Journal of Engineering and Technological Advances (JETA)
https://jeta.segi.edu.my/index.php/segi
2025 Vol. 10 No. 2

DYNAMIC POLYGON GEOFENCING FOR SPATIAL VALIDATION IN CAMPUS ATTENDANCE MONITORING

[1]Premitasari, M.*, [1]Diharja, R.M., [1]Yudha, M.K., [2]Maryanto, T.I.

[1] Informatics Engineering, Faculty of Industrial Engineering, Institut Teknologi Nasional, Jalan PH.H Mustapa No. 23, Bandung, Indonesia.

[2] Geodetic Engineering, Faculty of Civil Engineering and Planning, Institut Teknologi Nasional, Jalan PH.H Mustapa No.23, Bandung, Indonesia.

* Corresponding Author: marisa@itenas.ac.id TEL: (022)-7272215

Received: 26 September 2025; Accepted: 25 November 2025; Published: 31 December 2025
doi: 10.35934/segi.v10i2.144

Highlights:

- Dynamic geofencing adapts boundaries using real-time geotag data.

- Multi-vertex polygon formed from outermost geotag points.

- Achieves up to 87% alignment with benchmark geofence.

- Stable under ±10% distance tolerance.

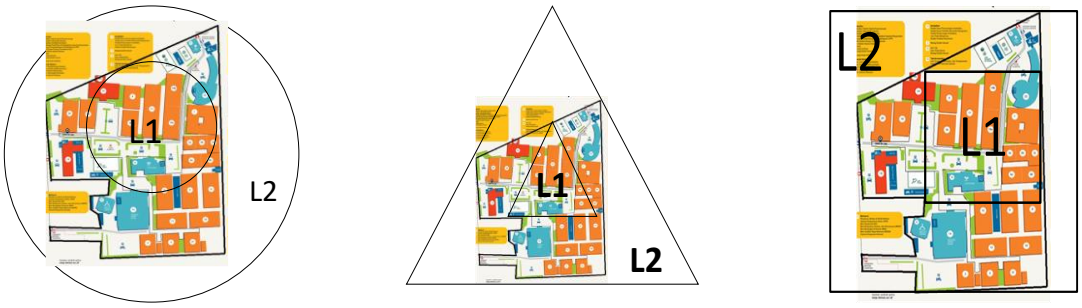
Abstract: Geofencing systems commonly rely on circular boundaries, which are often insufficient for representing irregular spatial environments such as university campuses. Polygon-based geofencing provides greater spatial flexibility; however, static polygon boundaries become ineffective when user activity areas change over time. This study proposes a dynamic polygonal geofencing approach that adapts its boundary based on user geotagging data. The proposed system constructs a multi-vertex polygon by identifying outermost geotag points relative to a centroid and validating them using proximity-based rules within a predefined distance tolerance. Distance calculations are performed using the Haversine formula. A static polygon derived from benchmark coordinates is used as a reference for evaluation. Experimental results from six validation scenarios demonstrate that the dynamic geofence progressively improves its spatial alignment with user activity as more geotag data become available. Partial-data testing achieves vertex alignment rates between 33% and 67%, while whole-data testing reaches up to 87% alignment with the six-vertex baseline. Sensitivity analysis further shows that the system remains stable under ±10% tolerance variation. These findings indicate that the proposed method provides a stable and adaptive solution for spatial presence verification in dynamic campus environments.

Keywords: Dynamic Geofencing; Polygon Boundary; Attendance Validation; Geotagging; Spatial Clustering

---

1. Introduction

Many users utilize geofencing for specific purposes, such as monitoring people, animals, and items, whether they are in motion or not. In essence, a geofence is designed to scan objects within a virtual region and will lose its ability to detect them once they move outside that region (Cho et al., 2025). To construct a geofence, a region must first be defined. Once it is determined, the system allows users to generate an area as a geofence boundary (Everbridge, 2025). Most users prefer circular geofences as a generic shape because they are simple to construct; they require only a centroid point and a radius to define the area (Garcia, 2022). Even when the centroid position changes, the circular geofence remains valid as long as the radius is maintained (Gokhale et al., 2022). The other shapes, such as triangular and rectangular, were rarely implemented, but they still have simple methods for calculating area using one or two parameters. However, the generic shape of geofences often leaves unused areas on fields, especially on irregular spaces like university campuses, where buildings and boundaries rarely align with these simple geometries. Figure 1 shows a geofence using generic shapes which resulting to non-optimal geofence as there is area that are undesirable, if we use small area like L1, there will be area that are not covered by the geofence, on the other hand if we use big area like L2 there will be area that are undesirable but inside the geofence.



(a) Circular **(b)** Triangular **(c)** Rectangular

Figure 1. The generic shape of a virtual geofence placed to a campus sitemap

Figure 1 illustrates the unused spatial areas produced by simplified geofencing boundaries, including circular (L1 and L2) on Figure 1a, triangular (Figure 1b), and rectangular (Figure 1c) configurations. Such unused areas indicate potential misclassification of user presence when the geofence does not conform to the actual spatial layout of the classroom or campus environment. To accurately validate attendance during teaching and learning activities, the geofence boundary should represent the true spatial shape of the room before users perform

