A Self-Contained Spatio-Temporal Anomaly Detection Application for Travel Safety

Kwasi Edwards
Department of Computing and Information Technology
The University of the West Indies
St. Augustine, Trinidad and Tobago
kwasiedwards@gmail.com

Patrick Hosein
Department of Electrical and Computer Engineering
The University of the West Indies
St. Augustine, Trinidad and Tobago
patrick.hosein@uwi.edu

***Abstract*—We describe a mobile application designed to enhance safety when using public transportation. During the training phase the application learns typical travel patterns and times by analyzing historical route data stored locally on the user's device. During operation, it identifies spatio-temporal anomalies in real-time by comparing a user's current location against their historical profile. If a significant deviation is detected, an alert is sent via SMS to the user's emergency contacts. All data and processing occurs strictly on the device itself and nothing is shared with the Cloud. It does not require an Internet connection (except for when it has to be installed in which case public WiFi can be used) making it useful for those who cannot afford cellular data plans. This paper provides a Proof of Concept of this application and includes typical use cases to illustrate efficacy.**

***Index Terms*—location based services, privacy in GPS, travel safety, mobile safety**

I. Introduction

Individuals face increasing risks associated with unforeseen circumstances and potentially dangerous situations. Several applications have safety features, such as location sharing and emergency assistance, but many of them rely on consistent internet connectivity to enable these features. This reliance on a persistent internet connection creates vulnerabilities for travelers, especially those in areas with limited or no internet access, those who live in rural areas, or during emergencies. This vulnerability is also intensified for citizens in developing countries, where access to affordable mobile data is limited.

While existing location-based safety applications offer emergency alerts and location sharing capabilities for travelers, these features are dependent on a stable internet connection. This dependency renders these solutions ineffective in situations when network access is unavailable, unreliable, or financially inaccessible, leaving individuals vulnerable when communication is most needed.

To address these limitations, we introduce SayfTrip, an Android application designed to provide proactive traveller safety through a fully self-contained, offline solution. SayfTrip learns individual travel patterns by recording and analyzing historical route data, enabling it to detect deviations from expected routes in real-time without requiring internet connectivity. This approach offers a significant advantage in situations where network access is unavailable.

The key contributions of this work are an innovative offline anomaly detection system for traveler safety and a practical implementation of SayfTrip for Android devices. This aims to offer a reliable and accessible safety solution for individuals in areas with limited or unreliable network connectivity. The remainder of this paper is organized as follows: Section 2 reviews related work on anomaly detection and location-based safety. Section 3 details the extent of the problem in Small Island Developing States (SIDS). Section 4 outlines our methodology and approach to solving the problem. Section 5 details the design and implementation of SayfTrip, including the algorithm and system architecture. Section 6 discusses the implications of this research, and Section 7 highlights the future directions of this research.

II. Literature Survey

Reliable location tracking independent of network connectivity is critical for ensuring the safety and well-being of children, individuals with medical conditions, and other vulnerable populations. This literature review explores the current state of research in offline location tracking and anomaly detection, focusing on techniques that address the challenges of privacy concerns, limited connectivity, and resource constraints.

Traditional location-based services (LBS) rely primarily on cloud infrastructure for data processing and analysis. While effective in many scenarios, these systems suffer from several security and privacy concerns. Firstly, they are vulnerable to network outages and disruptions, rendering them unreliable in emergencies or remote locations. Secondly, continuous data transmission to centralized cloud servers raises significant privacy concerns, as sensitive location data is susceptible to breaches and unauthorized access. This vulnerability stems from the need to store and process data on potentially untrusted third-party servers. Researchers have addressed these concerns with various privacy-preserving techniques. For instance, [1] proposes an efficient scheme using homomorphic encryption to enable secure LBS queries without revealing user location to the cloud server. Similarly, [2] demonstrates the importance of forward security and content privacy in spatiotemporal data by developing a scheme for secure and efficient search

978-1-6654-7808-3/25/$31.00 ©2025 IEEE

over encrypted data, addressing concerns about data breaches and past exposure. Similarly, [3] highlights the challenges of protecting access patterns and supporting high-dimensional queries in spatial crowdsourcing, offering solutions to prevent information leakage even with complex data filtering. These advancements demonstrate a growing interest in mitigating the inherent privacy risks associated with centralized cloud-based LBS architectures by shifting towards secure computation and encryption-based approaches.

