International Journal of Informatics and Data Science
Volume 1, No 2, June 2024, Page: 55−63
ISSN 3026-7315 (media online)
https://journals.adaresearch.or.id/ijids/index
DOI: doi.org/10.64366/ijids.v1i2.45

Clustering of YouTube Viewer Data Based on Preferences using Leiden Algorithm

Erlin Windia Ambarsari[1,*], Aulia Paramita[1], Desyanti[2]

[1]Faculty of Engineering and Computer Science, Informatics Engineering, Universitas Indraprasta PGRI, Jakarta, Indonesia
[2]Informatics Engineering, Sekolah Tinggi Teknologi Dumai, Riau, Indonesia
Email: [1]erlinunindra@gmail.com, [2] aulia.pps@gmail.com, [3]desyanti734@gmail.com
Correspondence Author Email: erlinunindra@gmail.com

Abstract−This study aims to analyze YouTube viewer engagement patterns by applying the Leiden algorithm for clustering based on user interactions such as likes, dislikes, and subscription behaviors in correlation with video duration. Therefore, the method that we used begins with data cleaning to ensure completeness, followed by selecting relevant features and applying z-score normalization to equalize their contributions. A similarity graph is constructed using cosine similarity, representing instances as nodes and their relationships as edges. The Leiden algorithm is then applied to optimize modularity and extract clusters, with results integrated into the original dataset for analysis. Dimensionality reduction using PCA facilitates cluster visualization, while statistical summaries and distribution plots provide deeper insights into cluster characteristics. Subsequently, we obtained a dataset sourced from the YouTube content creator @ArmanVesona, which includes 237 instances with ten features: Shares, Comments Added, Dislikes, Likes, Subscribers Lost, Subscribers Gained, Views, Watch Time (hours), Impressions, and Click-Through Rate (%). The analysis reveals two distinct clusters: Cluster 0, characterized by lower engagement and stable audience, and Cluster 1, exhibiting higher engagement but higher subscriber churn. The findings highlight the effectiveness of the Leiden algorithm in detecting well-connected communities and provide insights into viewer behavior, aiding in the development of improved content strategies and targeted marketing approaches.

Keywords: YouTube Viewer Engagement; Leiden Algorithm; Clustering; Viewer Behavior; Content Strategy

1. INTRODUCTION

Television, patented by Philo Farnsworth in 1928, reached its zenith in popularity from the 1950s to the 1980s, serving as the primary medium for entertainment and information dissemination. In Indonesia, the analog television broadcasting system transitioned to digital in November 2022, providing enhanced picture quality and more efficient frequency utilization [1]. As television technology evolved, television programs began to rely heavily on the TV Audience Measurement (TAM) system to quantify viewership. This rating system allows television producers to secure advertising revenue without prioritizing the quality of the broadcasted content.

However, the emergence of smartphones, which support social media platforms more flexibly and portably than television, has significantly altered the media landscape. This shift has also impacted YouTube, a video platform accessible anywhere via the internet. Consequently, even television celebrities have started creating their channels on YouTube. This platform enables direct interaction between viewers and creators through comments, likes, and shares, fostering a stronger and more engaged community.

Viewers can like, dislike, and subscribe to a video based on its duration. The like and dislike features enable viewers to express their opinions about a video while subscribing allows them to follow channels they enjoy. However, viewers may also unsubscribe if videos made by creators do not align with their preferences [2], which can affect monetization. Therefore, in this study, we analyze how viewer interactions change over time and how this affects the popularity and engagement of the channels.

To analyze YouTube viewer behavior, we utilized a clustering method. Several options were considered, including K-Means clustering [3]–[8], K-Medoids clustering [9], [10], DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [11]–[13], Fuzzy C-Means clustering [14]–[16], Agglomerative Hierarchical clustering [17], [18], the Louvain algorithm [19], [20], and the Leiden algorithm [21], [22]. Ultimately, we selected the Leiden algorithm to categorize viewers based on their interactions, such as likes, dislikes, and subscription behaviors, with video duration.

We chose the Leiden algorithm for several reasons based on previous studies. The study by [23] highlighted the advantages of the Leiden algorithm in providing more stable and higher-quality community detection results compared to the Louvain algorithm. The Leiden algorithm proved faster and more accurate by refining partitions and ensuring well-connected communities, especially for large-scale networks. This improvement is crucial for analyzing extensive datasets like YouTube viewer interactions, where the quality of community detection directly impacts the understanding of viewer behavior and preferences.

The study [24] demonstrated that the Leiden algorithm outperformed the Louvain algorithm in terms of execution time and the modularity of the detected communities. The Leiden algorithm's ability to manage dynamic and evolving networks was particularly emphasized, making it suitable for real-time social media analysis. This capability is essential for YouTube data, where viewer engagement and preferences can change rapidly.

Additional studies, such as the one by [21], further reinforce the Leiden algorithm's superiority in community detection tasks. The Leiden algorithm addresses shortcomings observed in earlier methods by identifying well-connected communities and ensuring convergence to locally optimal solutions. This robustness is particularly

Copyright © 2024 **Author's**, Page 55
This Journal is licensed under a Creative Commons Attribution 4.0 International License

International Journal of Informatics and Data Science
Volume 1, No 2, June 2024, Page: 55−63
ISSN 3026-7315 (media online)
https://journals.adaresearch.or.id/ijids/index
DOI: doi.org/10.64366/ijids.v1i2.45

beneficial when dealing with diverse datasets across different domains, including social, biological, and information networks.

