I.J. Modern Education and Computer Science, **2025, 3, 92-105**
Published Online on June 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2025.03.07

Performance Evaluation of Various Machine Learning Algorithms for User Story Clustering

Bhawnesh Kumar*
Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun, India
Email: bhawneshmca@gmail.com
ORCID iD: https://orcid.org/0000-0002-2464-3401
*Corresponding Author

Umesh Kumar Tiwari
Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun, India
Email: umeshtiwari22@gmail.com

Dinesh C. Dobhal
Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun, India
Email: dineshdobhal@gmail.com

Received: 23 July, 2024; Revised: 03 September, 2024; Accepted: 20 October, 2024; Published: 08 June, 2025

Abstract: In agile development, user stories are the primary method for defining requirements. These days, managing user stories effectively is difficult because software projects typically contain a large number of them. A project can involve a large amount of user stories, which should be clustered into different groups based on their functionality's similarity for systematic requirements analysis, effective mapping to developed features, and efficient maintenance. Unfortunately, the majority of user story clustering methods now in use require a great deal of manual work, which is error-prone and time-consuming. In this research, we suggest an automated framework that uses a family of machine learning algorithms to classify user stories. First, preprocessing the data is done in order to examine user stories and extract keywords from them. After that, features are taken out, which allow user stories to be automatically grouped into distinct categories. We use four feature extraction algorithms and six clustering algorithms. According to our experimental results, K-means and BIRCH clustering outperform other clustering methods, whereas cosine similarity and distance are the best feature extraction for user stories categorization to form the more balanced clusters as they both have the standard deviation is 3.08. In case of user stories cohesion, the silhouette coefficient value is 0.225 for spectral with (cosine similarity and cosine distance feature extraction) is best outcome than other clustering algorithms. The usefulness and applicability of the suggested framework are demonstrated by this study. Additionally, it offers some useful recommendations for enhancing the effectiveness of user stories clustering, for example through parameter adjustments for enhanced feature extraction and clustering.

Index Terms: User Story, Agile Development, Clustering, Standard Deviation, Silhouette Coefficient

1. Introduction

Agile development embraces the continuous evolution of user requirements throughout the entire development process and strives for rapid software development [1]. In agile development, user stories are the primary method for gathering and eliciting requirements. User stories can be written easily; if they are complicated, attention must be paid to simplify them. For systematic requirements analysis, effective mapping to developed features, and efficient maintenance, user stories should be grouped into distinct groups based on their functionality similarity. A project may involve a large number of user stories. However, the majority of the current user story clustering is done manually, which is time-consuming and subject to human bias. Studies have shown that about 80% of agile methods utilize user stories to describe customers' requirements for the system [2]. User stories have been the foundation for a number of major agile development technologies, such as release planning and iteration. User story follows the simple format not

only to interact with customers even also do the requirements elicitation for software developers. They developed the similar functions of the application and system software. Mapping of user story has become the powerful method for managing the user stories in agile development. A story map will give a pictorial representation to development team more emphasis on outcomes of customer's requirement. Similarity between functional requirements of software can be seen in user story. Similar user stories can also be assigned to similar group of development team so development team can plan accordingly [3]. It helps developers to develop the software as soon as possible. Mapping of user story is a between requirements division and methods usability.

Additionally, the maintenance of user stories and the entire development project depend on the user story clustering [4]. The reason for this is that with user story clustering, a set of unlabeled user story data is identified, and then user stories with similar or identical functional descriptions are grouped together. Clearly planning the agile development process can be helpful. User story grouping is another name for user story clustering, which is the process of grouping user stories that are similar to one another. User story clustering has been the subject of some preliminary research.

As far as we are aware, there isn't a straightforward, effective, automatic, or at least semi-automatic method for grouping user stories. Automating the process of forming similar user stories for large systems is challenging and time-consuming. Prior efforts in natural language processing for Agile Requirements Engineering (ARE) have aimed to automate this process, particularly in clustering user stories to decompose system requirements early on. We present a methodology in this study for clustering and analyzing user stories. Modern machine learning approaches, such as feature extraction algorithms and clustering techniques, provide the foundation of the system. A vast array of empirical research has been carried out using publicly available datasets to evaluate the framework and its associated approaches. The experimental findings demonstrate that various feature extraction and clustering techniques have various effects on user story clustering.

The following are this article's main contributions:

- We suggest a framework for organising user stories that combines a variety of feature extraction and clustering techniques.
- To assess how well various user story clustering strategies operate, we have carried out a number of comparative studies.
- After analysing the trial data, recommendations for enhancing user story clustering efficacy are produced.

2. Related Work

Lucassen et al. invented a conceptual model that elicitation concept for user story and form clusters using semantic similarity [5]. Wautelet et al. [6] generates the relation diagram to manage the user stories and clustered through merging and decomposition. Dimitrijevic et al. proposed that need of tool for management of user stories [7]. Carl Yang et al. [8] applied k-means with silhouette analysis on the feature combinations and created four clusters. Sarwosri et al. [9] suggested to identify the conflict between user story using clustering. The k-means and single linkage are used to find out conflict using silhouette values. Takwa Kochbati et al. [10] employed the Hierarchical Agglomerative Clustering algorithm to group the semantically similar requirements and provide an early decomposition of the system. Bo YANG et al. [2] gives a study and framework for user story clustering where k-means, spectral and DBSCAN clustering algorithms are used and features extractions TF-IDF, doc2vec and TCNN are considered. Rizqy Ahsana Putri et al. [11] suggested that k-means clustering improves the performance of user story clustering in agile development. Anže Mihelič et al. [12] identified the key activities, artifacts and roles in agile engineering of secure software by using hierarchical clustering. Bo YANG et al. [13] evaluated the assessment of various machine learning based user story clustering where 5 feature extraction and 7 clustering algorithms are used. As per result analysis, k-means and hierarchical clustering performs better than others. Fredy H. Vera-Rivera et al. [14] gives semantic grouping of user stories to identifying microservices while optimizing for low coupling, high cohesion, and high semantic similarity. Some other requirement tools are available such as Kanboard, Leangoo and visual diagram to manages the user requirements [15]. As a basis for clustering, software requirements analysts read and evaluate user story intent information. This manual work can be completed quickly if there are fewer user stories. However, when the number of user stories grows, manual clustering may produce incorrect or inaccurate clustering results. It's important to note that this procedure takes a lot of time. A method for automatically clustering user stories is urgently required because it will greatly reduce energy consumption and promote the construction of user story mapping [16]. However, to the best of our knowledge, there is not yet a straightforward, effective, automated, or at least semiautomatic, method for clustering user stories. An approach based on Natural Language Processing (NLP) is presented in this paper. Here in this paper, propose a framework for clustering of user story to improve the performance of agile software development. In the approach, format of user story considered first for analysing and extract the task section from given user story. After extraction, similarity of user story is being calculated. The contribution of the paper is described below.