However, the efficacy of these sophisticated privacy-enhancing technologies is often contingent upon reliable network connectivity, as such techniques are computationally expensive and are not feasible to run on resource-constrained devices such as mobile phones. This prerequisite is not universally available, particularly within underserved communities, rural areas, or developing nations. To address this, research has begun to explore on-device, offline methods for spatial anomaly detection. Approaches leveraging machine learning offer a promising path due to their ability to identify anomalous patterns without relying on external data sources. For example, [4] presents a hybrid machine learning system that uses clustering (HDBSCAN) and supervised classification to identify and exclude anomalous GNSS observations performed locally. Similarly, [5] developed GPSvas, a visual analytics system that utilizes conditional random fields for anomaly detection in streaming GPS data, enabling human experts to refine the model and improve accuracy through interactive visualization. Although GPSvas requires adaptation for fully offline operation, its visual analytics approach offers potential for on-device refinement of anomaly detection models.

However, a significant gap in the existing literature is the lack of focus on fully offline, self-contained anomaly detection systems specifically designed for mobile devices. While several studies have explored personal safety using machine learning, the computational demands of complex machine learning models often make their deployment on resource-constrained mobile platforms infeasible unless performance compromises are made. Existing approaches often rely on pre-processed data or cloud connectivity for anomaly scoring. For instance, [6] presents a grid-based Local Outlier Factor (LOF) algorithm for detecting anomalies in spatio-temporal traffic data. While effective in identifying unusual patterns, this method, like many others in the field, is designed for analysis on larger datasets and does not address the challenges of running complex computations directly on a mobile device with limited resources. This highlights the need for anomaly detection algorithms specifically designed for the constraints of offline mobile operation.

The proposed research addresses this gap by developing an innovative anomaly detection algorithm specifically designed for resource-constrained mobile devices. This system aims to provide real-time safety monitoring independent of network connectivity with minimal compute resource requirements in an effort to enhance personal security in challenging environments. This research offers a unique solution tailored for fully offline operation.

III. Problem Description

Small Island Developing States (SIDS) face significant challenges in achieving reliable digital connectivity. Mobile phone ownership in SIDS reached 74% in 2023 [7]. This trend is close to the global average of 78%. However, mobile-broadband penetration rates lag considerably, averaging 63 subscriptions per 100 people in SIDS compared to 87 globally. This disparity is particularly pronounced in regions like the Pacific SIDS, with a subscription rate of only 28.4, compared to 81.9 in African SIDS [7]. These connectivity gaps are magnified by high operational costs for telecommunication providers as they face challenges stemming from high import expenses, environmental vulnerabilities, and inadequate infrastructure. As a result, only 18 of 50 SIDS meet the affordability criteria for data-only mobile broadband.

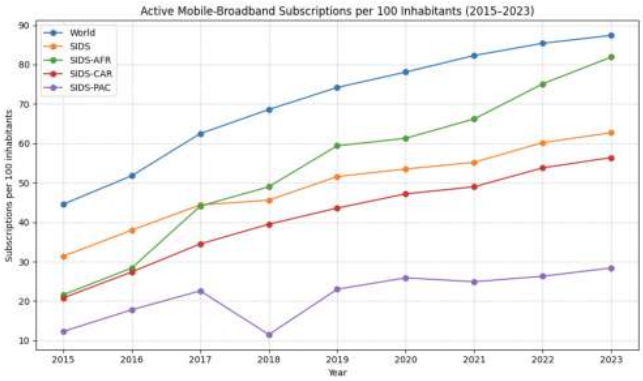

Fig. 1. Active mobile-broadband subscriptions per 100 inhabitants in SIDS

An example of a SIDS in the Caribbean, Trinidad and Tobago, tells a similar story. While the number of cellular mobile connections compared to the population is oversubscribed at 132 percent [8], the mobile data affordability index is low at 44.2 [9]. This evidence reinforces that SIDS face unique challenges in accessing reliable data connectivity.