A recent empirical study by [25] focused on the strengths and weaknesses of the Louvain and Leiden algorithms using modularity metrics. The study demonstrated that when executed iteratively, the Louvain algorithm often results in poorly connected communities. In contrast, the Leiden algorithm was designed to address these weaknesses, showing better modularity and running time performance. The study applied these algorithms to synthetic and natural networks, confirming that the Leiden algorithm consistently outperformed the Louvain algorithm.

The study by [26] explored the use of the Leiden and Louvain algorithms in delineating healthcare service areas, focusing on spatially constrained community detection. Their study highlighted that the Leiden algorithm, due to its iterative refinement and improved partitioning, produced more cohesive and better-defined service areas compared to the Louvain algorithm. This study's findings further substantiate the Leiden algorithm's applicability to various spatial and real-world networks, making it a robust choice for analyzing geographically distributed data.

Therefore, given the Leiden algorithm's strengths in handling large-scale, dynamic networks, its application to clustering YouTube viewer data based on preferences is well-founded. The ability to detect viewer communities based on interactions such as likes, dislikes, and subscription behaviors offers insights into viewer engagement patterns. By leveraging the Leiden algorithm, we can achieve a more accurate and efficient categorization of viewer data, facilitating improved content recommendations and more precise marketing strategies.

This study has a gap in the literature regarding applying the Leiden algorithm for clustering YouTube viewer interactions, particularly likes, dislikes, and subscription behaviors correlated with video duration. While the Leiden algorithm has been widely recognized for its ability to detect well-connected communities in various datasets, its application has largely focused on static or spatial datasets. There is limited exploration of how this algorithm performs in highly dynamic environments like YouTube, where viewer behaviors evolve rapidly and are influenced by multiple factors, including content type, duration, and creator engagement strategies.

Furthermore, existing applications of the Leiden algorithm often focus on optimizing technical metrics such as modularity or execution time, without addressing the practical implications for understanding user engagement patterns. Little attention has been given to how the algorithm can uncover meaningful insights about user interactions, particularly in digital platforms where engagement metrics (likes, dislikes, subscriptions) are closely tied to content strategies and monetization efforts.

Therefore, this study addresses this gap by applying the Leiden algorithm to analyze YouTube viewer interactions. Unlike existing studies, this research leverages the clustering results to interpret user engagement patterns concerning video characteristics, providing actionable insights for content creators and marketers.

2. RESEARCH METHODOLOGY

2.1 Research Stages

This study used a dataset from the YouTube content creator @ArmanVesona. The dataset comprises 237 instances and includes ten features: 'Shares,' 'Comments Added,' 'Dislikes,' 'Likes,' 'Subscribers Lost,' 'Subscribers Gained,' 'Views,' 'Watch Time (hours),' 'Impressions,' and 'Click-Through Rate (%).' Figure 1 illustrates the steps of the study:

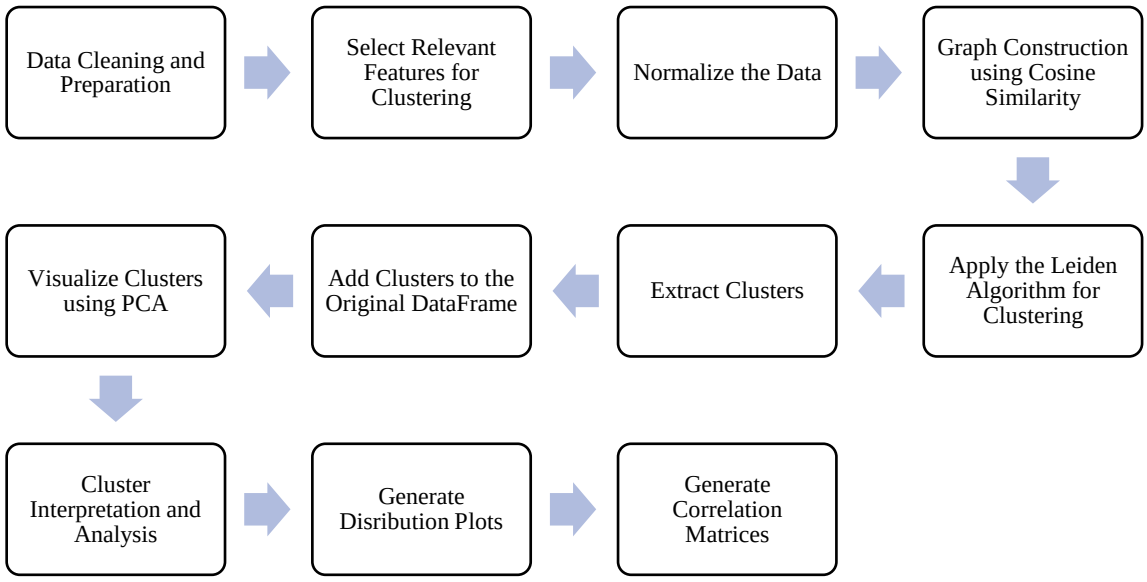


Figure 1. Study Stages for Clustering Construct and Analysis

Copyright © 2024 **Author's**, Page 56
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Volume 1, No 2, June 2024, Page: 55−63
ISSN 3026-7315 (media online)
https://journals.adaresearch.or.id/ijids/index
DOI: doi.org/10.64366/ijids.v1i2.45

The study stages in Figure 1 are detailed as follows:

a. Data Cleaning and Preparation
Data cleaning and preparation involve removing rows with missing values to avoid errors and inconsistencies. This step ensures that the dataset is complete and ready for further analysis.

