MALAYSIAN JOURNAL OF SCIENCE
MJS
ISSN 1394 - 3065

5th International Conference on Applied & Industrial Mathematics and Statistics 2025 (ICoAIMS 2025)

Special Issue

Malaysian Journal of Science 45 (Special Issue 2): 83-94(November 2026) https://mjs.um.edu.my

Simulation Study of Agglomerative Clustering Algorithms for Detecting Outliers in Univariate Circular Data

Nur Syahirah Zulkipli[1a], Siti Zanariah Satari[2a*] and Wan Nur Syahidah Wan Yusoff[3a]

Abstract: Outlier detection techniques for univariate circular data are distinct from those designed for bivariate or multivariate contexts due to differences in data structure and dimensionality. In earlier research, clustering-based procedures specifically those utilizing agglomerative algorithms have proven effective for detecting outliers in bivariate circular datasets. This study aims to evaluate the performance of agglomerative clustering algorithms such as single-linkage (SL), complete-linkage (CL) and average-linkage (AL) by conducting a simulation study for a clustering-based outlier detection (COD) procedure for univariate circular data. The performance of the clustering algorithms from COD procedure assessed across various simulated data conditions under a specific outlier scenario. In the simulation study, outliers are embedded within synthetic datasets using a defined outlier scenario, where the mean direction is shifted by varying degrees of contamination to distinguish outliers from inliers. The simulation study revealed that CL-Satari/Di algorithm has the best performance on detecting outliers correctly with minimal masking effect under specific data conditions. Meanwhile, the performance of SL-Satari/Di algorithm found to be the best algorithm with low swamping effect even though not performing well in detecting correct outliers. This result also revealed that the concentration parameter, percentage of outliers, and sample size affect the performance of the clustering algorithms. The findings demonstrate that the algorithms effectively detect outliers with low masking and swamping effects when the concentration parameter is large. The simulation results also found that the algorithms performed well in detecting outliers when percentage of outliers is small with low masking effect but producing high swamping effect. Moreover, large sample size was found to be less sensitive to masking effect, while a smaller sample size is less sensitive to swamping effect. This work extends existing research on identifying outliers within univariate circular datasets by employing a clustering-based procedure as its core framework.

Keywords: Circular data, Agglomerative clustering algorithms, Outlier detection, Outlier scenario, Simulation study.

1. Introduction

Numerous techniques have been established for identifying outliers in univariate linear data. However, approaches specific to univariate circular data have received comparatively less attention. This is primarily due to the fundamental differences in data structure, as the strategies used for linear data are not directly applicable to circular data owing to its periodic and bounded nature. Detecting outliers in circular data is crucial for accurate analysis with the suitable procedure for both univariate and multivariate datasets. Outliers can cause problems such as in modelling and forecasting because they affect how parameters are estimated (Ramlee et al., 2024; Sert & Kardiyen, 2023; Satari et al., 2021; Satari & Ku Khalif, 2020). In circular data, the concept of an outlier as an extreme value is not valid due to its closed and bounded range (Abuzaid, 2020). Since circular data has a different geometric structure from linear data, detecting outliers in it needs special methods (Sert & Kardiyen, 2023). In some cases, such as in medical research or fraud detection, outliers may represent important discoveries rather than errors. Therefore, Mahmood et al. (2017a) suggested using robust statistical methods or adjusting outliers using clean data points, instead of simply removing or ignoring them.

Several studies have explored the detection of outliers in univariate circular data. Researchers such as Mahmood et al. (2017b), El et al. (2019), Abuzaid (2020), and Rattanawong et al. (2023) have proposed various methods specifically designed for this purpose. Among these, the row deletion technique has been widely adopted and frequently cited in the literature (Collett, 1980; Abuzaid et al., 2009; Hung et al., 2012; Mahmood et al., 2017b). However, a key limitation of these approaches is their focus on detecting only one or a few outliers at a time (Abuzaid, 2020; Satari & Ku Khalif, 2020). Sebert et al. (1998) highlighted the usefulness of clustering in outlier detection, noting that clustering algorithms can group outlying observations into a distinct cluster. In particular, agglomerative clustering has been recognized for its sensitivity to outliers and commonly used for identifying the outliers.

Satari (2015) demonstrated that agglomerative clustering could effectively detect multiple outliers by introducing a clustering-based approach specifically for bivariate circular data. Later, Di et al. (2017a, 2017b) extended the method using single-linkage clustering for circular-circular regression models, incorporating novel circular similarity measures along with a

Authors information:
[a]Pusat Sains Matematik, Universiti Malaysia Pahang Al-Sultan Abdullah, Lebuh Persiaran Tun Khalil Yaakob, 26300 Kuantan, Pahang, MALAYSIA. E-mail: syahirahzulkipliwork@gmail.com[1], zanariah@umpsa.edu.my[2], wnsyahidah@umpsa.edu.my[3]

*Corresponding Author: zanariah@umpsa.edu.my

Received: October 28, 2025
Accepted: May 6, 2026
Published: November 15, 2026

DOI: https://doi.org/10.22452/mjs.vol45sp2.9
Malaysian Journal of Science 45 (Special Issue 2): 83-94(November 2026)

stopping rule proposed by Satari (2015). Furthermore, Satari et al. (2019) evaluated the applicability of the single-linkage algorithm for Down and Mardia circular-circular regression models under three different outlier scenarios. This line of research was further extended by Satari et al. (2021) to compare several agglomerative clustering algorithms for detecting multiple outliers in the Down and Mardia circular-circular regression models. This study motivated to compare single-linkage, complete-linkage and average-linkage algorithms using various circular distances, with performance evaluated through simulations across different outlier scenarios, concentration levels, sample sizes, and contamination degrees.

However, the existing clustering-based outlier detection procedures for bivariate circular data were not directly applicable to univariate circular data due to differences in dimensional structure and data generation processes. Recognizing this limitation, Satari and Ku Khalif (2020) highlighted the need to develop a new clustering-based method specifically tailored for univariate circular data. In response, Zulkipli et al. (2024a) proposed and successfully developed the clustering-based outlier detection (COD) procedure and applied to the real-life univariate circular biological data. Accordingly, this study aims to assess the performance of algorithms within the COD framework under a range of data conditions through simulation. To perform the simulation study, the location of outlier will be set using a simulated data generation procedure with a specific outlier scenario formulation. As defined by Sebert et al. (1998), an "outlier scenario" refers to how outlying data points are positioned relative to the core (or “clean”) observations. In the context of univariate circular data, outliers are typically introduced by altering the mean direction, which is a key parameter (Collett, 1980; Abuzaid et al., 2012; Rambli et al., 2012). This study adopts Outlier Scenario 1 (OS1), introduced by Zulkipli et al. (2022), to simulate data under various experimental conditions. At the end of this study, the most effective clustering algorithms and the optimal confidence interval for the stopping rule will be identified, along with the best performing data conditions such as sample size, concentration parameter, and percentage of outliers.

2. Methodology

2.1 Clustering-Based Outlier Detection (COD) Procedure for Univariate Circular Data

Several studies have utilized clustering-based procedure to identify outliers in bivariate circular data, as widely reported in the literature. In this study, the clustering-based procedure for univariate circular data namely COD procedure are outlined in three stages. As a first step, appropriate circular similarity measures were employed to compute the dissimilarity between observations for each clustering algorithm. The similarity distance is calculated by using Satari, Di and Chang distances. According to Zulkipli et al. (2022), the similarity measures produced by the Satari and Di distances are equivalent when dealing with univariate circular datasets. Hence, these circular distance measures are collectively denoted as the Satari/Di distance throughout the study. The formulas for the Satari/Di and Chang-chien similarity distances are given in Eq. 1 and Eq. 2.

Satari/Di distance (Zulkipli et al., 2022):

$$d_{ij(Satari)} = \sum_{k=1}^{p}\left(\pi - \left|\pi - \left|\theta_{ik} - \theta_{jk}\right|\right|\right) \quad (1)$$

Chang-chien distance (Hung et al., 2012):

$$d_{ij(Chang-Chien)} = \sqrt{\sum_{k=1}^{p}\left(1-\cos(\theta_{ik} - \theta_{jk})\right)^2}. \quad (2)$$

At the second stage, the agglomerative clustering algorithms are chosen which are single-linkage (SL), complete-linkage (CL), and average-linkage (AL) algorithms to detect the outliers in univariate circular data. The *N* observations are clustered by using SL, CL and AL clustering algorithms and the following are the steps for grouping the observations given by Johnson and Wichern (2014):

Step 1: It starts with *N* clusters and each cluster contain a single observation, $\theta_1, \theta_2, \ldots, \theta_n$.

Step 2: Calculate the distance matrix between all possible pairs of clusters using circular distance in Eq. 1 and Eq. 2, respectively according to circular similarity distance.

