Research Paper

Journal of Engineering and Technological Sciences

Pixel Value Graphical Password Scheme: Analysis on Time Complexity performance of Clustering Algorithm for Passpix Segmentation

Mohd Sidek Fadhil Mohd Yunus*, Mohd Rizal Mohd Isa, Mohd Afizi Mohd Shukran, Norshahriah Wahab, Syarifah Bahiyah Rahayu & Amalina Farhi Ahmad Fadzlah

Department of Defence Science, National Defence University of Malaysia, Bangunan Lestari, Kem Perdana Sungai Besi, 57000 Kuala Lumpur, Malaysia

Corresponding author: sidek@upnm.edu.my

Abstract

Passpix is a key element in pixel value access control, containing a pixel value extracted from a digital image that users input to authenticate their username. However, it is unclear whether cloud storage settings apply compression to prevent deficiencies that would alter the file's 8-bit attribution and pixel value, causing user authentication failure. This study aims to determine the fastest clustering algorithm for faulty Passpix similarity classification, using a dataset of 1,000 objects. The source code for the K-Means, ISODATA, and K-Harmonic Mean scripts was loaded into a clustering experiment prototype compiled as Clustering.exe. The results demonstrate that the number of clusters affects the time taken to complete the clustering process, with the 20-cluster setting taking longer than the 10-cluster setting. The K-Harmonic Mean algorithm was the fastest, while K-Means performed moderately and ISODATA was the slowest of the three clustering algorithms. The results also indicate that the number of iterations did not affect the time taken to complete the clustering process. These findings provide a basis for future studies to increase the number of clusters for better accuracy.

Keywords: *access control; cybersecurity; image clustering; graphical password; k-means; pixel value.*

Introduction

In the era of Industry 4.0, particularly in the context of smart campus initiatives [1], cloud computing applications [2], integrated information systems [3], and mobile applications [4] are being developed at a rapid pace, often relying on a password authentication system. Pixel value access control (PVAC) [5] is a graphical password technique that utilizes a pixel value as the password. This password scheme is known as Password Pixel or *PassPix*,[6], where PVAC extracts the pixel value from a designated digital image file. An example of a login interface for PVAC is shown in Figure 1, which is almost identical to common login interfaces, except the password textbox has been replaced with a file browsing control.

In a cloud-based environment, PVAC can run on multiple server-based platforms. Cloud-based environments have become essential in daily life, providing flexibility and accessibility, especially during the COVID-19 pandemic [7]. Most industries have been forced to adopt cloud-based systems that allow for internet-based access to all files and systems. However, this trend can pose challenges for PVAC users storing their PassPix, as cloud storage conditions and settings are often unclear. Service providers may use compression to prevent storage insufficiencies, delays in service, and data processing consumption. Unfortunately, this can unintentionally alter the pixel values of digital images, resulting in authentication failure. Li et al. [8] found that these circumstances can particularly affect multimedia files, such as digital sound, digital video, and digital images, by changing the file's 8-bit attribution.

ISSN: 2337-5779

J. Eng. Technol. Sci. Vol. 55, No. 1, 2023, 52-59
DOI: 10.5614/j.eng.technol.sci.2023.55.1.6

DOI: 10.5614/j.eng.technol.sci.2023.55.1.6

Figure 1 The log-in interface of pixel value access control.

Uploading, storing, and sharing image files in messaging applications is a popular option for netizens since it is multi-platform and free. Most messaging applications apply file compression on every image file uploaded to the application repository. As a result, transfer and compression of the image file creates a pixel value difference when compared to the original image [9]. However, since the pixel value difference is within a reasonable range, we present a digital image clustering algorithm as a fault tolerance mechanism for the flawed *PassPix* issue problem in this research. In theory, digital image clustering algorithms may be integrated with pixel value access control, since both methods process the digital image computation, which includes the extraction of pixel values. Table 1 lists all the required clustering features needed to enable a fault tolerance mechanism for PVAC.

Table 1 Required clustering features for PVAC.

