Reuse Environment for Software Components

Sunidhi Vashisth, Iqbaldeep Kaur

Chandigarh Engineering College, Department of Computer Science, Landran (Mohali)

***Abstract*-**Component Based Development is the branch of software engineering in which a new system is designed by using already existing software components. The software component may be a code file, a plug-in or any other entity which are independent in nature. Component search is a very common activity in software development practice. To implement program functionality, developers can reuse a software component, a large-scale searches component library. Many component search tools have been proposed over the years that help developers. Existing methods usually treat the Software component as a text document and use the information retrieval model that retrieves the relevant code snippet that matches the given query. A number of traditional approaches have been perform to recommend things over a query. The growth of the development industry is increasing very rapidly and designers usually require component and codec to run their software utility. The research problem of this thesis is to develop a software repository based on certain search parameters. In this research work, a software repository system is developed to store locations from which software components might be retrieved and implement on a computer system. As per requirement, the ranking list is created. The designed repository model comprises of four categories named Software, Architecture, Network and DBMS. An additional admin panel is created for adding the components according to the domains. The added components are stored into the mat repository file. In the searching phase, the client fills the search details and the matching takes place according to the designed rule set. Pre-processing is done in order to remove the stop words. The classification process is done using an ANN (Artificial neural network) technique. If the target component is matched then the ranking goes high otherwise goes low.

Keywords-Code search, ANN, search engine, recommendation system**.**

I. INTRODUCTION

Component based software engineering (CBSE) is the branch of software engineering in a new system is developed by using already existing software components. CBSE is the most commonly used technique as in the object oriented programming, the objected are very complex and hence not allow plug play as it is provided by CBSE technique. A CBSE is an assembly of software components that follows an engineering procedure which conforms to CBSE principles. Several languages are utilized to write these components, those can be executed in various operational platforms, and when on the web, they can be distributed across vast geographic distances [3]. Software reuse is the most effective way to get better software processes, quality of software and application consistency, and hence decrease improvement and maintenance. Component-based software engineering is a method of software development by reusing the present components. A set of pre-defined components are made available for the specific architectural style. The application is then built by assembling the components in that particular architectural style. The component based systems are easy to build; therefore less time is used for development [2]. A software component is then a software element which is independent and ready to use, it conforms the component model and deployed and use without any modification in structure or code files. There are some examples of software components like .html file, .exe code file, plugins and API etc.Software components must have these three properties: Component adaptation**:** component adaptation is the process in which some of the mismatching or errors are fixed in reusable components . These components must be adapted to meet the needs of the architecture and replaced by other, more suitable components. Component composition: The component composition means that there need to be the architecture structure or design checks in reusable components. Component update: On the time of implementation the update is complicated by the third party (the organization).Process of CBSE:CBSE is partially related to object oriented technology. The process works in parallel with Domain Engineering and Component Based Development. Domain Engineering explores an application domain with the intent of finding functional, behavioral and data components that are candidates for reuse. These components are placed in Reuse libraries. The intent of domain Engineering is to identify, construct, catalog and disseminate a set of software components that have applicability to existing and future software in a particular application domain. Component based development collects the requirements from the customers, selects an appropriate architectural **d**esign to meet the requirement of the system to be built and then select potential components for reuse, qualifies the components to be sure that they properly fit the architecture of the system, Adapts components, if modifications must be made to properly integrate them and, Integrates the components to form subsystems and the application as a whole [12].

Retrieval: The term retrieval means the fetching of software components from the library or repository in which components are stored. Software repository is a collection of data files or software components in organized manner**.** In this research work, we are using a software repository system for storing position from which the software packages can be extracts and installed on a computer system. In other words, software repository can be defined as the integration of software components. In CBSE, the components are known as part of code, software module etc. During the designing of software components, the software designed faces various problems such as complexity, project delivery time, and cost etc. By using repository system in CBSE the problem of delivery time has been overcome as the developer select the components from the repository. Software component retrieval is a term which is used to retrieve the software component from the software components. There are many techniques already developed for retrieval like: Keyword based retrieval, semantic based retrieval. Text based search method: In this method, the whole text or paragraph is searched and if the component found valid according the given search, they are displayed .Keyword based search method: In this method user just have to give the input in keywords, once the keywords are filled in the syaytem checks for all the keywords and related keywords are found. The system shall display the related components. Signature based search method Signature base search as name implies that the search is on the basis of signatures of the component. This type of search helps to provide with components with approximately the exact match from the requirement [6].