This paper addresses a critical need for reliable traveler safety in the absence of ubiquitous network connectivity. Existing location-based safety solutions rely heavily on cloud-based architectures, necessitating constant data transmission and rendering them ineffective during outages or in areas with poor coverage. This limitation poses substantial risks for vulnerable individuals, especially given the connectivity and affordability challenges prevalent in SIDS.

To mitigate this risk, we introduce SayfTrip, an innovative approach to spatiotemporal anomaly detection. Unlike traditional systems that rely on simple location-based alerts, SayfTrip learns and adapts to individual travel patterns. The system continuously captures a user's location as a background process, establishing a profile that encompasses spatial and temporal aspects of their movements.

The core challenge lies in differentiating between legitimate deviations, such as detours due to traffic, and anomalous behavior indicative of a safety risk. SayfTrip employs a real-time detection algorithm that compares a user's current

location against their established profile. Significant deviations exceeding predefined thresholds trigger alerts and initiate pre-configured safety protocols, such as notifying emergency contacts via SMS. The system's self-learning capability refines its understanding of typical behavior, improving accuracy and minimizing false alarms.

IV. Methodology

This methodology details the system architecture, implementation techniques, and the approach to anomaly detection employed within the application.

A. System Architecture and Implementation

SayfTrip is developed as a native Android application that utilizes the Flutter framework. This was chosen for its cross-platform compatibility and rapid development capabilities. The development environment consists of IntelliJ IDEA as the Integrated Development Environment (IDE). The application architecture centers around capturing and analyzing location data in the absence of network connectivity, relying entirely on onboard processing and storage.

The application allows users to define and save named routes, representing frequently traveled paths. Once a route is defined, users can initiate a "run" along that route. During a run, the application continuously captures the device's location using the Android operating system's location services. Location updates are acquired with the bestForNavigation setting, balancing accuracy and battery consumption. While higher accuracy levels are available that can poll the sensors as frequently as 1Hz, those settings increase the demand on the battery, leading to a shorter device runtime.

The captured location data consists of latitude, longitude, and timestamp, and is persistently stored locally on the device using a SQLite database. The SQLite database was chosen for its lightweight footprint, embedded nature, and efficient data management capabilities.

The application's database contains four primary tables that are designed to manage application state and historical travel data. The Settings tables persist application state, such as the status of an active route. This facilitates communication between the background service and the foreground application. The Route table stores user-defined routes. Instances of route traversals are recorded in the RouteRun table and finally the RouteRunPoint stores granular location data that's associated with a RouteRun. This aims to provide a time-series record of a user's journey along that route. The schema for the tables is detailed in Figure 2

B. Anomaly detection in well-known routes

The core of SayfTrip lies in its spatial anomaly detection algorithm. Upon receiving a new location update, the algorithm calculates the location's path deviation, which is the minimum distance between the current GPS coordinate and all points from all previously recorded trips on the route that's being traveled on. The calculated deviation is then ranked into one of four categories.

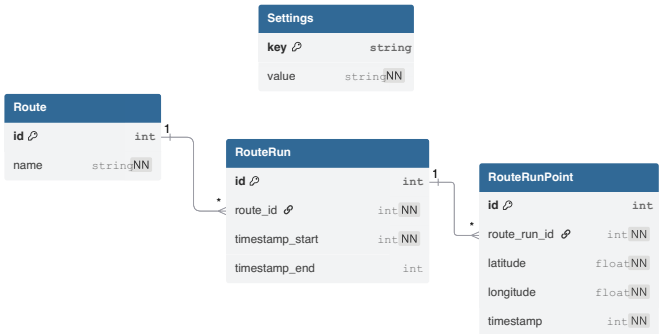

Fig. 2. Application SQLite database schema

- **Normal deviation**: This is when the path deviation does not exceed 150 meters
- **Minor deviation**: This is when the deviation exceeds 150 meters but does not exceed 500 meters.
- **Major deviation**: This is when the deviation exceeds 500 meters but not 1000 meters.
- **Critical deviation**: This is when the deviation exceeds 1000 meters.