Step 3: Find the smallest value of any $d_{\theta_i\theta_j}$ (distance between θ_i and θ_j) in distance matrix to calculate the nearest members and merge the corresponding observations, let say θ_1 and θ_2 are the nearest members and now they formed a cluster. The distance between cluster (θ_1, θ_2) and the other cluster or single observation, $\theta_1, \theta_2, \ldots, \theta_n$ will be calculated. Let a single observation (θ_3) be a cluster and the distance between cluster (θ_1, θ_2) and (θ_3) can be computed as given in Eq. 3 to Eq. 5 for SL, CL and AL, respectively. The distance of $d_{\theta_1\theta_3}$ and $d_{\theta_2\theta_3}$ are referred from the distance matrix in Step 2.

Single-linkage (SL):

$$d_{(\theta_1,\theta_2)\theta_3} = \min\left\{d_{\theta_1\theta_3}, d_{\theta_2\theta_3}\right\}. \quad (3)$$

Complete-linkage (CL):

$$d_{(\theta_1,\theta_2)\theta_3} = \max\left\{d_{\theta_1\theta_3}, d_{\theta_2\theta_3}\right\}. \quad (4)$$

Average-linkage (AL):

$$d_{(\theta_1,\theta_2)\theta_3} = \frac{\sum_i \sum_k d_{ik}}{N_{(\theta_1\theta_2)} N_{\theta_3}}, \quad (5)$$

where $N_{(\theta_1\theta_2)}$ and N_{θ_3} are the number of observations in clusters $(\theta_1\theta_2)$ and θ_3.

Step 4: The rows and columns in the distance matrix responding to the merged cluster(s) are deleted. Now, a new distance matrix is updated by re-calculating the distance matrix after forming new cluster in Step 3.

Step 5: Repeat Step 3 and Step 4 if there are more than one cluster remain in the dendrogram until all the clusters are merged into a single cluster.

DOI: https://doi.org/10.22452/mjs.vol45sp2.9
Malaysian Journal of Science 45 (Special Issue 2): 83-94(November 2026)

Next, for the third stage of COD procedure, dendrogram is generated as the output to visualize the clusters and stopping rule. The dendrogram serves to identify the optimal number of clusters, requiring a "cut" at a specific height to distinguish outlying observations from the main data cluster. The stopping rule can be calculated according to the different level of confidence interval as mention in Satari (2015). The specific multipliers λ_P are displayed in Satari (2015) for a selection of possibly *P*-values such as 0.90 and 0.95. Given the appropriate values of λ_P and their range of validity in terms of κ as in Eq. 6.

$$\begin{aligned} P=0.90,\ \lambda_P=1.69,\quad \kappa\geq 0.65\ \ (\rho\geq 0.31)\\ P=0.95,\ \lambda_P=2.06,\quad \kappa\geq 0.80\ \ (\rho\geq 0.37)\end{aligned} \tag{6}$$

Purposely, Satari (2015) choose a specific multiplier $\lambda_P=2.06$ and this study also utilizes 90% confidence interval by using multiplier $\lambda_P=1.69$. Thus, given the stopping rule used in this study as in Eq. 7 and Eq. 8.

$$\bar{h}+1.69s_h \tag{7}$$

$$\bar{h}+2.06s_h \tag{8}$$

The Eq. 7 and Eq. 8 show that $\bar{h}$ is the mean of the heights for all $N-1$ clusters, *N* is a total number of cluster which contain single observations and s_h is the circular standard deviation of the heights given by Eq. 9.

$$s_h=\sqrt{-2\log \bar{R}_h} \tag{9}$$

where $\bar{R}_h$ is the mean resultant length of the height for $N-1$ clusters. Thus, any cluster that exceeds the threshold defined by the stopping rules in Eq. 7 and Eq. 8 will be classified as a candidate outlier group. In summary, six clustering algorithms are employed in this study for identifying outliers in univariate circular dataset, as outlined in Table 1. Each COD algorithm is named based on the agglomerative hierarchical clustering and the circular similarity measures. For instance, CL-Satari/Di denotes the use of complete-linkage clustering combined with the Satari/Di distance.

Table 1. Clustering algorithms and stopping rule to identify outliers

Clustering method	Circular similarity measures	Algorithm name	Stopping rule
Single-linkage (SL)	Satari/Di distance	SL-Satari/Di	90% and 95%
	Chang-chien distance	SL-Chang	
Complete-linkage (CL)	Satari/Di distance	CL-Satari/Di	
	Chang-chien distance	CL-Chang	
Average-linkage (CL)	Satari/Di distance	AL-Satari/Di	
	Chang-chien distance	AL-Chang	

2.2 Outlier Scenario and Simulated Data Generation

In this study, simulated data for univariate circular data is generated with a specific outliers scenario. A simulated data is generated from all combinations of different sample size, n and concentration parameter, κ. To create the simulated data, the "clean" observations or called inliers were randomly generated from von Mises distribution, $VM(\mu,\kappa)$ with mean direction is fixed at 0. Seven values of concentration parameter, $\kappa=1.5,2,5,10,30,50,100$ with four sample size, $n=15,30,50,100$ are considered to generate the simulated univariate circular dataset. The outliers will be introduced in the dataset by 5%, 10%, 15% and 20%. In this study, the outliers will be placed by shifting the specific parameter which is mean direction, μ. The outliers were placed away from inlying observations at specified distance by shifting the mean direction with different degree of contamination, λ in the range of $0\leq\lambda\leq 1$ which defined by Eq. 10,

$$\bar{\theta}_{out}=\mu+\lambda\pi \tag{10}$$

where the mean direction, μ is fixed with 0. The formulation of outlier scenario in Eq. 10 namely as Outlier Scenario 1 (OS1) proposed by Zulkipli et al. (2022). Six contamination levels of $\lambda=0.0,0.2,0.4,0.6,0.8,1.0$ will be set to the simulated data with OS1. The procedure of generating the synthetic dataset with OS1 is briefly explained as follows:

i) Inliers are randomly generated from $VM(\mu,\kappa)$ with the mean direction, $\mu=0$ and the concentration parameter is set at $\kappa=1.5,2,5,10,30,50,100$.

ii) The outliers with 5%, 10%, 15% and 20% are randomly generated from $VM(\mu+\lambda\pi,\kappa)$, where the contamination value, $\lambda=0.0,0.2,0.4,0.6,0.8,1.0$. The mean direction, μ and the concentration parameter, κ are fixed according to condition used in (i).

iii) Inliers and outliers are combined together in one dataset called simulated dataset.

The simulated data is generated for large sample size up to $n=100$ and it is large enough for univariate circular data since this study are not focusing on big data. Moreover, the run time for simulating sample size more than 100 is time consuming. It takes a few days to run the simulation for sample size, $n>30$. The combination of data conditions to generate simulated dataset for simulation study are summarized in Table 2 and there will be 672 simulated datasets with OS1 are generated. In the Results and Discussion section, only selected data conditions will be presented to illustrate key findings and insights.

Table 2. Combination of data conditions and total synthetic datasets generated for simulation study

Sample size, n	Concentration parameter, κ	Percentage outliers	Contamination level, λ
15	1.5	5%	0.0
30	2	10%	0.2
50	5	15%	0.4
100	10	20%	0.6
	30		0.8
	50		1.0
	100		

DOI: https://doi.org/10.22452/mjs.vol45sp2.9
Malaysian Journal of Science 45 (Special Issue 2): 83-94(November 2026)

2.3 Simulation Study Design and Performance Indicator

A simulation study will be conducted using Python to assess the performance of the COD procedure in detecting outliers within univariate circular data. The number of outliers denoted as *out* will be introduced into each simulated dataset is determined by

$$out = out\%(n) \tag{11}$$

where $out\%$ is the percentage of outliers and n is sample size. The COD procedure is simulated for 5000 times for each outlier scenario to investigate its performance. In each iteration for simulation study, the six algorithms stated in Table 1 will be performed to identify the outliers from generated synthetic dataset. Three performance indicators defined by Sebert et al. (1998) will be used in this study to evaluate the performance of six algorithms in Table 1 according to each pair of condition in simulation study. The formulas of "success" probability as well as masking and swamping effect as performance indicators are displayed in Table 3.

Note that *s* is the total number of simulations. The closer the *pout* value to 1, the more successful the proposed COD procedure is at identifying all outliers, indicating that no masking has occurred. To make sure that the clustering algorithms from COD procedure performs very well, the masking rate need to be low. Masking becomes a more critical issue than swamping when the primary goal of the analysis is to identify influential subsets of observations. Meanwhile, *pswamp* occurs when an inlier is classified as an outlier by the clustering algorithms from COD procedure. Hence, this will be noted as "*false alarm"*. Similar with *pmask*, low *pswamp* values is considered to be good performance for the clustering algorithms from COD procedure to detect outliers.

3. Result and Discussion

3.1 Simulation Results

The simulation results as measured by the three performance indicators; *pout, pmask* and *pswamp* are presented in Table 4 to Table 6, respectively. The "success" probability of detecting all outliers (*pout*) result as shown in Table 4 found that as contamination level, λ increases, the *pout* also increases. It indicates that the probability of the procedures successfully identified all outliers is high as λ is approaching to 1. The results in Table 4 also show that the *pout* values are decreasing for all algorithms when the percentage of outliers are increasing. Moreover, the *pout* values are increasing when κ getting larger and all algorithms produce similar pattern. However, the *pout* values slightly decrease as n increase for most data conditions. A similar pattern was observed across all algorithms.

