GEO-FENCE BASED ALERT SYSTEM FOR WILDLIFE AND ASSET MONITORING USING QGIS

[1]A. Sowjanya, [2]P.S.N. Deepanvitha, [3]S. Iteesha, [4]T. Poornima, [5]V. Malleswararao

[1] Assistant Professor, [2]Student, [3] Student, [4] Student, [5] Student
[1]Electronics and Communication Engineering,
[1]Lendi Institute of Engineering and Technology, Vizianagaram, India

Abstract : This paper presents the development of a Geo-Fence Based Alert System for wildlife and asset monitoring using QGIS and Python. The system enables real-time location tracking and zone-based monitoring by defining geographical boundaries such as Safe Zone, Warning Zone, and Danger Zone. A point representing the tracked object is continuously monitored using spatial analysis techniques. The system uses the point-in-polygon method to determine the position of the object within predefined zones. Alerts are generated instantly using Telegram Bot API whenever the object crosses zone boundaries. The system also supports movement analysis including distance, speed, and time. The proposed solution is cost-effective, software-based, and eliminates the need for complex hardware. It is highly suitable for wildlife protection, forest monitoring, and asset tracking applications.

Index Terms **-** Geo-Fencing, QGIS, Wildlife Monitoring, Asset Tracking, Spatial Analysis, Point-in-Polygon, Telegram Alerts, Real-Time Monitoring.

I.INTRODUCTION

Geo-fencing is a location-based technology that enables the creation of virtual geographic boundaries for monitoring the movement of objects, animals, or assets. It has gained significant importance in applications such as wildlife monitoring, transportation, and security systems due to its ability to provide real-time tracking and automated alerts [1], [5].

In wildlife monitoring, tracking animal movement is essential to prevent them from entering hazardous zones such as highways or human settlements. Similarly, in asset tracking, it is important to ensure that valuable resources remain within predefined boundaries to avoid theft or unauthorized movement. Traditional tracking systems mainly rely on Global Positioning System (GPS) technology, which provides location information but lacks intelligent boundary detection and real-time alert mechanisms [6], [18].

With the advancement of Geographic Information Systems (GIS), spatial analysis and visualization have improved significantly. QGIS, an open-source GIS platform, enables the creation of geofence boundaries, spatial layers, and real-time monitoring systems. By defining multiple zones such as Safe Zone, Warning Zone, and Danger Zone using polygon layers, the system can effectively analyze the position of a tracked entity and determine its status based on location [2], [13], [15].

Whenever the tracked entity crosses predefined geofence boundaries, the system generates alerts and sends notifications to users. This integration of GIS with Python-based automation and messaging APIs provides a cost-effective and efficient solution for real-time monitoring compared to traditional systems [7], [19], [20]. In addition to monitoring and alert generation, geo-fencing systems also provide valuable insights into movement patterns and behaviour of tracked entities.

By analysing the interaction of objects with different geofence zones, it becomes possible to identify high-risk areas and take preventive measures in advance. Such analysis is particularly useful in wildlife conservation, where understanding movement patterns helps in reducing human-wildlife conflicts and improving habitat management [3], [4].

Furthermore, modern geo-fencing systems focus on automation and real-time communication to enhance system efficiency. The integration of programming tools such as Python enables continuous data processing and dynamic decision-making, while messaging platforms like Telegram provide instant notification capabilities. These advancements improve responsiveness and reduce manual intervention, making geo-fencing systems more reliable and scalable for large-scale monitoring applications [7], [19], [20].

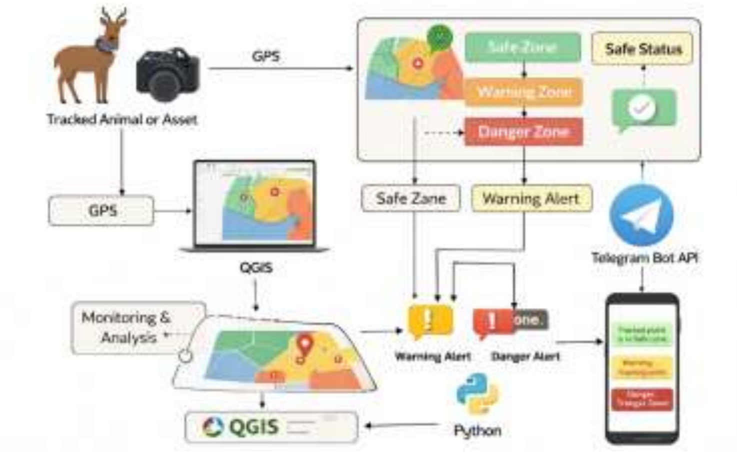

Fig.1 Represents the flow diagram of proposed project

II. NEED OF THE STUDY.

The rapid increase in environmental challenges and security concerns has created a strong need for efficient monitoring systems for wildlife and valuable assets. In wildlife conservation, animals often move into human-populated areas, agricultural lands, or dangerous zones such as highways, leading to accidents and conflicts. Similarly, in asset tracking, the movement of valuable resources outside predefined boundaries can result in loss, theft, or misuse. Therefore, continuous monitoring and timely detection of boundary violations are essential for ensuring safety and effective management [1], [5].

