JOURNAL OF INFORMATION TECHNOLOGY AND ITS UTILIZATION, VOLUME 9, ISSUE 1, MAY 2026
EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.9.1.6065. SUBMITTED: JANUARY 15, 2026; REVISED: FEBRUARY 9, 2026; ACCEPTED: APRIL 6, 2026

Comparative Analysis of Distance Measures in Bug Report Clustering using Agglomerative Hierarchical Clustering

Krisnawan Hartanto[1], Suprapto[2]

[1]Game Technology Study Program, Sekolah Tinggi Multi Media "MMTC" Yogyakarta, Indonesia
[2]Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia
krisnawan.hartanto@mmtc.ac.id[1*], sprapto@ugm.ac.id[2]
*Corresponding author

***Abstract*--Grouping bug reports into clusters can assist in verifying and validating bugs in the software development cycle. One of the clustering methods is Agglomerative Hierarchical Clustering (AHC). It relies on distance calculations to determine the degree of similarity between clusters. One of the distance calculations is the Jaccard coefficient. The Jaccard Coefficient method has the disadvantage that it only considers the same set of words between two documents but does not consider their importance. Previous research added Inverse Document Frequency (IDF) algorithm to the Jaccard coefficient to calculate the importance of word groups and in this research is referred to the weighted Jaccard coefficient. Clustering is carried out using a combination of AHC and that coefficient. The silhouette score is then compared with the silhouette score of AHC with the Jaccard coefficient. Results indicate that increasing term complexity reduces cluster quality, with silhouette scores dropping from 13.13% (bigram) to 0.45% (4-gram). Furthermore, many clusters exhibited negative silhouette scores, highlighting the difficulty of separating high-dimensional bug data using unsupervised methods. In contrast, the supervised classification baseline achieved significantly higher accuracy. This paper contributes a critical analysis demonstrating that while Weighted Jaccard captures semantic nuance, unsupervised clustering remains insufficient for this domain compared to supervised approaches.**

***Key words*: Agglomerative Hierarchical Clustering; AHC; Bug report; Cluster; IDF.**

I. INTRODUCTION

Bugs or defects are common problems in the world of software development. Both can appear in any stage of software development. A common challenge in the testing phase is the sporadic and unpredictable nature of bug reports. One method for dealing with the problems that arise is to group each bug that is automatically reported into several clusters. Grouping is based on similarities where bugs occur. The clustering method was chosen because it can categorize without having to do data labeling as in the classification method [1]. So that the groups of bugs that are formed are not limited to predetermined categories.

Agglomerative Hierarchical Clustering (AHC) is a clustering method based on hierarchical formation [2]. Starting with a large number of initial small clusters, the grouping was carried out in a bottom-up manner by combining a pair of clusters with the greatest similarity level according to certain criteria until the desired conditions are achieved. In determining clusters, AHC is highly dependent on the distance calculation algorithm.

The hierarchical model is used because the software architecture being tested resembles a hierarchy. Software consists of several services. Each service can consist of several modules, and each module has endpoints. In the bug search process, bugs were found at the smallest endpoints. The bugs found were grouped according to the smallest software component to the largest component. This treatment is similar to the Agglomerative Clustering method. For this reason, the AHC method was proposed as a clustering method for the bug report. Unlike the case with the K-mean method, which forms clusters by first determining the centroid randomly, AHC tends to produce more consistent clusters.

Research on clustering bug reports has been conducted using various methods such as: Hierarchical Clustering [3], NLP [4], and classification [5] to build a clustering system for bug reports. Some studies used AHC for clustering in addition to bug reports, such as unemployment analysis [6], optimal number of clusters [7], online social networks [2], Ward Linkage Data Clustering [8], Hierarchical Agglomerative Clustering Optimization for

EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.9.1.6065. SUBMITTED: JANUARY 15, 2026; REVISED: FEBRUARY 9, 2026; ACCEPTED: APRIL 6, 2026

Massive Data [9], used as a second method after clusterization using FCM [10], and pattern clustering of industrial time series [11].