In [8] a machine learning approach is proposed to categorize the document using KNN as a classification algorithm. From the existing survey it has been concluded that KNN perform well in terms of accuracy while dealing with the text categorization in medicine, email filtering etc. The proposed technique finds application like while an user enter keywords the similar words are categorized using KNN and displays the most important document for the user. Authors have also compared the accuracy obtained by three classification techniques such as Naïve Bayes, Term graph and KNN techniques and concluded that KNN perform well with an accuracy of 99.27 for the “places” database. In reference [13] Proposed k-mean clustering technique that has been used to define various clusters repeatedly and allocates required clusters to an un-clustered region. By using clustering technique the accuracy of detecting cancer has been improved along with the decrease in the clustering time. In this paper, SVM (support vector machine) has been used as a classification approach to distinguish between normal and abnormal cells. In[11] indicates the importance of software reusing repositories being developed. In this article, the authors have discussed various search techniques to efficiently retrieve components from reusable repositories. This paper illustrates the working process of the technology through its automated implementation, highlighting the concept of keyword-based search technology in a clear way. The implementation is done through a new development tool called ARE (Automated Repository Exploration) ver. 1.0.0. This technique helps to retrieve components related to user-submitted keywords as queries. Therefore, the sort results are presented to the user in the order of relevance of the searched components.In [7] have proposed a study on how to retrieve reusable components more easily. The implemented system also effectively stores various reusable components in manageable and understandable categories from which users can efficiently and efficiently extract the required components. In [10] has identified multi-voxel representations of different mode components that converge in the angle back during retrieval. It is essential that convergence occurs only after 24 hours of integration and during transfer testing, where the model material is applied to novel but relevant experiments. Therefore, the angle back seems to reassemble the integrated pattern components into a memory representation. In [9]has proposed the first point descriptor that captures the "topology" of the shape from a single point in a multi-scale and provable stable manner. The author has also demonstrated how to map a large class of topological signatures (including our topology signatures) to vectors, opening the door to much classic analysis and learning methods. The authors have also illustrated the performance of this approach in supervising shape markers and shape matching problems. The author has also considered their signature that provides additional information for existing signatures and allows for less performance using less training data in both applications . In reference [5] has proposed an embedded software online repository to facilitate component management and retrieval. ORES uses an ontology-based approach to facilitate repository browsing and efficient searching. To facilitate the browsing of ORES, the authors has analyzed the typical ontology of software components and has developed a merge and echo technique that transforms the ontology into a hierarchical structure suitable for browsing without losing any key semantic information contained in the ontology. An algorithm has been developed for grouping search results based on ontology. Therefore, search result groups to avoid showing a large number of search results or the risk of having to trim the results and reduce the recall factor can be validated. Another important aspect of embedded software is a set of non-functional requirements and attributes. In [1]presented an entropy based fuzzy k-mean clustering algorithm in order to retrieve components in repository system. The clustering helps to combine components into clusters, on the first choice of users’ search. Thus the users get the accurate results by comparing the components to the nearest component on the basis of preference. In this research work, 1000 random components have been generated along with single random search preferences. In order to control the fuzziness of the cluster the weighting exponents of (1.10 and 1.20) have been used. At last the search similarity has been determined which is approximately equal to 90%.

From the above literature survey it has been cleared that there are many techniques already used for the retrieval of software components. The proposed technique includes Neural Network for the retrieval propose.

- Neural Network is applied for searching the software components and then rank them.
- The software components then retrieved are relevant according to query.