geotagging activities. Otherwise, predefined or overly simplified boundaries may incorrectly detect users as being outside the designated area. Several previous studies have investigated campus attendance validation using user geotagging data, commonly relying on predefined circular or polygonal boundaries in static geofencing systems. Enikuomehin et al. (2021) validated student attendance using a predefined allowable range determined by instructors, which effectively corresponds to a circular geofence. Similarly, Eweoya et al. (2025) employed a predefined radius of approximately 100 m to support attendance validation in large lecture halls. While radius-based geofencing remains dominant in attendance systems published since 2020, polygonal boundary representations have been more extensively explored in other spatial monitoring contexts. For instance, Bu et al. (2021) applied convex and concave hull extraction techniques to GPS point collections for modelling activity spaces. However, the explicit application of static or dynamic polygonal geofences for attendance validation remains limited in existing literature. Sasaki et al. (2025) introduced a data-driven point-of-interest notifier using dynamic circular geofences optimized via genetic algorithms, demonstrating the potential of adaptive boundaries beyond static radius-based approaches. Building on this motivation, dynamic polygon-based geofencing offers a promising alternative for dense campus environments, where radius-based geofences may incorrectly validate attendance for users located in neighbouring buildings. This study introduces a rule-based edge-validation method for dynamic polygon geofencing, which updates boundary edges using clustered-distance logic rather than time-based updates or hull reconstruction. The proposed approach enables activity-triggered boundary adaptation inferred directly from real-time user geotagging data, allowing the geofence to remain constrained within spatial limits that reflect actual classroom or campus layouts. Table 1 presents the state of the art of related research in geofencing-based attendance and spatial validation systems. Although this study is demonstrated using a campus attendance scenario, the underlying logic is generalisable as a spatial presence verification framework. The proposed rule-based dynamic polygon mechanism can be extended to other application domains, such as industrial safety monitoring to restrict unauthorised access to hazardous zones and secure logistics systems for validating asset presence within irregular operational areas. These examples indicate that the contribution of this work extends beyond a single application context.

Table 1. State of the art of geofencing-based attendance and spatial validation research

Study	Geofence Shape	Boundary Update	Adaptation Level	Application
Enikuomehin et al. (2021)	Circular	Static	None	Attendance
Eweoya et al. (2025)	Circular	Static	None	Attendance
Bu et al. (2021)	Polygon	Static (Hull)	Global	Activity Space
Sasaki et al. (2024)	Circular	Dynamic (Radius)	Global	POI / Marketing
This study	Polygon	Dynamic (Rule-Based)	Edge-level	Spatial Attendance Verification

2. Materials and Methods

In this approach, geofence boundaries are generated by identifying the geotagged points farthest from the centroid and validating them through proximity-based clustering within a 10 m radius, in accordance with the SNI GPS accuracy standard (Kakalang, 2022). The validated clusters define edge points that form the polygon boundary, ensuring that the resulting geofence accurately represents the monitored area. Distance calculations between user coordinates are performed using the Haversine formula, while real-time location tagging enables continuous boundary refinement. Accordingly, geofence boundaries are refined using centroid-based clustering validation by adjusting individual area edges rather than modifying the entire boundary simultaneously. Figure 2 illustrates the general idea of said geofencing (example: triangular geofence as basic geofencing)

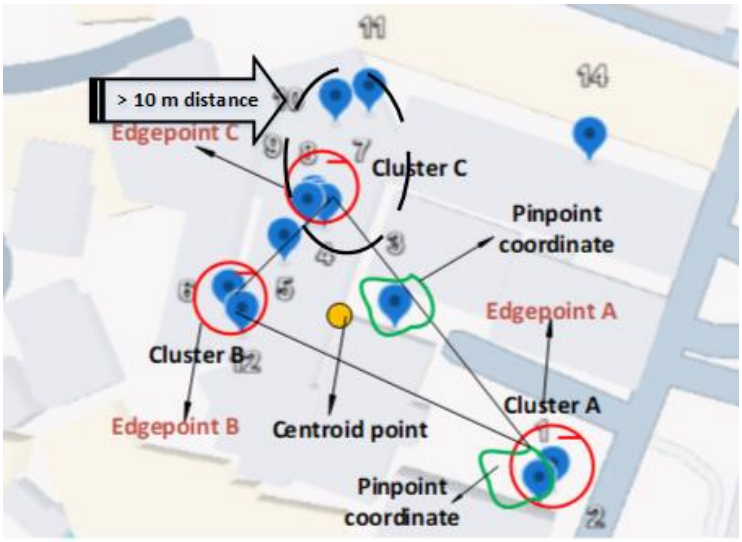


Figure 2. General idea of a Geofencing system

Figure 2 illustrates the spatial configuration of the geotagging process, including pinpoint

coordinates, edge points, centroid points, and their associated clusters. The example shows a simple geofence generated as a triangular polygon. Pinpoint coordinates are obtained by continuously logging users' live GPS positions, enabling dynamic adjustment of the monitored area for both on-campus and off-campus scenarios (Chaudhari, 2024)

2.1. A Static Polygon Boundary

A polygon-based boundary provides greater flexibility for geofencing applications, as it can represent complex spatial shapes more accurately than generic geometric forms. In this study, a static polygon boundary is defined using benchmark coordinates as fixed geographical reference points. Benchmarks are established geodetic markers that indicate precise spatial positions within a given area (Taufiqqurahman et al., 2025). These markers are often physically materialised to ensure long-term stability; even in the event of natural disturbances such as earthquakes, benchmarks remain reliable spatial references that enable affected regions to be realigned to their original coordinates (Tomaszewski, 2021).

The ITENAS campus benchmark dataset consists of 34 benchmark points (Premitasari, 2023). Among these, the ten outermost benchmarks (A–J) define the campus perimeter and are used to construct the static geofence. These outer boundary points serve as default edge points and form a ten-node polygon representing the campus area, as illustrated in Figure 3. This static polygon is used as a fixed spatial boundary for comparison and evaluation purposes throughout the study.

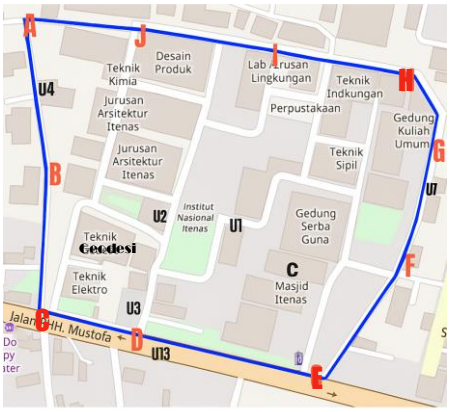


Figure 3. Ten Benchmarks of the Static ITENAS Boundary

As a static polygon scenario (Figure 3), the ten-node benchmark polygon is used to represent the campus boundary. In this configuration, the geofence shape remains fixed regardless of user movement. If a recorded position lies within the static boundary, attendance is considered valid; if the position falls outside the boundary, attendance is invalidated. Although this static

geofence configuration has not been experimentally evaluated, it is included as a reference representation covering an area of approximately 6.34 hectares. This static polygon is used solely for conceptual comparison with the proposed dynamic geofencing approach, which operates within a smaller, activity-driven target area of approximately 1.8 hectares based on user positioning, as shown in Figure 6.