The detailed implementation is outlined in Algorithm 1.

Algorithm 1 Anomaly Detection Algorithm

Require: *Input*: New location $(lat, lng, time, routeRunId)$
Ensure: *Output*: Deviation (Normal, Minor, Major, Critical)

Store $(lat, lng, time, routeRunId)$ in the database
Retrieve all historical *RouteRun* for identified by *routeRunId*
$minDistance \leftarrow \infty$
for each RouteRun R **do**
  for each TripPoint $p \in R$ **do**
    $d \leftarrow$ HaversineDistance$((lat, lng), (p.lat, p.lng))$
    if $d < minDistance$ **then**
      $minDistance \leftarrow d$
    end if
  end for
end for
if $minDistance \leq 150$ **then**
  Normal deviation
else if $minDistance \leq 500$ **then**
  Minor deviation
else if $minDistance \leq 1000$ **then**
  Major deviation
else
  Critical deviation
end if

C. Alerting Mechanism and Safety Protocol

As described previously, the anomaly detection algorithm continuously analyzes GPS data, identifying deviations from the user's expected route. A persistent notification is shown to the user while the background service is capturing and processing location data, giving them the ability to see the

status of their deviation from the path. This notification gets updated each time the deviation status changes. If the deviation status reaches the critical state, an alerting sequence is initiated. Details on the alerting sequence are as follows:

1) A device notification is displayed to the user, providing information about the severity of the deviation. This notification asks users to confirm if they are safe. The notification offsers the user two options so that they can indicate whether they're safe or not.
2) If the user indicates that they are not safe, or 30 seconds has elapsed since the user was notified via the in-app or device notification, the application automatically sends a pre-defined SMS message to the user's designated emergency contact. The message includes the user's current location (latitude, longitude), a link to a map displaying their location, the route that they were travelling on and a brief description of the detected anomaly.

V. Proof of concept: Design and Implementation

In the initial Proof of Concept (PoC) app, the anomaly detection algorithm is restricted to spatial only, instead of spatiotemporal, as, for many purposes, this is sufficient in scenarios such as kidnapping detection. Additionally, in this environment, we're focusing on users who take public transportation, which usually has a fixed route with minimal deviations. Additionally, for a truly offline experience, the map of Trinidad and Tobago comes pre-loaded into the app to minimize the post-install configuration. The features of the PoC app include:

1) Creation of routes
2) Initiation of Route Trips
3) Reviewing past trips
4) Saving emergency contact details

To evaluate the functionality of SayfTrip's anomaly detection algorithm, a series of runs was conducted by installing the application on a physical device (Google Pixel 9 Pro, API 36), and driving a route. While each route run may have had minor deviations in the data points collected, this would not affect the algorithm's performance as the temporal aspect of the data collected was not evaluated in this algorithm. The simulated route mirrored a typical traveller's route between Gasparillo and St. Augustine, Trinidad.

A. Experiment details

Two experiments were conducted to evaluate the functionality and performance of the SayfTrip application. In both experiments, a predefined route was implemented within the application and location data, representing a single route run, was loaded. Experiment 1 focused on verifying the accuracy of the anomaly detection algorithm, while Experiment 2 assessed the application's resource utilization. All experiments were performed on a Pixel 9 Pro device running Android SDK Level 36.

1) Experiment 1: Anomaly Detection Algorithm Verification: This experiment evaluated the algorithm's ability to correctly classify route deviations based on severity. Four distinct scenarios were designed to simulate varying degrees of deviation from the intended route:

- **Nominal Route**: The simulated user adhered strictly to the predefined route.
- **Minor Route Deviation**: The simulated user deviated from the route multiple times, with each deviation not exceeding 300 meters away from the route. This scenario represents minor navigational adjustments, such as choosing an alternate road to circumvent minor traffic congestion.
- **Major Route Deviation**: The simulated user deviated from the route multiple times, with each deviation not exceeding 750 meters away from the route. This simulates more substantial detours, potentially caused by significant traffic events or road closures.
- **Complete Deviation**: The simulated user deviated from the route by more than 1000 meters, representing a scenario potentially indicative of a hazardous situation.