Requirements			Descriptions
Security	C1	Tolerable Range	An image that resides within the query range is considered as the same *PassPix.*
Features	C2	Feature Extraction	PVAC is employing the pixel-based extraction method, thus the DIC algorithm is selected from a pixel-based DIC technique.
	C3	Color-Space	As PVAC is working on RGB color-space, the DIC algorithm must be suited to RGB color-space.
	C4	Logical-Grid Extractions	To preserve the password space strength, the clustering data is analyzed in two-dimensional data.

However, by adding the clustering process to the PVAC flow increases time consumption for the log-in process compared to a common textual password process. Thus, this study aimed to find a clustering algorithm that would be able to process *PassPix* style data with minimal time consumption.

Background Works

A clustering algorithm is proposed as the fault tolerance mechanism for the faulty *PassPix* problem, which was the purpose of this research. A digital image clustering algorithm is needed that is capable to compute the pixel value difference as specific as possible in order to avoid deception of *PassPix* authentication. In other words, the range of recognition between the faulty *PassPix* and the actual *PassPix* is limited off from fake *PassPix* values. During the inquiry procedure, only a *PassPix* value that falls within an acceptable range is acknowledged as authentic *PassPix*.

Even though there are also edge-based digital image clustering techniques and region-based digital image clustering techniques [10], Panda, Hassanien & Abraham [11] stressed that pixel-based or point-based segmentation is among the more uncomplicated ones. Even though this technique can become less efficient with high brightness or darkness content, it is able to detect color density and isolate objects from the background. However, the detection or the pixel value extraction task has already been accomplished by the pixel value access control extraction module. Therefore, the digital image clustering algorithm only needs to process the extracted data, which is an advantage, as it saves a lot of time. A simple structured pixel-based segmentation technique that consumes the least computational resources and produces an admissible segmentation output is therefore preferred among researchers as well as content-based image retrieval (CBIR) developers.

Researchhers [12-15] that studied digital image clustering algorithms categorized all algorithms based on the 3V characteristics as distinguished by Oracle Big Data, i.e., volume, variety, and velocity. Based on the 3Vs, the clustering algorithms are categorized into five subcategories: (i) density-based digital image clustering algorithms, (ii) grid-based digital image clustering algorithms, (iii) hierarchical-based digital image clustering algorithms, (iv) model-based digital image clustering algorithms, and (v) partition-based digital image clustering algorithms. Each digital image clustering algorithm category employs a different approach for the constructing clusters and object membership assignment processes.

For certain content-based image retrieval systems, the ability to filter out outliers is an essential feature to exclude them from the query. But for pixel value access control, an object that is an outlier will result in the faulty *PassPix* remaining an inauthentic *PassPix*, where the matching record for that *PassPix* is unavailable in the query parameter. Based on this requirement, the K-Means clustering algorithm was found to be the perfect match for the Pixel Value Graphical Password scheme. The findings on K-Means based selection for this research are presented in Table 2.

Table 2 Summary of clustering strategies selection.

Clustering Strategies	**Density-based**	**Grid-based**	**Hierarchical-based**	**Model-based**	**K-means-based**
Feature(s) of Concerns	Outlier object	Query based on grid	Query based on linkage	Outlier for certain data patterns	(1) Absorb all objects (2) Query for object distance
Description	Objects that do not absorb into a cluster will be unavailable in the query parameter.	A whole object that resides in a related grid will be identified as similar.	Query performed up to smallest neighborhood. Every object that links to the neighbourhood is identified as similar.	Non-concentrated spot object density will result in outliers that are unavailable in the query parameter.	(1) All objects are absorbed into clusters leaving no objects to become outliers. (2) The query for object similarity is based on object distance.
As for PVAC…	(i) The outlier *PassPix* will become unavailable during the query.	(ii) An identical fake *PassPix* is recognized as similar to the authentic *PassPix*	Same as (ii).	Same as (i).	This clustering strategy is obviously suitable for PVAC pixel fault tolerance, where feature (2) complies with (C1)

DOI: 10.5614/j.eng.technol.sci.2023.55.1.6

Macqueen [16,17] initially proposed Basic K-Means to classify a set of objects into predetermined categories. When a new object is added to the dataset, he advises using an iteration process to re-determine category attributes such as partition and centroid position. Similar to the pixel value extraction function in pixel value access, K-Means contains a pixel value extraction function that converts a digital image to a string of pixel values.