b. Select Relevant Features for Clustering
Selecting relevant features involves choosing specific columns from the dataset pertinent to the clustering analysis. These features should significantly contribute to the variability in the data and help distinguish between different clusters. Therefore, we use the selected features to form a feature matrix and thus process them in subsequent steps.

c. Normalize the Data
We ensure that each feature contributes equally to the clustering process and the data using z-score normalization. The formula for z-score normalization is [26]:

$$z = \frac{x-\mu}{\sigma} \quad (1)$$

Where z is the standardized value, x is the original value, μ is the mean, and σ is the standard deviation of the feature.

d. Graph Construction using Cosine Similarity
We construct a graph using cosine similarity, which involves calculating the cosine similarity matrix S. Cosine similarity is used to measure angular similarity between feature vectors, making it particularly effective for high-dimensional data like YouTube viewer interactions, where diverse behaviors can be accurately represented. The formula for cosine similarity is [27]:

$$S_{ij} = \frac{X_i X_j}{\|X_i\|\|X_j\|} \quad (2)$$

Where S_{ij} is the cosine similarity between instances i and j, X_i, and X_j are the feature vectors, and $\| \|$ denote the Euclidean norm. This step transforms the data into a similarity graph where each node represents an instance, and edges represent the similarity between instances.

e. Apply the Leiden Algorithm for Clustering
We use the Leiden algorithm to identify clusters within the graph. It optimizes a quality function such as modularity [23]:

$$Modularity\ (Q) = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j) \quad (3)$$

Where A_{ij} is the adjacency matrix, k_i, and k_j are the degrees of nodes i and j, m is the total number of edges, and δ (c_i, c_j) is the Kronecker delta function indicating whether nodes i and j are in the same cluster. This algorithm iteratively refines the clusters to maximize modularity. The Leiden algorithm iteratively refines clusters to maximize modularity, ensuring well-connected groups. Its robustness and efficiency make it ideal for large-scale datasets like YouTube interactions.

f. Extract Clusters
Extracting clusters involves obtaining a membership vector C that indicates the cluster assignment for each instance:

$$C = \{c_1, c_2, \ldots, c_m\} \quad (4)$$

Where c_i is the cluster label, for instance, i, this step assigns each instance to a specific cluster based on the results of the Leiden algorithm.

g. Add Clusters to the Original DataFrame
Adding the cluster information to the original data frame involves appending the membership vector C to the dataset. We integrated the cluster assignments with the original data and made further analysis and visualization.

h. Visualize Clusters using PCA.
Principal Component Analysis (PCA) reduces the dimensionality of the data while preserving its variance. In this study, PCA aids in visualizing clusters generated by the Leiden algorithm, enabling a clearer interpretation of user engagement patterns. The PCA transformation is as follows [28]:

$$Z = XW \quad (5)$$

Where Z is the matrix of principal components, X is the data matrix, and W is the matrix of eigenvectors of the covariance matrix of X. PCA projects the data onto a new coordinate system with the most observable variances we captured in the first few principal components.

i. Cluster Interpretation and Analysis
Cluster interpretation involves computing summary statistics for each cluster, such as mean, median, and standard deviation. These statistics provide insights into the characteristics of each cluster, helping to understand

This Journal is licensed under a Creative Commons Attribution 4.0 International License

International Journal of Informatics and Data Science
Volume 1, No 2, June 2024, Page: 55−63
ISSN 3026-7315 (media online)
https://journals.adaresearch.or.id/ijids/index
DOI: doi.org/10.64366/ijids.v1i2.45

the distribution and central tendency of the features within each cluster. By analyzing these statistics, one can draw meaningful conclusions about the patterns and behaviors represented in the data.

j. Generate Distribution Plots
Distribution plots visualize the distribution of features within each cluster. It provides a graphical representation of the data distribution, aiding cluster interpretation.

k. Generate Correlation Matrices
Correlation matrices quantify the linear relationships between pairs of features. The Pearson correlation coefficient
r calculated as follows [29]:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}} \quad (6)$$

where r_{xy} measures the strength and direction of the linear relationship between variables x and y, and $\bar{x}$ and $\bar{y}$ are the means of x and y, respectively.

3. RESULT AND DISCUSSION

3.1 PCA and Distribution Plots

Based on Figure 1, we had the clustering results and PCA visualization, illustrated in Figure 2. We divided two clusters (cluster 0 and cluster 1) using the Leiden algorithm. The PCA plot reveals two distinct clusters, with Cluster 0 represented in purple and Cluster 1 in yellow. Analyzing the spread and separation of these clusters, we observe that Cluster 1 displays a wider spread along the PCA Two-axis, indicating higher variability within this cluster. Conversely, Cluster 0 is more concentrated, suggesting less variability. Notably, there is an outlier in Cluster 1 with a very high PCA One value, suggesting a video with extreme feature values.