The results of masking effect (*pmask*) for OS1 are presented in Table 5. The results of *pmask* found that as λ increase, the *pmask* values gradually decrease and approaching to zero which indicates the algorithm is well-performed since there are no outliers falsely detected as inliers. The *pmask* values also decreasing as the value of κ getting larger for most data conditions. Meanwhile, the *pmask* values slightly increase as the percentage of outliers increase especially when $\lambda \geq 0.2$ for most data conditions. The *pmask* values also gradually increase when n increase especially when $\lambda < 0.6$ for most data conditions. The result of swamping effect (*pswamp*) for OS1 are presented in Table 6. The results show that as λ increase, the *pswamp* values gradually decrease and approaching to zero which indicates the algorithms are not falsely detecting inliers as outliers. Similar with *pmask* result, the *pswamp* values also decreasing as the value of κ getting larger for most data conditions. Moreover, the *pswamp* values gradually decrease when n increase for most data conditions. However, the *pswamp* values slightly increase as the percentage of outliers increase especially when $\lambda > 0.2$ for most data conditions

Table 3. Performance indicator to evaluate the performance of the COD procedure

Performance indicator	Formula
"Success" probability (*pout*)	$pout = \frac{"success"}{s}$ (12) where *pout* is the probability that all the planted outliers are successfully detected and *"success"* is number of dataset that the method successfully identified all the planted outliers.
Masking effect (*pmask*)	$pmask = \frac{"failure"}{(out)(s)}$ (13) where *pmask is* the probability of outliers are falsely detected as inliers, *"failure*" is the number of outliers in all data set that detected as inliers, *out* is the planted outliers.
Swamping effect (*pswamp*)	$pswamp = \frac{"false"}{(n-out)(s)}$ (14) where *pswamp* is the probability of inliers are falsely detected as outliers, "*false*" is the number of inliers in all data set that detected as outliers, *n* is sample size, *out* is the planted outliers.

DOI: https://doi.org/10.22452/mjs.vol45sp2.9
Malaysian Journal of Science 45 (Special Issue 2): 83-94(November 2026)

Table 4. "Success" probability (pout) as a performance measure for agglomerative clustering algorithms

Algorithm			SL-Satari/Di		SL-Chang		CL-Satari/Di		CL-Chang		AL-Satari/Di		AL-Chang	
Outlier %			5%	15%	5%	15%	5%	15%	5%	15%	5%	15%	5%	15%
n	κ	λ												
15	1.5	0.0	0.1368	0.0056	0.1424	0.0078	0.3538	0.0564	**0.3618**	**0.0600**	0.2130	0.0138	0.2136	0.0148
		0.2	0.1758	0.0164	0.1816	0.0204	0.3982	0.1038	**0.4064**	**0.1110**	0.2550	0.0440	0.2580	0.0434
		0.4	0.3122	0.0618	0.3192	0.0636	0.5488	0.2416	**0.5544**	**0.2502**	0.4094	0.1386	0.4144	0.1392
		0.6	0.4870	0.1500	0.4922	0.1564	0.6922	0.4340	**0.7006**	**0.4408**	0.5832	0.3200	0.5852	0.2996
		0.8	0.6354	0.2696	0.6424	0.2776	0.7898	0.5816	**0.7974**	**0.5924**	0.7350	0.4512	0.7362	0.4590
		1.0	0.7102	0.3294	0.7156	0.3442	0.8418	0.6362	**0.8480**	**0.6456**	0.7994	0.5106	0.8028	0.5200
	5	0.0	0.1544	0.0104	0.2316	0.0624	0.2828	0.0252	**0.2826**	**0.0252**	0.2192	0.0170	0.2240	0.0178
		0.2	0.3510	0.0722	0.4132	0.1074	0.4576	0.1998	**0.4570**	**0.1998**	0.4168	0.1430	0.4194	0.1468
		0.4	0.7158	0.3220	0.7146	0.3374	0.7158	0.6074	**0.7146**	**0.6066**	0.7614	0.5104	0.7600	0.5164
		0.6	0.9094	0.6738	0.9074	0.6800	0.9094	0.8734	**0.9074**	**0.8724**	0.9506	0.8364	0.9494	0.8360
		0.8	0.9926	0.8986	0.9926	0.8974	0.9870	0.9798	**0.9870**	**0.9802**	0.9954	0.9542	0.9950	0.9528
		1.0	0.9982	0.9624	0.9984	0.9630	0.9984	0.9956	**0.9986**	**0.9954**	0.9990	0.9840	0.9992	0.9840
50	1.5	0.0	0.0048	0.0008	0.0090	0.0060	**0.3448**	**0.1392**	0.3002	0.1252	0.1572	0.0614	0.1092	0.0440
		0.2	0.0092	0.0018	0.0130	0.0054	**0.3820**	**0.1722**	0.3452	0.1512	0.2228	0.0778	0.1630	0.0552
		0.4	0.0492	0.0088	0.0394	0.0096	**0.5336**	**0.2750**	0.5002	0.2488	0.3908	0.1662	0.3288	0.1284
		0.6	0.1788	0.0486	0.1434	0.0408	**0.7012**	**0.4370**	0.6796	0.4138	0.6242	0.3410	0.5632	0.2896
		0.8	0.3638	0.1486	0.3082	0.1310	**0.8014**	**0.5866**	0.7880	0.5634	0.7590	0.5090	0.7178	0.4594
		1.0	0.4696	0.2332	0.4100	0.2128	**0.8302**	**0.6508**	0.8200	0.6320	0.8302	0.5860	0.7984	0.5420
	5	0.0	0.0094	0.0026	0.2084	0.1040	**0.2510**	**0.1146**	0.1852	0.0862	0.1374	0.0568	0.0918	0.0388
		0.2	0.0568	0.0088	0.3004	0.1608	**0.4564**	**0.2068**	0.3758	0.1512	0.3346	0.1382	0.2472	0.0914
		0.4	0.3490	0.1294	0.3730	0.2522	**0.8144**	**0.5964**	0.7424	0.5252	0.7144	0.5258	0.5756	0.4056
		0.6	0.7530	0.5990	0.6536	0.5252	**0.9596**	**0.8956**	0.9314	0.8638	0.9316	0.8580	0.8500	0.7620
		0.8	0.9598	0.9254	0.9172	0.8876	**0.9958**	**0.9852**	0.9926	0.9810	0.9928	0.9786	0.9788	0.9496
		1.0	0.9922	0.9860	0.9784	0.9732	**0.9994**	**0.9980**	0.9990	0.9972	0.9996	0.9982	0.9972	0.9930

Table 5. Masking effect (pmask) as a performance measure for agglomerative clustering algorithms