This paper proposed a framework, analysis of user story is done and feature extraction of "I want" section from user story. Through similarity calculation, adjacency matrix is obtained. Prepare the clusters of user story using various clustering algorithms.

2.1 Motivation

To form a cluster of user stories takes heavy scenario procedure which normally a time-consuming process. Most of the time, the user story clustering is done by hand. When requirements change, a lot of user stories can be created and stored in the backlog, especially in many real-world scenarios. The productivity of agile development will suffer as a result of this manual clustering and mapping of user stories. As a result, it is imperative that a clustering approach be developed to automatically group the complex user stories in the backlog.

The manual clustering of these user stories might take 3-5 minutes in the previous small-scale example. However, if there are a lot of user stories (like Baidu's daily addition of more than 6,000 user story cards [7]), the manual clustering method will take a lot of time and be prone to errors. As a result, research into automated methods for clustering user stories is necessary

3. Approach

3.1 Clustering framework for user story

The proposed clustering framework for user story consists of feature extraction and clustering. In first step, "WHAT" section extracted from user story then find out the similarity measures and perform clustering approaches on similarity measures to obtained clusters of user stories.

3.2 Feature extraction

The requirement of features is necessary to identify the content of user story. It has stop word, lemmatization and word division. A user story contains the three sections: Who, What and Why. "WHO" section describes the role, "WHAT" section describes the task and "WHY" section describes the goal. Initially we have all three sections consideration and later feature extraction extracts the "What" section from user story which is part of word division. As "WHAT" section describes the task of the user story which is operational method. A user story defined as Who (as a), What (I want) and Why (So that). Clustering basically implemented on the basis of keywords appear in What section. Here the decomposition of irrelevant words from user stories as no reflection of functional needs. Due to usage of these words impact the performance of clustering. Stop words and lemmatization both also make the improvement in clustering of words during the feature extraction. Keywords appearances in user stories after preprocessing, may lead to creating the clusters of operational words.

3.3 Clustering of user story

User story clustering follows the below steps shown in figure. 1.

1. After data preprocessing, in feature extraction, finds out the count vectorization, TF-IDF, cosine similarity values of selected words.
   a. Countvectorizer(t, d) is used convert the text into numerical data.
   b. TF-IDF calculation
      TF= term frequency
      IDF=inverse document frequency

$$TF(i,j) = \frac{term\ i\ frequency\ in\ document\ j}{total\ words\ in\ document} \tag{1}$$

$$IDF(i) = \log_2 \frac{Total\ documents}{document\ with\ term\ i} \tag{2}$$

TF-IDF score for term i in document

$$\text{j} = TF(i,j) \times IDF(i) \tag{3}$$

$$\text{Cosine similarity} = \cos(\theta) = \frac{A.B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4}$$

   c. Cosine distance=1-cosine similarity
2. Adjacency matrix obtained by similarity measures.
3. Apply various clustering algorithms to have the clusters of selected words of user stories.
4. Find the silhouette coefficient and standard deviation parameters of clusters.

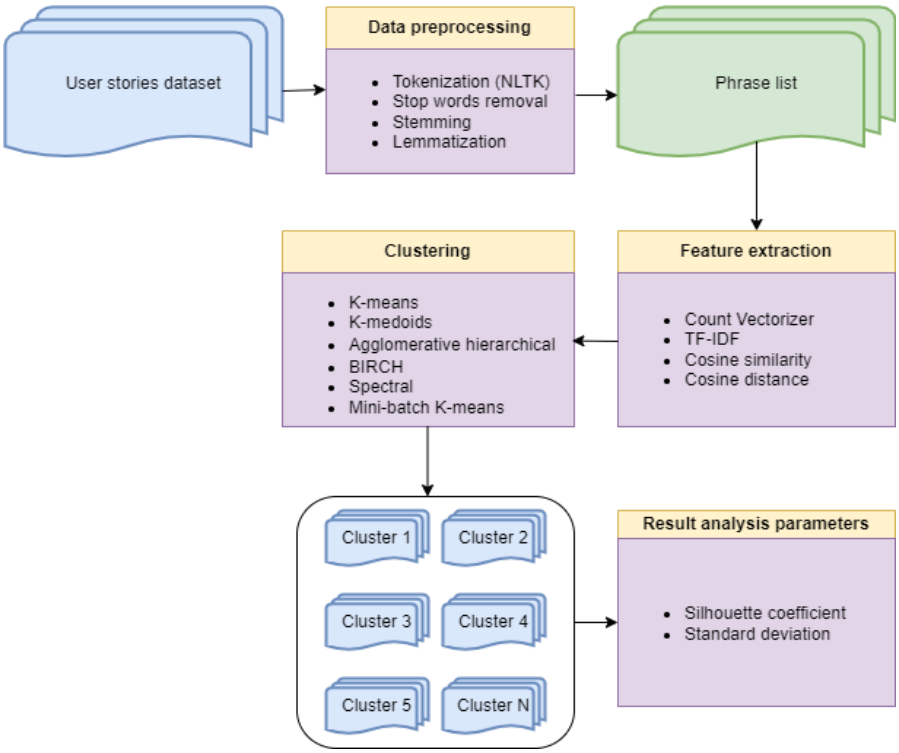

Fig. 1. Flow of proposed work

3.4 Example illustration

Let us have an example to elaboration of our approach. For example, suppose that 5 user stories are considered of user stories:

User story 1: As a Public User, I want to Search for Information, so that I can obtain publicly available information concerning properties, County services, processes and other general information.

User story 2: As a ProspectiveApplicant, I want to research requirements and to select a service, so that I can find the relevant service and/or application type to initiate via the online portal.

User story 3: As an Applicant, I want to Request PreApplication Assistance, so that I can receive a response to a request for a meeting or information that is a result of the preapplication assistance.

User story 4: As a Customer, I want to Create a Customer Portal User Account, so that I can log on to the Customer Portal and perform transactions that first require user authentication.

User story 5: As an Applicant, I want to Submit Application, so that I can provide my information, plans and/or documents to initiate a transaction with the County.

Firstly, extract the feature what section from given 5 user stories. The following output is obtained:

I want to Search for Information.

I want to research requirements and to select a service.

I want to Request Pre-Application Assistance.

I want to Create a Customer Portal User Account.

I want to Submit Application.

Secondly, at runtime stop words implemented to find the operational words only. Because the operational keywords are meaningful for functional requirements of user story and stop word have no worth. Operational keywords are search, information in story 1 and story 2 is having research, requirements, select and service. Under story 3, the keywords are request, preapplication and assistance. Keyword create, customer, portal, user and account are come under story 4 and submit, application are under story 5. In third step, obtain the adjacency matrix of similarity measures such as count vectors, TF-IDF, cosine similarity and distance as well. In fourth step, applying the distinguish clustering algorithms on similarity measures to form the clusters. In last step, obtain the silhouette coefficient value to get the closeness of user stories.

4. Experiment

We conduct the comparative experiments to assess the effectiveness of clustering algorithms.

4.1 Research Questions

This work basically answers the following questions.

