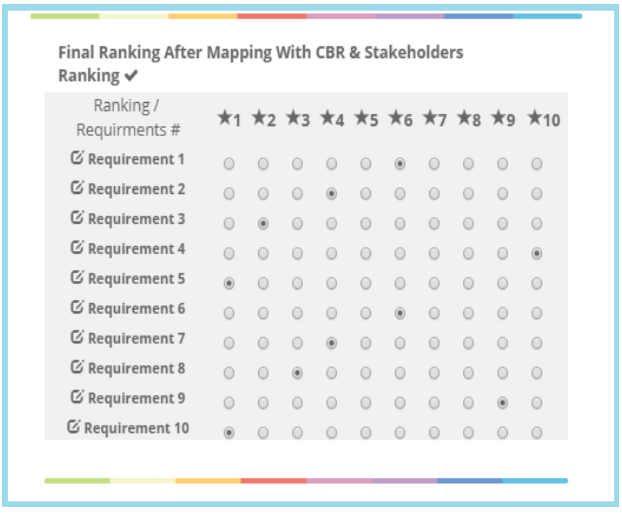

Figure 8: Final Prioritized Requirements after mapping stakeholders and CBR Ranking

Apply any sorting technique with respect to their ranking we will get the final sorted prioritized ranking of the requirements (shown in figure 9) which will decide the education order of the requirement in development phase

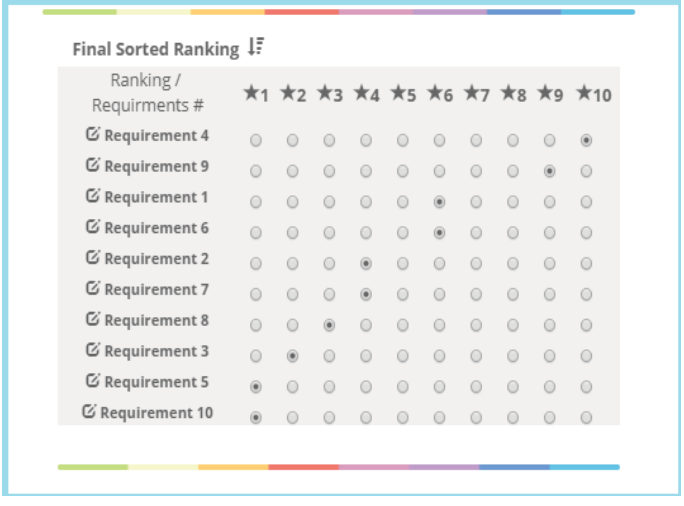

Figure 9: Sorted Prioritized Requirements

For the assessment of experiment performance, we conducted questioner based review from both ET and NET members. The review based on parametric analysis which based on existing literate i.e. user friendly (UF), usability (U), learnability (L), efficient (E), high effectiveness (HE), less human interaction (LHI), proficient knowledge management (PKM), efficient knowledge identification and retrieval (EKIR), requirements priority accuracy (RPA), enhance elicitation and prioritization (EEP), high productivity (HP) and higher user satisfaction (HUS).