Algorithm			SL-Satari/Di		SL-Chang		CL-Satari/Di		CL-Chang		AL-Satari/Di		AL-Chang	
Outlier %			5%	15%	5%	15%	5%	15%	5%	15%	5%	15%	5%	15%
n	κ	λ												
15	1.5	0.0	0.8632	0.8583	0.8576	0.8525	0.6462	0.6394	**0.6382**	**0.6345**	0.7870	0.7815	0.7864	0.7813
		0.2	0.8242	0.8261	0.8184	0.8201	0.5902	0.5920	**0.5836**	**0.5847**	0.7250	0.7366	0.7420	0.7460
		0.4	0.6878	0.7343	0.6808	0.7296	0.4512	0.4737	**0.4456**	**0.4642**	0.5906	0.6239	0.5856	0.6213
		0.6	0.5130	0.6000	0.5078	0.5911	0.3078	0.3213	**0.2894**	**0.3135**	0.4168	0.4410	0.4148	0.4371
		0.8	0.3646	0.4479	0.3576	0.4389	0.2002	0.2113	**0.1926**	**0.2043**	0.2650	0.2895	0.2638	0.2829
		1.0	0.2898	0.3768	0.2844	0.3681	0.1582	0.1737	**0.1520**	**0.1668**	0.2006	0.2306	0.1972	0.2243
	5	0.0	0.8456	0.8417	0.7684	0.7622	0.7172	0.6991	**0.7174**	**0.6993**	0.7808	0.7786	0.7760	0.7654
		0.2	0.6490	0.7187	0.5868	0.6752	0.5424	0.5796	**0.5430**	**0.5799**	0.5832	0.6341	0.5806	0.6292
		0.4	0.2666	0.4311	0.2436	0.4175	0.2842	0.2635	**0.2554**	**0.2641**	0.2386	0.2919	0.2400	0.2887
		0.6	0.0498	0.1815	0.0486	0.1773	0.0906	0.0679	**0.0526**	**0.0679**	0.0494	0.0761	0.0506	0.0751
		0.8	0.0074	0.0483	0.0074	0.0481	0.0130	0.0085	**0.0091**	**0.0095**	0.0046	0.0273	0.0050	0.0181
		1.0	0.0018	0.0151	0.0016	0.0163	0.0026	0.0023	**0.0014**	**0.0023**	0.0010	0.0054	0.0008	0.0055
50	1.5	0.0	0.8888	0.8869	0.9033	0.9015	**0.3406**	**0.3394**	0.3370	0.3790	0.5245	0.5268	0.5955	0.5961
		0.2	0.8513	0.8543	0.8687	0.8772	**0.3060**	**0.3085**	0.3002	0.3377	0.4593	0.4655	0.5342	0.5397
		0.4	0.7244	0.7589	0.7643	0.7956	**0.2175**	**0.2270**	0.2394	0.2480	0.3183	0.3284	0.3708	0.3755
		0.6	0.5178	0.5613	0.5765	0.6130	**0.1313**	**0.1512**	0.1418	0.1608	0.1683	0.1811	0.2052	0.2116
		0.8	0.3229	0.3514	0.3787	0.3913	**0.0849**	**0.1021**	0.0908	0.1085	0.0920	0.0985	0.1103	0.1147
		1.0	0.2458	0.2337	0.2920	0.2585	**0.0746**	**0.0829**	0.0786	0.0879	0.0616	0.0704	0.0739	0.0803
	5	0.0	0.8921	0.8895	0.6133	0.6125	**0.4335**	**0.4313**	0.5135	0.5154	0.5747	0.5730	0.6690	0.6679
		0.2	0.7015	0.7623	0.4927	0.5547	**0.2706**	**0.2993**	0.3403	0.3688	0.3691	0.4008	0.4595	0.4913
		0.4	0.3449	0.4587	0.3619	0.4650	**0.0866**	**0.1136**	0.1206	0.1427	0.1239	0.1290	0.1994	0.2096
		0.6	0.1037	0.1272	0.1619	0.2026	**0.0207**	**0.0320**	0.0313	0.0377	0.0204	0.0222	0.0565	0.0406
		0.8	0.0151	0.0145	0.0339	0.0304	**0.0026**	**0.0036**	0.0037	0.0047	0.0024	0.0028	0.0073	0.0066
		1.0	0.0027	0.0020	0.0076	0.0043	**0.0005**	**0.0013**	0.0006	0.0014	0.0001	0.0002	0.0009	0.0009

DOI: https://doi.org/10.22452/mjs.vol45sp2.9
Malaysian Journal of Science 45 (Special Issue 2): 83-94(November 2026)

Table 6. Swamping effect (pswamp) as a performance measure for agglomerative clustering algorithms

Algorithm			SL-Satari/Di		SL-Chang		CL-Satari/Di		CL-Chang		AL-Satari/Di		AL-Chang	
Outlier %			5%	15%	5%	15%	5%	15%	5%	15%	5%	15%	5%	15%
n	κ	λ												
15	1.5	0.0	**0.1404**	**0.1409**	0.1452	0.1464	0.3567	0.3575	0.3629	0.3619	0.2136	0.2119	0.2151	0.2138
		0.2	**0.1381**	**0.1351**	0.1433	0.1403	0.3590	0.3638	0.3644	0.3707	0.2115	0.2113	0.2137	0.2132
		0.4	**0.1301**	**0.1305**	0.1361	0.1350	0.3599	0.3656	0.3655	0.3746	0.2031	0.2114	0.2061	0.2143
		0.6	**0.1166**	**0.1134**	0.1216	0.1174	0.3548	0.3623	0.3632	0.3725	0.1958	0.1902	0.1981	0.1916
		0.8	**0.1013**	**0.1017**	0.1047	0.1050	0.3555	0.3599	0.3629	0.3726	0.1770	0.1725	0.1781	0.1781
		1.0	**0.0964**	**0.1006**	0.0991	0.1048	0.3453	0.3583	0.3544	0.3710	0.1659	0.1646	0.1690	0.1706
	5	0.0	**0.1560**	**0.1558**	0.2334	0.2387	0.3001	0.2979	0.3000	0.2978	0.2230	0.2243	0.2296	0.2287
		0.2	**0.1344**	**0.1258**	0.1957	0.1701	0.2801	0.2663	0.2800	0.2662	0.2029	0.1900	0.2078	0.1949
		0.4	**0.0725**	**0.0584**	0.0960	0.0665	0.2108	0.1641	0.2102	0.1639	0.1243	0.0924	0.1278	0.0957
		0.6	**0.0204**	**0.0165**	0.0222	0.0176	0.1156	0.0729	0.1130	0.0729	0.0423	0.0274	0.0437	0.0276
		0.8	**0.0038**	**0.0045**	0.0038	0.0045	0.0618	0.0396	0.0567	0.0402	0.0107	0.0077	0.0110	0.0077
		1.0	**0.0048**	**0.0022**	0.0048	0.0021	0.0509	0.0319	0.0496	0.0337	0.0048	0.0031	0.0048	0.0031
50	1.5	0.0	**0.1114**	**0.1114**	0.0955	0.0966	0.6626	0.6607	0.6231	0.6212	0.4765	0.4766	0.4019	0.4060
		0.2	**0.1105**	**0.1117**	0.0965	0.0960	0.6603	0.6678	0.6255	0.6279	0.4820	0.4865	0.4041	0.4129
		0.4	**0.1095**	**0.1130**	0.0946	0.0981	0.6623	0.6684	0.6256	0.6359	0.4772	0.4916	0.4119	0.4256
		0.6	**0.1083**	**0.1176**	0.0920	0.1038	0.6590	0.6645	0.6224	0.6371	0.4681	0.4847	0.4009	0.4200
		0.8	**0.1068**	**0.1206**	0.0894	0.1089	0.6509	0.6520	0.6174	0.6267	0.4629	0.4709	0.3949	0.4110
		1.0	**0.1033**	**0.1238**	0.0867	0.1146	0.6556	0.6509	0.6215	0.6247	0.4599	0.4657	0.3961	0.4076
	5	0.0	**0.1093**	**0.1096**	0.3872	0.3852	0.5660	0.5706	0.4837	0.4865	0.4276	0.4284	0.3339	0.3314
		0.2	**0.0966**	**0.0885**	0.3521	0.3298	0.5592	0.5583	0.4755	0.4705	0.4144	0.4038	0.3168	0.3123
		0.4	**0.0519**	**0.0519**	0.1483	0.1772	0.5120	0.5233	0.4187	0.4354	0.3124	0.2987	0.1931	0.1923
		0.6	**0.0204**	**0.0290**	0.0189	0.0325	0.4858	0.4849	0.3875	0.3979	0.2035	0.1828	0.0737	0.0716
		0.8	**0.0092**	**0.0137**	0.0045	0.0083	0.4696	0.4583	0.3665	0.3792	0.1290	0.1057	0.0302	0.0292
		1.0	**0.0058**	**0.0077**	0.0023	0.0041	0.4682	0.4567	0.3649	0.3761	0.1105	0.0867	0.0218	0.0216

3.2 Performance Analysis for COD Procedure

The performance analysis particularly with OS1 were performed by comparing the performance of clustering algorithms. The performance of the COD procedure also analyzed and compare in various data conditions and confidence interval (C.I) of stopping rule. The results were summarized in table and the graphs are plotted in various data conditions as measured by *pout, pmask* and *pswamp* to visualized:

i. the best clustering algorithm
ii. the best concentration parameter
iii. the best percentage of outliers
iv. the best sample sizes
v. the best confidence interval of stopping rule

3.2.1 Comparison Analysis of Clustering Algorithms

The performance of six clustering algorithms are analyzed and compared based on *pout, pmask* and *pswamp*. Table 7 show the best algorithms on identifying multiple outliers in univariate circular datasets across varying conditions as measured by *pout, pmask* and *pswamp* for OS1.

The results show that the performance of CL-Satari/Di is the best by referring to the *pout* value especially when $\kappa \le 2$ and $n \ge 30$ for any percentage of outliers and both 90% and 95% C.I. of stopping rule. CL-Chang has the best performance to successfully detect all outliers especially when concentration parameter, $\kappa \le 2$ and small sample size, $n = 15$ for all percentage of outliers and both 90% and 95% C.I. of stopping rule. Moreover, the simulation study found that, AL-Chang has the best performance for *pout* especially when $\kappa = 100$ and $n = 100$. Meanwhile, SL-Chang is the best algorithms when the dataset has large concentration parameter and sample size $(\kappa = 100,\ n = 100)$ for all percentage of outliers and both 90% and 95% C.I. of stopping rule.