The remaining paper has been structures as follows:The rest of the paper has been organized as follows: section 2 gives detailed information about the planned approach. Section 3 contains the results and performance of proposed technique. In section 4 the analysis and drawbacks from the study and section 5 gives conclusion and future scope of work.

II. MATERIEAL

In this section, the methodology along with the techniques and their algorithms has been discussed in detail. In the proposed system a software component repository is designed for storing the reusable components. Use of the repository as per user requirement including the ranking of the searched results. The creation of the repository includes an Admin panel and four categories are summarized: Software, Architecture, Network, DBMS (Database Management System).An admin addition panel is created in which the administrator decides that which data has to be added. The administrator has to provide the description of each component which is added to the repository. Algorithm for Proposed System:

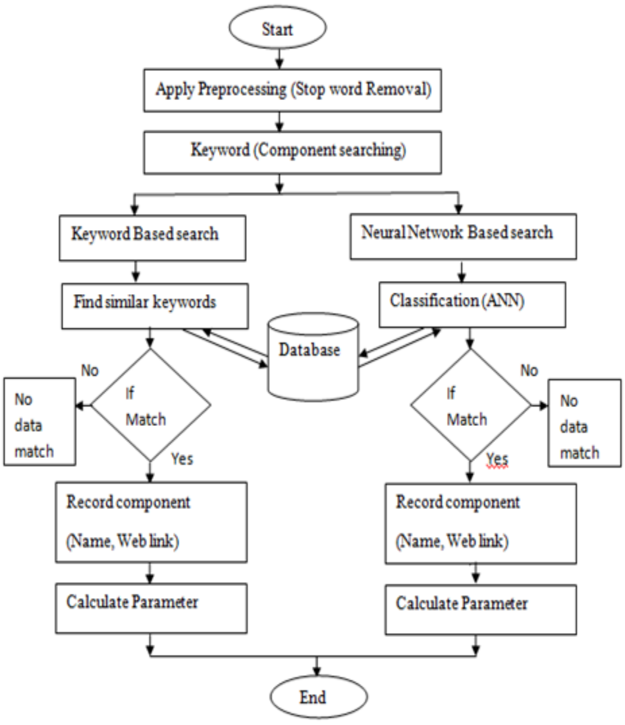


Figure- Flow graph of proposed techniques

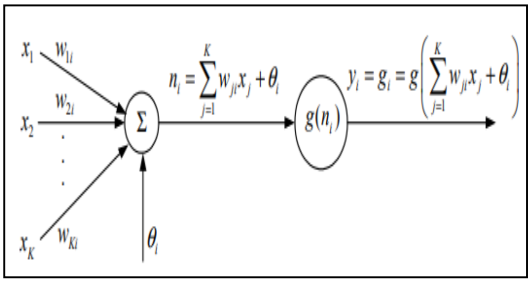
a. INTIAL WORK

In this research work the input query which is given in description format is processed to remove the anomalies and make it ready for further operation. For preprocessing the Stop word Removal technique is used. It is the first step for this research work. It is basically used to remove the stop words from the query. The resultant keyword is then searched by using Keyword based and Neural Network search.

In keyword based search the similar keywords are searched, if the query data is matched with the components stored in the database then the components are displayed with names and their web links. In neural network based approach, classification technique is applied on the searching for component (query). If data is matched then ranking is applied using Levenberge Maquard algorithm in order to obtain the relevant component from the repository. Determine parameters such as Precision, Recall, F-measure and Execution time.

a. Artificial Neural Network

The architecture of ANN used in the proposed work consists of four panels named as neural network, algorithm, progress and plots. The ANN used in the proposed work includes three layer named as input, hidden layer and output layer. The training algorithm used in the proposed work is “Levenberg-Marquardt”, the performance of which depends upon MSE (mean square error) value. In progress panel, the completion of ANN depends upon epoch (iteration), time, performance, gradient, and Mutation and validation checks. After the completion of any of the parameters Ann training is completed. The neural network used in the proposed work consists of a single input layer, a hidden layer and a output layer. The structure of the network is shown below:


x_1
W_{1i}
W_{2i}
x_2
x_K
W_{Ki}
$n_i = \sum_{j=1}^{K} w_{ji}x_j + \theta_i$
Σ
θ_i
$g(n_i)$
$y_i = g_i = g\left(\sum_{j=1}^{K} w_{ji}x_j + \theta_i\right)$