Traditional monitoring systems mainly rely on GPS-based tracking, which provides location information but lacks intelligent analysis and real-time alert mechanisms. These systems cannot identify whether the tracked entity is in a safe or risky zone, making them less effective for practical applications. Additionally, their performance may be affected in areas with weak signal strength, reducing reliability in certain environments [6], [18].

With the advancement of geospatial technologies, geo-fencing has emerged as a powerful approach for defining virtual boundaries and monitoring movement within specific regions. However, many existing systems still lack proper zone classification, visualization, and automated communication features. This creates a need for a system that can not only track location but also analyse movement relative to multiple zones and generate instant alerts [2], [8].

The proposed Geo-Fence Alert System addresses these challenges by integrating QGIS-based spatial analysis with Python automation and real-time communication through the Telegram Bot API. The system enables continuous monitoring, accurate zone detection, and instant alert generation, providing a cost-effective and efficient solution for wildlife monitoring and asset tracking applications [7], [20].
.

2.1 Related Work

Several research studies have been carried out in the field of location-based monitoring and tracking systems. Early approaches mainly relied on Global Positioning System (GPS) technology to monitor the movement of objects and assets. These systems were effective in providing location information; however, they lacked intelligent boundary detection and real-time alert mechanisms, which limited their practical usability in dynamic environments [1], [5].

With the advancement of location-based services, geofencing techniques were introduced to define virtual boundaries and monitor movement within specific geographic areas. These techniques have been applied in various applications such as vehicle tracking, safety monitoring, and attendance systems. However, many of these implementations were limited to simple boundary detection and did not fully utilize advanced spatial analysis capabilities for accurate decision-making [6], [8].

In recent years, the integration of Geographic Information Systems (GIS) with geofencing has significantly improved monitoring accuracy and visualization. GIS tools such as QGIS provide powerful features for creating multiple zones and performing spatial analysis operations. These systems enable better understanding of movement patterns and improve the efficiency of monitoring applications [2], [13], [15].

Furthermore, modern systems have started incorporating automation and real-time communication using programming languages and APIs. Python-based processing and messaging platforms allow automatic alert generation and instant communication. However, many existing solutions still face challenges such as lack of proper integration, high cost, and dependency on hardware components, which reduce their scalability and efficiency [7], [19], [20].

2.2 Problem Identification

Although several tracking and monitoring systems have been developed using GPS and geofencing technologies, they still suffer from multiple limitations. Traditional GPS-based systems mainly focus on providing location coordinates and do not perform intelligent analysis of the tracked entity's position with respect to predefined zones. As a result, these systems are unable to detect critical situations such as boundary violations or entry into dangerous areas in real time [6], [18].

Many existing geofencing systems provide only basic boundary detection without classifying regions into multiple levels such as Safe, Warning, and Danger zones. This lack of detailed zone classification reduces the effectiveness of monitoring, especially in applications like wildlife tracking where different levels of risk must be identified clearly. Additionally, these systems often lack proper visualization tools, making it difficult for users to understand movement patterns and spatial relationships [5], [8].

Another major limitation is the lack of real-time alert mechanisms and automation. In many systems, alerts are delayed or require manual monitoring, which reduces responsiveness and increases the risk of damage or loss. Furthermore, some solutions depend on additional hardware components or complex configurations, leading to higher cost and reduced accessibility for large-scale deployment [1], [5].

To overcome these limitations, there is a need for a cost-effective, software-based system that integrates geofencing with GIS-based spatial analysis and real-time communication. The proposed Geo-Fence Alert System addresses these challenges by using QGIS for visualization and spatial processing, along with Python and Telegram API for automated real-time alert generation. This improves accuracy, efficiency, and usability compared to existing systems [2], [7], [20].

III. RESEARCH METHODOLOGY

The proposed Geo-Fence Alert System follows a structured approach for monitoring the movement of a tracked entity and generating alerts based on its spatial position. Initially, location data in the form of latitude and longitude coordinates is obtained from a GPS source and used as input for the system [18].

The collected data is integrated into the QGIS environment, where geospatial processing and visualization are performed. Geo-fence boundaries are created using polygon layers and categorized into different zones such as Safe Zone, Warning Zone, and Danger Zone. These zones represent different levels of risk and are used to monitor the movement of the tracked entity effectively [2], [13], [15].

In the next stage, spatial analysis is carried out using techniques such as the point-in-polygon method. This technique determines whether the tracked point lies inside or outside a specific geofence zone. Based on this analysis, the system identifies the status of the entity in real time and updates it continuously as the position changes [5], [8].