Some research on bug clustering focused on clustering techniques comparison for bug triaging models using Random Forest, SVM, Logistic Regression, Decision Tree, Neural Networks [12], and Bidirectional Encoder Representations from Transformer [13].

This research refers to the research that has been done before, which uses Hierarchical Clustering [3] to perform clustering on bug reports. As a distance measurement algorithm, it is proposed to add the Improved Jaccard Similarity Coefficient [14]. Inverse Document Frequency (IDF) calculation was added to calculate the importance of word clusters for certain documents. This addition is inspired by the TF-IDF algorithm. An innovation model called the Weighted Jaccard Coefficient, is expected to increase the accuracy of main topic extraction.

Existing research often applies clustering algorithms to bug reports under the assumption that natural, distinct groups exist. However, few studies have critically analyzed the limitations of distance measures like Jaccard when applied to noisy, unstructured bug data. The research gap addressed in this study is the lack of comparative evaluation between weighted distance metrics and supervised baselines. We aim to determine if a weighted Jaccard coefficient can improve separability or if the data require supervised learning to be effectively managed.

II. METHOD

A. System Analysis

The data consisted of 220 columns and 651 rows. Columns containing bug information data include Summary, Bug Location, Status, Resolution, Description, and Issue Type. Columns other than those previously mentioned were irrelevant to the study, some of which have no data. The “Project key” and “Project name” columns have the same value for all data. The “Issue key” and “Issue ID” columns were data identity numbers. The data used are shown in Table I.

TABLE I
Sample Dataset Documents before Pre-Processing

Summary	Status	Description	Priority	Issue Type	Bug Location
[Dev][Demo] [Admin Portal][Edit Bundle items]: ...	To Do	This issue occurs on *Add new bundle items.*\n...	P1	Bug	Admin Portal
[Dev][Admin Portal][Creat e New][Create From ex...	In Pro gress	This issue occurs on Create new Role and Creat...	P1	Bug	Admin Portal
Duplicate items while using Search items by ke...	To Do	While making script for cleanup, we found that...	P2	Bug	Automat ion Test

The priority and issue type fields have no significant values, so they can be excluded. Then, the columns used are Summary, Status, Description, and Bug Location. After getting the column that was used for the clustering process, the process of checking which column has a null value is carried out. In Fig. 1, it can be seen that the Description column has 59 rows with a null value.

```
Summary          0
Status           0
Description     59
Priority         0
Issue Type       0
Bug Location     0
dtype: int64
```

Fig. 1. Column with null value

To prevent errors from occurring when the program run, the null value in the description was overwritten with a string value, this was deleted during the pre-processing. The string value used is "0".

B. System Design

In this section, we discuss the design of the AHC clustering model using the calculation of the Weighted Jaccard Coefficient which was built and the test plan that was carried out. Fig. 2 shows a flowchart of the model design, which consists of input documents in the form of a collection of bug reports, pre-processing, and calculation of Improved Jaccard Coefficient, to the formation of a distance matrix.

EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.9.1.6065. SUBMITTED: JANUARY 15, 2026; REVISED: FEBRUARY 9, 2026; ACCEPTED: APRIL 6, 2026

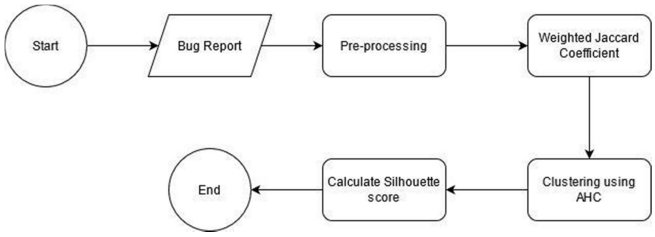

Fig. 2. Model design diagram

1. Input Document

The data used were bug report data taken from the repository PT. AccelByte Technology Indonesia.