There are three major K-Means variants based on theoretical findings and analysis of other research works, i.e., K-Means (basic), ISODATA and K-Harmonic Mean (KHM). Even though KHM and ISODATA were developed to enhance some basic features of K-Means, K-Means was theoretically found to be better than its successors for certain features [18,19]. KHM is believed to have slower object convergence than K-Means and ISODATA consumes more time to complete the clustering process than K-Means [20,21]. Besides, ISODATA requires the user to specify a value for seven parameters as compared to three parameters in K-Means and KHM.

There is still debate on the three algorithms, specifically about which one can perform computational queries with minimal time consumption from a theoretical perspective. Thus, we studied statistical data from experimental findings to find the most appropriate clustering algorithm to match the aim of this study.

Experiment

Experimental Setup

In a clustering algorithm experiment, the primary objective is to determine which clustering method can complete the process in the shortest time. Two statements remained inconclusive after our literature review, which led to the need for this experiment. The first statement concerns the ISODATA method's ability to reduce initialization computational time but not the overall clustering process, when compared to K-Means. The second statement relates to the slow object convergence using KHM, which can impact the entire KHM clustering computation.

These three algorithms were challenged with a dataset containing 1,000 objects (referred to as '1kD') to find which algorithm could complete the clustering faster than the others (ISODATA vs K-Means or KHM vs others). The experiment for clustering algorithm speed setup is presented in Figure 2.

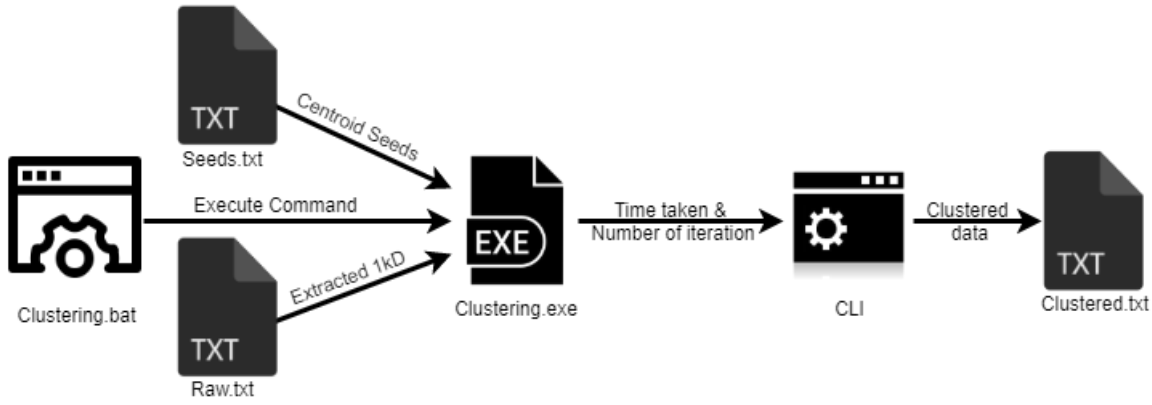

Figure 2 Clustering experiment setup.

The source code for K-Means, ISODATA and KHM script were written in C. Thus, the prototype used for the clustering experiment was compiled with Microsoft Visual Studio as a Microsoft Windows executable file to generate a program named Clustering.exe. It requires the parameters to be specified and then sends the command to execute Clustering.exe. It executes from the command line interface (CLI) console and a process summary appears on the console upon completion, showing the time taken and the number of iterations.

The clustering result is recorded in a file called Clustered.txt, which contains the centroid position and the list of objects sorted by cluster number. The application was developed with the capability to process multiple objects listed in a file called Raw.txt in just one session to replace the manual feed, which is tedious because the data needs to be fed one by one.