2.2 Haversine formula

The Haversine formula is used to compute the great-circle distance between two geographic points on the Earth's surface, assuming a spherical Earth model (Noviarianto et al., 2023; Yunardi et al., 2024). The input parameters consist of latitude and longitude values obtained from GPS measurements and expressed in decimal degrees. These values are converted into radians (Radian), as required by the trigonometric operations in the Haversine formulation, as shown in Equation (1).

$$\text{Radian} = \frac{\text{Deg} * \pi}{180^\circ} \tag{1}$$

Given the number π =3.141592653589793, latitude and longitude values in degrees (Deg) are converted into radians before the calculation. After conversion, the system computes the central angle between two points on the Earth's surface. (Equation 2) which represents the angular distance used to calculate the great-circle distance

$$\Delta\sigma = 2\arcsin\left(\sqrt{sin^2\left(\frac{\Delta\varphi}{2}\right) + \left(cos\varphi_A cos\varphi_B sin^2\left(\frac{\Delta y}{2}\right)\right)}\right) \tag{2}$$

φ described as latitude in radian, φ_A equal to the user's latitude in radian, and φB describes the centroid latitude in radian with $\Delta=\varphi_B-\varphi_A$. Parameter y equal to longitude in radians with $\Delta y = y_B - y_A$ (Ikasari et al., 2021). Equation 3 shows the final output related to the separate distance between two points

$$D = R * \Delta\sigma \tag{3}$$

D means distance in km, and R is specified as the Earth's radius, with an average radius equal to 6.371 km, and denotes the central angle.

2.3 System Design

The system design updates the polygon boundary by selecting new edge points derived from user geotag coordinates. An edge point candidate must satisfy two conditions: (1) it is farther from the current centroid than previously selected points, declared as **centroid distance** on Figure 4 , and (2) it maintains local consistency by having at least one neighbouring point within the predefined 10 m tolerance (**cluster distance**) as defined in the static polygon boundary section (Min Allah et al., 2021; Premitasari et al., 2023). When at least two valid edge points are identified within the same region, the system connects them to update the polygonal boundary. The centroid is then recalculated based on the updated polygon, allowing the geofence to adapt dynamically to changes in user spatial distribution. Figure 4 illustrates the overall workflow of this process, starting from GPS coordinate acquisition via the web interface to dynamic boundary updates

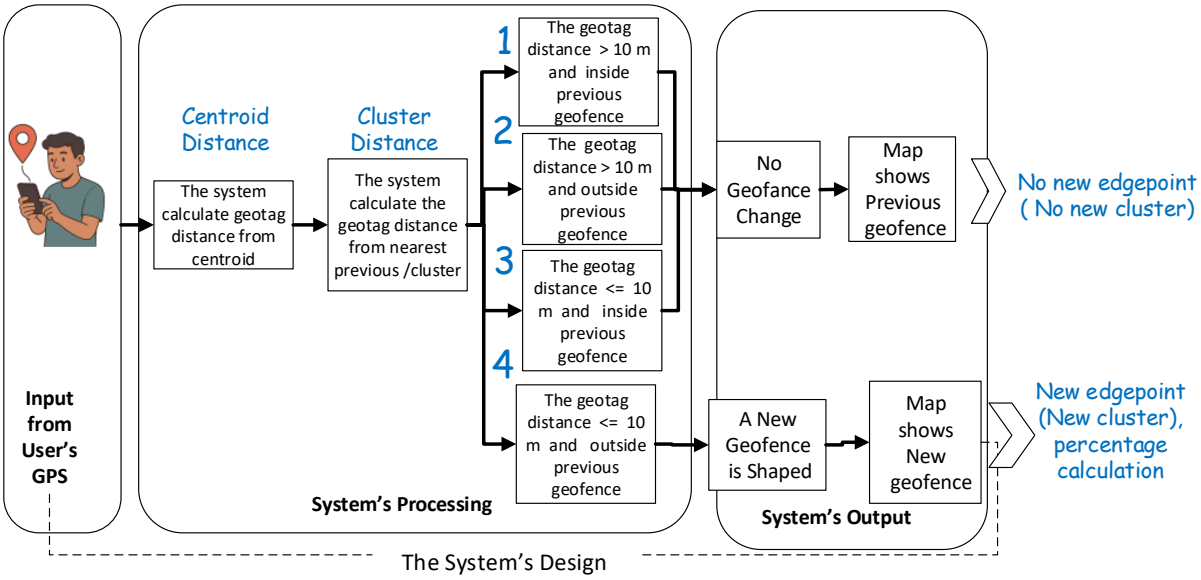


Figure 4. Block diagram of the proposed system

As shown in Figure 4, the system executes a decision logic (shown in a parallel box labelled 1, 2, 3, and 4) to determine whether the geofence boundary remains unchanged or needs reshaping. The logic is defined as the number to follow:

1. **If the geotag distance > 10 m but inside previous geofence** → No geofence change resulting in displaying the map with the last boundary
2. **If the geotag distance > 10 m and outside previous geofence** → A potential new edge point is detected, but still results in displaying the map with the previous boundary
3. **If the geotag distance ≤ 10 m and inside previous geofence** → The point may be located inside the geofence, and no boundary update

4. **If the geotag distance ≤ 10 m and outside previous geofence** → A new geofence boundary is identified, resulting in a new geofence

It begins when the user inputs coordinates, which are then displayed as pinpoints on a map. For each pinpoint, the distance to the centroid is calculated; if the new distance exceeds the previous one, the system measures the distance between the nearest pair of pinpoints. Points within 10 m of each other are grouped into a cluster. For each cluster, the system identifies edge points, defined as the points farthest from the centroid. If there are at least two edge points, the system draws a new polygon representing the geofence boundary. A new centroid is then recalculated—if it remains unchanged, no update occurs; if it shifts, a new geofence is generated. Figure 5 illustrates the system flowchart.