The processed spatial data is then handled by a Python-based processing module, which evaluates the zone conditions and applies logical rules for decision-making. The system continuously checks for any boundary crossing or zone transition to ensure timely detection of critical situations [7], [19]. Finally, when a change in zone is detected, alert messages are generated and sent to users through the Telegram Bot API. This ensures instant communication and allows users to take immediate action. The overall methodology provides a cost-effective, scalable, and efficient solution for real-time geospatial monitoring and alert generation [20].

Fig.2 Represents the methodology of proposed project

IV. SYSTEM ARCHITECTURE AND DESIGN

The architecture of the proposed Geo-Fence Alert System is designed as a modular framework that enables efficient data processing, spatial analysis, and real-time alert generation. The system consists of multiple interconnected modules, each responsible for a specific function in the overall workflow.

The process begins with the data acquisition module, which collects location coordinates in the form of latitude and longitude from a GPS source. This data serves as the primary input for the system and is continuously updated to track the movement of the entity [18].

The collected data is then passed to a Python-based processing module, which performs initial computations and prepares the data for spatial analysis. This module also manages communication with external services such as the Telegram API for alert generation [7], [19].

Next, the processed data is integrated into the geospatial analysis module implemented using QGIS. In this module, geo-fence boundaries are created using polygon layers and categorized into Safe, Warning, and Danger zones. QGIS provides powerful tools for visualization and spatial operations, enabling efficient monitoring of the tracked entity [2], [13], [15].

The system then applies a zone detection mechanism using spatial techniques such as the point-in-polygon method. This determines the exact position of the tracked entity relative to the defined zones and classifies its status accordingly [5], [8].

Finally, the alert generation module evaluates the zone status and triggers notifications whenever a boundary crossing occurs. Alerts are sent to users in real time using the Telegram Bot API, ensuring immediate communication and response [20].

Overall, the system architecture integrates geospatial analysis, data processing, and communication modules to provide a scalable, cost-effective, and efficient solution for real-time geo-fence monitoring.