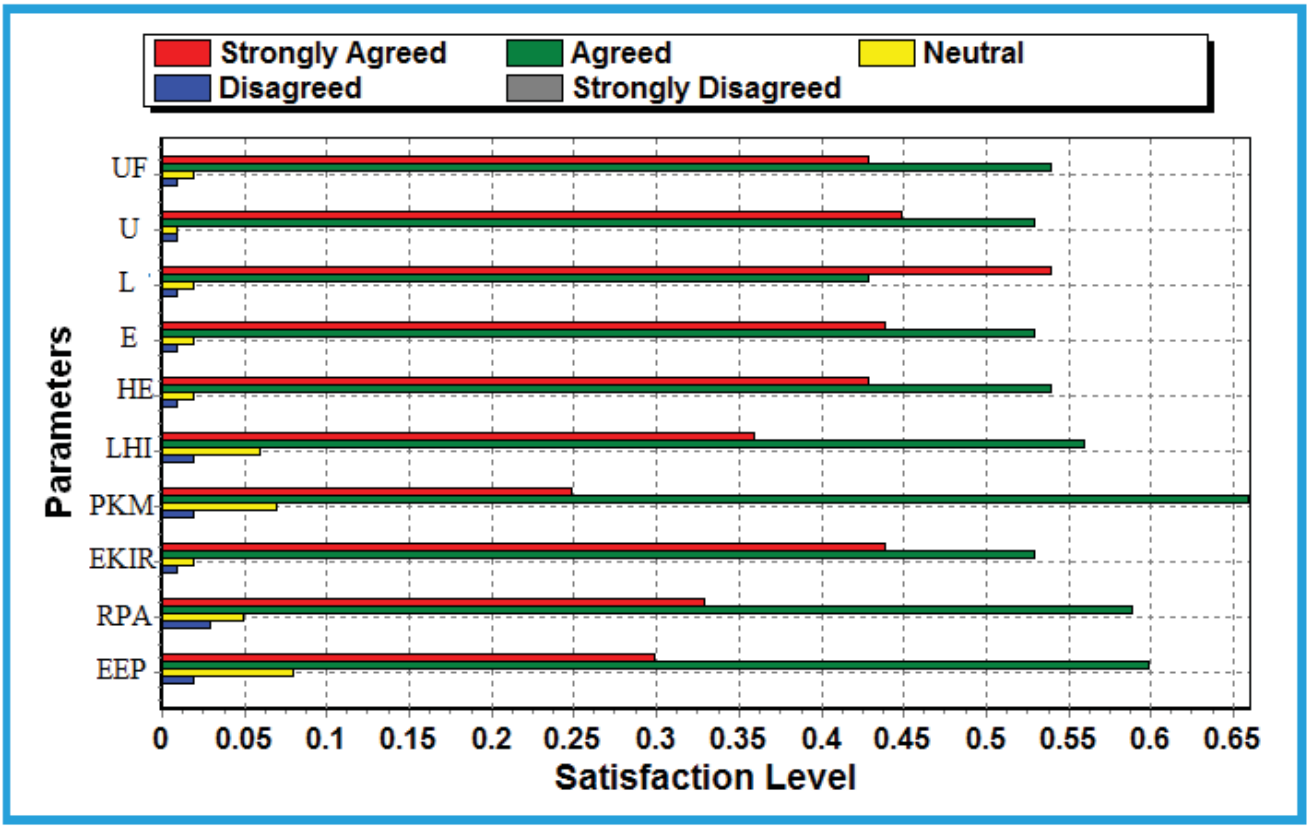

Figure 10: Review Analysis

The overall analysis result on the parameters implementing both tools i.e. IRPR and TT as demonstrate in figure 10. The figure 10 shows the satisfaction ratio of users on left side vertically with more than 50 percent satisfaction ratio and parameters review on the Y-axis.

Table 1: Comparative Analysis

Techniques	Participators				
	PM	TL	RA	Ds	QE
Experimental Treatment of P-A (ET P-A)	0.7	0.69	0.63	0.86	0.76
Non- Experimental Treatment of P-A (NET P-A)	0.32	0.36	0.45	0.27	0.38
Experimental Treatment of P-B (ET P-B)	0.80	0.70	0.60	0.76	0.86
Non-Experimental Treatment of P-A (NET P-B)	0.35	0.40	0.45	0.37	0.28

The users of project A in which experimental treatment (ET) is applied, are more satisfied and gained better results than the participants of non-experimental treatment (NET). Whereas; same is the case with participants of project B. The members of experimental treatment (ET) of B give better quality and competence.

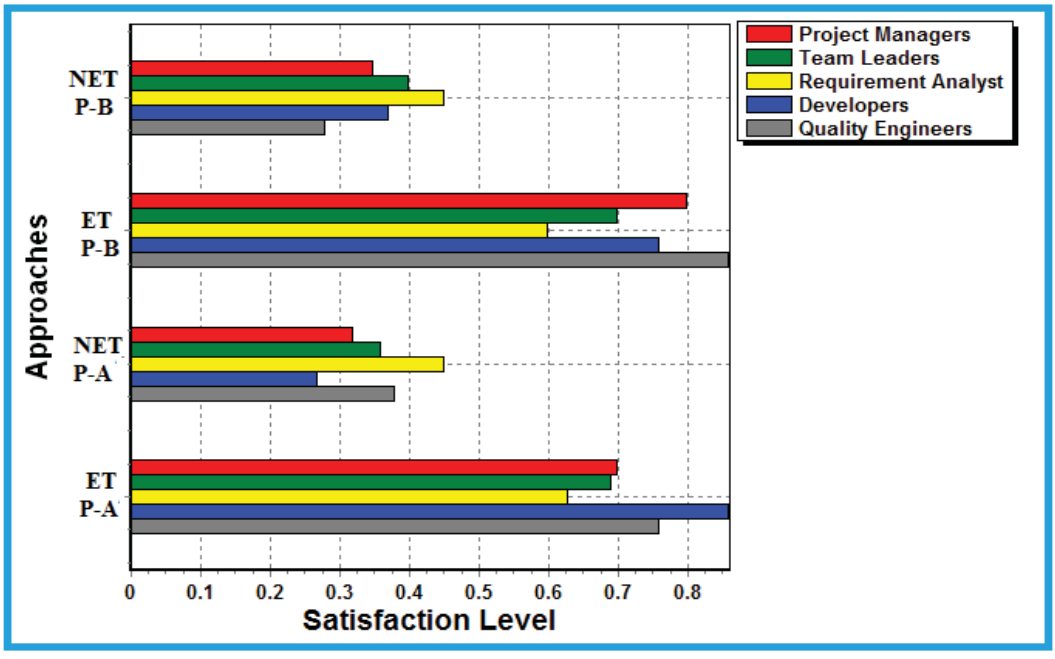

Figure 11: Comparative analysis results

We also illustrate the comparative analysis of both projects with experimental treatment (ET) and non-experimental treatment (NET) in Figure 11. Figure 11 represent the participants' satisfaction level. The y-axis labels each project development approaches while x-axis explains the satisfaction level of each user. The results present that our proposed framework's performance and satisfaction for quality and customer needs.