As Clustering.exe is a console-based program, the Clustering.bat file triggers Clustering.exe by sending the kick start command that contains the algorithm parameters, such as *k*, the seed file name, and the list of extracted 1kD together with the clustering algorithm mode (K-Means, ISODATA or KHM). Only a single algorithm mode is used in each clustering session, which means that each digital image clustering algorithm runs the clustering process in a separate session.

The centroid seed value for the seed parameter is obtained from either Seeds10.txt or Seeds20.txt, where Seeds10.txt clusters objects into ten categories and Seeds20.txt uses twenty categories. Overall, six sessions were performed in this experiment, as the three clustering algorithms were each tested with $k = 10$ and $k = 20$. The test was run in six sessions, recorded into six different text files called ClusteredKM10.txt, ClusteredKM20.txt, ClusteredISO10.txt, ClusteredISO20.txt, ClusteredKHM10.txt and ClusteredKHM20.txt, which also contain the clustering process time taken and the number of iterations.

Experiment Procedure

The clustering experiment procedure in six sessions is illustrated in Figure 3 and the step-by-step procedure is briefly explained in (1) to (6).

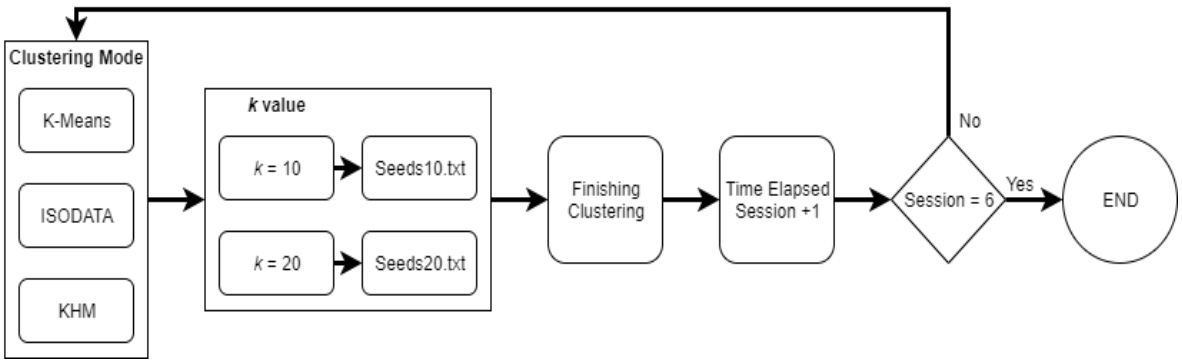

Figure 3 Flow of the clustering experiment procedure.

(1) The variable clustering mode in the Clustering.bat file is set to K-Means with $k = 10$ and the seeds are obtained from Seeds10.txt.
(2) Clustering.bat is set to execute Clustering.exe with the clustering settings from Clustering.bat and obtains the raw data from Raw.txt.
(3) When the clustering process is finished, the time taken, the number of iterations, and the object cluster ID are recorded as the results. This session is counted as +1.
(4) In the second session, procedure (1) is modified with $k = 20$ and the seeds are obtained from Seeds20.txt with the same clustering mode = K-Means and this session is counted as +1, which makes the current session number equal to 2.
(5) Procedures (2) to (3) are repeated with the settings mentioned in (4).
(6) Procedures (1) to (5) are repeated with the mode changed to ISODATA and KHM until the session number is equal to 6.

Result

The clustering process times taken and the number of iterations are presented in Table 3.

DOI: 10.5614/j.eng.technol.sci.2023.55.1.6

Table 3 Clustering time taken result.