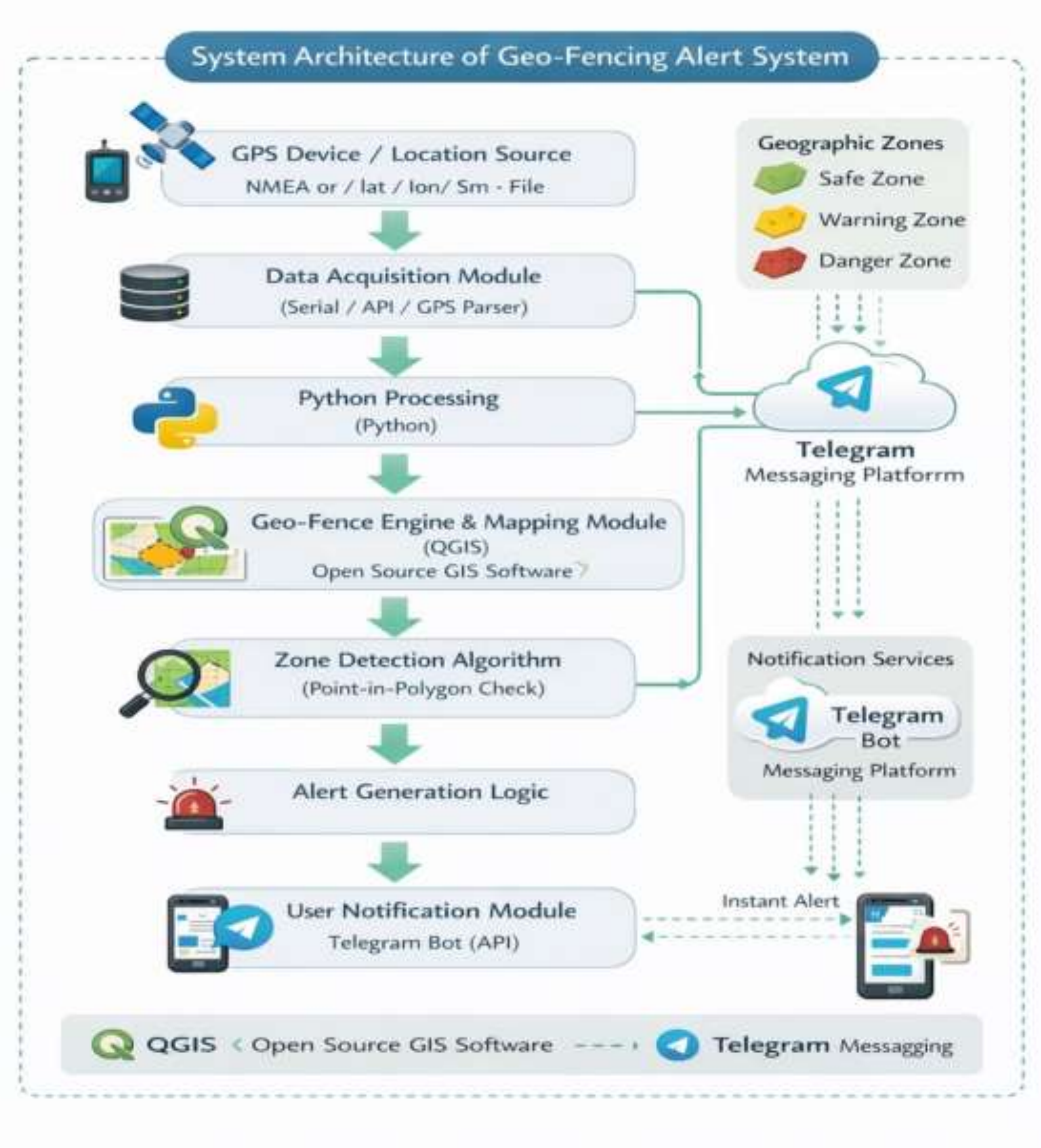

Fig.3 System architecture of the proposed project

VI. WORKING PRINCIPLE

The proposed Geo-Fence Alert System operates based on real-time location tracking and spatial analysis. Initially, the system receives the location coordinates of the tracked entity in the form of latitude and longitude from a GPS source. This data is continuously updated and used as the primary input for monitoring movement [18].

The received location data is then integrated into the QGIS environment, where predefined geo-fence zones such as Safe Zone, Warning Zone, and Danger Zone are created using polygon layers. The system performs spatial analysis to determine the position of the tracked point relative to these zones. This is achieved using techniques such as point-in-polygon analysis, which helps in identifying the exact zone in which the entity is located [5], [8].

Based on this analysis, the system classifies the current status of the tracked entity. If the point lies within the safe zone, the system maintains normal status. When the point moves into the warning zone, a warning condition is detected, and if it crosses the boundary into the danger zone, a critical alert condition is triggered. This process is carried out continuously to ensure accurate real-time monitoring [6].

Once a zone transition is detected, the processed data is passed to the Python module, where alert conditions are evaluated. The system then generates alert messages and sends them to users through the Telegram Bot API, ensuring instant communication and timely response [7], [20].

Thus, the system provides an automated and efficient mechanism for monitoring, analyzing, and alerting users about the movement of tracked entities, making it suitable for wildlife monitoring and asset tracking applications.

VI. TECHNIQUES USED

The proposed Geo-Fence Alert System is developed using a combination of geospatial technologies, programming frameworks, and communication tools to achieve accurate real-time monitoring and alert generation. The system integrates Geographic Information Systems (GIS), spatial analysis techniques, and software-based automation to provide an efficient and scalable solution for tracking and monitoring applications.

One of the primary techniques used in this system is Geographic Information System (GIS) analysis through QGIS. QGIS is an open-source platform that enables the creation and management of geospatial data using layers. In this system, polygon layers are used to define geo-fence boundaries such as Safe Zone, Warning Zone, and Danger Zone. QGIS provides powerful visualization and spatial processing capabilities, allowing users to clearly understand the movement of tracked entities within different regions. This enhances decision-making and improves monitoring efficiency [2], [13], [15].

Another important technique employed is spatial analysis, specifically the point-in-polygon method. This method is used to determine whether a given point lies inside or outside a defined polygon. By applying this technique, the system can accurately classify the position of the tracked entity into different zones. This classification plays a crucial role in detecting boundary crossings and identifying risk levels associated with the movement of the entity. Spatial analysis ensures precision and reliability in zone detection, which is essential for real-time monitoring systems [5], [8].

Python programming is used as the core processing and automation tool in the system. Python scripts are integrated with QGIS to process location data, evaluate spatial conditions, and implement logical decision-making. The automation provided by Python enables continuous monitoring without manual intervention, improving system efficiency and reducing response time. It also facilitates integration with external APIs for communication and alert generation [7], [19].

For real-time communication, the system uses the Telegram Bot API, which allows automatic transmission of alert messages to users. Whenever the tracked entity crosses a geo-fence boundary or enters a warning or danger zone, the system instantly sends notifications through Telegram. This ensures timely communication and allows users to take immediate action. The use of messaging APIs enhances the responsiveness and usability of the system in practical applications [20].

Additionally, Global Positioning System (GPS) technology is used to provide accurate location coordinates of the tracked entity. These coordinates serve as the primary input for the system and are continuously updated to reflect real-time movement. The integration of GPS with GIS and software-based processing enables effective tracking and monitoring without the need for complex hardware infrastructure. However, the accuracy of the system depends on the quality of GPS signals, which may vary under different environmental conditions [18].

Overall, the combination of GIS-based spatial analysis, Python automation, GPS tracking, and real-time communication through Telegram API results in a cost-effective, scalable, and efficient system. These techniques collectively enhance the system's ability to perform accurate monitoring, instant alert generation, and detailed movement analysis for wildlife and asset tracking applications.