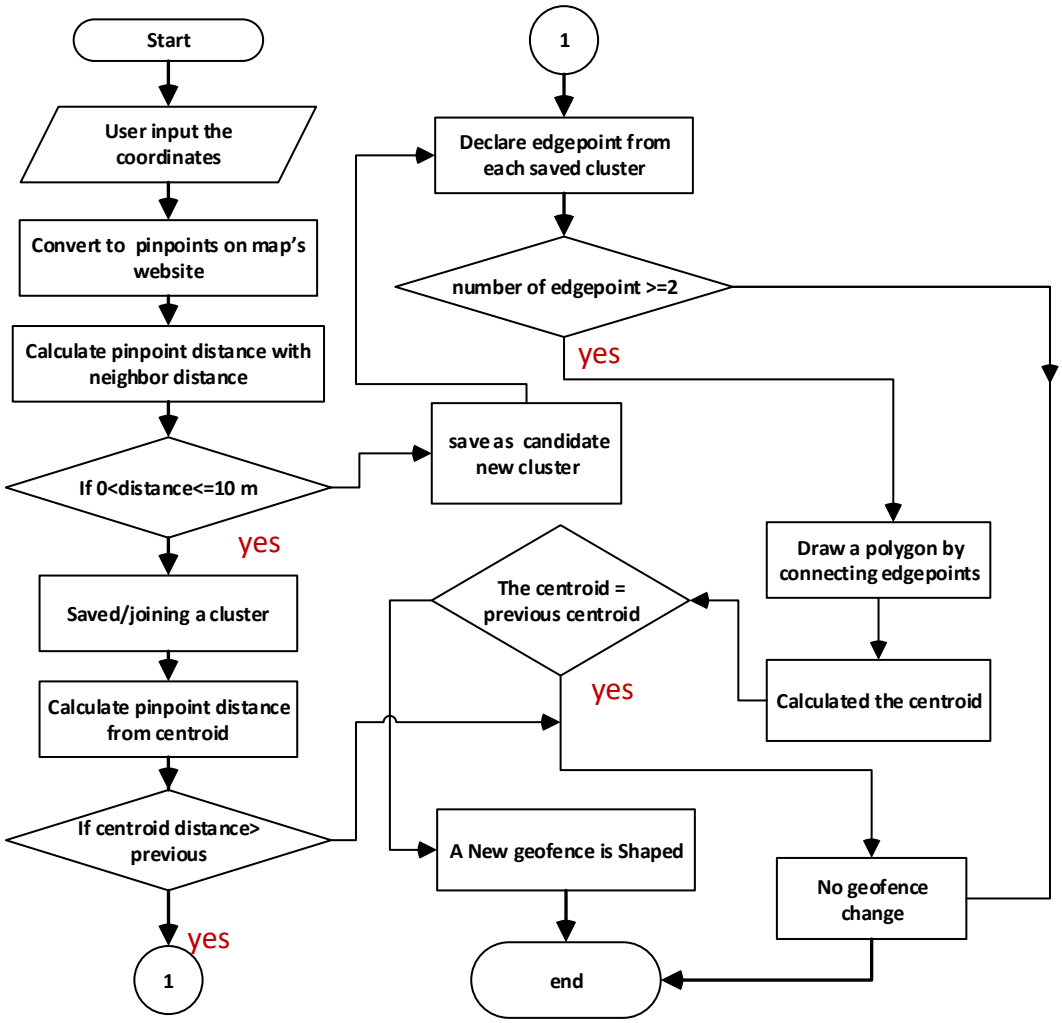


Figure 5. System Flowchart

This study implements a dynamic geofencing approach, which Sasaki et al. (2024) also explored, in which geofence boundaries can change and shift based on the radius of a point of interest (POI) relative to the user's location. The work by Sasaki et al. focuses on analysing

user geolocation behaviour for marketing purposes**.** However, Sasaki et al. do not employ polygon-based geofencing**.** The geofences generated in their study are limited to circular shapes, in which the geofence dynamically updates its centre and radius but does not support vertex- or edge-level boundary adaptation. The step by step below shows simple pseudocode of general system as abstraction algorithm for Dynamic Geofence

Require: *P* **=** Set of geotag points
***Ensure*: G =** geofence polygon
1: Cluster *P* into C
2: Initialize an empty edge set *E*
3: For each $C_i \in C$, **do**
4: Apply rules (R1–R6) to select valid edge point Vi
5: If $|V_i| \geq 2$, then add V_i to edge set *E*
8: End **for**
9: Construct polygon *G* from E
10: Return *G*
Hint: *P* (pinpoint), **G** (Geofence shape), ***C*** (Clusters). ***E*** (Edge), and ***V*** is for Vertex.

3. Result and Discussion

Based on repeated geotagging experiments involving 40 participants, a total of 232 pinpoint locations were collected from multiple attendance sessions. The spatial distribution of these pinpoints was used to generate a heatmap representing areas of high user presence. This heatmap provides an empirical visualization of point concentration, which supports the selection of baseline edge points. Figure 6a illustrates the heatmap derived from the accumulated geotag distribution, while Figure 6b shows the corresponding six-vertex baseline polygon constructed to enclose the dominant activity region.

(a) Pinpoint heatmap (b) Six vertex Baseline

Figure 6. Pinpoint heatmap and six vertex baselines generated by GPS users' mobile phone

From Figure 6, it can be concluded that the baseline reference is constructed by computing the centroid of aggregated high frequency geotag points, which represents the dominant centre of user activity. No clustering algorithm is applied at this stage, as the objective is not to identify spatial clusters but to define a representative reference location. For GPS accuracy, satellite signal strength is visually represented using bar indicators provided by the GPS Test application. These indicators reflect the number of available GNSS satellites contributing to positioning. In accordance with GNSS positioning principles, at least three satellites are required to obtain horizontal coordinates, while four satellites are required to estimate altitude (Kaplan, 2023). Based on the accumulated geotagging data, the resulting spatial distribution converges toward a six-edge configuration, as illustrated in Figures 6a and 6b. The reduction from the original ten benchmark points is primarily due to sampling limitations, since the experimental data cover only the classroom area within the broader ITENAS campus boundary. Consequently, a six-vertex polygon is adopted as the baseline edge configuration for subsequent evaluation, representing the dominant activity region. This six-vertex baseline serves as a reference to evaluate whether the system-generated dynamic geofence can reproduce a comparable spatial shape during testing. The configuration remains valid under the predefined constraint that the number of edge nodes does not exceed the original ten-node campus benchmark, as discussed earlier