The above figure consists of x_k number of inputs with weight values varies from w_{1i} tow_{ki}. These weight values are multiplied with the number of inputs and then added in the summation block with a constant value named as biased valueθ_i. The resultant output is forwarded to the activation function g_i which is a hyperbolic tangent function and mathematically can be defined as:

$$\tanh(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

No data found

Then the output obtaine twork becomes:

$$y_i = g_i = g\left(\sum_{j=1}^{k} w_{ji}x_j + \theta_i\right)$$

Here, the activation function is used to choose the values of parameters such as weight and bias along with the training algorithm metrics namely error rate, maximum epoch etc are decided. During simulation process, neural network is called for classification and the output is compared with the input data. When the output data is matched with the input data accurate results are obtained. In plot panel, the performance by plotting the MSE with respect to number of epoch is plotted, when we click on training state and regression graph the plots are appear on the screen as shown below. MSE value is calculated by using the given formula written below:

$$Error_{w=\frac{1}{R}\sum_{n}\sum_{k=1}^{c}|y_k(x_n;W)-t_{kn}|r.|t_{kn}|^{r}}$$

Where,

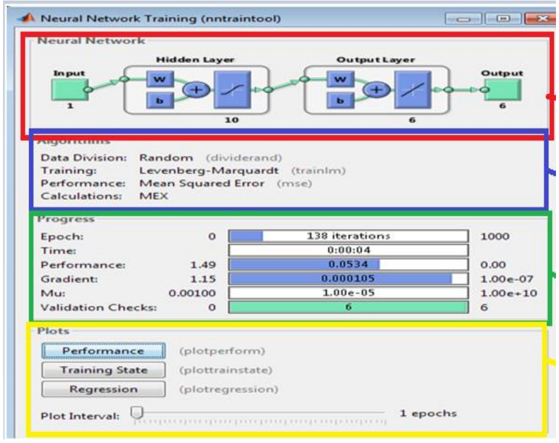
$Error_w$- weighted error of neural network

R- Murkowski-R error

n-Number of neurons

t-Tension force

ANN uses regression model as well as the back propagation model for the same. The architecture of ANN used in the proposed work is shown in figure:



Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MEX

Epoch:	0	138 iterations	1000
Time:		0:00:04	
Performance:	1.49	0.0534	0.00
Gradient:	1.15	0.000105	1.00e-07
Mu:	0.00100	1.00e-05	1.00e+10
Validation Checks:	0	6	6

Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)
Plot Interval: 1 epochs

Fig: ANN Architecture.

III. SIMULATED RESULTS

The flow of algorithms for the proposed work is given below:

Algorithm 1: Keyword based Retrieval Algorithm

Step-1: Load the components in the Database with description and name.
Step-2: Search with any keyword with length minimum characters.
Step-3: Matching components will display in a Long list based on the searched keyword.
Step-4: Calculate parameters F-measure, Precision, Recall and Execution time.
Step-5: Repeat steps 2, 3 and 4.
Step-6: Stop.

Algorithm 2: Neural Network based Retrieval Algorithm

Step-1: Load database of components with description and names of components**.**
Step-2: Load trained ANN structure as trained ANN with their position.
Step-3: Load the types of operating systems and domain of the system.
Step-4: Search with description and free text in search panel**.**
Step-5: Select its operating system and domain of component for advanced search.
Step-6: Description is tested with the trained data for ranking.
Step-7: Resulted components are again ranked through neural network.
Step-8: A precise list of components will display.
Step-9: Calculate parameters F-measure, Precision Recall and Execution time.
Step-10: Repeat steps 4 to 9 for searching with ANN.

