Design and Implementation of A Real-Time Log-Based AI Personalized Voice Coaching and Dynamic Safe Route Recommendation System

Minjeong Yu[1], Gunhee Park[1], Donghyeok Kim[1], Hyuncheul Park[1], Seungjae Lee[2]

Department of Computer Science, Sunmoon University, Korea

DOI: https://doi.org/10.47772/IJRISS.2026.100700716

Received: 26 July 2026; Accepted: 31 July 2026; Published: 11 August 2026

ABSTRACT

With the rapid advancement of wearable devices and location-based services, personalized fitness tracking has gained significant attention. However, conventional running applications often fail to provide real-time adaptive feedback and dynamic route optimization based on runners' immediate physiological states and environmental hazards. To address these limitations, this paper proposes a real-time, log-based AI personalized voice coaching and dynamic safe route recommendation system. The proposed architecture systematically collects multi-modal telemetry logs—including biometric data and spatial coordinates—to analyze runners' performance profiles in real time. By leveraging advanced machine learning models, the system delivers context-aware voice coaching tailored to individual fitness levels and fatigue metrics. Furthermore, it incorporates a dynamic routing algorithm that evaluates environmental safety factors to recommend secure and optimal running paths. Experimental results demonstrate that the proposed system enhances user safety, improves workout efficiency, and delivers an optimized personalized running experience.

Keywords— Intelligent Healthcare, AI Voice Cloning, Spatial Database, Dijkstra Algorithm, Real-time Data Processing, Personalized Support System

INTRODUCTION

The modern health-management paradigm is evolving beyond mere treatment and prevention of disease toward mobile-healthcare and Quantified-Self systems that precisely datafy an individual's everyday physical activity and provide real-time, tailored feedback on that basis. Outdoor running in particular has become a representative aerobic exercise that anyone can easily take up without special physical equipment, and worldwide demand is surging for tracking applications that analyze a user's exercise trajectory, cumulative distance, elapsed time, and pace changes using the GPS and inertial measurement unit (IMU) built into smartphones [1].

However, most currently commercialized mobile running applications remain passive recorders that log the user's trajectory on a 2-D planar map and summarize statistical metrics after exercise. Unlike an indoor treadmill, the running environment is exposed to countless uncontrollable variables—disruption of biological rhythm due to waiting at crosswalks, cardiopulmonary load from steep elevation gains, and the risk of safety accidents from insufficient illumination during night running [2]. Existing general-purpose map services and commercial APIs (e.g., Google Maps, Kakao routing) are optimized for the “shortest distance” or “least time” arrival of vehicle navigation, and are therefore algorithmically unsuitable for the multi-condition search that must guarantee a pedestrian's comfort and safety, or for generating circular routes whose start and end points coincide [3], [5]. The audio feedback provided during training likewise remains uniform mechanical text-to-speech (TTS) reading, showing clear limits in eliciting emotional immersion and sustained motivation [8].

Just as a dynamic-difficulty-adjustment (DDA) system in a game environment analyzes real-time log data to provide support matched to the player's level and thereby prevents game churn, real-world running activity should likewise have its route and coaching feedback adjusted intelligently and dynamically according to the user's past records, that day's condition, and the risk level of the surrounding terrain. To bridge this technical and experiential gap, this study proposes RunMate, a next-generation real-time personalized dynamic support system.

This system completely escapes dependence on external map APIs and independently develops a Dijkstra-based multi-weight safe-route recommendation engine by fusing precise OSM road-network data with public data (streetlights, crosswalks, and elevation data) into its own spatial database (PostGIS) [4], [6]. In addition, considering that runners have limited capacity for visual feedback during exercise, it introduces a GPT-SoVITS v4-based AI voice-cloning pipeline that provides real-time running guidance in the user's own recorded voice or the familiar voice of an acquaintance [9]. The system analyzes the user's running state in a multidimensional manner through a real-time GPS multi-stage filtering mechanism and geometric bearing computation, and applies a priority-based audio-interrupt architecture to deliver appropriate turn guides and pace briefings in a timely manner when entering complex urban intersections [7].

This report describes in detail the overall distributed-architecture design of this AI coaching and dynamic route-support system and the implementation principles of its spatial-computation algorithms. It also examines in depth, from the dual-database design for large-scale I/O processing to the technical challenges derived during the integration of mobile hardware and AI servers and the engineering techniques used to overcome them, in order to present academic and practical standards toward which user-centric mobile-healthcare systems should advance.