3.1 Testing Result

The developed system was evaluated using 232 pinpoints collected across the campus area. Six validation rules (R1–R6) were applied, reflecting typical classroom conditions with 10–20 students and spatial constraints based on classroom dimensions (20 m × 25 m). The rules are defined as follows: R1 (10 points, 10 m radius), R2 (15 points, 10 m radius), R3 (10 points, 15 m radius), R4 (15 points, 15 m radius), R5 (10 points, 20 m radius), and R6 (15 points, 20 m radius). The dataset was divided into five testing scenarios comprising 50, 50, 50, 50, and 32 pinpoints, respectively. Two evaluation strategies were employed. In the partial-data approach, each batch was processed independently without incorporating data from previous tests. In contrast, the whole-data approach cumulatively incorporated pinpoints from prior batches, resulting in progressive evaluations of 50, 100, 150, 200, and finally 232 pinpoints. At each testing step, clusters were formed according to the defined rules, and qualified data points were identified and evaluated against the centroid and nearest benchmark to determine their eligibility as new edge points. Table 2 summarizes the number of qualified data points and validated clusters for both partial and whole data scenarios

Table 2. Partial Data (1) VS Whole Data (2) for Number of Cluster.

Rule	Test 1				Test 2				Test 3				Test 4				Test 5			
Testing & Result	Qd		C		Qd		C		Qd		C		Qd		C		Qd		C	
	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
R1	14	14	5	5	19	36	5	10	2	55	2	17	8	63	2	19	0	70	-	21
R2	9	9	3	3	10	22	3	6	0	41	-	11	0	41	-	11	0	47	-	12
R3	38	38	5	5	31	74	6	13	16	108	4	21	14	123	5	27	0	138	-	30
R4	26	26	2	2	16	46	3	7	4	81	2	16	0	81	-	16	0	91	-	19
R5	40	40	5	5	31	77	5	11	20	120	5	7	22	142	7	25	0	161	-	28
R6	28	28	2	2	18	52	2	5	19	113	5	17	0	114	-	18	0	132	-	23

In Table 2, Qd denotes the number of qualified data points, C represents the number of validated clusters, while subscripts 1 and 2 indicate partial-data and whole-data evaluations, respectively.The number of clusters resulting on Table 2 generated varies, affected by the rule of validation, with conclusion if there is no qualified data in both test , yet the number of cluster recorded as null. Those clusters then act as the benchmark that is used to construct the geofencing area. Tables 3 and 4 show the results of each test which each rule represents Intersection of Union against six baseline vertexes (Figure 6b).

Table 3. The testing result with the first way of using data (Partial Data)

Rule	Test 1 (50)	Test 2 (50)	Test 3 (50)	Test 4 (50)	Test 5(32)
R1 (IoU = 0.20)					
R2 (IoU = 0)					
R3 (IoU=0.58)					
R4 (IoU =0.15)					
R5 (IoU = 0.58)					
R6 (IoU = 0.15)					

Tables 3 and 4 show all rules in Test 1 and Test 2, concluding that no geofence was formed because the system detected fewer than three edge points. The blue area represents the geofence shape once the minimum number of edge points is reached. The total area of each geofence was not calculated, as this information is not required for constructing a dynamic polygon in the campus attendance system. To determine whether a pinpoint lies inside or outside the previously detected geofence, the system evaluates each point relative to the current polygon boundary, using centroid distance as a global reference and cluster-based proximity as a local consistency check. Table 4 shows the results from the test using the second method, with each column in the table including the cluster found in the test, the new benchmark (if applicable), and the new centroid (if applicable) along with the IoU, following Table 3.

Table 4. The testing result with the second way of using data (Whole Data)

Rule	Test 1 (50)	Test 2 (100)	Test 3 (150)	Test 4 (200)	Test 5(232)
R1 (IoU = 0.38)					
R2 (IoU = 0.07)					
R3 (IoU = 0.44)					
R4 (IoU = 0.32)					
R5 (IoU = 0.58)					
R6 (IoU = 0.69)					

Based on the analysis of Table 2 and Table 3 which utilizes partial datasets, shows that geofencing shape changes predominantly occur under Rules R1, R3, and R5. These results indicate that these rules provide stronger indicators for spatial validation in campus attendance systems compared to R2, R4, and R6, which consistently produce identical geofencing shapes across Test 1 to Test 6. In contrast, the second way of using data (Table 2 and Table 4), reveals that geofencing changes do not occur only under Rule R2, while the remaining rules demonstrate adaptive boundary formation (Table 4). This finding indicates that the evaluation based on the whole data is more reliable and can serve as a robust reference for most validation rules (except R2). Moreover, this approach can generate up to six edge points, matching the number of desired edge points, whereas the partial-data approach results in more limited geofence adaptations. This metric was calculated by averaging the distance comparison results for each collected pinpoint and the benchmark. To ensure that the geofence operates in an optimal manner, a minimum number of users is required, as adding more users may affect the resulting geofence shape. Based on the results shown in Tables 3 and 4, at least 100 users (pinpoints) are required for the dynamic geofence to function effectively. The effectiveness of the dynamic shape becomes evident after Test 2, when the number of pinpoints reaches 100 in both evaluation scenarios. Table 5 summarizes the comparison of edge-point detection results obtained from the first and second evaluations using both partial-data and whole-data approaches. For each rule (R1–R6), edge-point detection outcomes are reported for the six baseline vertices (A–F). When the results from the first and second evaluations are identical for a given edge point, a single symbol (✓ or ✗) is shown. When the outcomes differ between

the two evaluations, the result is expressed using paired notation (✗/✓ or ✓/✗), where the first symbol corresponds to the first evaluation and the second symbol corresponds to the second evaluation. The 'Edge Points' column reports the total number of detected edge points relative to the six baseline vertices, expressed as first/second evaluation results. Similarly, the 'Frequency' row indicates how often each edge point appears across all validation rules (R1–R6) for both evaluations. The 'Edge Points' column reports the total number of detected edge points relative to the six baseline vertices, expressed as first/second evaluation results. Similarly, the 'Frequency' row indicates how often each edge point appears across all validation rules (R1–R6) for both evaluations Table 5 summarizes the results from two separate evaluation phases. Phase I (1st) represents baseline testing with 40 users (232 geotag points), while Phase II (2nd) represents extended testing with 110 users (409 geotag points) collected from combined indoor and outdoor environments.