V. RESULTS AND DISCUSSION

The proposed Geo-Fence Alert System was successfully implemented using QGIS, Python, and Telegram Bot API. The system was tested for key functionalities such as geo-fence zone detection, real-time alert generation, and movement analysis. The results demonstrate that the system accurately monitors the position of the tracked entity and generates alerts based on predefined geo-fence boundaries. The integration of GIS-based spatial analysis and automated communication ensures reliable and efficient performance in real-time monitoring applications [2], [7], [20].

5.1 Geo-fence Zone Detection

The geo-fence zones were successfully created in QGIS using polygon layers representing Safe, Warning, and Boundary regions. The tracked point is displayed on the map, allowing clear visualization of its position relative to these zones. In the observed case, the point is located outside the boundary, indicating a critical alert condition. This confirms that the system can accurately detect the position of the entity and identify boundary violations in real time using spatial analysis techniques [5], [8].

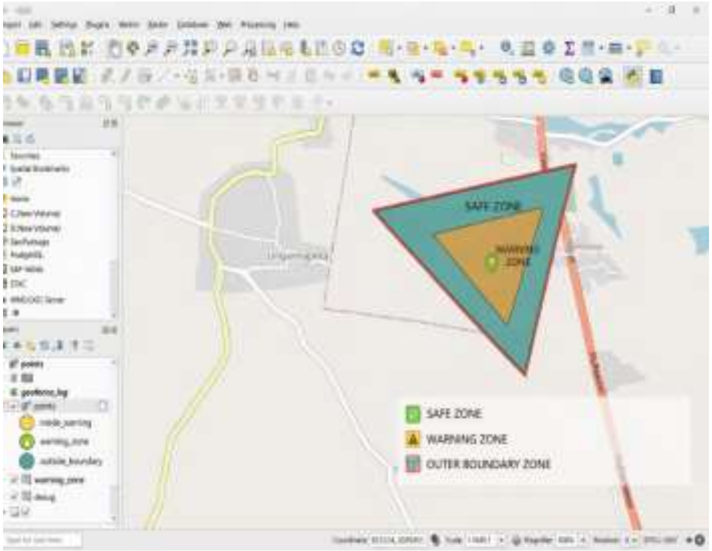

Fig. 4 Geo-Fence zone detection in QGIS showing the tracked point position

5.2. Real-time Alert Notifications

The system generates real-time alert notifications using the Telegram Bot API. Based on the zone classification, messages such as “SAFE,” “WARNING,” and “ALERT” are sent to users instantly. This demonstrates the system’s ability to provide immediate updates whenever a boundary crossing occurs, improving responsiveness and enabling quick decision-making [7], [20].

Fig.5 Real-time alert notifications generated using Telegram Bot API

5.3. ZONE ANALYSIS

The bar graph represents the number of occurrences in each zone category, including Safe, Warning, and Danger zones. This analysis helps in understanding how frequently the tracked entity moves between zones. The results indicate that the system effectively records and classifies events, providing useful insights into movement patterns and risk levels.

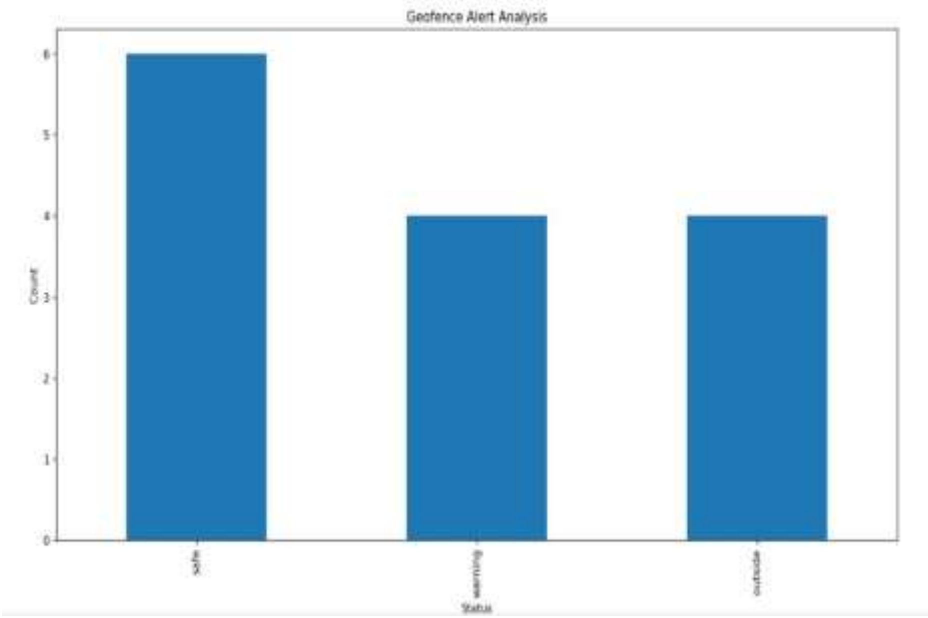

Fig.6 Analysis of geo-fence alerts based on zone classification

5.4. Events Over Time

The line graph illustrates the variation of geo-fence events over time. It shows how the tracked entity transitions between zones at different time intervals. This helps in identifying patterns of movement and detecting periods of high activity or frequent boundary crossings.