RQ1: How to affect the effectiveness of user story clustering?
RQ2: How similarity measures are helpful to perform the user story clustering?
RQ3: Is cluster formation done in order to assess the effectiveness of agile development?
RQ4: Which similarity measure for user story clustering is more efficient than others?
RQ5: To improve the cohesion between user stories which clustering algorithm is best?

4.2 Dataset Description

The proposed model for clustering need of user story collection in order to make clusters. The collection of user stories available publicly approximate 18 files [17], where each file contains approximate in between 47 to 138 user story entries. The statistical representation of dataset where description, size, words, WHO, WHAT and WHY entries described in Table 1.

Table 1. Characteristics of dataset

SR. NO.	DESCRIPTION	SIZE	WORDS	WHO	WHAT	WHY
1	online stage for conveying straightforward data on US government spending	98	2091	98	97	49
2	electronic system for managing land in the Loudoun County, Virginia	57	1579	57	57	57
3	online platform to encourage recycling of waste	51	1287	51	51	45
4	website that makes it easy to see how much the government spends	53	1640	53	53	52
5	first version of the website for the scrum alliance	66	1750	66	66	62
6	the redesign and discovery of content for the new version of the NSF website	97	2581	97	97	94
7	App for parents and camp administrators	73	1749	73	73	56
8	first rendition of the planning-poker site	47	1397	47	47	47
9	stage to find share distribute information on the web	67	1844	67	67	67
10	the board data framework for duke university	66	1537	65	65	4
11	simplified toolbox for quick and easy Hadoop development	64	1627	64	64	4
12	Portal for the management of research data at Oxford University and Southampton	102	2213	102	102	20
13	personal assistant with interactive features for active aging and independent living	138	2445	138	138	9
14	platform for registration and management for the conferences	67	1793	67	67	67
15	plans for machine-adjustable data management	83	2246	83	83	83
16	achieving information system accessible online	56	876	56	56	0
17	university of Bath's official repository for data	100	2016	100	100	1
18	Cornell University's digital content management system	115	3314	115	115	62

4.3 Pre-processing

Tokenization, stop words, stemming and lemmatization are used to make dataset ready for the further process in the pre-processing phase of proposed work. So that relevant keywords of user story are focused at the time of clustering and requirement implementation. NLP provides text preparation techniques, converting raw text into cleaned tokens. Tokens, individual or grouped words, are then counted by frequency, serving as analysis features. Cleaning and tokenization, or text data purification, stop word removal, stemming or lemmatization, and four more steps make up the preprocessing process shown in figure 2. The fact that a word might have several interpretations based on its context or part of speech is a challenge for stemming and text analytics in general. This issue is addressed by lemmatization, which incorporates the part of speech into the rules that categorize word roots.

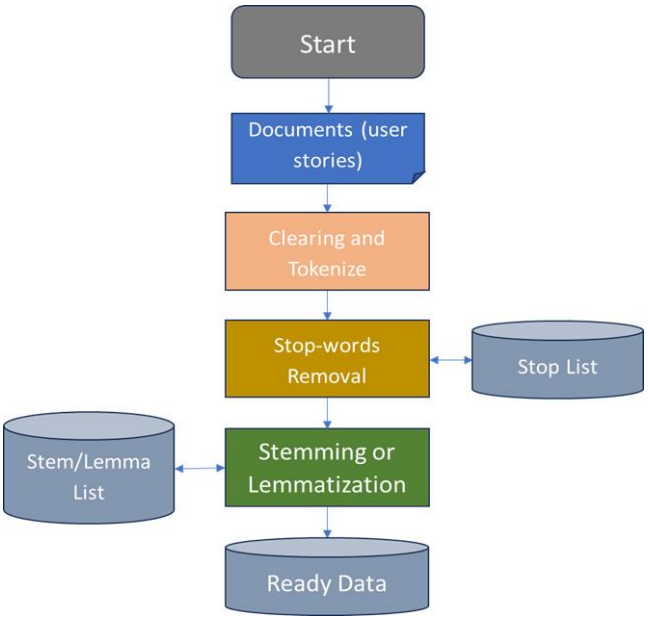

Fig. 2. Pre-processing phases

4.4 Feature extraction

To describe proposed work, various feature extraction techniques are used. To translate the documents into numerical representation, feature extraction techniques are used to represents the documents in the form of metric. When two-dimensional data is ready, easy to implement the clustering algorithms along with x and y axis. Feature extraction is used to work with words frequency. Feature extraction algorithms are described below:

Count vectorizer [18]: - It is used to convert the document text into numerical data which is term vectors or token count. In which involvement to have number of occurrences of words in a document. It has the format of matrix where rows and columns are represented as text token and numerical value of word appearance. Each word can be appeared in different sentences and document as well.

TF-IDF [19]: - It stands for term frequency inverse document frequency, and used to calculate how relevant those words are to given document. It has two statistics TF and IDF, that used to reflect how a word is significant to a document in corpus. It allows for a more accurate representation and support of intuitive approach to weight words.

Cosine similarity [20]: - It is a popular method to find out that how vectors of two words are similar. Usage of cosine is to have the angle between two vectors to obtain how two or more documents are similar. To find out the cosine similarity, initially find the vectorization then TF-IDF. In cosine similarity, dot product between vectors and magnitude calculated. In case of trigonometry purpose, cos(theta) angle lies between 0 and 1.

Cosine distance [21]: - Cosine distance defined as 1- cosine similarity. For example, if two vectors are in same direction then similarity is 1 where cos(theta)=1 and theta angle is 0 whereas distance is 0 which is 1-1=0.

4.5 Clustering algorithms

Grouping of unlabeled user story dataset is not easy task, need to extract the labeled feature from existing dataset. Feature extraction converts the textual dataset into numerical value so that cluster algorithms are able to implement on that data. Clustering is used to form the clusters of user stories on behalf of similarity measures. Clustering techniques are better suited for more complex processing tasks, such as organizing large textual datasets into clusters of user stories. They are useful for identifying previously undetected patterns in user story and can help identify features useful for categorizing the user stories. Following algorithms are used to form the clusters of user stories to make development easy.

K-means [22]: - The aim of this clustering algorithm is to partition the n datapoints into k clusters. It is type of unsupervised learning which works for unlabeled data and also a centroid based where calculate the distance between datapoint and centroid. Each datapoint belongs to the cluster with the nearest mean and find the optimal centroid.

K-medoid [23]: - It is an extended version of K-means clustering algorithm and also works for unlabeled datapoints. It is used in domains that require robustness to outlier data and arbitrary distance metrics.

Agglomerative hierarchical [24]: - It supports bottom-up approach where each datapoint considered as cluster later two similar datapoints together makes a cluster. This procedure follows iteration for all datapoints to be the part of clusters. Distance between two clusters if closest then it will be important to form clusters, it is known as linkage method. Hierarchical clustering follows the tree like structure used to hold each iteration a memory.