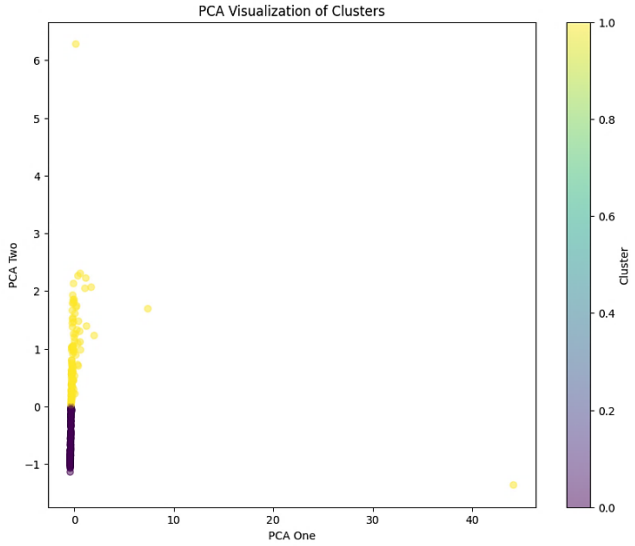
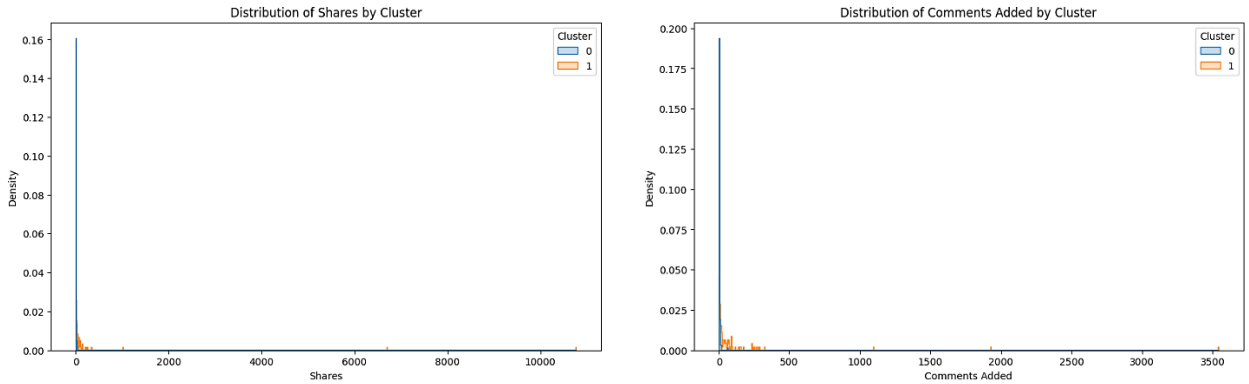


Figure 2. YouTube Viewer Data Clustering

Then, we added distribution visualization to understand clustering in Figure 2. The distribution involves ten features illustrated in Figure 3.



Copyright © 2024 **Author's**, Page 58
This Journal is licensed under a Creative Commons Attribution 4.0 International License

International Journal of Informatics and Data Science
Volume 1, No 2, June 2024, Page: 55−63
ISSN 3026-7315 (media online)
https://journals.adaresearch.or.id/ijids/index
DOI: doi.org/10.64366/ijids.v1i2.45