The summary result of *pmask* in Table 7 shows that CL-Satari/Di has the best performance on detecting outliers with low masking effect for most of data conditions. However, the result shows that CL-Chang is less sensitive to masking effect especially when the dendrogram is cut at 95% C.I. of stopping rule for small sample size, $n = 15$ when $\kappa < 10$. The simulation study found that AL-Chang has the best performance for *pmask* especially when $\kappa = 100$ and $n \ge 50$ for any percentage outliers and both 90% and 95% C.I. of stopping rule. This indicates that when the dataset has large concentration parameter and sample size, AL-Chang is the best to detect outliers in univariate circular data for OS1 without or less likely to falsely detect outliers as inliers. The result in Table 7 also show that the performance of SL-Satari/Di is the best for all data conditions by referring to the *pswamp* value, indicating that this algorithm is less sensitive to swamping effect. Thus, SL-Satari/Di is the best clustering algorithms on detecting outliers in univariate circular biological data without or less likely to falsely detect inliers as outliers for OS1 regardless of concentration parameter, sample size, percentage of outliers and C.I. of stopping rule. Figure 1 shows a few plots of all clustering algorithms performance for OS1 with combination of certain conditions such as $n = 15$, $\kappa = 1.5$ with 5% of outliers and $n = 50$, $\kappa = 5$ for 15% of outliers.

Generally, the curves of *pout, pmask* and *pswamp* show similar pattern for both 5% and 15 % of outliers. It can be seen

DOI: https://doi.org/10.22452/mjs.vol45sp2.9
Malaysian Journal of Science 45 (Special Issue 2): 83-94(November 2026)

Table 7. The best algorithms on identifying multiple outliers in univariate circular data for highlighted conditions

95% C.I	***pout***							
	5% Outliers				15% Outliers			
Sample size	$\kappa=1.5$	$\kappa=5$	$\kappa=50$	$\kappa=100$	$\kappa=1.5$	$\kappa=5$	$\kappa=50$	$\kappa=100$
$n=15$	CL-Chang	CL-Chang	SL-Chang	AL-Chang	CL-Chang	CL-Chang	SL-Chang	AL-Chang
$n=50$	CL-Satari/Di	CL-Satari/Di	SL-Chang	AL-Chang	CL-Satari/Di	CL-Satari/Di	SL-Chang	AL-Chang
90% C.I	***pout***							
	5% Outliers				15% Outliers			
Sample size	$\kappa=1.5$	$\kappa=5$	$\kappa=50$	$\kappa=100$	$\kappa=1.5$	$\kappa=5$	$\kappa=50$	$\kappa=100$
$n=15$	CL-Satari/Di	CL-Satari/Di	SL-Chang	AL-Chang	CL-Satari/Di	CL-Satari/Di	SL-Chang	AL-Chang
$n=50$	CL-Satari/Di	CL-Satari/Di	SL-Chang	AL-Chang	CL-Satari/Di	CL-Satari/Di	SL-Chang	AL-Chang
95% C.I	***pmask***							
	5% Outliers				15% Outliers			
Sample size	$\kappa=1.5$	$\kappa=5$	$\kappa=50$	$\kappa=100$	$\kappa=1.5$	$\kappa=5$	$\kappa=50$	$\kappa=100$
$n=15$	CL-Chang	CL-Chang	SL-Chang	AL-Chang	CL-Chang	CL-Chang	SL-Chang	AL-Chang
$n=50$	CL-Satari/Di	CL-Satari/Di	SL-Chang	AL-Chang	CL-Satari/Di	CL-Satari/Di	SL-Chang	AL-Chang
90% C.I	***pmask***							
	5% Outliers				15% Outliers			
Sample size	$\kappa=1.5$	$\kappa=5$	$\kappa=50$	$\kappa=100$	$\kappa=1.5$	$\kappa=5$	$\kappa=50$	$\kappa=100$
$n=15$	CL-Satari/Di	CL-Satari/Di	SL-Chang	AL-Chang	CL-Satari/Di	CL-Satari/Di	SL-Chang	AL-Chang
$n=50$	CL-Satari/Di	CL-Satari/Di	CL-Satari/Di	AL-Chang	CL-Satari/Di	CL-Satari/Di	CL-Satari/Di	AL-Chang
95% C.I	***pswamp***							
	5% Outliers				15% Outliers			
Sample size	$\kappa=1.5$	$\kappa=5$	$\kappa=50$	$\kappa=100$	$\kappa=1.5$	$\kappa=5$	$\kappa=50$	$\kappa=100$
$n=15$	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di
$n=50$	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di
90% C.I	***pswamp***							
	5% Outliers				15% Outliers			
Sample size	$\kappa=1.5$	$\kappa=5$	$\kappa=50$	$\kappa=100$	$\kappa=1.5$	$\kappa=5$	$\kappa=50$	$\kappa=100$
$n=15$	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di
$n=50$	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di	SL-Satari/Di

from Figure 1, the curves of *pout* gradually increase and approaching to 1 as λ increase. The larger the sample size, the faster the curves approach to 1. Besides, it can be seen that the *pmask* gradually decreases as the λ increase and as n increase, the faster the curves of *pmask* approaching to zero. Similarly happen to the curves of *pmask* when κ increase, the faster the curves approaching to zero. Meanwhile, the plots of *pswamp* display in Figure 1 show that the value of *pswamp* remain constant as the value of λ increase. However, as n and κ increase, the curves slightly approaching to zero as the value of λ increase especially for SL-Chang, AL-Satari/Di, and AL-Chang algorithms.

3.2.2 Comparison Analysis of Concentration Parameter

There are seven concentration parameters considered in this study, $\kappa=1.5,2,5,10,30,50,100$. In this study, $\kappa=1.5$ is considered small concentration parameter which means that the dataset has a very large spread, meanwhile $\kappa=100$ is considered large concentration parameter which means that the dataset is concentrated at mean angle. The simulation study found that $\kappa=100$ is the best concentration parameter value on detecting multiple outliers in univariate circular data with various conditions as measured by *pout*, *pmask* and *pswamp* for OS1. This result shows that when $\kappa=100$, the outliers are most likely to detect correctly without masking and swamping effects for OS1 using both 90% and 95% C.I of stopping rule. In other words, $\kappa=100$ is large enough to explain how concentrated the dataset towards the mean angle and make the outliers obviously separated from the others. Hence, the clustering algorithms are able to detect outliers correctly without masking and swamping effects regardless of sample size, percentage of outliers and confidence interval of cutting height.

Figure 2 show the plots of *pout*, *pmask* and *pswamp* versus λ for OS1 using SL-Satari/Di and CL-Satari/Di clustering algorithms by looking at different value of κ with selected conditions using 95% C.I of stopping rule. Generally, the performance of all clustering algorithms shows similar pattern for *pout*, *pmask* and *pswamp* for most data conditions. The value of *pout* gradually increases and approaching to 1 as the value of λ increases for all value of κ for any other data conditions. The pout values approaching to 1 faster even with small λ when $\kappa\geq 30$ for all clustering algorithms. Meanwhile, the plots of *pmask* in Figure 2 show that the *pmask* values decreasing and approaching to zero faster even with small λ especially when $\kappa\geq 30$. Similarly, the *pswamp* values also declining and gradually approaching to zero as the value of λ increases especially when $\kappa=100$. The plots of *pswamp* show that the values remain constant at specific value as

DOI: https://doi.org/10.22452/mjs.vol45sp2.9
Malaysian Journal of Science 45 (Special Issue 2): 83-94(November 2026)

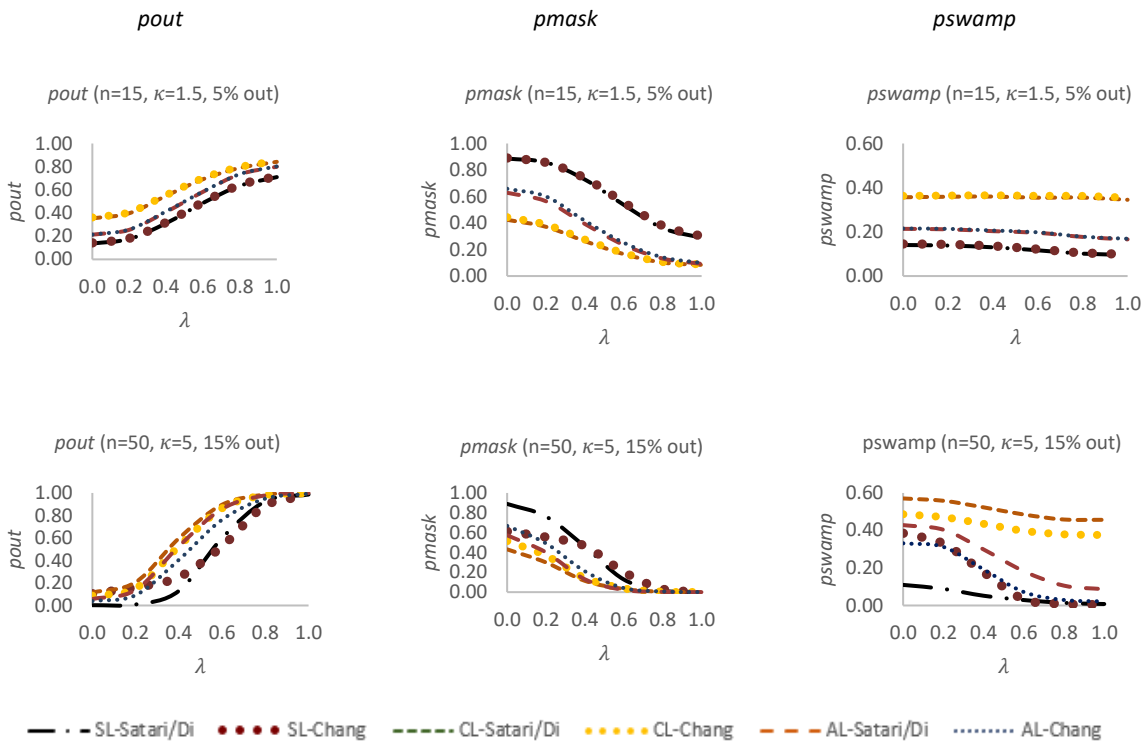


Figure 1. Plots *pout*, *pmask* and *pswamp* versus λ with selected condition ($n=15$, $\kappa=1.5$ with 5% outliers and $n=50$, $\kappa=5$ for 15% of outliers) using 95% C.I stopping rule