BIRCH [25]: - To focus on the running time and large dataset, regular clustering does not scale well. So, Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) is able to perform better than others on large dataset. It is

also advanced than k-means clustering for noise datapoints. In this clustering no need to require dataset in advance at initial stage. Parameters used to implement BIRCH are threshold, factor of branching and number of clusters.

Spectral [26]: - It is focused on the graph partitioning problem, where each datapoint worked as graph node. It is a flexible and also allows to make clusters of non-graph data. It uses the information from spectrum of eigenvalues of matrices from the dataset. Is used for both types of data graph and arbitrary data.

Mini-batch K-means [27]: - It is also version of K-means algorithm which is used for small, fixed and random size of batches to store in memory. In each iteration, randomly datapoints is collected and used to make the clusters. It reduces the computational cost to find out a cluster. It performs on large dataset to get better performance in terms of run time than K-means algorithm.

K-means is best used on smaller data sets because it iterates over all of the data points. K-medoids minimizes a sum of pairwise dissimilarities instead of a sum of squared Euclidean distances. BIRCH algorithm works better on large data sets than the k-means algorithm. Spectral clustering can be more robust to noise and outliers in the data. Agglomerative clustering is best at finding small clusters. Mini-batch K-means useful when the data is too large to fit in the memory and traditional K-means can't be applied.

In this proposed work, first considered a feature extraction algorithm and then implemented clustering algorithms to form the clusters of user stories. There are combinations of clustering algorithms with feature extraction to get the performance level of user stories. In experiment phase of proposed work, 6 clustering and 4 feature extraction techniques have been used to analyze the performance.

4.6 Performance metrics

To obtain the effectiveness of clustering algorithm need to have silhouette coefficient value metric[28]. This value indicates that how datapoints of its own cluster closed each other which is cohesion and also compared to other clusters. The value lies in-between -1 and +1, where 1 means closeness and compactness of datapoints under clusters and -1 means worst value which indicates the overlapping. silhouette coefficient value is calculated with mean of intra cluster distance denoted by A and mean of nearest cluster distance denoted B for each datapoint. The silhouette coefficient value is denoted by S(i) for each datapoints, which as follows.

$$S(i) = \frac{B(i)-A(i)}{\max\{A(i),B(i)\}} \tag{5}$$

Where A(i) is denoted as average distance between i and all the remaining datapoints with in the cluster and B(i) is for average distance from i to all clusters.

Another parameter is standard deviation which is used to know that how the balanced clusters of user stories are formed. A smaller standard deviation means your data points cluster closely around the mean, while a larger one suggests more spread. In case of user story clustering, low standard deviation means user stories are closer to other with in cluster. The formula for standard deviation as follows.

$$Standard\ Deviation = \sqrt{\frac{\sum_{i=0}^{n}(x_i-\bar{x})^2}{n-1}} \tag{6}$$

where: x_i=Value of the ith point in the data set
$\bar{x}$=The mean value of the data set
n=The number of data points in the data set

Dunn index and Davies–Bouldin index parameters are also used to validate and analysis the clusters. Where Dunn index identifies groups of clusters that exhibit both compactness and low variance among their members and Davies–Bouldin index expresses the similarity between clusters. But proposed work focused on the similarity of user stories of cluster and how the user stories are binding with in a cluster.

5. Result Analysis and Evaluation

RQ1: How to affect the effectiveness of user story clustering?

Usually, user story has three terms which are who, what and why. Each term has their importance to define the user story which affects the user story clustering performance. This paper extracts these terms from user story shown in figure 3, where extraction shows the completeness of the user stories.

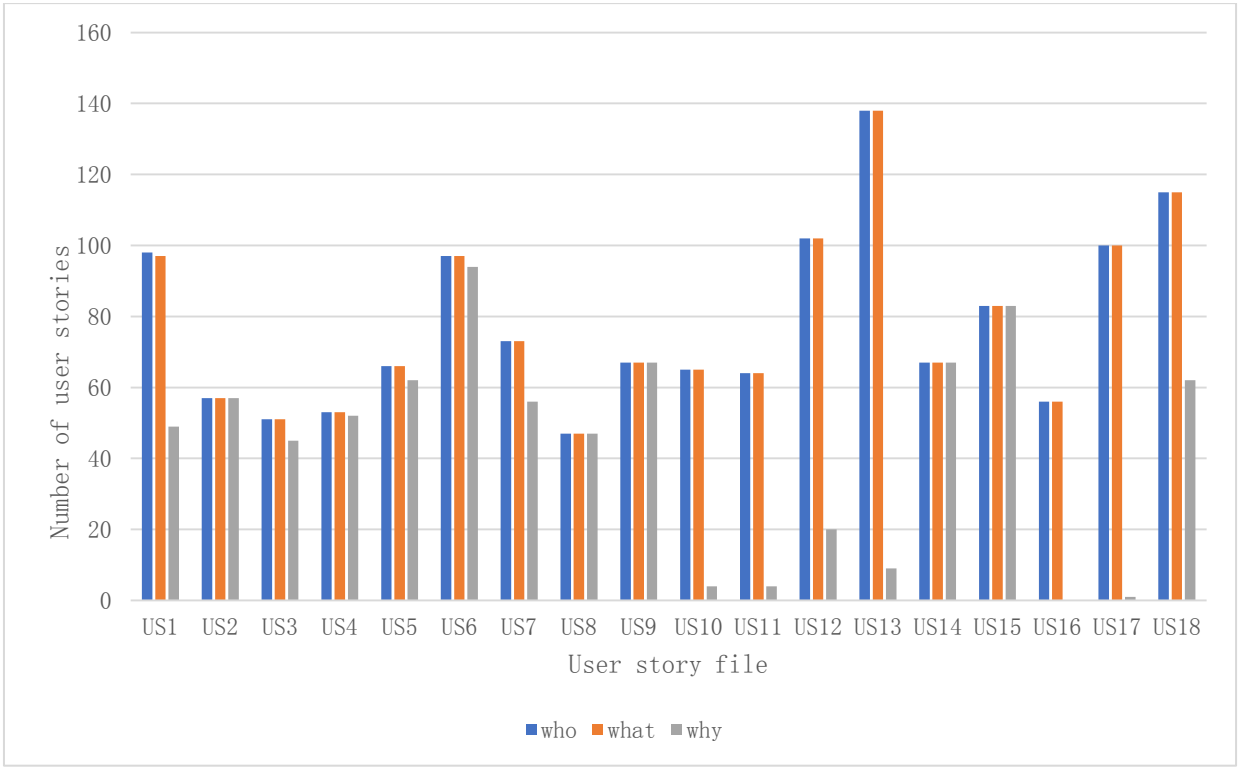

Fig. 3. Three terms (who, what and why) of user story

As completeness level of user story depends on all three terms. User story text file US2, US3, US4, US5, US6, US7, US8, US9, US14 and US15 have the high-level completeness and, US1 and US18 are on mid-level completeness whereas US10, US11, US12, US13, US16 and US17 are coming under low level completeness due to terms missing. High-level completeness of user story files lies between 75 and 100%, mid-level percentage is in between 50 and 74% and for low level range is in between 0 to 49%. The number of user story files with their completeness level shown in figure 4.