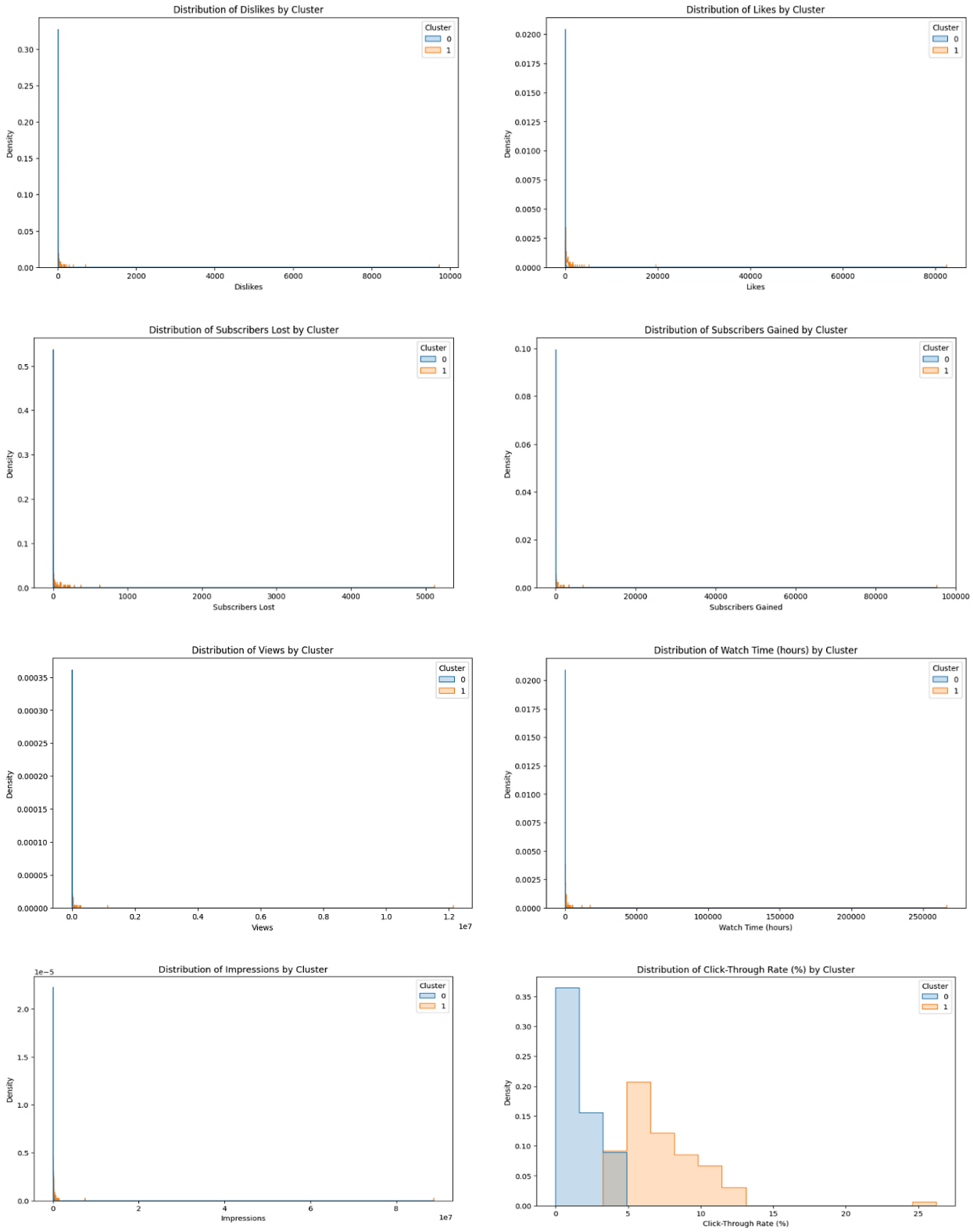


Figure 3. YouTube Viewer Distribution Based on Cluster

Based on Figure 3, we analyze the details as follows:

a. Distribution of Shares

The density plot of shares indicates that Cluster 0 exhibits a high-density peak at zero shares, suggesting that most videos in this cluster receive very few or no shares. This pattern highlights the low engagement level of content within this cluster. Conversely, Cluster 1 displays a broader distribution, with several videos achieving higher share counts. This observation suggests that Cluster 1 contains more shareable and engaging content, distributed more frequently among viewers.

Copyright © 2024 **Author's**, Page 59
This Journal is licensed under a Creative Commons Attribution 4.0 International License

International Journal of Informatics and Data Science
Volume 1, No 2, June 2024, Page: 55−63
ISSN 3026-7315 (media online)
https://journals.adaresearch.or.id/ijids/index
DOI: doi.org/10.64366/ijids.v1i2.45

b. Distribution of Comments Added
The analysis of the comments added reveals a similar trend. Cluster 0 shows a high-density peak at zero comments, indicating that most videos do not generate significant viewer interaction. In contrast, Cluster 1 shows a more varied distribution, with more videos receiving higher comment counts. This trend suggests that Cluster 1's content is more interactive and engages viewers in discussions, reflecting higher levels of viewer engagement.

c. Distribution of Watch Time (hours)
The distribution of watch time further differentiates the two clusters. Cluster 0 has a high density at low watch times, indicating shorter viewer retention. Videos in this cluster fail to hold the audience's attention for extended periods. On the other hand, Cluster 1 encompasses videos with longer watch times, indicating better viewer retention and more engaging content that captures and holds viewers' attention for longer durations.

d. Distribution of Views
The view distribution plot illustrates that Cluster 0 has a high density at lower view counts, signifying less popular content with limited reach. Cluster 1, however, displays a more dispersed distribution with higher view counts, indicating that the videos are more widely viewed and popular among audiences.

e. Distribution of Subscribers Lost
After analyzing the subscribers lost, Cluster 0 shows a high density with low subscriber loss values, indicating a stable audience base. Cluster 1 includes videos with higher subscriber loss, reflecting more polarizing content. Despite this, the higher engagement metrics in Cluster 1 suggest that the content, while polarizing, can also attract significant viewer interest.

f. Distribution of Subscribers Gained
Cluster 0 exhibits a high density at low subscriber gains, indicating limited audience growth. In contrast, Cluster 1 shows a broader distribution with higher subscriber gains, signifying that the content effectively attracts new viewers and expands the audience base.

g. Distribution of Impressions
The impression distribution shows that Cluster 0 has a high density at low impression counts, indicating limited visibility. Cluster 1, conversely, encompasses videos with higher impression counts, suggesting better reach and visibility. This trend is consistent with the higher engagement metrics observed in Cluster 1.

h. Distribution of Likes
The analysis of likes further distinguishes the clusters. Cluster 0 shows a high density at low like counts, indicating less popular content. Cluster 1 features a more varied distribution with higher like counts, reflecting more popular and well-received content.

i. Distribution of Dislikes
The dislike distribution reveals that Cluster 0 has a high density at low dislike counts, indicating less polarizing content. In contrast, Cluster 1 includes videos with higher dislike counts, suggesting more polarizing content that elicits stronger reactions from viewers.

j. Distribution of Click-Through Rate (%)
The click-through rate distribution highlights that Cluster 0 has a high density at low click-through rates, indicating less effective content in converting impressions to views. Cluster 1, however, shows a broader distribution with higher click-through rates, suggesting that the content is more effective in engaging viewers and converting impressions to views.