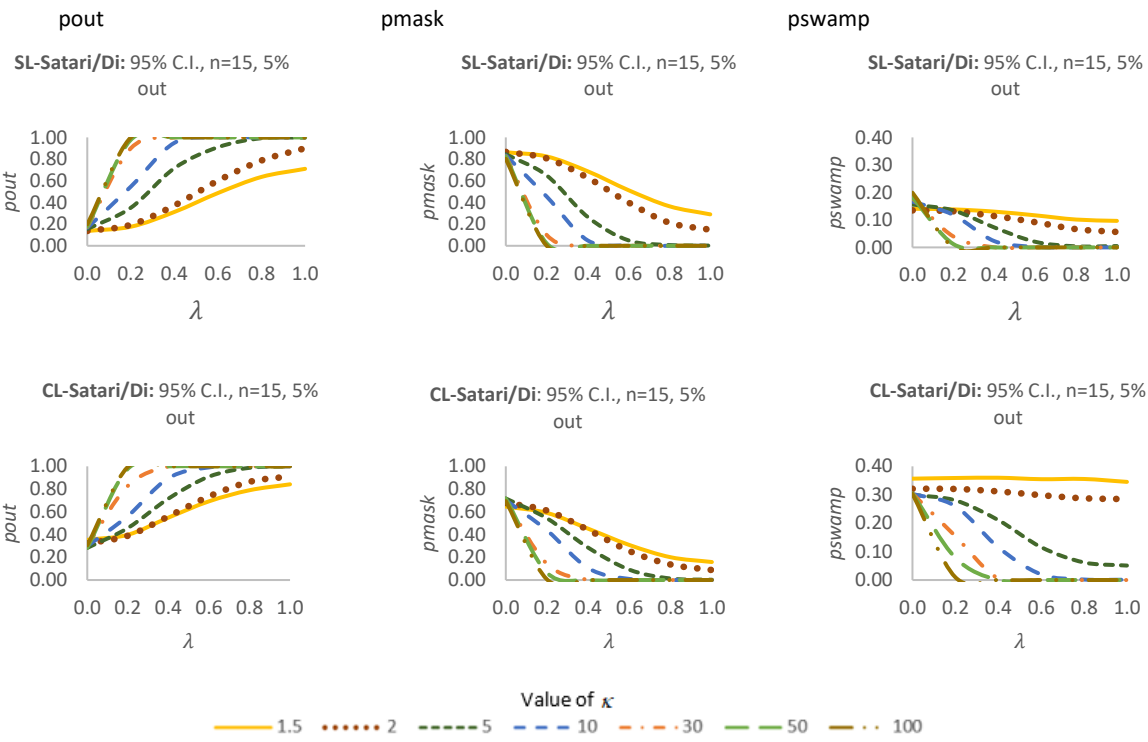


Figure 2. OS1: Plot of *pout*, *pmask* and *pswamp* versus λ for SL and CL algorithms with selected conditions ($n=15$ with 5% of outliers using 95% C.I of stopping rule

the value of λ increases particularly when $\kappa \leq 2$. Hence, the plots in Figure 2 show that the performance of *pout*, *pmask* and *pswamp* for OS1 are at the best even with small λ when $\kappa=100$ for all clustering algorithms. This study also found that the results for the best concentration parameter are similar when applying the 90% C.I of stopping rule.

DOI: https://doi.org/10.22452/mjs.vol45sp2.9
Malaysian Journal of Science 45 (Special Issue 2): 83-94(November 2026)

3.2.3 Comparison Analysis of Percentage Outliers

In simulation study, the outliers are planted in simulated data generated from von Mises distribution with outliers strategically placed at a specified distance from the inlying observations using the formulation of OS1. Four percentage of outliers are chosen which are 5%, 10%, 15% and 20% from the sample size. Table 8 summarized the best percentage outlier for *pout*, *pmask* and *pswamp* by considering the highlighted data conditions for all clustering algorithms using both 90% and 95% C.I of stopping rule based on the indicators *pout*, *pmask* and *pswamp*, respectively.

found that 5% is the best percentage of outliers for all clustering algorithms by using 90% and 95% C.I of cutting height regardless any n and κ. It indicates that the probability for the clustering algorithms falsely detected outliers as inliers is lower when the outliers are not more than 5% from sample size. Table 8 also shows that SL-Satari/Di and SL-Chang might have low swamping effect (*pswamp*) when minimal outlier presence (5%), particularly when $n \geq 30$. In contrary, CL-Satari/Di, CL-Chang, AL-Satari/Di and AL-Chang might have small values of *pswamp* when the dataset consists of 20% of outliers across most data conditions. It indicates that when the percentage of outliers are high and the

Table 8. Summary of the best percentage outlier based on three performance indicators by considering the highlighted sample size for all and both 90% and 95% C.I of stopping rule

Clustering algorithms	*pout*		*pmask*		*pswamp*	
	$n=15$	$n \geq 30$	$n=15$	$n \geq 30$	$n=15$	$n \geq 30$
SL-Satari/Di	5%	5%	5%	5%	20%	5%
SL-Chang	5%	5%	5%	5%	5%	5%
CL-Satari/Di	5%	5%	5%	5%	20%	20%
CL-Chang	5%	5%	5%	5%	20%	20%
AL-Satari/Di	5%	5%	5%	5%	20%	20%
AL-Chang	5%	5%	5%	5%	20%	20%

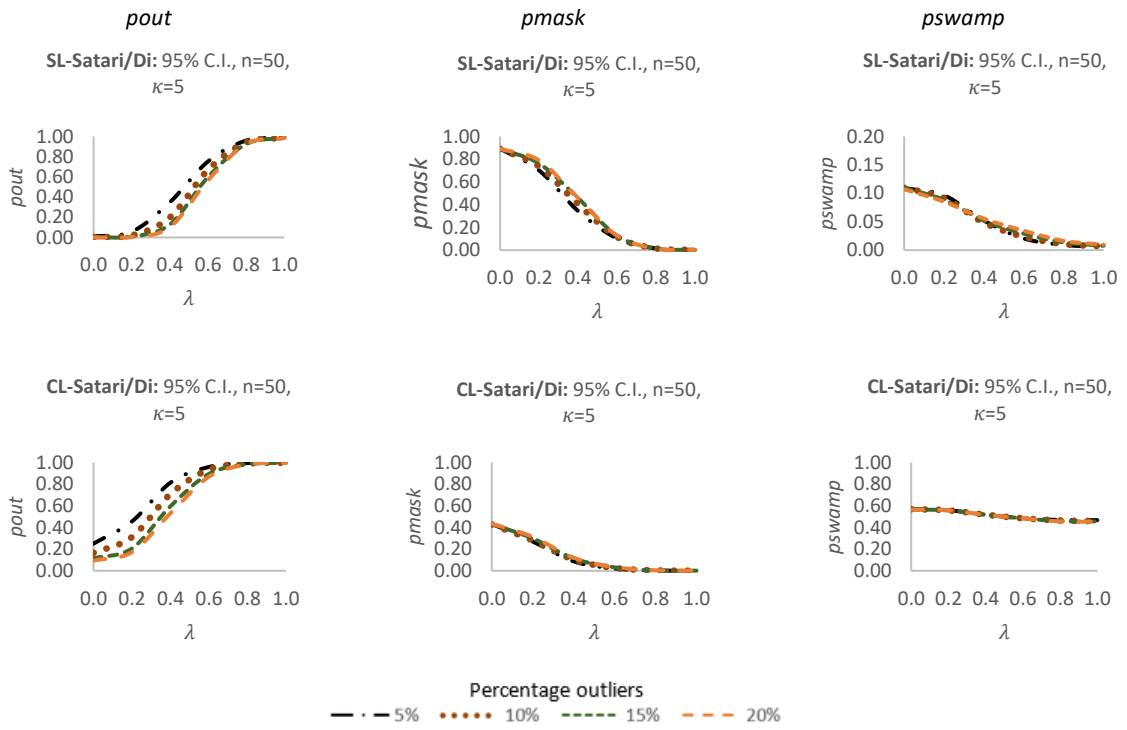


Figure 3. Plots *pout, pmask* and *pswamp* versus λ for different percentage of outliers with selected clustering algorithms for dataset $n = 50$ with $\kappa = 15$ using 95% C.I stopping rule

The simulation results show that 5% of outliers is the best condition for any clustering algorithms to successfully detect the correct outliers in univariate circular data in any data conditions using both 90% and 95% C.I of stopping rule . The performance of *pout* indicates that the outliers are more successfully detected when the dataset have small percentage of outliers for any clustering algorithms regardless any n and κ for both 90% and 95% C.I of stopping rule. Similarly, the results based on *pmask* probability of falsely detecting inliers as outliers is very low for CL-Satari/Di, CL-Chang, AL-Satari/Di and AL-Chang.