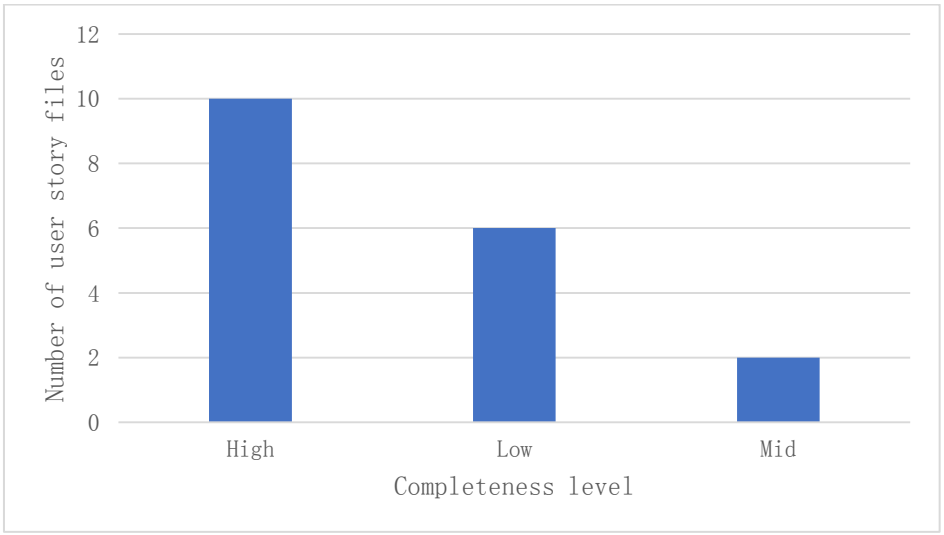

Fig. 4. Completeness level of user stories

RQ2: How similarity measures are helpful to perform the user story clustering?

To determine how similar two user stories are, similarity measures are used. Several similarity measures are frequently utilized in clustering. The significance of measuring distances and similarity in clustering helps to rectify the relationships and patterns between user stories. It improves the accuracy of clustering algorithms by calculating the similarity between user stories and can make clusters of similar user stories and also minimize the number of misclassified user stories. By using similarity measures, clustering algorithms can work quickly and accurately on huge datasets which reduces the complexity of computation. Similarity measures enable to implement different clustering

algorithms. Proposed worked on count vectorization, TF-IDF, cosine similarity and, cosine distance. As the comparison between distance measures, cosine similarity is more robust than others and make efficient computation. Cosine similarity is best suited for text classification and information retrieval.

RQ3: Is cluster formation done in order to assess the effectiveness of agile development?

In order to form the clusters of user stories, different algorithms are implemented on dataset. These clustering algorithms formed the clusters on the basis on following similarity measures such as count vectorization, TF-IDF, cosine similarity and, cosine distance as show in figures. The number of clusters is taken as 10. Each cluster formed with their similar user story on the basis of measures. As comparison, spectral clustering is formed cluster more effectively than others when CV is used as similarity measure shown in figure 5, it has standard deviation is 3.85 which is lowest.

When TF-IDF is considered similarity measure then agglomerative clustering is best than shown in figure 6. The standard deviation in case of TF-IDF is 6.39 which is lowest in comparison with others.

Fig. 5. CV based cluster formation

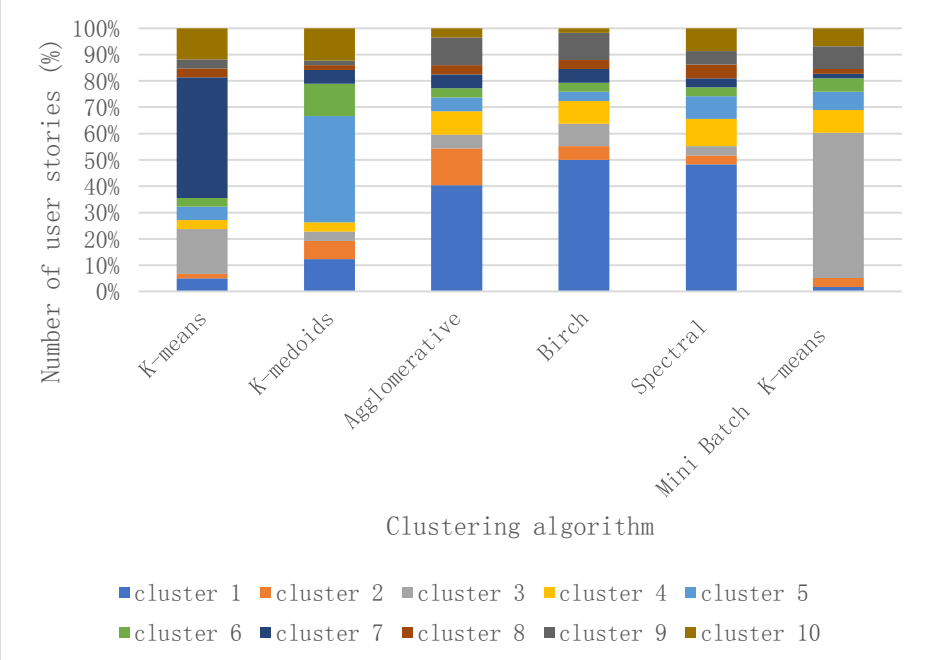

Fig. 6. TF-IDF based cluster formation

When cosine similarity is considered as similarity measure then K-means and BIRCH clustering both are giving best result to form clusters shown in figure 7. The standard deviation for K-means and BIRCH cluster is 3.08 which is same for both.