3.2 Correlation Matrices

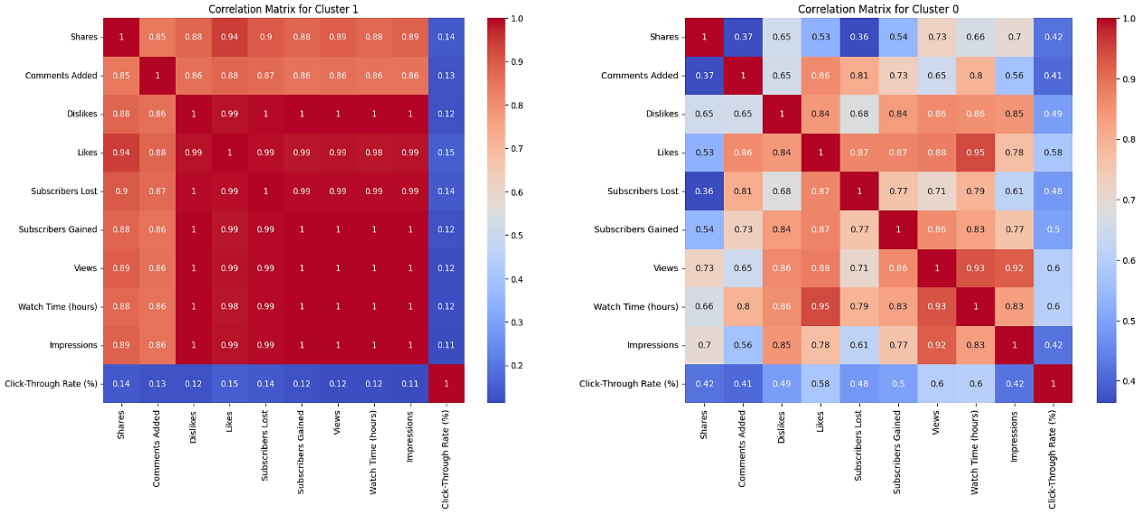


Figure 4. YouTube Viewer Correlation Matrices

Copyright © 2024 **Author's**, Page 60
This Journal is licensed under a Creative Commons Attribution 4.0 International License

International Journal of Informatics and Data Science
Volume 1, No 2, June 2024, Page: 55−63
ISSN 3026-7315 (media online)
https://journals.adaresearch.or.id/ijids/index
DOI: doi.org/10.64366/ijids.v1i2.45

We can interpret clusters based on Figure 4 as follows:

a. Correlation Matrix for Cluster 0

The correlation matrix for Cluster 0 elucidates the relationships between various metrics such as shares, comments, dislikes, likes, subscribers lost, subscribers gained, views, watch time, impressions, and click-through rate. Several significant correlations are evident within this cluster. Notably, a strong positive correlation between likes and watch time (0.95) indicates that videos with higher likes tend to retain viewers for longer durations. Similarly, the correlation between likes and views (0.88) suggests that videos receiving more likes attract higher view counts. Additionally, the relationship between subscribers gained and likes (0.87) highlights that gaining subscribers is closely associated with the number of likes a video receives.

Moderate correlations are observed between comments added and likes (0.86), suggesting that videos generating more comments also tend to receive more likes. The correlation between dislikes and likes (0.84) indicates that videos with more likes also receive more dislikes, possibly due to higher visibility and engagement. Furthermore, the correlation between views and impressions (0.92) underscores the effectiveness of visibility in attracting viewers, as more impressions typically lead to higher view counts. However, the click-through rate shows a relatively lower correlation with likes (0.58), implying that while there is some relationship, it is less pronounced than other metrics.

b. Correlation Matrix for Cluster 1

The correlation matrix for Cluster 1 reveals extremely high correlations across most metrics, indicating a highly interconnected relationship among the features within this cluster. Notably, there is an almost perfect correlation between likes and dislikes (0.99), suggesting that videos with high likes also receive high dislikes, potentially due to their higher visibility and polarizing nature. The correlation between subscribers gained and likes (0.99) further emphasizes that gaining subscribers is strongly related to the number of likes a video receives. Additionally, the perfect correlation between views and impressions (1.00) indicates that every impression almost directly translates to a view, reflecting the convenient nature of the content in this cluster.

Other significant correlations include the relationship between watch time and views (0.99), demonstrating that longer watch times are associated with higher view counts. The correlation between subscribers lost and likes (0.99) highlights that the video's reception highly influences changes in subscriber counts in terms of likes. However, the correlations between the click-through rate and other metrics (around 0.12-0.15) are relatively lower, indicating that while the click-through rate is essential, it is less strongly correlated with other engagement metrics within this cluster.

3.3 Discussion

Cluster 0 exhibits lower engagement across most metrics, including shares, comments, and likes. This cluster's content does not resonate strongly with the audience. The subscriber churn in Cluster 0 is low, suggesting that the content is not polarizing and does not negatively impact viewer retention. However, the growth within this cluster is limited, as reflected by the low number of subscribers gained. It indicates that the content is not attracting new viewers effectively. While impressions are moderate, both the click-through rate and view counts are low, suggesting that the content is impractical in converting impressions to views. The shorter watch times further indicate lower viewer retention, implying that the videos fail to maintain the audience's interest for extended periods. Therefore, Cluster 0 is characterized by less popular and less engaging content but a more stable audience.

In contrast, Cluster 1 demonstrates significantly higher engagement across all metrics, including shares, comments, likes, and views, indicating that the content resonates well with the audience. However, this cluster also experiences higher subscriber churn, suggesting that the content is polarizing and elicits strong reactions from viewers, both positive and negative. Despite the higher churn, the growth in this cluster is robust, with a high number of subscribers gained, demonstrating the content's ability to attract new viewers. High impressions coupled with a high click-through rate indicate that the content in Cluster 1 is effective in converting visibility into actual views. Additionally, longer watch times point to strong viewer retention, suggesting that the videos in this cluster can hold the audience's attention for extended periods. Thus, Cluster 1 reflects more popular, engaging, and widely viewed content despite being more polarizing.

Therefore, to enhance engagement and retention, it is recommended that the high engagement strategies observed in Cluster 1 be leveraged for content production. Content creators should continue producing similar content to maintain high engagement levels. However, efforts should be made to address the reasons for higher dislikes and subscriber loss in Cluster 1 to improve viewer satisfaction. Investigating the factors contributing to higher churn and dislikes can provide insights into improving content quality and viewer experience.

For Cluster 0, identifying and incorporating successful elements from Cluster 1 can help improve engagement. Experimenting with different content strategies, such as varying content formats, styles, and topics, can increase shares, comments, and likes. Enhancing content appeal and relevance can drive higher engagement and attract new subscribers.