Algorithm 4: K-mean

Step-1: Crate Dataset and estimated centroid.
Step-2: Calculate the size of dataset.
Step-3: If dataset is in group 1, then data is in cluster 1.
Step-4: If dataset is in group 2, then data is in cluster 2.
Step-5: Else data is in cluster n, which is grouped.
Step-6: Repeat steps 4 and 5.
Step-7: Stop.

The work has been carried out in MATLAB simulator tool. The performance of the proposed work is examined by determining the various parameters such as execution time, precision, recall and F-measure. The values along with the graphical representation are illustrated below.

Table 1: Computed parameters with keyword based searching

Description for components	**Execution time (Sec)**	**Precision**	**Recall**	**F-measure**
The architecture guide map	7.1	0.500	0.500	0.507
The code file for uml	8.3	0.500	0.497	0.498
Wireless network dataset	5.8	0.488	0.507	0.497

Java file for cloud computing	8.42	0.500	0.499	0.5
The routing protocol for wired network	11.08	0.505	0.497	0.501

Table 2 Computed parameters with Neural network based searching

Description for components	Execution time (Sec)	Precision	Recall	F-measure
The architecture guide map	42.6	0.948	0.978	0.969
UML CODE	46.5	0.952	0.978	0.970
Network dataset	40.8	0.949	0.979	0.970
Java file for cloud computing	44.7	0.952	0.979	0.971
The routing protocol	48.22	0.951	0.978	0.970

The comparison of keyword based searching technique along with neural network based searching algorithm is shown in the graphical form on the basis of execution time.

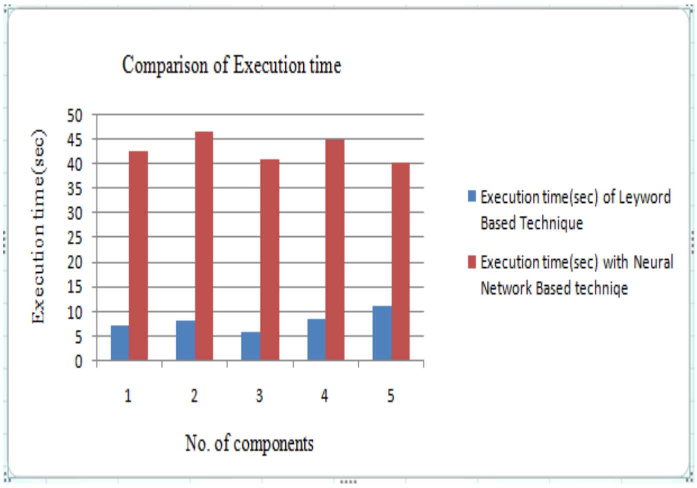


Figure 3 Comparison of Execution Time

From the Figure 3 it is clear that the execution time required by the keyword based technique is less as compared to the classification algorithm.

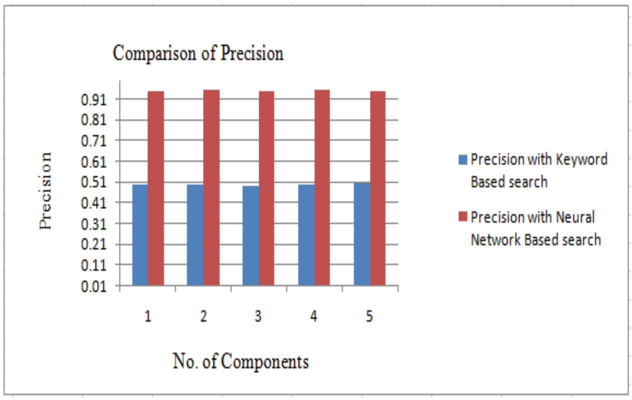


Figure 4Comparison of Precision

The value of precision obtained for the different components searching while using keyword based algorithm and neural network is shown in the form of graph. X-axis defined the number of components and y-axis represents the values of precision obtained for the proposed work. From the figure displayed in 6, it is concluded that the average values obtained for keyword based technique and neural network based technique are 0.4986 and 0.9504 respectively. Therefore, it is concluded that when neural network is used in the proposed work, the value of precision is increased by 90.61%.