Algorithms	*k* = 10, seeds = Seeds10.txt			*k* = 20, seeds = Seeds20.txt		
	Number of Clusters	Iterations	Time Taken	Number of Clusters	Iterations	Time Taken
K-Means	10	15	10 seconds	20	17	22 seconds
ISODATA	5	10	23 seconds	15	10	54 seconds
KHM	10	5	2 seconds	20	2	3 seconds

KHM was obviously the fastest digital image clustering algorithm with recorded time at 2 seconds with 5 iterations for 10 clusters (k = 10) and 3 seconds with 2 iterations for 20 clusters (k = 20). The number of clusters (k) was relatively affected by the time taken by all three algorithms to complete the clustering process, where each of them took more time for the k = 20 setting than for the k = 10 setting. Basic K-Means took 10 seconds for k = 10 and 120% additional time taken (12 seconds) for k = 20, which makes 22 seconds in total. ISODATA with the default settings took more time than K-Means to complete the clustering process, i.e., 23 seconds for k = 10 and 54 seconds (additional 93%) for k = 20.

The results shown in Figure 4 indicate that ISODATA took more time to complete the clustering process. Furthermore, it required an additional 9.3% for every k increment compared to K-Means, with an additional 12% for every additional k. KHM was found to be the most efficient digital image clustering algorithm, requiring only 11 seconds to complete the clustering process for k = 100. Another way to analyze the linear graph is through the slope (Θ) of the Pythagoras theorem.

As a result, the Θ of K-Means was 50.19º, ISODATA's Θ was 72.19º and KHM's Θ was 5.71 º. The Θ degree is an effective and accurate way to describe the relation between the number of k and the time taken, where the digital image clustering algorithm that can produce the lowest Θ value is the best in terms of time consumption against the increase of k. KHM is the least steep digital image clustering algorithm in terms of Θ compared to the other clustering algorithms, which leads to the conclusion that KHM has the lowest impact on time taken with the increase of k.

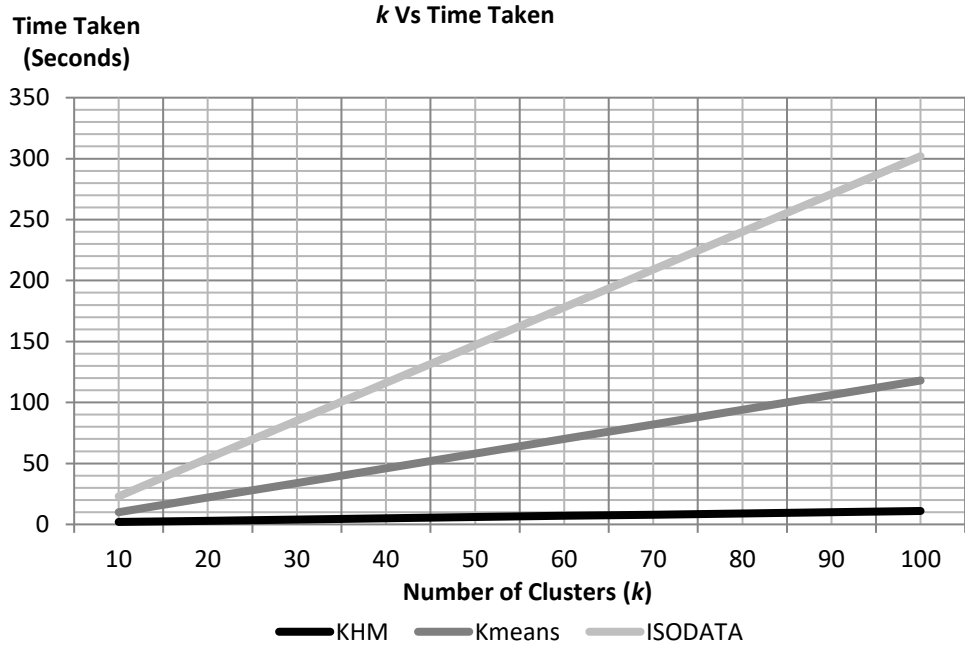

Figure 4 Line graph of k vs clustering time taken.