2. Pre-processing
   a. Lower case conversion

This stage ensures that the text is in a form that is easy to manipulate. Activity at this stage was to convert every character in the document to lowercase. This step is necessary so that the same word written in lowercase and uppercase is not considered two different words, for example "Bug" and "bug".

   b. Tokenization

Documents that have been in lowercase letters were then converted into a list of tokens. Token is the smallest part of a string that can be letters, words, or sentences. For this research, the token needed is in the form of words.

   c. Noise removal

At this stage, checking of existing tokens was carried out. Tokens in the form of irrelevant text, or so-called stop words, were deleted. Strings containing HTML addresses were also removed. So only the relevant tokens remain.

   d. Stemming

The remaining relevant tokens were converted into their base form. To perform this step, a stemming rule is needed, which is an association rule from a token with its basic form.

3. Weighted Jaccard Coefficient Calculation
   a. Find A intersection B

A and B are the sets of tokens representing each document. In this process, the program looks for a token that is a member of set A and is also a member of set B.

   b. Calculate the Jaccard Coefficient

The Jaccard coefficient method is mostly used for binary variables where the values 0 and 1 do not have the same frequency of occurrence. Jaccard coefficients can also be generalized to be applied to non-binary variables. In that case, the Jaccard coefficient is calculated based on set theory [16]. Suppose there are two sets A and B. Then, the Jaccard Coefficient between these two sets is calculated using Equation (1).

$$Jaccard = |A \cap B| / |A \cup B| \quad (1)$$

Where $A \cap B$ is the set of all objects belonging to set A, which are also members of set B. While $A \cup B$ is the set of all objects belonging to set A and members of set B.

   c. Calculate IDF A intersection B

To calculate the level of importance in words, the IDF formula was added to add value to Jaccard. IDF calculations borrow similarities from TF-IDF but have differences. The IDF calculation in this study was the average IDF value in the word group that is the intersection of the two documents being compared. To calculate the IDF, we used equation (2).

$$idf_j = log(D/df_j) \quad (2)$$

Where D is the number of documents in the collection, and df_j is the number of documents containing the searched word. The equation can produce a lower limit of weights of 0 when a word appears in the entire document collection. To overcome this, 1 can be added to the denominator. So that it became like equation (3).

$$idf_j = log(D/(df_j+1)) \quad (3)$$

The reason for adding 1 in the denominator is to avoid a situation where a word appears in all documents, so the weight is 0. However, even adding the number 1 does not close the emergence of new problems. The problem that arises is that if the value of df_j is D-1, df_j is log 1, the weight remains 0. For this reason, it is necessary to ensure that the dataset used does not find the value of $df_j = D$-1.

   d. Calculate the Weighted Jaccard Coefficient

The combination of equations (1) and (3) is called the Weighted Jaccard Coefficient (WJC). To calculate WJC, equation (4) can be used.

$$wjc = jaccard \times idf \quad (4)$$

EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.9.1.6065. SUBMITTED: JANUARY 15, 2026; REVISED: FEBRUARY 9, 2026; ACCEPTED: APRIL 6, 2026

4. Clustering using Agglomerative Hierarchical Clustering

Agglomerative Hierarchical Clustering (AHC) is a clustering algorithm that adopts the Bottom-Up system. Clusters are built by combining smaller clusters called singleton clusters based on the similarity from the bottom to the top. AHC requires determining a measure of similarity between data items and a computational scheme of similarity between clusters.

Fig. 3 illustrates the data clustering process with AHC [15]. Initially, there are 6 separate data as input. Each of these data is considered a single cluster called a singleton, thus forming a number of 6 clusters containing a single data. After the initialization process, all the correct pairs are formed from the cluster list, and the degree of similarity between clusters is based on distance calculations.

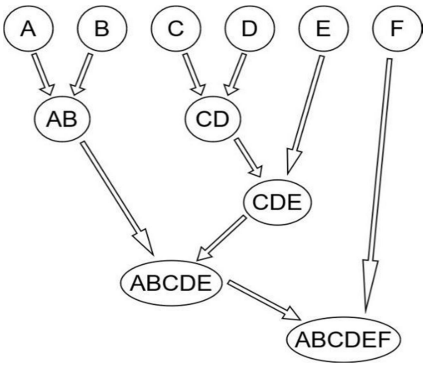

Fig. 3. Clustering process flow scheme with AHC

In the AHC algorithm, the data item clustering process is carried out by combining the initial clusters. Each appropriate pair of clusters is formed, and the degree of similarity is calculated. The cluster pair with the highest degree of similarity is selected and merged into one larger cluster. When combining cluster pairs into a single cluster, the number of clusters is reduced by one. The process is repeated until the desired number of clusters is reached.