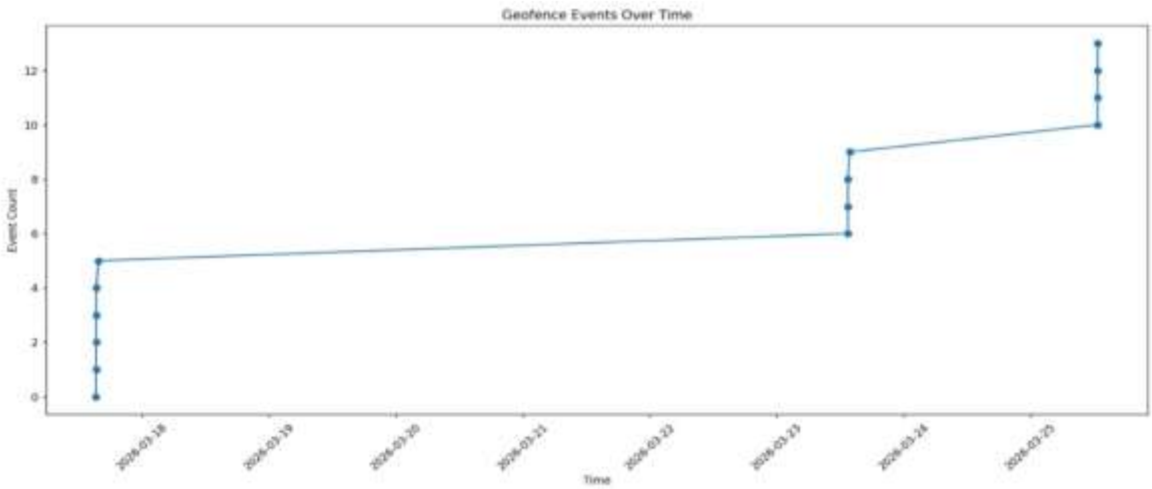

Fig.7 Geo-fence events variation over time

5.5. Zone Distribution

The pie chart shows the percentage distribution of time spent by the tracked entity in different zones. It provides a clear overview of how the entity interacts with various regions. This visualization helps in evaluating overall system behavior and identifying dominant zones.

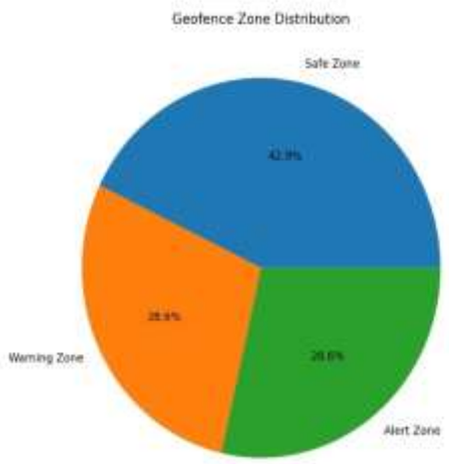

Fig.8 Percentage distribution of zone classifications

5.6. Distance Analysis

The distance analysis graph shows the variation in distance travelled by the tracked entity over time. It helps in understanding movement behavior and identifying sudden changes in displacement, which may indicate abnormal activity or boundary crossings.