To ensure controlled experimentation and mitigate the influence of external variables, the primary route and deviation scenarios were created using Geographic Information System (GIS) software. This approach allowed for precise control over route geometry and simulated GPS data. The generated routes were exported as GPX files and streamed to the SayfTrip application using a mock-location application to simulate real-time GPS updates. The accuracy of the anomaly classification was then evaluated. The results of this experiment are tabulated in I below.

TABLE I
DEVIATION CLASSIFICATION EVALUATION

Scenario	Expected Severity	Detected Severity
Nominal Route	Normal Deviation	Normal Deviation
Minor Route Deviation	Minor deviation	Minor deviation
Major Route Deviation	Major deviation	Major deviation
Complete Deviation	Critical deviation	Critical deviation

2) Experiment 2: Application Resource Intensity: This experiment aimed to quantify the computational resources consumed by the SayfTrip application during typical operation. Understanding the application's resource demands is crucial for optimizing performance and ensuring a positive user experience, particularly on devices with limited computational resources or battery capacity. The primary metrics of interest were CPU utilization, memory consumption, and battery drain.

Data collection was performed during a 67-minute field trial involving physical traversal of a predefined route. Application performance was profiled using Perfetto, a system-wide tracing tool, to capture CPU usage and memory allocation patterns. Battery statistics were obtained using the 'adb shell dumpsys batterystats' command, providing detailed information on power consumption during the trial.

TABLE II
RESOURCE CONSUMPTION METRICS

Metric	Minimum	Maximum	Average
CPU Usage	0%	29.6%	4.65%
Memory Consumption	340.4 MB	439.4 MB	353.7 MB

TABLE III
BATTERY USAGE BREAKDOWN

Component / State	Energy (mAh)	Duration / Notes
Total app usage	4.68	–
Foreground (fg)	0.545	7m 52s
Background (bg)	0.000241	–
Foreground service (fgs)	2.15	1h 7m 54s
CPU	1.61	–
CPU foreground	0.200	–
CPU background	0.000241	–
CPU foreground service	1.41	–
Screen	1.98	screen on
Wakelock	1.04	total (includes fg/fgs)
GPU	0.0529	total

VI. DISCUSSION OF RESULTS

The goal of the simulations was to evaluate the system's ability to accurately identify and classify anomalous events within the defined operational parameters.

A. Anomaly Detection Algorithm Verification

A high level of precision is critical in this application as false alarms could lead to user desensitization and a decreased willingness to respond to threats. From the results shown in Table I, the system was able to accurately classify all tested deviation scenarios, demonstrating a strong correlation between the expected and detected anomaly severity. This suggests the algorithm is functioning as designed and effectively distinguishes between expected, minor navigational adjustments, significant detours, and critical deviations indicative of potential hazards. A significant limitation of simulated routes is that it lacks the spontaniety of real world travel, and it ignores events such as drift due to atmospheric conditions and changes in routes due to unforseen circumstances. However, it was sufficient in testing that the classification algorithm is functional. The consistent and accurate classification observed across all scenarios indicates a robust anomaly detection capability. However, future work should explore the algorithm's performance under more complex conditions, such as varying speeds, GPS signal degradation, and the presence of multiple simultaneous deviations.

B. Application Resource Intensity

Additionally, a minimal utilization of resources such as CPU, memory and battery is essential to ensure that the operating system does not terminate the application due to resource over-utilization. This is possible in a number of cases, namely

1) When a device's memory is low and the OS terminates background processes to free up RAM.
2) Due to battery optimization: The Android OS implements aggressive battery saving mechanisms. When an app is running in the background or when the battery percentage drops below a certain threshold, services and apps may be terminated to conserve power. Additionally, some device manufacturers have even stricter battery management policies that can be more aggressive in terminating apps

The results from Experiment 2 shows that the application demonstrates a relatively low resource footprint during typical operation. The average CPU utilization of 4.65% suggests that the application does not significantly burden the device's processing capabilities, allowing for other applications to run concurrently without performance degradation. Similarly, the average memory consumption of 353.7 MB is well within reasonable limits for modern mobile devices. The battery usage breakdown in Table III indicates that foreground service operations, likely related to location tracking and anomaly detection, contribute the most to power consumption. However, the overall energy usage of 4.68 mAh during the 67-minute trial is comparatively low, suggesting that the application is reasonably efficient in its power management. Optimizations targeting the foreground service processes could further reduce battery drain.