Figure 3 show the plot of *pout, pmask* and *pswamp* performance against level of contamination, λ for OS1 using SL-Satari/Di and CL-Satari/Di algorithms. The results are illustrated across varying percentages of outliers, based on selected data conditions and a 95% confidence interval applied to the stopping rule. The plots of *pout* for all clustering algorithms show that 5%

DOI: https://doi.org/10.22452/mjs.vol45sp2.9
Malaysian Journal of Science 45 (Special Issue 2): 83-94(November 2026)

of outliers performed the best compared to others. It is also show that the *pmask* values decreasing as λ increasing and it can be seen that 5% of outliers producing the lowest value of *pmask* compare to others for all algorithms. The *pswamp* plots in Figure 3 also show various pattern and it is found that the *pswamp* values are almost the same for all percentage outliers. The *pswamp* plots show that the *pswamp* value remain constant at specific value as the value of λ increases especially when the data has small sample size and concentration parameter for most of clustering algorithms. This study also found that the plots presented in Figure 3 have similar pattern to the results obtained using the 90% C.I of stopping rule.

3.2.4 Comparison Analysis of Sample Sizes

Four sample sizes, $n=15,30,50,100$ are chosen in this study where $n=15$ is considered small sample size and $n=100$ is considered as large sample size. This study found that the best sample size using the 90% C.I of stopping rule is similar to that obtained with the 95% C.I of stopping rule. The results from simulation study found that the probability of algorithm to detect outliers correctly (*pout*) when $n=15$ using 90% and 95% C.I of stopping rule. Meanwhile, the algorithms are mostly to detect outliers with low masking effect (*pmask*) when $n=100$ regardless any conditions for both C.I of stopping rule. This indicates that when sample size is large, any clustering algorithms used in this study have less sensitive to masking effect which occurs when outliers falsely detected as inliers regardless any conditions. The simulation study also found that the best sample size for outlier detection with low swamping effect is when $n=15$ by using any clustering algorithms for most data conditions. This indicates that when sample size is small, the clustering algorithms use in this study are less sensitive to swamping effect which occurs when inliers falsely detected as outliers regardless any conditions. Generally, the performance of all clustering algorithms shows similar pattern for *pout, pmask* and *pswamp* for most data conditions using both 90% and 95% C.I of stopping rule. Figure 4 visualized the plots of *pout, pmask* and *pswamp* against varying λ for OS1 using SL-Satari/Di and AL-Chang algorithms. The results are shown for selected data conditions with a 95% confidence interval applied to the stopping rule. The plots of *pout* in Figure 4 show that SL-Satari/Di achieved superior performance when $n=15$ using 95% C.I of stopping rule. The plot of *pmask* for SL-Satari/Di and AL-Chang in Figure 4 show the *pmask* performance have slightly differences among all sample size. Moreover, the plots of *pswamp* in Figure 4 found that most of clustering algorithms value remain constant at certain value as the value of λ increases. In conclusion, the plots in Figure 4 indicate that most clustering algorithms perform the best in detecting outliers with minimal swamping effects under small sample conditions but might lead to high masking effect.

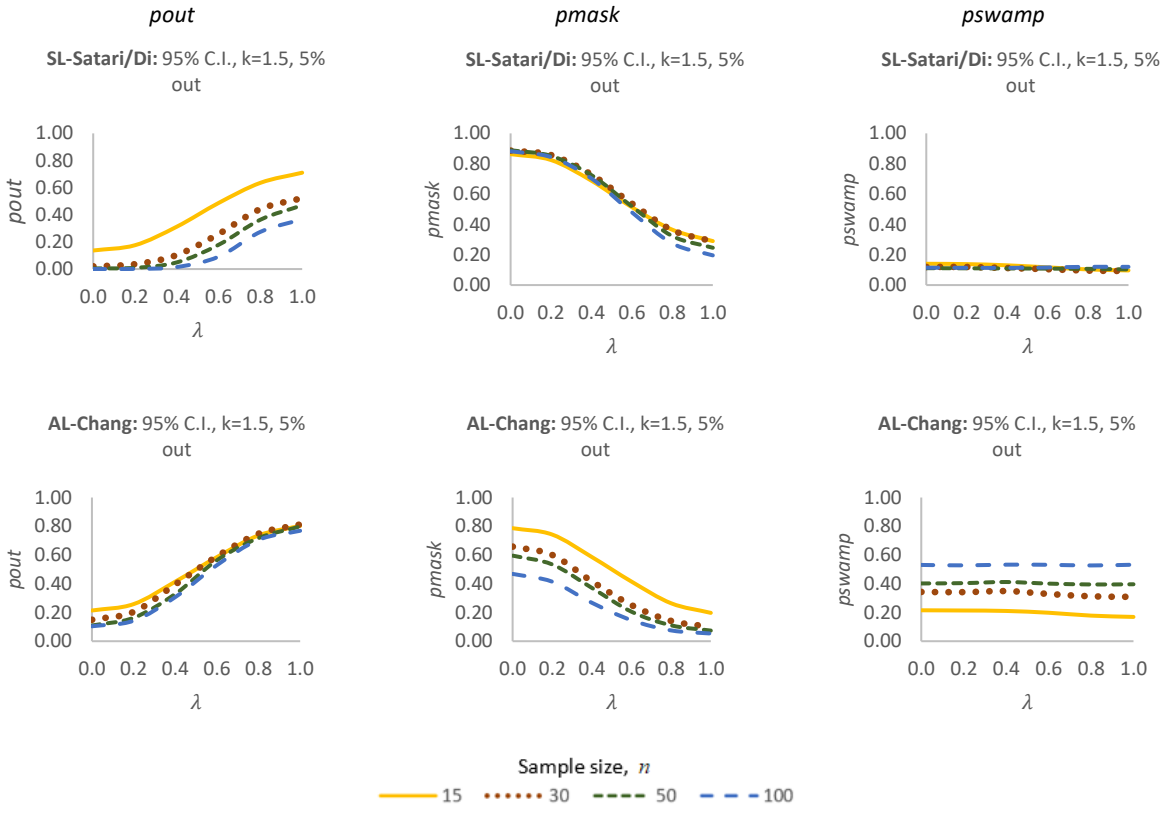


Figure 4. Plots *pout*, *pmask* and *pswamp* versus λ for selected clustering algorithms for selected dataset ($\kappa=1.5$ with 5% of outliers) using 95% C.I of stopping rule

3.2.5 Comparison Analysis of Confidence Interval of Stopping Rule

The results based on *pout* and *pmask* indicate that a 90% C.I for the stopping rule provides the most effective performance in identifying multiple outliers within univariate circular data across

DOI: https://doi.org/10.22452/mjs.vol45sp2.9
Malaysian Journal of Science 45 (Special Issue 2): 83-94(November 2026)

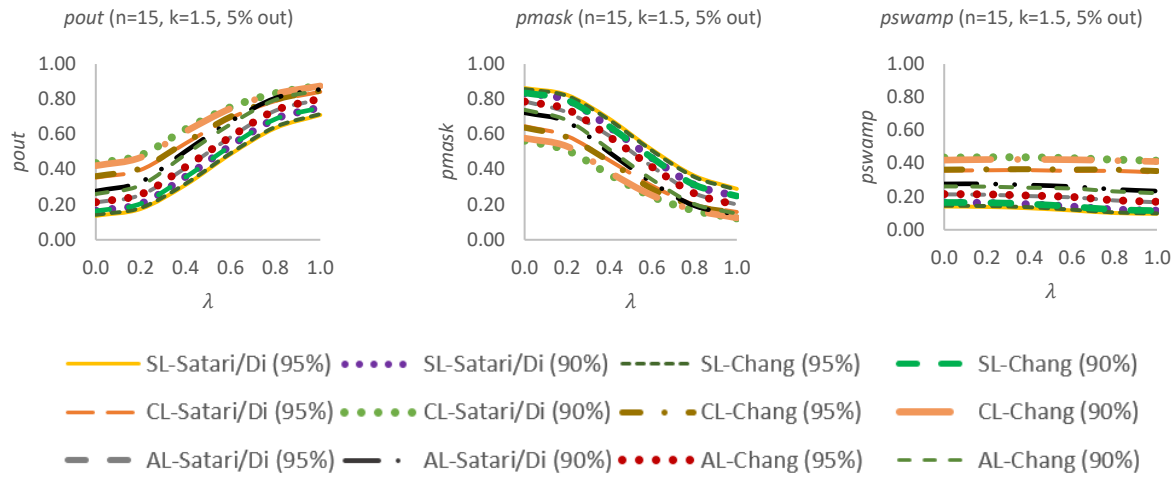


Figure 5. Plots *pout*, *pmask* and *pswamp* versus λ for all algorithms for $n=15$, $\kappa=1.5$ with 5% of outliers

most data conditions, regardless of the clustering algorithm applied. The plots of *pout* and *pmask* against value of λ in Figure 5 show that 90% C.I of stopping rule from all clustering algorithms has better performance even though the differences with 95% C.I are slightly noticeable. This indicates that by cutting the height of dendrogram at 90% C.I, any clustering algorithms used in this study are most likely to successfully detect outliers correctly without or with low masking effect. However, 95% C.I is the best for most data conditions by referring to the *pswamp* value as visualized in Figure 5 which indicates that 95% C.I of stopping rule is less sensitive to swamping effect by using any clustering algorithms.