There are various hierarchical clustering methods, including the single linkage method, complete linkage method, and average linkage method [16]. Single linkage measures cluster proximity based on the distance between the closest cluster members. Complete linkage measures cluster proximity based on the distance between the most distant cluster members. The average linkage measures cluster proximity based on the average distance between pairs of cluster members. This research used single linkage method.

5. Calculating the Silhouette Score

The silhouette score validates the cluster by considering two clustering criteria: separation (i.e., the average distance to the other closest cluster) and compactness (i.e., the average distance within the cluster) [17]. To calculate the silhouette score, it is first necessary to find the distance of objects in a cluster with all objects in neighboring clusters. As shown in equation (5).

$$b(i) = min\ c\ (i,C) \quad (5)$$

Suppose i is the object of concern that belongs to cluster A. C is a cluster that does not contain i. a(i) is defined as the average difference between i and all other objects in cluster A. While c(i,C) is the average difference between i and all objects in C. Thus, the silhouette score can be calculated using equation (6).

$$s(i) = (b(i) - a(i))/max\{a(i),b(i)\} \quad (6)$$

The best cluster is marked with a value close to 1 because it indicates that object i is closer to objects in the same cluster. Silhouette value 0 indicates that the distance of each cluster is not too significant, while the cluster value close to -1 indicates that the distance i to neighboring clusters is closer than the cluster itself.

To rigorously evaluate the quality of AHC clustering, we benchmarked the dataset against a supervised classification model (e.g., Naïve Bayes/SVM). The classification model serves as a baseline for 'learnability'. If the classification model performs well while the clustering model performs poorly, it proves that the bug reports contain distinguishable patterns, but the unsupervised distance measures (like Jaccard) cannot capture them without labeled guidance.

C. Testing and Evaluation

The test compared the results of clustering using AHC with two different distance calculations, namely, the Jaccard Coefficient and the Weighted Jaccard Coefficient. The test was conducted in the following stages:

1. The comparison test of cluster quality with word parameter unigram on the AHC model using Jaccard Coefficient and Weighted Jaccard Coefficient.

EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.9.1.6065. SUBMITTED: JANUARY 15, 2026; REVISED: FEBRUARY 9, 2026; ACCEPTED: APRIL 6, 2026

2. Comparison test of cluster quality with word parameter bigram on the AHC model using Jaccard Coefficient and Weighted Jaccard Coefficient.
3. Comparison test of cluster quality with word parameter n-gram = 3 on the AHC model using Jaccard Coefficient and Weighted Jaccard Coefficient.
4. Comparison test of cluster quality with word parameter n-gram = 4 on the AHC model using Jaccard Coefficient and Weighted Jaccard Coefficient.
5. Testing the quality of the cluster with word parameter unigram on the AHC model using Cosine, Euclidean, and Manhattan distance measurements.
6. Testing by running the classification model. The methods used are naive Bayes, logistic regression, random forest, and decision tree.

III. RESULT AND DISCUSSION

The testing process was carried out by clustering with the word unigram, bigram, n-gram = 3, n-gram = 4 parameters, and by measuring the distances of Cosine, Manhattan, and Euclidean. The test was conducted by displaying the silhouette score of each cluster formed.

As shown in Table II, the experimental results yielded low and occasionally negative silhouette scores. A negative silhouette score indicates that data points were assigned to the wrong cluster, suggesting significant overlap between bug report topics in the vector space.