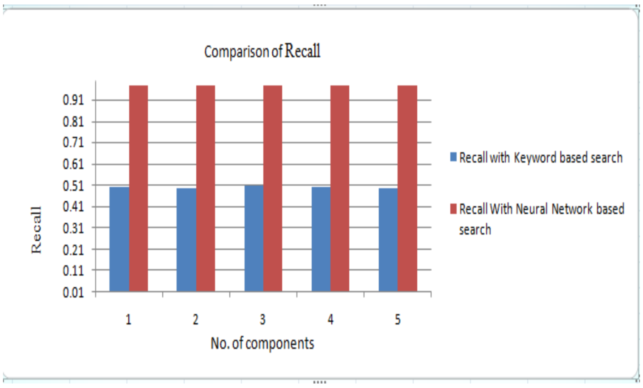


Figure 5 Comparison of Recall

The Figure 5 represents the values of recall obtained for keyword based approach as well as recall values obtained by using neural network based search approach. In the above graph, blue and yellow bar represents the values of recall obtained for keyword based and neural network based search technique respectively. The average values obtained for the keyword and neural network based approach are 0.5% and 0.97 % respectively. Therefore, it is concluded that when neural network is used in the proposed model the recall rate increased by 94%.

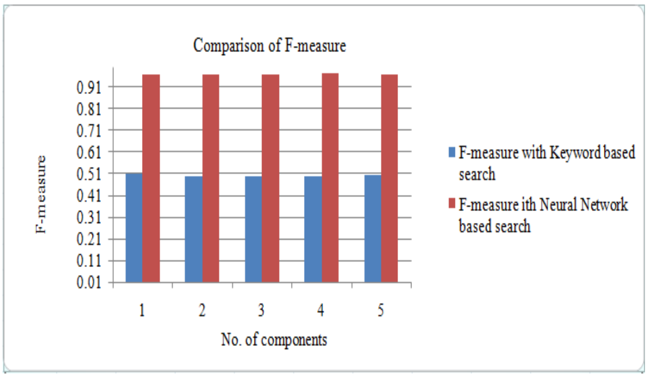


Figure 6 Comparison of F-measure

The figure 6 represents the comparison of F-measure obtained for keywords based and neural network based approach for the proposed work. From the above figure it is clear that the average value obtained for keyword and neural network are 0.50 % and 0.97 % respectively. So, there is an increase in the F-measure by 94%.

IV. STATE OF THE ART

In reference [4] proposed a scheme used for determine and extracting useful software components from large scale reuse libraries. In this paper, the comparison between different semantic based approaches such as signature based, keyword based, semantic based and blended based approach have been performed on the basis of precision parameter. From the experiment it has been observed that semantic based approach perform best among other techniques. The average value of precision obtained for semantic approach for 12 queries is 0.59.

Table 3: Comparison of proposed work with existing work

Proposed work (ANN)	Alnusair et al. [4]	Rasolofo et al [14]
0.95	0.49	0.21

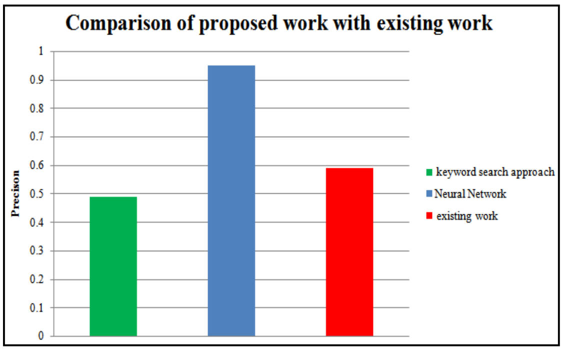


Figure 9 Comparison of proposed work with Existing work

The figure 9 represents the comparison of proposed technique (Neural network) with existing technique. Here, green, blue and red bar graph represents the average values of precision obtained for keyword based search, neural network based search performed by different authors Alnusair et al. [4] and Rasolofo et al [14]respectively. From the above graph it is clear that, the proposed technique performs better in terms of precision and the value of precision has been increased by 61.02 % from the existing work.