Clustering.exe provided another piece of data that allowed us to track the number of iterations taken during the clustering process for both values of *k*. However, we can conclude that the number of iterations did not affect the time taken for the digital image clustering algorithms to complete the process. In the case of K-Means with *k* = 20, the clustering process required seventeen iterations, compared to fifteen iterations for *k* = 10. This suggests that as the number of iterations increased, so did the time taken. However, the ISODATA algorithm exhibited a different pattern of the iterations-time taken relation, as both *k* values were completed with ten iterations, but the time taken increased with the increment of *k*. This finding is supported by the observation that the number of iterations was opposite to the time taken pattern for KHM, where only two iterations were required for *k* = 20 compared to five iterations for *k* = 10. These observations demonstrate that the number of iterations does not affect the time taken to complete the clustering process.

Conclusion

In summary, this study mainly focused on finding the least time-consuming clustering algorithm to be adopted as *PassPix* fault tolerance mechanism. This is of concern in order to avoid increasing the time consumption for the identity verification process when this mechanism is adopted in PVAC. It is concluded that KHM is the fastest digital image clustering algorithm, while K-Means is average, and ISODATA is the slowest among these three digital image clustering algorithms in completing the clustering process. Some researchers [22,23] disagree with ISODATA parameters, because using too many parameters for the DIC algorithm goes against the true purpose of unsupervised learning.

Even though the slope data of *k* versus time taken and *k* above 20 were not part of the scope of this research, it was shown that the value of *k* affects the time taken ratio of a digital image clustering algorithm in centroid based queries as proved and described in the clustering experiment.

The search for the fastest clustering algorithm has a great impact on PVAC, as one of PVAC's design aims was to reduce the complexity of the user login authentication process. A speedy clustering algorithm was used for the proposed *PassPix* fault tolerance mechanism to reduce the process complexity and time taken for handling a faulty *PassPix*. Due to several settings and restrictions imposed on current cloud storage infrastructure by service providers, it is almost inevitable for *PassPix* to stay immune to unintended pixel value alteration. In addition, the data on the value of Θ is of great help for using higher values of *k* for better accuracy in a future study. It is believed that the accuracy will increase if the number of objects is queried in a cluster that is more concentrated rather than a more varied number. The PVAC faulty tolerance similarity can then be reduced to only highly similar, which is can also reduce the success attempt rate for a wrong *PassPix* values.

Acknowledgement

Thanks go to Universiti Pertahanan Nasional Malaysia (UPNM) / National Defence University of Malaysia, especially the Faculty of Defence Science and Technology (FSTP) for their support and sponsorship of this study

References

[1] Mustafa, M.F., Isa, M.R.M., Rauf, U.F.A., Ismail, M.N., Shukran, M.A.M., Khairuddin, M.A. & Safar, N.Z.M., *Student Perception Study on Smart Campus: A Case Study on Higher Education Institution*, Malaysian Journal of Computer Science, **Special Issue 1**, pp. 1-20, 2021.

[2] Susandi, A., Tamamadin, M., Pratama, A., Faisal, I., Wijaya, A.R., Pratama, A.F. & Widiawan, D.A., *Development of Hydro-Meteorological Hazard Early Warning System in Indonesia*, Journal of Engineering & Technological Sciences, **50**(4), pp.461-478, 2018.

[3] Cafiso, S., di Graziano, A. & Pappalardo, G., *A Collaborative System to Manage Information Sources Improving Transport Infrastructure Data Knowledge*, Journal of Engineering & Technological Sciences, **51**(6), pp. 855-868, 2019.

[4] Navada, B.R. & Venkata, S.K., *Design of Mobile Application for Assisting Color Blind People to Identify Information on Sign Boards*, Journal of Engineering & Technological Sciences, **49**(5), pp.671-688, 2017.