VII. CONCLUSION / FUTURE WORK

This research has demonstrated the feasibility of utilizing spatial anomaly detection for enhanced personal safety. However, several avenues exist for further refinement and expansion of the SayfTrip application. Future work will focus on enhancing the core algorithm, optimizing app performance, and transitioning to real-world validation.

The current anomaly detection algorithm relies solely on spatial coordinates (latitude and longitude) to identify deviations from expected routes. While timestamps are collected in the app, they are not currently utilized within the anomaly detection process. Future development will extend this to a spatiotemporal model, incorporating the temporal dimension (timestamps) and directional data. This will involve:

- **Implementation of a Spatiotemporal Prediction Model:** Instead of simply assessing deviations from a predefined route, the algorithm will predict likely future locations based on historical trajectory data and expected travel patterns. This can be achieved through techniques such as Hidden Markov Models (HMMs), Kalman Filters, or Recurrent Neural Networks (RNNs), each offering different tradeoffs in terms of complexity and predictive accuracy.
- **Directional Awareness:** Incorporating heading data (direction of travel) will allow the algorithm to differentiate between intentional turns and unintended deviations.

While this would involve polling additional sensors such as the magnometer, acceleromets and gyroscope, which will increase the resource intensiveness of the app, it may improve the precision of anomaly detection.

- **Personalized Anomaly Thresholds:** Recognizing that travel behavior varies significantly between individuals, the system will implement personalized anomaly thresholds based on user-specific historical data. This will improve the algorithm's ability to distinguish between legitimate deviations and genuine threats. Machine learning techniques, such as clustering or anomaly detection algorithms, can be employed to learn these personalized thresholds.
- **Real-World Validation** This application would be released to a small controlled user group to conduct studies and evaluate the application's effectiveness in real-world scenarios. These studies will assess the accuracy of anomaly detection, the usability of the application, and the impact on user perception of safety. The results of th is will be used to improve the application.

REFERENCES

[1] H. Zhu, R. Lu, C. Huang, L. Chen, and H. Li, "An efficient privacy-preserving location-based services query scheme in outsourced cloud," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 7729–7739, 2016.

[2] Z. Li, J. Ma, Y. Miao, X. Wang, J. Li, and C. Xu, "Enabling efficient privacy-preserving spatiotemporal location-based services for smart cities," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 5288–5300, 2024.

[3] F. Song, J. Liang, C. Zhang, Z. Fu, Z. Qin, and S. Guo, "Achieving efficient and privacy-preserving location-based task recommendation in spatial crowdsourcing," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 4006–4023, 2024.

[4] Y. Xia, S. Pan, X. Meng, W. Gao, F. Ye, Q. Zhao, and X. Zhao, "Anomaly detection for urban vehicle gnss observation with a hybrid machine learning system," *Remote sensing*, vol. 12, no. 6, p. 971, 2020.

[5] Z. Liao, Y. Yu, and B. Chen, "Anomaly detection in gps data based on visual analytics," in *2010 IEEE Symposium on Visual Analytics Science and Technology*, 2010, pp. 51–58.

[6] Q. Wang, W. Lv, and B. Du, "Spatio-temporal anomaly detection in traffic data," in *Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control*, ser. ISCSIC '18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: https://doi.org/10.1145/3284557.3284725

[7] Facts and figures: Focus on small island developing states. [Online]. Available: https://www.itu.int/itu-d/reports/statistics/facts-figures-for-sids/

[8] Digital 2025 - trinidad and tobago. [Online]. Available: https://datareportal.com/reports/digital-2025-trinidad-and-tobago

[9] Gsma mobile connectivity index - trinidad and tobago. [Online]. Available: https://www.mobileconnectivityindex.com/index.html?utm_source=kepios&utm_medium=partner#year=2024&zoneIsocode=TTO&analysisView=TTO