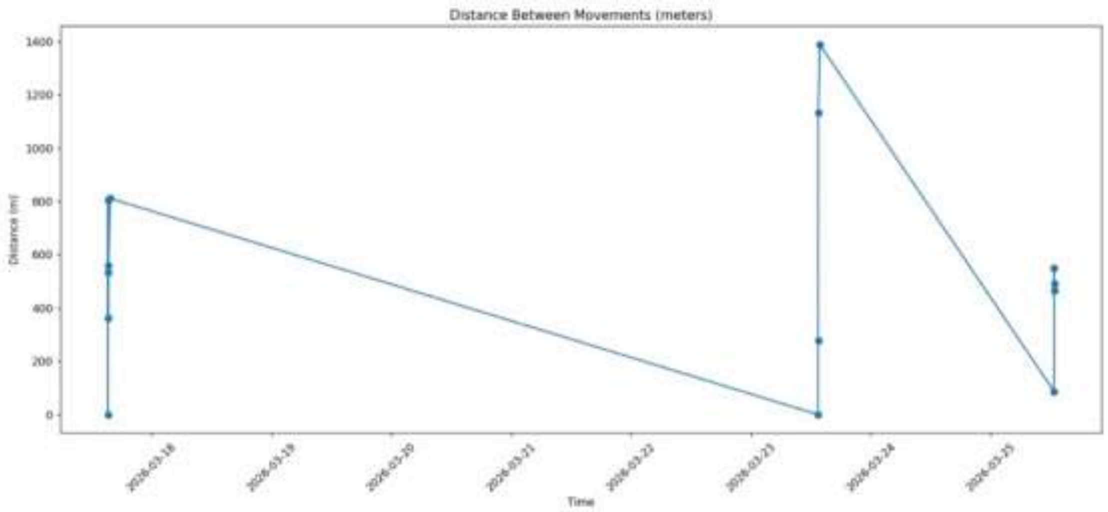

Fig.9 Distance variation of the tracked entity

5.7. Speed Analysis

The speed analysis graph represents the variation in speed of the tracked entity over time. Sudden increases in speed may indicate unusual movement, while consistent speed represents normal behavior. This helps in detecting potential risks and abnormal activity.

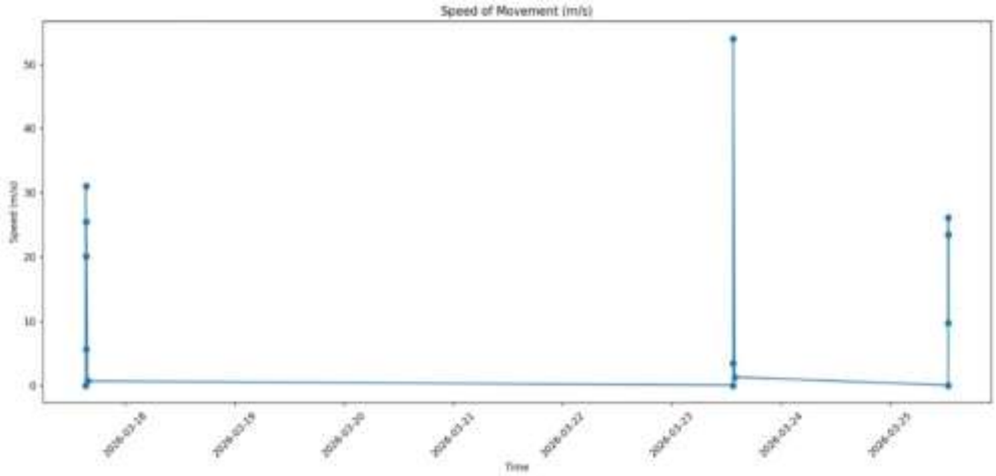

Fig.10 Speed analysis of the tracked entity

5.8. Attribute Table

The attribute table displays recorded data including timestamp, coordinates, and zone classification. This confirms that the system maintains detailed logs of movement, which can be used for further analysis and validation of results.

geofence_log — Features Total: 14, Filtered: 14, Selected: 0

	date&time	status	x	y	warning_count	outside_count	safe_count
1	17-03-2026 15:...	warning	83.3996156700...	18.0194634894...	1	0	0
2	17-03-2026 15:...	outside	83.3944213749...	18.0185407193...	0	1	0
3	17-03-2026 15:...	safe	83.4017276362...	18.0206033798...	0	0	1
4	17-03-2026 15:...	warning	83.3984169866...	18.0214175796...	1	0	0
5	17-03-2026 15:...	outside	83.3942501344...	18.0187035623...	0	1	0
6	17-03-2026 15:...	safe	83.4016710804...	18.0205621831...	0	0	1
7	23-03-2026 13:...	safe	83.4016710804...	18.0205621831...	0	0	1
8	23-03-2026 13:...	warning	83.3990484673...	18.0204929067...	1	0	0
9	23-03-2026 13:...	outside	83.3883394639...	18.0209778439...	0	1	0
10	23-03-2026 13:...	safe	83.4013796790...	18.0195230305...	0	0	1
11	25-03-2026 12:...	safe	83.4016728959...	18.0187937683...	0	0	1
12	25-03-2026 12:...	outside	83.3967461131...	18.0172320441...	0	1	0
13	25-03-2026 12:...	warning	83.3974440740...	18.0213705829...	1	0	0
14	25-03-2026 12:...	safe	83.4017960654...	18.0198088829...	0	0	1

Fig.11 Attribute table showing geo-fence log data

Overall, the results confirm that the proposed Geo-Fence Alert System provides accurate real-time monitoring, effective zone detection, and reliable alert generation. The integration of QGIS for spatial analysis and Python for automation ensures efficient system performance. The use of Telegram API enables instant communication, making the system suitable for practical applications such as wildlife monitoring and asset tracking [2], [7], [20].

VI. CONCLUSION

The proposed Geo-Fence Alert System for wildlife and asset monitoring has been successfully developed and implemented using QGIS and Python. The system provides accurate real-time tracking of the monitored entity and effectively classifies its position into Safe, Warning, and Danger zones using spatial analysis techniques such as point-in-polygon [2], [5]. This enables precise detection of boundary crossings and improves situational awareness in monitoring applications.