V. CRUX OF THE WORK

This work has proposed a research method that makes it easier for users to retrieve reusable components by proposing and implementing an interface to do so. The implemented system also effectively stores various reusable components in a manageable and understandable category from which users can efficiently and efficiently extract the required components. In this research work, a component repository system has been designed in a well organized manner using MATLAB simulator tool. In this research work, different software components is classified on the basis of the content of appropriate software articles that has been accomplished by designing a categorizing model that is capable to classify the software. The different software categories have been studied repeatedly to extract valuable information and estimate their movement. Textual data such as software editorial has better information, therefore, utilizing textual information can enhance the input to the data of numeric time series and the better predictions are imagined from this type of input than only from statistical data. It has been concluded that the improved artificial neural network performs better. To know the efficiency of the proposed system, the parameters named as precision, recall, true positive and false positive have been measured. The average value of precision, recall, and F-measure obtained for the proposed work are 90.61%, 94% and 97% respectively. In future, any optimization algorithm such as genetic algorithm, Cuckoo search can be used along with ANN and SVM and comparison of both the hybridization techniques can be shown to know the efficiency of the algorithms.

VI. REFERENCES

[1]. Stylianou, Constantinos, and Andreas S. Andreou. "A hybrid software component clustering and retrieval scheme using an entropy-based fuzzy k-modes algorithm." In Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on, vol. 1, pp. 202-209. IEEE, 2007.

[2]. Qureshi, M. RizwanJameel, and S. A. Hussain. "A reusable software component-based development process model." *Advances in engineering software* 39, no. 2 (2008): 88-94.

[3]. ArvinderKaur, and Kulvinder Singh Mann. "Component based software engineering." *International Journal of Computer Applications* 2, no. 1 (2010): 105-108.

[4]. Alnusair, Awny, and Tian Zhao. "Retrieving reusable software components using enhanced representation of domain knowledge." In *Recent Trends in Information Reuse and Integration*, pp. 363-379. Springer, Vienna, 2012.

[5]. Singh, Shekhar. "An experiment in software component retrieval based on metadata and ontology repository." *International Journal of Computer Applications* 61, no. 14 (2013).

[6]. AmandeepBakshi, SeemaBawa, “Development of a Software Repository for the Precise Search and Exact Retrieval of the Components”, International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 8, August 2013.

[7]. Bakshi, Amandeep. "Development of a Software Repository for the Precise Search and Extract Retrieval of the Component." PhD diss., 2013.

[8]. Bijalwan, Vishwanath, Vinay Kumar, PinkiKumari, and Jordan Pascual. "KNN based machine learning approach for text and document mining." *International Journal of Database Theory and Application* 7, no. 1 (2014): 61-70.

[9]. Carriere, Mathieu, Steve Y. Oudot, and MaksOvsjanikov. "Stable topological signatures for points on 3d shapes." In *Computer Graphics Forum*, vol. 34, no. 5, pp. 1-12. 2015.

[10]. Wagner, Isabella C., Mariët van Buuren, Marijn CW Kroes, Tjerk P. Gutteling, Marieke van der Linden, Richard G. Morris, and GuillénFernández. "Schematic memory components converge within angular gyrus during retrieval." *Elife* 4 (2015): e09668.

[11]. Ram Chatterjee, and HimaniRathi. "To ameliorate component searching by automating keyword search technique." In *Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on*, pp. 560-565. IEEE, 2015.

[12]. Roger S. Pressman, Software engineering, “A practitioner’s approach” sixth edition 2016.

[13]. Bansal, A., Sharma, M., &Goel, S. (2017). Improved k-mean clustering algorithm for prediction analysis using classification technique in data mining. *International Journal of Computer Applications*, *157*(6), 0975-8887.

[14]. Rasolofo, Y., & Savoy, J. (2003, April). Term proximity scoring for keyword-based retrieval systems. In European Conference on Information Retrieval (pp. 207-218). Springer, Berlin, Heidelberg.