Table 5. Edge-point comparison test results (partial data vs. whole data)

	The Partial Data (1st/2nd)							The Whole Data (1st/2nd)						
Rule	**A**	**B**	**C**	**D**	**E**	**F**	**Edge Points**	**A**	**B**	**C**	**D**	**E**	**F**	**Edge Points**
R1	✓	×/✓	×	×	✓	×	**2/3**	✓	✓	×	×	✓	✓	**4/4**
R2	✓	×/✓	✓/×	×	×	×	**2/2**	✓	×/✓	✓	×	×	×	**2/3**
R3	✓	×/✓	×	✓	✓	✓	**4/5**	✓	✓	×	✓	✓	✓	**5/5**
R4	✓	×/✓	✓	✓	×	×	**3/2**	✓	✓	×/✓	✓	×	✓	**4/5**
R5	✓	×/✓	×	✓	✓	✓	**4/5**	✓	✓	×	✓	✓	✓	**5/5**
R6	✓	×	✓	✓	×	×	**3/3**	✓	✓	×	✓	×	✓	**4/4**
Frequency	**6/6**	**0/5**	**3/1**	**4/3**	**3/3**	**2/2**		**6/6**	**4/6**	**1/2**	**4/4**	**3/3**	**5/5**	

Based on the experimental results, Rule R2 consistently yields the lowest number of detected edge points across both evaluation phases. In the partial-data evaluation, R2 detects only two vertices in both Phase 1 and Phase 2. In the whole-data evaluation, R2 also detects two vertices in Phase 1 and increases to only three vertices in Phase 2. This result aligns with Table 6, where higher spatial dispersion (SD) corresponds to higher accuracy; notably, R2, which shows the lowest accuracy, also has the lowest SD (1.3157356921833E-5). Higher SD values provide better suitability for real campus applications, as GPS capture during class sessions requires greater tolerance for positional variation caused by indoor environments and the dispersed nature of classroom activities.

Table 6. Average SD Value

Rule	Spatial Dispersion

R1	0,000014627313799341
R2	0,000013157356921833
R3	0,000013847786226812
R4	0,000013586507238485
R5	0,000014745045134639
R6	0,000014924666051805

To further validate the results, performance evaluation was conducted using the Root Mean Square Error (RMSE), which achieved its lowest value at 100 data points, with an RMSE of 5.8 m (Figure 7). RMSE is commonly used to evaluate GNSS positioning accuracy because it emphasizes larger positional deviations, which are critical in assessing spatial reliability. As discussed by Amerudin (2024), RMSE is particularly suitable for GNSS performance evaluation due to its sensitivity to error magnitude and its ability to reflect the impact of positioning noise. This characteristic makes RMSE appropriate for assessing spatial dispersion and centroid stability in location-based clustering.

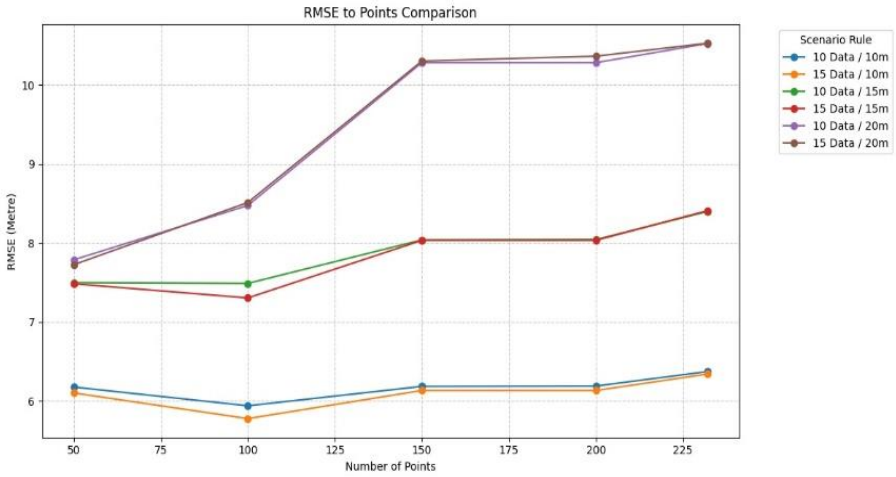


Figure 7. RMSE-to-point comparison

This characteristic is particularly relevant in location-based systems where positional errors are not uniformly distributed. Table 7 compares accuracy results between Phase 1 (first baseline testing) and Phase 2 (second non-baseline testing) under partial- and whole-data evaluations. Rules R3 and R5 consistently achieve the highest accuracy, reaching 83% in whole-data testing for both phases, indicating stable geofence reconstruction as data accumulate. Other rules (R1, R4, and R6) show moderate performance, with accuracy generally improving from partial to whole data and from Phase 1 to Phase 2. As previously analysed in Table 5, Rule R2 yields the lowest number of detected edge points. This limitation is reflected in Table 7, where R2 consistently records the lowest accuracy across both Phase 1 and Phase 2.

Table 7. Accuracy comparison test results

Rules	Baseline Testing (1st Testing)		Non-Baseline Testing (2nd Testing)	
	Partial Data	**Whole Data**	**Partial Data**	**Whole Data**
R1	33%	67%	50%	67%
R2	33%	33%	33%	50%
R3	67%	83%	83%	83%
R4	50%	67%	33%	83%
R5	67%	83%	83%	83%
R6	50%	67%	50%	67%

Another way to evaluate the geofence is by analysing the polygon area. As described by Det Petris et al. (2024), polygon area is derived from vertex coordinates, and its computation includes uncertainty that propagates from the uncertainty of those coordinates. Each polygon area is calculated and then compared with the static geofence area (Figure 4) and the desired or target geofence area (Figure 5). The evaluation is performed using the Area Difference Ratio (ADR) and Area Improvement Ratio (AIR). The static and desired geofence areas have values of 21,229.1424 and 18,497.7228, respectively. Both static and desired benchmark coordinates represent approximate areas rather than exact values. Area values obtained from all generated geofences for each rule under the partial-data evaluation are compared with both static and desired benchmark areas, producing ADR and AIR values as shown in Table 8 for partial data.