4. Conclusions

This study aims to assess the effectiveness of the COD procedure in detecting outliers within univariate circular datasets. The simulation was carried out using synthetic datasets, where outliers were introduced using the formulation of OS1. The outliers from OS1 are introduced by shifting the mean direction of the inliers by a specific distance defined by the contamination level. The mean direction was initially set to zero and the outliers are placed at a specific distance from the inliers depending on the contamination level, λ. The value of λ defines how far the outliers are moved from the central of inlying data points or called inliers. There are six different λ used in the simulation study and this study shows that different magnitudes of contamination in OS1 affect the outlier detection process. The simulation results show that the *pout* values gradually increase and approaching to one as λ and κ increase. Meanwhile, the *pout* values gradually decreasing as the percentage of outliers and n increase. The *pmask* values gradually decreasing and approaching to zero as λ and κ increase. As the percentage of outliers and n increase, the *pmask* values are slightly increasing. As conclusion from *pswamp* results for OS1, the values of *pswamp* gradually decreasing and approaching to zero as the values of λ, κ and n increase. However, the *pswamp* values slightly increase as the percentage of outliers increase. Generally, the results of *pout*, *pmask* and *pswamp* performance revealed that all clustering algorithms have similar result pattern. Table 9 presents a summary of the best-performing results from the simulation study under Outlier Scenario 1 (OS1) using three performance measure which are *pout, pmask* and *pswamp*

The result from simulation study with OS1 found that CL-Satari/Di has the best performance on detecting outliers with low masking effect for particular data conditions. Meanwhile, the performance of SL-Satari/Di is not good compared to other algorithms for *pout* and *pmask* but this algorithm found to be the best algorithm based on *pswamp*. This indicates that SL-Satari/Di is less sensitive to swamping effect for OS1. This study found that SL-Satari/Di is convincing to identify outliers in univariate circular data without or less likely to misidentified inliers as outliers for OS1. The results show that the best sample size for detecting outliers varies depending on the clustering algorithm and data conditions. It is found that large sample size is less sensitive to masking effect, while a smaller sample size is less sensitive to swamping effect. This study also shows that the mean direction and contamination levels play a crucial role in determining how outliers are generated. Thus, the topic on outlier scenario is interesting to study deeper as it can be extended by adjusting the concentration parameter to define the new outliers scenario and compare with OS1 from this study.

In addition, this paper focuses on simulation-based validation to assess the performance of the COD procedure under various conditions. A companion studies from Zulkipli et al. (2024a, 2024b) present a real-life data application of the method to univariate circular biological data. Together, these studies provide both theoretical and practical validation of the COD procedure. Future studies should consider to extend the evaluation by comparing the best algorithm, CL-Satari/Di from COD procedure

DOI: https://doi.org/10.22452/mjs.vol45sp2.9
Malaysian Journal of Science 45 (Special Issue 2): 83-94(November 2026)

against other existing outlier detection methods for univariate circular data.

5. Acknowledgments

The authors sincerely appreciate the thoughtful assessments and constructive suggestions provided by the editors and reviewers, which have played an important role in improving the overall quality of this manuscript. Financial support for this research was provided by Universiti Malaysia Pahang Al-Sultan Abdullah and the Ministry of Higher Education Malaysia through the UMPSA Internal Grant (PGRS210328), RDU1901168, and FRGS/1/2019/STG06/UMP/02/6.

6. References

Abuzaid, A. H. (2020). Detection of outliers in univariate circular data by means of the outlier local factor (LOF). *Statistics in Transition New Series, 21*(3), 39–51.

Abuzaid, A. H., Hussin, A. G., Rambli, A., & Mohamed, I. (2012). Statistics for a new test of discordance in circular data. *Communications in Statistics—Simulation and Computation*, *41*, 1882–1890.

Abuzaid, A. H., Mohamed, I. B., & Hussin, A. G. (2009). A new test of discordancy in circular data. *Communications in Statistics - Simulation and Computation*, *38*(4), 682–691.

Collett, D. (1980). Outliers in circular data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *29*(1), 50–57.

Di, N. F. M., & Satari, S. Z. (2017). The effect of different distance measures in detecting outliers using clustering-based algorithm for circular regression model. *AIP Conference Proceedings*, *1842*, 030016. https://doi.org/10.1063/1.4982059

Di, N. F. M., Satari, S. Z., & Zakaria, R. (2017). Detection of different outlier scenarios in circular regression model using single-linkage method. *Journal of Physics: Conference Series*, *890*(1). https://doi.org/10.1088/1742-6596/890/1/012088.

El, H.I., Ahmed, S., Abuzaid, A.H., & Awar, I.I. (2019). Detection of Outliers in Circular Data using Kernel Density Function. *Life Sciences: An International Journal (LSIJ)*, *1*(1), 1–11.

Hung, W. L., Chang-Chien, S. J., & Yang, M. S. (2012). Self-updating clustering algorithm for estimating the parameters in mixtures of von Mises distributions. *Journal of Applied Statistics*, *39*(10), 2259–2274.

Johnson, R., & Wichern, D. (2014). *Applied multivariate statistical analysis* (6th ed.). Pearson.

Mahmood, E. A., Rana, S., Hussin, A. G., & Midi, H. (2017a). Adjusting outliers in univariate circular data. *Pertanika Journal of Science & Technology*, *25*(4), 1147–1158

Mahmood, E. A., Rana, S., Midi, H., & Hussin, A. G. (2017b). Detection of outliers in univariate circular data using robust circular distance. *Journal of Modern Applied Statistical Methods*, *16*(2), 418–438.

Rambli, A., Ibrahim, S., & Mohamed, I. (2012). On discordance test for the wrapped normal data. *Sains Malaysiana*, *41*(6), 769–778.

Ramlee, I. M., Ibrahim, S., Zhe, L. W., & Yusoff, M. I. (2024). Detection of Outliers in Circular Regression Model via DFBETAcIS Statistic. *Sains Malaysiana*, *53*(4), 935–951.

Rattanawong, B., Papukdee, N., & Srisodaphol, W. (2023). Detection of Outliers in Univariate Circular Data Using New Cut-off Points for the Circular Distance. *Philippine Journal of Science, 152*(5), 2019–2026.

Satari, S. Z. (2015). *Parameter estimation and outlier detection for some types of circular model* (Doctoral dissertation, University of Malaya).

Satari, S. Z., & Ku Khalif, K. M. N. (2020). Review on outliers identification methods for univariate circular biological data. *Advances in Science, Technology and Engineering Systems*, *5*(2), 95–103.

Satari, S. Z., Muhammad Di, N. F., Zubairi, Y. Z., & Hussin, A. G. (2021). Comparative study of clustering-based outliers detection methods in circular-circular regression model. *Sains Malaysiana*, *50*(6), 1787–1798.

Sebert, D. M., Montgomery, D. C., & Rollier, D. A. (1998). A clustering algorithm for identifying multiple outliers in linear regression. *Computational Statistics and Data Analysis*, *27*, 461–484.

Sert, S., & Kardiyen, F. (2023). A method for detecting outliers in linear-circular non-parametric regression. *PLoS ONE*, *18*(6 June), 1–17.

Zulkipli, N. S., Satari, S. Z., & Wan Yusoff, W. N. S. (2022). The effect of different similarity distance measures in detecting outliers using single-linkage clustering algorithm for univariate circular biological data. *Pakistan Journal of Statistics and Operation Research*, *18*(3), 561–573.

Zulkipli, N. S., Satari, S. Z., & Wan Yusoff, W. N. S. (2024a). The Multiple Outliers Detection for Circular Univariate Data Using Different Agglomerative Clustering Algorithms. *AIP Conference Proceedings,* 2895(1): p. 090007.

Zulkipli, N. S., Satari, S. Z., & Wan Yusoff, W. N. S. (2024b). Abnormalities detection in Apert syndrome using hierarchical clustering algorithms. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, *44*(4), 135–174.

DOI: https://doi.org/10.22452/mjs.vol45sp2.9
Malaysian Journal of Science 45 (Special Issue 2): 83-94(November 2026)