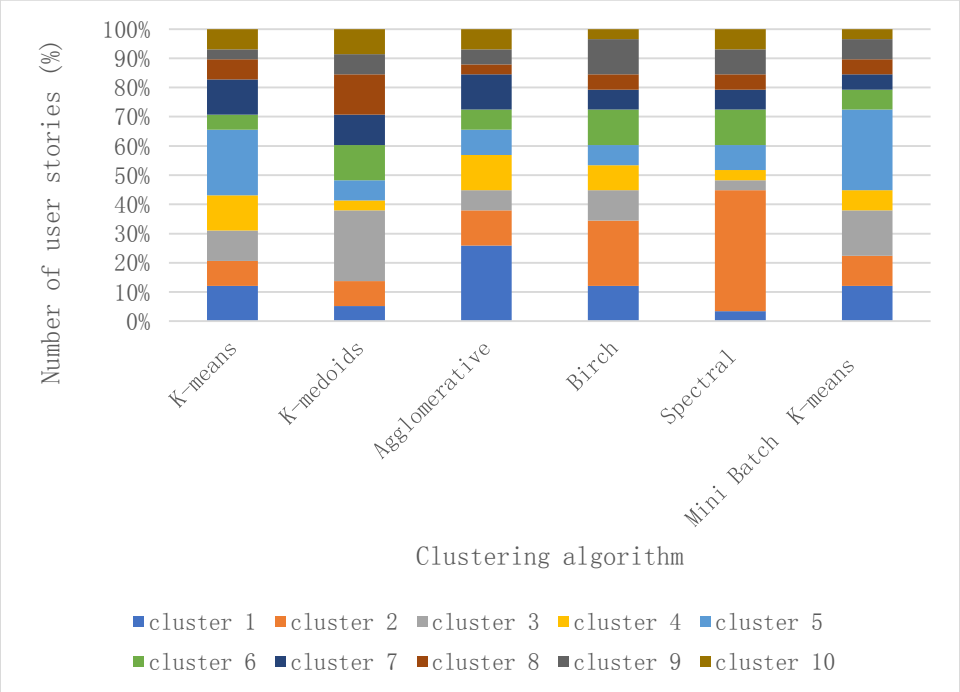

Fig. 7. Cosine similarity-based cluster formation

When cosine distance is considered similarity measure then only BIRCH clustering gives output to form clusters shown in figure 8. The standard deviation for BIRCH cluster is 3.08.

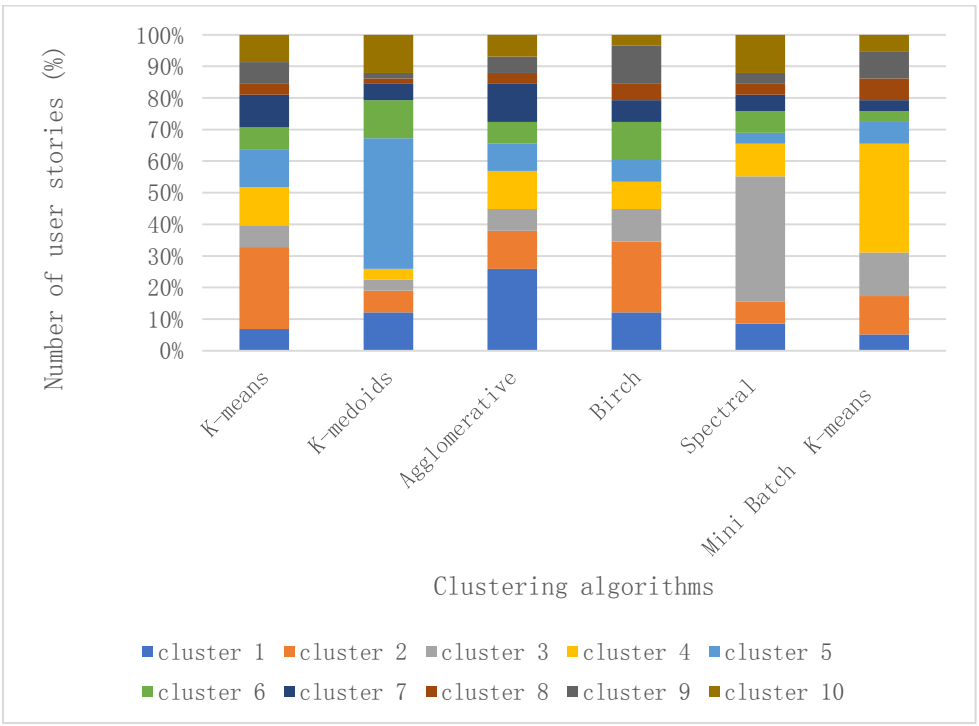

Fig. 8. Cosine distance-based cluster formation

RQ4: Which similarity measure for user story clustering is more efficient than others?

Data with a low standard deviation are grouped around the mean, while data with a high standard deviation are more dispersed. Among similarity measures, cosine similarity has low standard deviation which is more efficient than others shown in figure 9. For K-means and BIRCH, the standard deviation is 3.08, for K-medoids, 3.39, 3.68 for agglomerative, 6.59 for spectral and 4.16 for mini batch K-means. The lowest standard deviation means that K-means and BIRCH are more efficient than other for user story clustering to form the clusters more balanced.

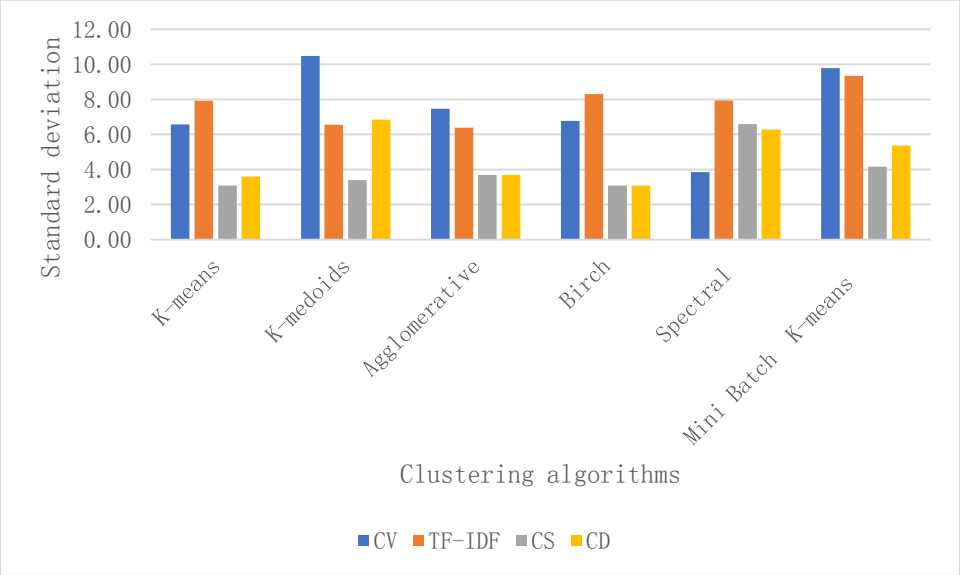

Fig. 9. Standard deviation on different features extraction

RQ5: To improve the cohesion between user stories which clustering algorithm is best?

As figure 10 shows that cosine similarity and distance both marked silhouette coefficient value on and above 0.20 as comparison to CV and TF-IDF using clustering algorithms. The comparison between clustering algorithms shows that spectral clustering is giving best result than others which has 0.225 silhouette coefficient value for cosine similarity and distance. Mini-batch K-means is also performed better which has 0.221 silhouette coefficient value for cosine similarity and distance. Agglomerative also has the optimized results which has 0.216 silhouette coefficient value for cosine similarity and distance.