Table 8. Partial data geofence area evaluation

Rules	Area	ADR	AIR
R1	4636.0348815692	0.7493726691364353,	-4.074902280920429
R2	0	1.0,	-5.772200922097497
R3	9810.8935914653	0.4696161402946299	-2.18033485833516
R4	2953.5878728922	0.8403269466334544	-4.690862922854453
R5	9810.8935914653	0.4696161402946299	-2.18033485833516
R6	3141.5564445468	0.8301652324523731	-4.622045752707244

ADR evaluates area accuracy by measuring how different the resulting geofence is from the desired geofence; a smaller value indicates closer agreement with the desired geofence. As shown in Table 8, Rule R2 records an ADR value of 1.0, which is the highest value, indicating that the geofence generated under R2 in the partial-data evaluation differs substantially from the desired geofence. In contrast, Rules R3 and R5 yield the lowest ADR value of 0.4696, indicating that these geofences are the closest to the desired geofence. AIR performs better when its value is closer to 1; however, almost all AIR values in Table 9 are negative, except R4 and R6 with the lowest value observed for R2 and higher values for R3 and R5. This occurs because the generated geofence areas are relatively smaller than the desired geofence and significantly smaller than the static geofence.

Table 9. Whole data geofence area evaluation

Rules	Area	ADR	AIR
R1	8376.933293481	0.5471370501301015	-2.705322035404777
R2	1753.983395462	0.9051784147416004	-5.130050094975817
R3	8802.903878779	0.5241087784395492	-2.5493699526277074
R4	16270.38849805	0.12041126834689307	0.1845506974702416
R5	8802.903878779	0.5241087784395492	-2.5493699526277074
R6	17600.17652052	0.0485219855437758	0.6713993647584401

Table 9 shows the area, ADR, and AIR values of the geofence generated using the whole-data evaluation. Like the partial data results, Rule R2 records the highest ADR value, indicating that the generated geofence differs substantially from the desired geofence. In contrast to the partial-data evaluation, Rule R6 achieves the lowest ADR value (0.0485), indicating the closest approximation to the desired geofence. A similar pattern is observed for AIR values, where R2 records the lowest value and R6 records the highest value. While Bu et al. (2021) employ concave hull–based boundary extraction to model activity spaces from continuous GPS trajectories, the proposed approach focuses on attendance validation using discrete geotag points. Under a convex-hull formulation, Rule R2 exhibits unstable behaviour due to an insufficient number of valid outer points for reliable boundary formation, which is consistent with its low vertex detection and accuracy results. In addition, GPS Test measurements collected from 52 volunteers indicate an average positioning accuracy of 3.5 m and a mean signal-to-noise ratio (SNR) of 26.375, confirming that GNSS accuracy conditions during the experiments were well within the tolerance range used in this study. Table 10 presents a sensitivity analysis using different GPS accuracy tolerance values of 9 m, 10 m, and 11 m.

Table 10. Sensitivity Data Analysis

Tolerance (T) (m)	**Inside Rate (%)**	**Outside Rate (%)**	**Truncated RMSE (m)**
9	71.12	28.88	5.75
10 (SNI baseline)	74.57	25.43	6.19
11	77.16	22.84	6.59

By adopting this convex-hull–based boundary construction with rule-based constraints on outermost points, the proposed method prioritises robustness and boundary stability rather than fine-grained mobility representation. In practical attendance implementations, lecturer and student identifiers may be incorporated as validation constraints, where a valid cluster requires at least two pinpoints within a 10 m radius, including one lecturer and one student.

4. Limitation and Recommendation

The current study does not consider the altitude of the coordinate and the result generated in daytime campus working hours. It might take different if the data collected at night hours

(18.00-06.00). Future research should include the altitude of the coordinate to further increase the accuracy of each coordinate.

5. Conclusion

Experiments conducted with 40 users in baseline testing and 110 users in non-baseline testing demonstrate that Rules R3 and R5 consistently achieve the highest accuracy, reaching up to 83% in whole-data evaluations. These results indicate that rule configurations with moderate spatial dispersion enable sufficient edge-point formation for stable polygon reconstruction. In contrast, Rule R2 consistently records the lowest accuracy, improving only from 33% to 50%, due to its limited number of detected edge points. Convex-hull stability improves after approximately 100 accumulated pinpoints, coinciding with the lowest observed RMSE (≈5.8 m), indicating that instability in R2 is primarily caused by insufficient outer-point formation rather than GNSS positional error. Sensitivity analysis confirms that the system remains stable under a ±10% variation around the 10 m distance tolerance, supporting its suitability for indoor GNSS-based attendance validation. GPS Test measurements from 52 volunteers further indicate an average positioning accuracy of 3.5 m and a mean signal-to-noise ratio (SNR) of 26.375, confirming that the results are not affected by poor GNSS signal quality.

Credit Author Statement

Conceptualization and methodology, Premitasari, M.; software and validation, Dihardja, R.M., Yudha., M.K., resources and data curation, Premitasari, M., Dihardja. R.M., Maryanto T.I.; writing—original draft preparation, Diharja, R.M; writing—review and editing, Premitasari, M., Diharja. R.M., Maryanto T.I

Conflicts of Interest

The authors declare no conflict of interest.

Declaration of AI Use

Artificial intelligence tools (ChatGPT) were used to assist in paraphrasing to increase clarity and enhance manuscript readability. All sources, theories, and scientific statements were manually verified using original publications.