DOI: 10.5614/j.eng.technol.sci.2023.55.1.6

[5] Shukran, M.A.M. & Yunus, M.S.F.M., *Method and System for Authenticating User Using Graphical Password for Access Control*, Malaysia Patent MY-167835-A, 4 September 2018. (Patent Technical Report)
[6] Yunus, M.S.F.M., Shukran, M.A.M. & Abdullah, M.N., *Pixel-Based Graphical Password Scheme: Password from Digital Image File*, Kuala Lumpur: UPNM Press, 2019.
[7] Reza, M.S., Hasan, A.K., Ahmed, A.S., Afroze, S., Bakar, M.S.A., Islam, S.N. & Darussalam, B., *COVID-19 Prevention: Role of Activated Carbon,* Journal of Engineering and Technological Sciences, **53**(4), pp. 627-638, 2021.
[8] Li, C., Bai, J., Yi, C. & Luo, Y., *Resource and Replica Management Strategy for Optimizing Financial Cost and User Experience in Edge Cloud Computing System*, in Information Sciences, **516**, pp. 33-55, 2020.
[9] Shukran, M.A.M., Yunus, M.S.F.M., Abdullah, M.N., Ismail, M.N. & Isa, M.R.M., *Pixel Value Graphical Password: A PassPix Clustering Technique for Password Fault Tolerance*, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, **8**(3), pp. 2973-2975, 2019.
[10] Yuheng, S. & Hao, Y., *Image Segmentation Algorithms Overview*, arXiv preprint arXiv:1707.02051, 2017.
[11] Panda, M., Hassanien, A.E. & Abraham, A., *Hybrid Data Mining Approach for Image Segmentation Based Classification*, Biometrics: Concepts, Methodologies, Tools, and Applications, pp. 1543-1561, August 2016.
[12] Sajana, T., Rani, C.S. & Narayana, K.V., *A Survey on Clustering Techniques for Big Data Mining*, Indian journal of Science and Technology, **9**(3), pp. 1-12, 2016.
[13] Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A.Y., Foufou, S. & Bouras, A., *A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis*, IEEE Transactions on Emerging Topics in Computing, **2**(3), pp. 267-279, 2014.
[14] Cai, Z., Yang, X., Huang, T. & Zhu, W., *A New Similarity Combining Reconstruction Coefficient with Pairwise Distance for Agglomerative Clustering*, Information Sciences, **508**, pp. 173-182, 2020.
[15] Oracle Inc., *Oracle Big Data*, Oracle Inc., https://www.oracle.com/big-data/guide/what-is-big-data.html. (7 April 2018).
[16] MacQueen, J., *Some Methods for Classification and Analysis of Multivariate Observations*, in Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967.
[17] Garg, N., & Gupta, R.K., *Exploration of Various Clustering Algorithms for Text Mining*, Int. Educ. Manag. Eng., **4**, pp. 10-18, 2018.
[18] Dunn, J.C., *A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters*, pp. 32-57, 1973.
[19] Nayini, S.E.Y., Geravand S. & Maroosi, A., *A Novel Threshold-Based Clustering Method to Solve K-Means Weaknesses, in 2017 International Conference on Energy*, Communication, Data Analytics and Soft Computing (ICECDS), 2017.
[20] Ayech, M.W. & Ziou, D., *Terahertz Image Segmentation Using K-Means Clustering Based on Weighted Feature Learning and Random Pixel Sampling*, Neurocomputing, **17**, pp. 243-264, 2018.
[21] Khanmohammadi, S., Adibeig, N. & Shanehbandy, S., *An Improved Overlapping K-Means Clustering Method for Medical Applications*, Expert Systems with Applications, **67**, pp. 12-18, 2017.
[22] Jin, X. & Han, J., *K-Means Clustering*, in Encyclopedia of Machine Learning and Data Mining, pp. 695-697, 2017.
[23] Wahab, N. S., Rusiman, M.S., Mohamad, M., Azmi, N.A., Him, N.C., Kamardan, M.G. & Ali, M., *A Technique of Fuzzy C-Mean in Multiple Linear Regression Model toward Paddy Yield*, Journal of Physics: Conference Series, **995**(1), 012010, 2018.

Manuscript Received: 24 October 2022
Revised Manuscript Received: 2 January 2023
Accepted Manuscript: 3 January 2023