TABLE II
Comparison of Algorithm Jaccard and Algorithm Weighted Jaccard

Method	Avg Silhouette score			
	Unigram	**Bigram**	**N-gram = 3**	**N-gram = 4**
Jaccard Coefficient	-0.74	-0.85	-0.73	-0.71
Weighted Jaccard Coefficient	-0.77	-0.74	-0.69	-0.71

The resulting silhouette score for all tests has a negative value. This indicates that the cluster is not well formed. For this reason, further testing is needed using other distance calculations. The distance calculations include Cosine, Euclidean, and Manhattan.

TABLE III
Comparison with Cosine, Euclidean, and Manhattan

Method	***Silhouette score***
Jaccard	-0.74
Weighted Jaccard	-0.77
Cosine	-0.15
Euclidean	0.03
Manhattan	0.17

Based on the test results shown in Table III, it can be seen that clustering using Cosine, Euclidean, and Manhattan distance measurements gives better results than the Jaccard and weighted Jaccard. Manhattan has the best value among the three distance measurement methods with a silhouette value of 0.34 and some clusters formed 2.

In addition, testing was conducted using the classification method, because the results of clustering tests for the bug report dataset showed an indication that a good cluster was not formed. Thus, it is suspected that the dataset is not suitable for the clustering process. The dataset also has a column that can be used as a label, namely Bug Location. The classification methods used are naive Bayes, logistic regression, random forest, and decision tree.

TABLE IV
Test results with the classification method

Method	**Accuracy**	**Precision**	**Recall**	**F1**
Naive Bayes	51.53	51.53	51.53	51.53
Logistic Regression	69.94	69.94	69.94	69.94
Random Forest	91.41	91.41	91.41	91.41
Decision Tree	92.64	92.64	92.64	92.64

Based on the test results using the classification method shown in Table IV, it can be seen that the decision tree shows the best performance with an accuracy value of 92.64 and F1 92.64 compared to other classification methods. From the test results, it can be concluded that the dataset is more suitable for the classification process.

IV. CONCLUSION

Results show that the weighted Jaccard slightly coefficient outperforms standard Jaccard in bigram and 3-gram settings. However, both

EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.9.1.6065. SUBMITTED: JANUARY 15, 2026; REVISED: FEBRUARY 9, 2026; ACCEPTED: APRIL 6, 2026

metrics ultimately produced low silhouette scores, indicating that weighting terms alone cannot overcome the sparsity of the data. Manhattan has the best value among all distance measurement methods with a silhouette value of 0.34 with some clusters formed 2.

All clustering results have a negative silhouette score and only two clusters are formed, which indicates that the clusters are not well formed. This is because the dataset has several unique documents that have no proximity value to any document. After testing with the classification method, it can be concluded that the dataset is more suitable for the classification method. The decision tree shows the best performance with an accuracy value of 92.64 and F1 92.64 for testing with a classification model.

V. REFERENCES

[1] N. Limsettho, H. Hata, A. Monden, and K. Matsumoto, “Unsupervised Bug Report Categorization Using Clustering and Labeling Algorithm,” International Journal of Software Engineering and Knowledge Engineering, vol. 26, no. 7, pp. 1027–1053, 2016, doi: 10.1142/S0218194016500352.

[2] V. Venkatesh and A. S. Vindhya, "Predicting the Accuracy of Fractionation of Patron's Activities in Online Social Networks Using Novel K-Means Clustering Algorithm Comparing with Agglomerative Hierarchical Clustering Algorithm," 2023 6th International Conference on Contemporary Computing and Informatics (IC3I), Gautam Buddha Nagar, India, 2023, pp. 2563-2566, doi: 10.1109/IC3I59117.2023.10397982.

[3] S. G. Jindal and A. Kaur, "Automatic Keyword and Sentence-Based Text Summarization for Software Bug Reports," in IEEE Access, vol. 8, pp. 65352-65370, 2020, doi: 10.1109/ACCESS.2020.2985222.

[4] M. I. Nawaz Tarar, F. Ahmed and W. H. Butt, "Automated Summarization of Bug Reports to speed-up software development/maintenance process by using Natural Language Processing (NLP)," 2020 15th International Conference on Computer Science & Education (ICCSE), Delft, Netherlands, 2020, pp. 483-488, doi: 10.1109/ICCSE49874.2020.9201846.