References

Abdirahman, A. A., Hashi, A. O., Dahir, U. M., Elmi, M. A., & Rodriguez, O. E. R. (2023). Enhancing vehicle tracking through SMS: A cost-effective approach integrating GPS and

GSM. *SSRG International Journal of Electrical and Electronics Engineering*, *10*(9), 29–39. https://doi.org/10.14445/23488379/IJEEE-V10I9P104

Amerudin, S. (2024). Recent methods for evaluating GNSS receiver accuracy and reliability. https://people.utm.my/shahabuddin/?p=7901

Andrews, J. G., Claussen, H., Dohler, M., Rangan, S., & Reed, M. C. (2021). Modeling and analyzing millimeter wave cellular systems. *IEEE Communications Magazine*, *59*(2), 40–46. https://doi.org/10.1109/MCOM.001.2000003

Arief, R., Renaldi, F., & Umbara, F. R. (2020). Dynamic geofencing in supervision of seller performance. In *Proceedings of the 5th NA International Conference on Industrial Engineering and Operations Management* (pp. 1389–1395). IEOM Society International.

Bu, J., Yin, J., Yu, Y., & Zhan, Y. (2021). Identifying the daily activity spaces of older adults living in a high-density urban area: A study using smartphone-based global positioning system trajectory in Shanghai. *Sustainability*, *13*(9), 5003. https://doi.org/10.3390/su13095003

Chaudhari, B. (2024). A data-driven approach to dynamic geofencing for sustainable and profitable fisheries. *International Journal of Innovative Science and Research Technology*, *9*(9), 2026–2034. https://doi.org/10.38124/ijisrt/IJISRT24SEP1401

Cho, Y., Shin, M., Man, K. L., & Kim, M. (2025). SafeWitness: Crowdsensing-based geofencing approach for dynamic disaster risk detection. *Journal of Sensor and Actuator Networks*, *9*(3), 156. https://doi.org/10.3390/jsan9030156

Enikuomehin, A. O., & Dosumu, O. U. (2021). Geofencing based attendance monitoring system. *Research Inventy: International Journal of Engineering and Science*, *11*(1), 42–46. http://www.researchinventy.com

Everbridge. (2025). *Everbridge control center 5.73 reference guide*.

Eweoya, I., Adeniyi, O. J., Awoniyi, A. O., Mgbeahuruike, E., Adewuyi, J. O., Adigun, T., & Mensah, Y. A. (2025). Design and implementation of a university attendance management system using geo-fencing. *Asian Journal of Computer Science and Technology*, *14*(1), 28–46. https://doi.org/10.70112/ajcst-2025.14.1.4323

García, M. B. (2022). Location-based marketing using mobile geofencing: Lessons learned from a user-centered application development research. *International Journal of Technology Marketing*, *17*(1), 1–29. https://doi.org/10.1504/IJTMKT.2022.10047566

Gokhale, P., Rasal, V., Amberkar, S., & Sonawane, S. (2022). Patient monitoring using geofencing. *International Journal for Research in Applied Science and Engineering Technology, 10*(11), 427–430. https://doi.org/10.22214/ijraset.2022.47339

Ikasari, D., Widiastuti, & Andika, R. (2021). Implementation of the Haversine formula to determine the shortest path using a web-based application for a case study of high school zoning in Depok. *American Journal of Software Engineering and Applications, 10*(2), 19–31. https://doi.org/10.11648/j.ajsea.20211002.11

Kakalang, P. J. (2022). *Determining the position of motor vehicles using the polygon method in a geofencing application* [Undergraduate thesis, Institut Teknologi Nasional Bandung, Indonesia].

Kaplan, E. D., & Hegarty, C. J. (2020). *Understanding GPS/GNSS: Principles and applications* (3rd ed.). Artech House.

Min-Allah, N., Alahmed, B. A., Albreek, E. M., Alghamdi, L. S., Alawad, D. A., Alharbi, A. S., Al-Akkas, N., Musleh, D., & Alrashed, S. (2021). A survey of COVID-19 contact-tracing apps. *Computers in Biology and Medicine, 137*, 104787. https://doi.org/10.1016/j.compbiomed.2021.104787

Muliana, A., & Abdul Aziz. (2024). Using GPS-based learning media to improve understanding of map concepts in geography lessons in the independent curriculum in middle schools in Aceh Province. *International Journal of Education and Computer Studies, 4*(3), 119–129. https://doi.org/10.35870/ijecs.v4i3.3594

Noviarianto, N., Taufiqurrahman, M., & Pribadi, T. (2023). Implementation of low-cost real-time GPS using the Haversine method. In *Proceedings of the 12th International Conference on Sensor Networks (SENSORNETS 2023)* (pp. 97–104). SCITEPRESS.

Patil, I. R., Giri, M. A., Pakhale, P. J., Phalake, A. S., & Phargaonkar, A. A. (2024). Implementation of location-based services using geofencing. *International Journal of Scientific Research in Engineering and Management, 8*(4), 1–5. https://doi.org/10.55041/IJSREM30164

Premitasari, M., Ungkawa, U., & Kakalang, P. J. (2023). Metoda kalibrasi untuk sistem geofencing dengan poligon tertutup. *Rekayasa Hijau, 7*(2), 111–122. https://doi.org/10.26760/jrh.v7i2.112-122

Sasaki, I., Arikawa, M., Lu, M., Utsumi, T., & Sato, R. (2024). Data-driven geofencing design for point-of-interest notifiers utilizing genetic algorithm. *ISPRS International Journal of Geo-Information, 13*(6), 174. https://doi.org/10.3390/ijgi13060174

Taufiqurrahman, M. F., Susilo, Y., Prabawa, S. E., & Yahya, F. (2025). Optimasi waktu pengukur jaring kerangka kontrol horisontal dengan receiver global navigation satellite system. *Jurnal Geodesi Undip, 14*(1), 11–20.

Tomaszewski, B. (2021). *Geographic information systems (GIS) for disaster management* (2nd ed.). Routledge.

Yunardi, E., Magdalena, L., & Febima, M. (2024). Implementation of the Haversine formula method in geographic information systems for searching the nearest sea freight expedition services in East Jakarta. *JAIEA Journal of Artificial Intelligence and Engineering Application*, *4*(1), 11–20. https://doi.org/10.59934/jaiea.v4i1.622

Yunus, N. I. B., & Nasir, S. A. M. (2023). Development of E-Travel mobile application using geofencing technique. *Applied Mathematics and Computational Intelligence, 12*(2), 15–27.