Developing strategies to reduce churn in Cluster 1 while maintaining high engagement is crucial. Providing exclusive content can help retain subscribers and attract new ones. Understanding the factors driving subscriber dynamics and tailoring strategies to address these can significantly improve overall performance.

Copyright © 2024 **Author's**, Page 61
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Volume 1, No 2, June 2024, Page: 55−63
ISSN 3026-7315 (media online)
https://journals.adaresearch.or.id/ijids/index
DOI: doi.org/10.64366/ijids.v1i2.45

4. CONCLUSION

In this study, we employed the Leiden algorithm to analyze YouTube viewer engagement patterns based on preferences, explicitly focusing on likes, dislikes, and subscription behaviors correlated with video duration. Our primary objective was to evaluate the performance of the Leiden algorithm in detecting well-connected communities, interpret and analyze the resulting clusters, and visualize their distribution and correlation to inform better content recommendations and targeted advertising strategies. The results indicated that the Leiden algorithm effectively categorized YouTube viewer interactions into distinct clusters with varying engagement levels. Cluster 0, characterized by lower engagement metrics, represented content that did not resonate strongly with the audience. In contrast, Cluster 1 exhibited significantly higher engagement across all metrics, suggesting that the content in this cluster was more popular and widely viewed despite being more polarizing. The analysis revealed that Cluster 1's content had higher shares, comments, likes, and views, indicating that it was more engaging and retained viewers' attention for extended periods. However, this cluster also experienced higher subscriber churn, reflecting the polarizing nature of the content. Cluster 0 demonstrated lower engagement and limited audience growth but maintained a more stable viewer base. These findings highlight the effectiveness of the Leiden algorithm in identifying viewer communities based on their interactions, providing valuable insights into viewer engagement patterns. By leveraging these insights, content creators can enhance engagement and retention by adopting successful strategies observed in high-engagement clusters. Therefore, future research should further explore the application and optimization of the Leiden algorithm in various contexts of YouTube viewer data analysis. Comparative studies with other community detection algorithms, such as Louvain, can evaluate its performance across different scenarios. Integrating the Leiden algorithm with machine learning techniques, including deep learning for data preprocessing, could enhance clustering accuracy. Applying the algorithm to broader metrics like viewer demographics and geographic locations may provide deeper insights into viewer behavior. Longitudinal studies to analyze the algorithm's effectiveness over time and real-time implementation for quick response to emerging trends are promising areas. Exploring its use in different content types, such as live streams versus pre-recorded videos, could further tailor content strategies for improved viewer retention and satisfaction.

REFERENCES

[1] Afgiansyah, *Televisi vs Youtube, Benarkah TV Akan Mati? Kumpulan Esai Seputar TV di Era Digital*. Proxy Media, 2022.

[2] S.-G. Jung, J. Salminen, and B. J. Jansen, “The Effect of Hiding Dislikes on the Use of YouTube’s Like and Dislike Features,” in *14th ACM Web Science Conference 2022*, New York, NY, USA: ACM, Jun. 2022, pp. 202–207. doi: 10.1145/3501247.3531546.

[3] S. Rashid, A. Ahmed, I. Al Barazanchi, and Z. A. Jaaz, “Clustering algorithms subjected to K-mean and gaussian mixture model on multidimensional data set,” *Periodicals of Engineering and Natural Sciences*, vol. 7, no. 2, pp. 448–457, 2019.

[4] S. Sieranoja and P. Fränti, “Adapting k-means for graph clustering,” *Knowl Inf Syst*, vol. 64, no. 1, pp. 115–142, Jan. 2022, doi: 10.1007/s10115-021-01623-y.

[5] J. Baarsch and M. E. Celebi, “Investigation of internal validity measures for K-means clustering,” in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2012, pp. 471–476.

[6] E. Patel and D. S. Kushwaha, “Clustering Cloud Workloads: K-Means vs Gaussian Mixture Model,” *Procedia Comput Sci*, vol. 171, no. 2019, pp. 158–167, 2020, doi: 10.1016/j.procs.2020.04.017.

[7] I. Ramadhaniati, “Product Clustering using K-MEANS Method in CV. JAYA ABADI,” *Jurnal TAM (Technology Acceptance Model)* , vol. 14, no. 1, pp. 91–97, 2023.

[8] H. Humaira and R. Rasyidah, “Determining The Appropiate Cluster Number Using Elbow Method for K-Means Algorithm,” in *Proceedings of the 2nd Workshop on Multidisciplinary and Applications (WMA) 2018, 24-25 January 2018, Padang, Indonesia*, EAI, 2020. doi: 10.4108/eai.24-1-2018.2292388.

[9] T. Wang, Q. Li, D. J. Bucci, Y. Liang, B. Chen, and P. K. Varshney, “K-Medoids Clustering of Data Sequences With Composite Distributions,” *IEEE Transactions on Signal Processing*, vol. 67, no. 8, pp. 2093–2106, 2019, doi: 10.1109/TSP.2019.2901370.

[10] I. Fatma, H. S. Tambunan, and F. Rizki, “Analisis Metode K-Medoids Cluster Dalam Mengelompokkan Siswa Yang Berprestasi,” *Bulletin of Informatics and Data Science*, vol. 1, no. 1, 2022, [Online]. Available: https://ejurnal.pdsi.or.id/index.php/bids/index

[11] R. Adha, N. Nurhaliza, and U. Soleha, “Perbandingan Algoritma DBSCAN dan K-Means Clustering untuk Pengelompokan Kasus Covid-19 di Dunia,” *Jurnal Sains, Teknologi dan Industri*, vol. 18, no. 2, pp. 206–211, 2021, [Online]. Available: https://covid19.who.int.