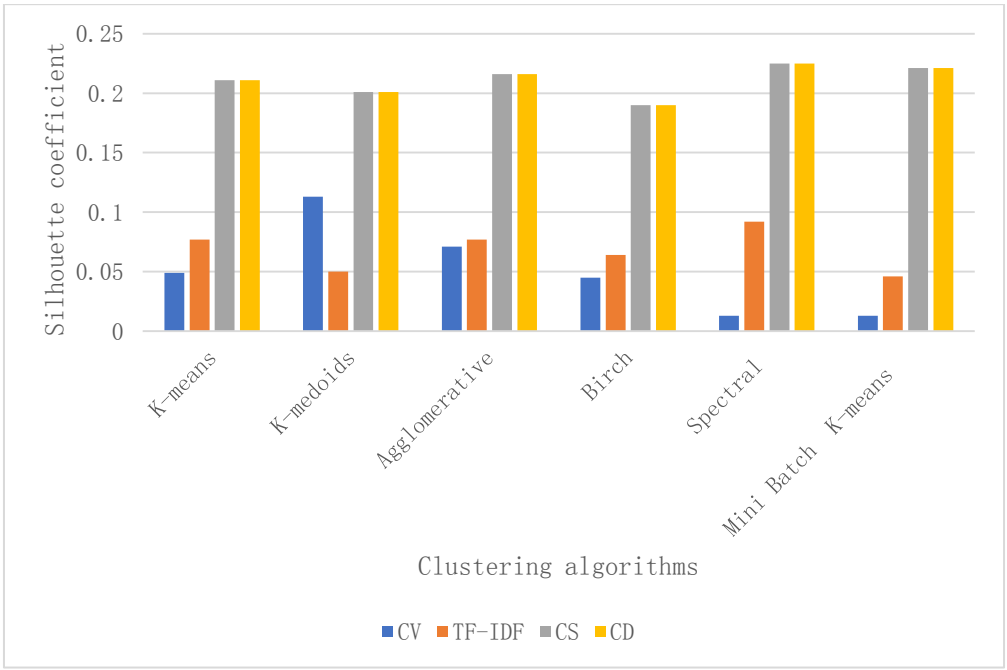

Fig. 10. Silhouette coefficient values on different features extraction

6. Discussion

Our research shows that when it comes to user story clustering, proposed work can assist clients and developers in getting the best possible outcomes. The user-centered design should be reflected in the user stories that are produced. Furthermore, the related function must pass the user acceptability test when it is developed. Because of this, user stories should be created iteratively, involving collaboration between engineers and clients. Better results in the user stories and created features may be attained by clustering user stories according to the customer and developer's discussions.
For the result analysis, two parameters standard deviation and silhouette coefficient are used in user story clustering. When clusters of user stories are formed then first check whether the clusters are formed balanced or not which depends on the standard deviation value. If the standard deviation value is minimum then formed balanced otherwise not. For K-means and BIRCH with cosine similarity measure, the standard deviation value is 3.08 which shows that balanced clusters. The cosine similarity is best suited to form the balanced clusters using K-means and BIRCH clustering algorithms. Another parameter is silhouette coefficient where high value indicates the high cohesion and low indicates the low cohesion between user stories with in cluster. The silhouette coefficient value of spectral clustering is 0.225 which higher than other cluster algorithms with cosine similarity and distance measures. That shows the high cohesion between the user stories of a cluster. Another high silhouette coefficient value is 0.221 which is of mini-batch K-means also has the second highest value for high cohesion with same measure.

7. Conclusion

In this paper, we suggested a framework based on four feature extraction approaches and six clustering algorithms to categorize user stories. A series of experiments were carried out utilizing these techniques. The outcomes of the experiment demonstrated how well our approach may assist in clustering user stories. Furthermore, we offer recommendations on how to modify the framework's method's parameters. Proposed worked on count vectorization, TF-IDF, cosine similarity and, cosine distance. As the comparison between distance measures, cosine similarity is more robust than others and make efficient computation. Cosine similarity is best suited for text classification and information retrieval. As comparison, spectral clustering is formed cluster more effectively than others when CV is used as similarity measure shown in figure 4, it has standard deviation is 3.85 which is lowest. When TF-IDF is considered similarity measure then agglomerative clustering is best than others. The standard deviation in case of TF-IDF is 6.39 which is lowest in comparison with others. When cosine similarity is considered as similarity measure then K-means and BIRCH clustering both are giving best result to form clusters. The standard deviation for K-means and BIRCH clustering is 3.08. The lowest standard deviation means that K-means and BIRCH are more efficient than other for user story clustering to form the clusters more balanced. The comparison between clustering algorithms shows that spectral clustering is giving best result than others which has 0.225 silhouette coefficient value for cosine similarity and distance.

By automatically clustering user stories and extracting features to aid in the speedy generation of user story mapping, this study is a part of a human-computer collaboration to produce a user story clustering. Additionally, using an ensemble might be a viable course of action. We will think about how to apply the ensemble strategy to enhance user story clustering performance. This work will also be helpful where the large number of user stories presented which improves the development phases of agile software.

References

[1] S. Nazir, B. Price, N. C. Surendra, and K. Kopp, “Adapting agile development practices for hyper-agile environments: lessons learned from a COVID-19 emergency response research project,” *Information Technology and Management*, vol. 23, no. 3, pp. 193–211, Sep. 2022, doi: 10.1007/s10799-022-00370-y.

[2] B. Yang, X. Ma, C. Wang, H. Guo, H. Liu, and Z. Jin, “User story clustering in agile development: a framework and an empirical study,” *Front Comput Sci*, vol. 17, no. 6, Dec. 2023, doi: 10.1007/s11704-022-8262-9.

[3] Z. Masood, R. Hoda, and K. Blincoe, “How agile teams make self-assignment work: a grounded theory study,” Empir Softw Eng, vol. 25, no. 6, pp. 4962–5005, Nov. 2020, doi: 10.1007/s10664-020-09876-x.

[4] T. Georges, L. Rice, M. Huchard, M. König, C. Nebut, and C. Tibermacine, “Guiding Feature Models Synthesis from User-Stories: An Exploratory Approach,” in Proceedings of the 17th International Working Conference on Variability Modelling of Software-Intensive Systems, New York, NY, USA: ACM, Jan. 2023, pp. 65–70. doi: 10.1145/3571788.3571797.

[5] N. Bik, G. Lucassen, and S. Brinkkemper, “A reference method for user story requirements in agile systems development,” Proceedings - 2017 IEEE 25th International Requirements Engineering Conference Workshops, REW 2017, no. September, pp. 292–298, 2017, doi: 10.1109/REW.2017.83.

[6] D Y. Wautelet, S. Heng, M. Kolp, and I. Mirbel, “Unifying and Extending User Story Models,” in International Conference on Advanced Information Systems Engineering, 2014, pp. 211–225.

[7] S. Dimitrijević, J. Jovanović, and V. Devedžić, “A comparative study of software tools for user story management,” Inf Softw Technol, vol. 57, pp. 352–368, Jan. 2015, doi: 10.1016/j.infsof.2014.05.012.

[8] C. Yang, X. Shi, L. Jie, and J. Han, “I Know You’ll Be Back: Interpretable New User Clustering and Churn Prediction on a Mobile Social Application,” in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA: ACM, Jul. 2018, pp. 914–922. doi: 10.1145/3219819.3219821.