The integration of Geographic Information Systems with Python-based automation enhances the system's capability to perform continuous monitoring and real-time analysis. Additionally, the use of the Telegram Bot API ensures instant communication by sending alerts to users whenever a zone transition occurs. This significantly improves response time and reduces the risk associated with delayed notifications [7], [20].

Compared to traditional GPS-based tracking systems, the proposed solution offers better visualization, improved accuracy, and intelligent decision-making through zone-based classification. The system is fully software-based, which eliminates the need for complex hardware components, making it cost-effective and easy to deploy in various real-world scenarios such as wildlife conservation and asset tracking [18].

Overall, the system demonstrates high efficiency, reliability, and scalability in real-time geo-fence monitoring. The combination of GIS tools, spatial analysis, and automated communication makes it a practical and effective solution for modern tracking and monitoring applications.

VII. FUTURE SCOPE

Although the proposed Geo-Fence Alert System performs efficiently in real-time monitoring and alert generation, there is scope for further enhancement. One of the major improvements can be the integration of real-time GPS devices and IoT-based sensors to enable automatic data collection instead of manual input. This would make the system more robust and suitable for large-scale deployment in real-world environments [18].

Future work can also focus on improving location accuracy, especially in areas with weak GPS signals such as dense forests or urban environments. Advanced filtering techniques and positioning algorithms can be incorporated to enhance precision and reliability of tracking data [1]..

In addition, the system can be extended by developing a mobile application or web-based dashboard for better visualization and user interaction. This would allow users to monitor movement, view alerts, and analyse data more conveniently in real time.

Furthermore, advanced features such as machine learning and predictive analytics can be integrated to analyse movement patterns and predict future behaviour of tracked entities. This would help in early detection of potential risks and improve decision-making in applications such as wildlife conservation and asset security [3], [4].

The system can also be expanded to support offline alert mechanisms and alternative communication methods to reduce dependency on internet connectivity. Overall, these enhancements will improve scalability, performance, and practical usability of the system for various monitoring applications [7], [20].

REFERENCES

[1] Z. Özdemir and B. Tuğrul, "Geofencing on the real-time GPS tracking system and improving GPS accuracy with moving average, Kalman filter and logistic regression analysis," in *Proc. IEEE Int. Symp. Measurement and Instrumentation (ISMSIT)*, 2019.

[2] N. E. D. Mohamed, M. A. Shafea, and M. M. Abdelhakam, "Anomaly-resilient geofencing and predictive navigation in IoT environments," *Scientific Reports*, vol. 16, 2026.

[3] I. Sasaki *et al*., "Data-driven geofencing design for point-of-interest notifiers utilizing genetic algorithm," *ISPRS Int. J. Geo-Information*, vol. 13, no. 6, 2024.

[4] Y. Shevchenko and U.-D. Reips, "Geofencing in location-based behavioral research: Methodology, challenges, and implementation," *Behavior Research Methods*, 2024.

[5] N. Prasad and S. Swetha, "Geofencing: Location-based services," *Int. J. Adv. Res. Ideas Innov. Technol.*, vol. 5, no. 2, 2019.

[6] A. Nait-Sidi-Moh *et al*., "On the use of location-based services and geofencing concepts for safety and road transport efficiency," *Communications in Computer and Information Science*, 2013.

[7] "Fleet tracking and geofencing using Internet of Things (IoT)," in *Proc. IEEE Conf.*, 2023.

[8] "Application of geofencing technology for spatial analysis in mobile navigation," in *Proc. IEEE Conf.*, 2016.

[9] "Attendance management system using geofencing technology," in *Proc. IEEE Conf.*, 2024.

[10] V. Hodge *et al*., "Analysing ultra-wide band positioning for geofencing in safety assurance systems," 2022.

[11] V. Vagal *et al*., "A new approach to complex dynamic geofencing for unmanned aerial vehicles," 2021.

[12] W. Zhuo *et al*., "Semi-supervised learning for geofencing services using RF signals," 2022.

[13] M. Goodchild, *Geographic Information Systems and Science*, 3rd ed. Wiley, 2018.

[14] P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind, *Geographic Information Systems and Science*. Wiley, 2015.

[15] QGIS Development Team, "QGIS Geographic Information System," Open Source Geospatial Foundation, 2024. [Online]. Available: https://qgis.org

[16] J. Schiller and A. Voisard, *Location-Based Services*. Morgan Kaufmann, 2004.

[17] S. Shekhar and S. Chawla, *Spatial Databases: A Tour*. Prentice Hall, 2003.

[18] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *GPS: Theory and Practice*. Springer, 2001.
[19] Python Software Foundation, "Python Language Reference," Version 3.12, 2024. [Online]. Available: https://www.python.org
[20] Telegram, "Telegram Bot API," 2024. [Online]. Available: https://core.telegram.org/bots/api