[12] B. Nurina Sari, A. Primajaya, and J. H. Ronggowaluyo Teluk Jambe Karawang, “Penerapan Clustering DBSCAN Untuk Pertanian Padi di Kabupaten Karawang,” *Jurnal Informatika dan Komputer*, vol. 4, no. 1, pp. 28–34, 2019, [Online]. Available: www.mapcoordinates.net/en.

Copyright © 2024 **Author’s**, Page 62
This Journal is licensed under a Creative Commons Attribution 4.0 International License

[13] N. Selvia, E. Windia Ambarsari, and N. Dwitiyanti, "Shortest Path Clustering Dalam Menyaring Tingkat Kepadatan Arus Lalu Lintas," *JURIKOM (Jurnal Riset Komputer)*, vol. 10, no. 2, pp. 396–403, 2023, doi: 10.30865/jurikom.v10i2.5979.

[14] I. A. Atiyah, A. Mohammadpour, N. Ahmadzadehgoli, and S. M. Taheri, "Fuzzy C-Means Clustering Using Asymmetric Loss Function," *Journal of Statistical Theory and Applications*, vol. 19, no. 1, pp. 91–101, 2020, doi: 10.2991/jsta.d.200302.002.

[15] O. Amira, J.-S. Zhang, and J. Liu, "Fuzzy c-means clustering with conditional probability based K–L information regularization," *J Stat Comput Simul*, vol. 91, no. 13, pp. 2699–2716, Sep. 2021, doi: 10.1080/00949655.2021.1906243.

[16] K. V. Rajkumar, A. Yesubabu, and K. Subrahmanyam, "Fuzzy clustering and Fuzzy C-Means partition cluster analysis and validation studies on a subset of CiteScore dataset," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 4, pp. 2760–2770, 2019, doi: 10.11591/ijece.v9i4.pp2760-2770.

[17] F. AlMahamid and K. Grolinger, "Agglomerative Hierarchical Clustering with Dynamic Time Warping for Household Load Curve Clustering," in *2022 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2022, pp. 241–247. doi: 10.1109/CCECE49351.2022.9918481.

[18] M. Jafarzadegan, F. Safi-Esfahani, and Z. Beheshti, "An Agglomerative Hierarchical Clustering Framework for Improving the Ensemble Clustering Process," *Cybern Syst*, vol. 53, no. 8, pp. 679–701, Nov. 2022, doi: 10.1080/01969722.2022.2042917.

[19] M. -, A. B. Mutiara, S. Wirawan, T. Yusnitasari, and D. Anggraini, "Expanding Louvain Algorithm for Clustering Relationship Formation," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 1, 2023, doi: 10.14569/IJACSA.2023.0140177.

[20] J. C. Paolillo, S. Ghule, and B. P. Harper, "A Network View of Social Media Platform History: Social Structure, Dynamics and Content on YouTube," in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019, pp. 2632–2641. [Online]. Available: https://hdl.handle.net/10125/59701

[21] V. A. Traag, L. Waltman, and N. J. van Eck, "From Louvain to Leiden: guaranteeing well-connected communities," *Sci Rep*, vol. 9, no. 1, p. 5233, 2019, doi: 10.1038/s41598-019-41695-z.

[22] S. Güldal, "The Effect of Scoring Factor for Leiden Algorithm," *Afyon Kocatepe Üniversitesi Fen Ve Mühendislik Bilimleri Dergisi*, vol. 21, no. 3, pp. 559–564, 2021, doi: 10.35414/akufemubid.870835.

[23] S. H. H. Anuar *et al.*, "Comparison between Louvain and Leiden Algorithm for Network Structure: A Review," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Dec. 2021. doi: 10.1088/1742-6596/2129/1/012028.

[24] K. Sliwa, E. Kusen, and M. Strembeck, "A Case Study Comparing Twitter Communities Detected by the Louvain and Leiden Algorithms During the 2022 War in Ukraine," in *Companion Proceedings of the ACM on Web Conference 2024*, New York, NY, USA: ACM, May 2024, pp. 1376–1381. doi: 10.1145/3589335.3651892.

[25] S. H. Hairol Anuar, Z. Abal Abas, N. Md Yunos, M. F. Mukhtar, T. Setiadi, and A. S. Shibghatullah, "Identifying Communities with Modularity Metric Using Louvain and Leiden Algorithms," *Pertanika J Sci Technol*, vol. 32, no. 3, pp. 1285–1300, Apr. 2024, doi: 10.47836/pjst.32.3.16.

[26] M. R. Firmansyah, "Stroke Classification Comparison with KNN through Standardization and Normalization Techniques," *Advance Sustainable Science, Engineering and Technology*, vol. 6, no. 1, p. 02401012, Jan. 2024, doi: 10.26877/asset.v6i1.17685.

[27] D. M. Amin and A. Garg, "Performance Analysis of Data Mining Algorithms," *J Comput Theor Nanosci*, vol. 16, no. 9, pp. 3849–3853, Sep. 2019, doi: 10.1166/jctn.2019.8260.

[28] N. Salem and S. Hussein, "Data dimensional reduction and principal components analysis," *Procedia Comput Sci*, vol. 163, pp. 292–299, 2019, doi: 10.1016/j.procs.2019.12.111.

[29] J. J. Berman, "Understanding Your Data," in *Data Simplification*, Elsevier, 2016, pp. 135–187. doi: 10.1016/B978-0-12-803781-2.00004-7.

Copyright © 2024 **Author's**, Page 63
This Journal is licensed under a Creative Commons Attribution 4.0 International License