[9] Sarwosri, U. L. Yuhana, and S. Rochimah, "Identification of Conflicts in User Story Requirements Using The Clustering Algorithm," in 2022 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), IEEE, Nov. 2022, pp. 1–5. doi: 10.1109/CENIM56801.2022.10037416.
[10] T. Kochbati, S. Li, S. Gérard, and C. Mraidha, "From user stories to models: A machine learning empowered automation," MODELSWARD 2021 - Proceedings of the 9th International Conference on Model-Driven Engineering and Software Development, pp. 28–40, 2021, doi: 10.5220/0010197800280040.
[11] R. A. Putri, U. L. Yuhana, and T. C. Amri, "K-Means and Feature Selection Mechanism to Improve Performance of Clustering User Stories in Agile Development," in 2023 International Conference on Modeling & E-Information Research, Artificial Learning and Digital Applications (ICMERALDA), IEEE, Nov. 2023, pp. 39–43. doi: 10.1109/ICMERALDA60125.2023.10458165.
[12] A. Mihelič, T. Hovelja, and S. Vrhovec, "Identifying Key Activities, Artifacts and Roles in Agile Engineering of Secure Software with Hierarchical Clustering," Applied Sciences, vol. 13, no. 7, p. 4563, Apr. 2023, doi: 10.3390/app13074563.
[13] B. Yang, H. Guo, and H. Liu, "Evaluation and assessment of machine learning based user story grouping: A framework and empirical studies," Sci Comput Program, vol. 227, Apr. 2023, doi: 10.1016/j.scico.2023.102943.
[14] F. H. Vera-Rivera, E. G. Puerto Cuadros, B. Perez, H. Astudillo, and C. Gaona, "SEMGROMI—a semantic grouping algorithm to identifying microservices using semantic similarity of user stories," PeerJ Comput Sci, vol. 9, p. e1380, May 2023, doi: 10.7717/peerj-cs.1380.
[15] A. Jarzebowicz and P. Weichbroth, "A Qualitative Study on Non-Functional Requirements in Agile Software Development," *IEEE Access*, vol. 9, pp. 40458–40475, 2021, doi: 10.1109/ACCESS.2021.3064424.
[16] M. A. Kuhail and S. Lauesen, "User Story Quality in Practice: A Case Study," *Software*, vol. 1, no. 3, pp. 223–243, Jun. 2022, doi: 10.3390/software1030010.
[17] F. Dalpiaz, "Requirements data sets (user stories)," in Mendeley Data, V1, 2018. doi: 10.17632/7zbk8zsd8y.1.
[18] V. A. Kozhevnikov and E. S. Pankratova, "RESEARCH OF THE TEXT DATA VECTORIZATION AND CLASSIFICATION ALGORITHMS OF MACHINE LEARNING.," *Theoretical & Applied Science*, vol. 85, no. 05, pp. 574–585, May 2020, doi: 10.15863/TAS.2020.05.85.106.
[19] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, pp. 11–21, 1972.
[20] A. R. Lahitani, A. E. Permanasari, and N. A. Setiawan, "Cosine similarity to determine similarity measure: Study case in online essay assessment," in *2016 4th International Conference on Cyber and IT Service Management*, IEEE, Apr. 2016, pp. 1–6. doi: 10.1109/CITSM.2016.7577578.
[21] B. Li and L. Han, "Distance Weighted Cosine Similarity Measure for Text Classification," in *International Conference on Cyber and IT Service Management*, 2013, pp. 611–618. doi: 10.1007/978-3-642-41278-3_74.
[22] Y. Li and H. Wu, "A Clustering Method Based on K-Means Algorithm," *Phys Procedia*, vol. 25, pp. 1104–1109, 2012, doi: 10.1016/j.phpro.2012.03.206.
[23] W. Y. E, A. Jian, L. Yan, and W. HongGang, "Optimization of K-medoids Algorithm for Initial Clustering Center," *J Phys Conf Ser*, vol. 1487, no. 1, p. 012011, Mar. 2020, doi: 10.1088/1742-6596/1487/1/012011.
[24] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," *arXiv:1109.2378v1*, 2011.
[25] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: A New Data Clustering Algorithm and Its Applications," *Data Min Knowl Discov*, vol. 1, no. 2, pp. 141–182, 1997, doi: 10.1023/A:1009783824328.
[26] U. von Luxburg, "A Tutorial on Spectral Clustering," *Stat Comput*, vol. 17, no. 4, 2007.
[27] A. Feizollah, N. B. Anuar, R. Salleh, and F. Amalina, "Comparative study of k-means and mini batch k-means clustering algorithms in android malware detection using network traffic analysis," in *2014 International Symposium on Biometrics and Security Technologies (ISBAST)*, IEEE, Aug. 2014, pp. 193–197. doi: 10.1109/ISBAST.2014.7013120.
[28] K. R. Shahapure and C. Nicholas, "Cluster Quality Analysis Using Silhouette Score," in *IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, 2020. doi: 10.1109/DSAA49011.2020.00096.

Authors' Profiles

Bhawnesh Kumar, Research Scholar, Department of Computer Science and Engineering in Graphic Era Deemed to be University, Dehradun, Uttarakhand, India. He had received MCA from Kurukshetra University, Kurukshetra, Haryana, India, M.Tech (CSE) from R. G. T. U. Bhopal, M.P (State University), India. His areas of interest are agile software development, wireless sensor network, computer graphics and machine learning.

Umesh Kumar Tiwari, Ph.D, he is working as an Associate Professor in Department of Computer Science and Engineering in Graphic Era Deemed to be University, Dehradun. He had received his Ph.D. in 2016. He has more than 12 years of experience in teaching/research of UG and PG level degree courses as a Lecturer/Assistant Professor/ Associate Professor in various academic/research organizations. He is supervisor of 02 PhD and 5 M. Tech students who are working on specific domains of software engineering and network security. His research is on multidisciplinary topics and he has published 17 journal papers in reputable international and national journals, and 12 conference papers in reputed international and national conferences. His research interests are Wireless Communication Networks, Network Security, and Software Engineering

topics with improved modeling, interaction-integration complexities, testing and reliability models.

Dinesh C. Dobhal, Ph.D, he is working as an Associate Professor in Department of Computer Science and Engineering in Graphic Era Deemed to be University, Dehradun. He had received his Ph.D. in 2017. He has more than 20 years of experience in teaching/research of UG and PG level degree courses as a Lecturer/Assistant Professor/ Associate Professor in various academic/research organizations. He has a vast experience in the area of teaching and research. He is an author of a number of research papers in SCOPUS indexed Journals/Conferences. His research interests are Cyber Security, Malware Analysis, Mobile Ad hoc Network and Software Engineering.

How to cite this paper: Bhawnesh Kumar, Umesh Kumar Tiwari, Dinesh C. Dobhal, "Performance Evaluation of Various Machine Learning Algorithms for User Story Clustering", International Journal of Modern Education and Computer Science(IJMECS), Vol.17, No.3, pp. 92-105, 2025. DOI:10.5815/ijmecs.2025.03.07