[5] S. Mukhtar, S. Lee and J. Heo, "A Multidocument Summarization Technique for Informative Bug Summaries," in IEEE Access, vol. 12, pp. 158908-158926, 2024, doi: 10.1109/ACCESS.2024.3487443.

[6] M. Wati, D. Adela and M. Jamil, "Implementation of Hierarchical Agglomerative Clustering Method to East Kalimantan Unemployment Analysis," 2023 IEEE 7th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Purwokerto, Indonesia, 2023, pp. 395-399, doi: 10.1109/ICITISEE58992.2023.10405078.

[7] P. Patel, B. Sivaiah and R. Patel, "Approaches for finding Optimal Number of Clusters using K-Means and Agglomerative Hierarchical Clustering Techniques," 2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP), Hyderabad, India, 2022, pp. 1-6, doi: 10.1109/ICICCSP53532.2022.9862439.

[8] S. Galdino and J. Dias, "Interval-valued Data Ward's Hierarchical Agglomerative Clustering Method: Comparison of Three Representative Merge Points," 2021 International Conference on Engineering and Emerging Technologies (ICEET), Istanbul, Turkey, 2021, pp. 1-6, doi: 10.1109/ICEET53442.2021.9659628.

[9] W. Dai and M. Zhang, "Hierarchical Agglomerative Clustering Optimization for Massive Data," 2024 IEEE Smart World Congress (SWC), Nadi, Fiji, 2024, pp. 2514-2519, doi: 10.1109/SWC62898.2024.00379.

[10] N. Anuar, N. K. K. Baharin, N. H. M. Nizam, A. N. Fadzilah, S. E. M. Nazri and N. M. Lip, "Determination of Typical Electricity Load Profile by Using Double Clustering of Fuzzy C-Means and Hierarchical Method," 2021 IEEE 12th Control and System Graduate Research Colloquium (ICSGRC), Shah Alam, Malaysia, 2021, pp. 277-280, doi: 10.1109/ICSGRC53186.2021.9515295.

[11] T. Yuan, J. Zhou and Y. Chen, "Identification of Industrial Process Time Series Events Based on Agglomerative Hierarchical Clustering*," 2023 China Automation Congress (CAC), Chongqing, China, 2023, pp. 6495-6499, doi: 10.1109/CAC59555.2023.10451693.

[12] F. -q. Meng, R. Huang and J. -d. Wang, "Clustering for Bug Triage Based on Developer Work Capabilities," 2025 11th International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, India, 2025, pp. 1743-1748, doi: 10.1109/ICCSP64183.2025.11088794.

[13] C. E. Laney, A. Barovic and A. Moin, "Automated Duplicate Bug Report Detection in Large Open Bug Repositories," 2025 IEEE 49th Annual Computers, Software, and Applications Conference (COMPSAC), Toronto, ON, Canada, 2025, pp. 450-458, doi: 10.1109/COMPSAC65507.2025.00065.

[14] S. Guo et al., “Developer Activity Motivated Bug Triaging: Via Convolutional Neural Network,” Neural Processing Letters, vol. 51, no. 3, pp. 2589–2606, Jun. 2020, doi: 10.1007/s11063-020-10213-y.

[15] C. Wu and B. Wang, “Extracting Topics Based on Word2Vec and Improved Jaccard Similarity Coefficient,” Proceedings - 2017 IEEE 2nd International Conference on Data Science in Cyberspace, DSC 2017, pp. 389–397, 2017, doi: 10.1109/DSC.2017.70.

[16] S. Bandyopadhyay and S. Saha, Unsupervised classification: Similarity measures, classical and metaheuristic approaches, and applications, vol. 9783642324512. Springer-Verlag Berlin Heidelberg, 2013. doi: 10.1007/978-3-642-32451-2.

[17] T. Jo, Text Mining: Concepts, Implementation, and Big Data Challenge, vol. 45. Cham: Springer International Publishing, 2019. doi: https://doi.org/10.1007/978-3-319-91815-0.