5 CONCLUSION

In this research, we proposed a framework for requirement ranking for software product line using CBR. The proposed framework uses J48 to find out the Cs and Vs from requirement document and then apply CBR on these requirements to find their final ranking. We have performed a tool based evaluation to evaluate our framework. Our results show noteworthy improvement in terms of satisfaction level for various parameters as compared to traditional approaches of ranking in SPL. The proposed research provides direction to industry and researchers to manage software prioritization.

References

[1] Bhushan, Megha, Shivani Goel, and Karamjit Kaur. "Analyzing inconsistencies in software product lines using an ontological rule-based approach." Journal of Systems and Software 137 (2018): 605-617.

[2] Khalique, F., Butt, W.H. and Khan, S.A., 2017, December. Creating domain non-functional requirements software product line engineering using model transformations. In 2017 International Conference on Frontiers of Information Technology (FIT) (pp. 41-45). IEEE.

[3] Xue, Y., Xing, Z., Jarzabek, S.: Feature location in a collection of product variants. In: IEEE 19th RE Conference, pp. 145–154 (2012)

[4] Ra'Fat, A., Seriai, A., Huchard, M., Urtado, C., Vauttier, S. and Salman, H.E., 2013, June. Feature location in a collection of software product variants using formal concept analysis. In International Conference on Software Reuse (pp. 302-307). Springer, Berlin, Heidelberg.

[5] D. Zowghi and C. Coulin, Requirements Elicitation: A Survey of Techniques, Approaches, and Tools, pp. 19–46. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[6] Arias, M., Buccella, A. and Cechich, A., 2018. A Framework for Managing Requirements of Software Product Lines. Electronic Notes in Theoretical Computer Science, 339, pp.5-20.

[7] Hasan, M. S., Mahmood, A. Al, Alam, J., & Hasan, S. N. (2010). An Evaluation of Software Requirement Prioritization Techniques. International Journal of Computer Scienceand Information Security, 8(9), 83–94.

[8] Ali, S., Iqbal, N. and Hafeez, Y., 2018. Towards Requirement Change Management for Global Software Development using Case Base Reasoning. Mehran University Research Journal of Engineering and Technology, 37(3), pp.639-652.

[9] Kaur, G. and Chhabra, A., 2014. Improved J48 classification algorithm for the prediction of diabetes. International Journal of Computer Applications, 98(22).

[10] White, J., Galindo, J.A., Saxena, T., Dougherty, B., Benavides, D. and Schmidt, D.C., 2014. Evolving feature model configurations in software product lines. Journal of Systems and Software, 87, pp.119-136.

[11] Khalique, F., Butt, W.H. and Khan, S.A., 2017, December. Creating domain non-functional requirements software product line engineering using model transformations. In 2017 International Conference on Frontiers of Information Technology (FIT) (pp. 41-45). IEEE.

[12] Barney, S., Aurum, A. and Wohlin, C., 2008. A product management challenge: Creating software product value through requirements selection. Journal of Systems Architecture, 54(6), pp.576-593.

[13] Ye, H. and Liu, H., 2005. Approach to modelling feature variability and dependencies in software product lines. IEE Proceedings-Software, 152(3), pp.101-109.

[14] Inoki, M., Kitagawa, T. and Honiden, S., 2014, August. Application of requirements prioritization decision rules in software product line evolution. In Requirements Prioritization and Communication (RePriCo), 2014 IEEE 5th International Workshop on (pp. 1-10). IEEE.

[15] Kaindl, H. and Mannion, M., 2018, August. Software Reuse and Reusability based on Requirements: Product Lines, Cases and Feature-Similarity Models. In 2018 IEEE 26th International Requirements Engineering Conference (RE) (pp. 510-511). IEEE.

[16] Port, D., Nikora, A., Hayes, J. H., & Huang, L. (2011, January). Text mining support for software requirements: Traceability assurance. In System Sciences (HICSS), 2011 44th

Hawaii International Conference on (pp. 1-11). IEEE.

[17] Asif, S. A., Masud, Z., Easmin, R., & Ul, A. (n.d.). arXiv : 1801 . 00354v1 [cs . SE] 31 Dec 2017 SAFFRON : A Semi-Automated Framework for Software Requirements Prioritization, 1–21

[18] Perini, A., Susi, A., & Avesani, P. (2013). A Machine Learning Approach to Software Requirements Prioritization, 39(4), 445–461.

[19] Patil, T.R. and Sherekar, S.S., 2013. Performance analysis of Naive Bayes and J48 classification algorithm for data classification. International journal of computer science and applications, 6(2), pp.256-