RELATED WORK

To secure the academic and technical validity of this system's core engine design and algorithm choices, the literature on GPS-trajectory big-data analysis, spatial databases and path-search algorithms, and the application of state-of-the-art generative-AI speech synthesis (TTS) models was reviewed as follows.

GPS-Trajectory-based Running Environment Variables and Spatial-Preference Modeling

A user's outdoor running activity is greatly affected not only by the individual's physical ability but also by surrounding built-environment and natural-environment variables. With recent advances in big-data collection technology, research that analyzes demographic exercise patterns and spatial preferences using large-scale GPS-trajectory data extracted from crowdsourcing-based mobile exercise applications (e.g., Strava, MapMyFitness) is being actively conducted [1]. According to several studies, runners tend to overwhelmingly prefer continuous, natural routes connected to tree-lined green space or waterfront blue space rather than merely smooth, paved asphalt roads. With respect to surface texture as well, it has been empirically demonstrated through GPS speed data that runners prefer shock-absorbing unpaved roads or track surfaces.

Furthermore, studies that elucidate the nonlinear ripple effects of environmental constraints on running trajectories using spatial machine-learning models are drawing attention [2]. It has been proven through time-series trajectory analysis that during night running, parks and narrow alleys frequently used in the daytime are thoroughly avoided for reasons of crime and safety, and that securing visibility through streetlight and security-light illumination acts as the top priority for route selection. In addition, environmental factors act as highly complex variables—such as the pattern of preferring dense tree shade to avoid direct sunlight in hot climates.

Attempts to convert users' qualitative experience into a quantified route-recommendation engine are also continuing. A study that cross-validated in-depth interviews of runners with objective GPS trajectories strongly suggests that runners strongly prefer low-traffic, pedestrian-only roads that allow them to avoid vehicle exhaust and noise, as well as continuous paths without waits caused by crosswalks, and that such preferences should be reflected as core weights in a recommendation system [3]. Existing studies commonly emphasize the importance of the nonlinear interaction that arises when multiple environmental variables (streetlights, gradient, crosswalks, etc.) are combined, and this body of literature provided a strong theoretical foundation for designing our own algorithm that computes parameters such as elevation gain, crosswalk count, and illumination density in combination—not just road distance—to derive user-preference weights.

Spatial Databases and Dijkstra-based Networking Algorithms

For dynamic and flexible user-customized route search, a sophisticated graph-database structure is essential—one that goes beyond simply connecting two points to dividing roads into nodes and edges and binding complex attribute data to each segment. To this end, academia and industry have continuously studied and validated the use of PostGIS, the spatial extension module of PostgreSQL, together with pgRouting, a network-analysis-specialized library that operates on top of that data. pgRouting supports high-performance shortest-path algorithms such as Dijkstra, A*, and Floyd–Warshall as SQL built-in functions, and enables precise computations such as bidirectional edge search, travel-distance constraints, and turn restrictions, so it is widely used from logistics-network analysis to pedestrian wayfinding [6].

Among prior studies addressing the practical application of path-search algorithms, the field of emergency-vehicle routing systems has a logical structure very similar to that of this study. Techniques have been proposed that integrate various edge-cost parameters—such as the number of intersections, the number of lanes, and real-time road-construction conditions—into the Dijkstra algorithm to minimize travel time in order to control the variables of vehicle movement [4]. This shares the same context as our scheme of converting uphill gradient into fatigue distance, or substituting crosswalks as a wait-time penalty, and integrating these into Dijkstra weights.

In addition, research that implemented a pedestrian-centric wayfinding algorithm using OSM-based spatial data points out the clear limitations of commercial APIs [5]. That study demonstrated that commercial APIs cannot finely control the special safety requirements of tourists or small-scale local pedestrians, and benchmarked the superiority and algorithm-execution speed of PostGIS-based self network analysis that reflects regional characteristics. The results of this prior literature served as a key academic basis for our project's architectural decision to control complex user-set weights directly with the Dijkstra algorithm on a self-built road graph, rather than relying on a one-way commercial map API.

AI-based Personalized Speech Synthesis (TTS) and Mobile-Healthcare Application

Speech synthesis (TTS, Text-to-Speech) technology is evolving beyond the simple mechanical reading of text of the past into an intelligent audio interface that leads emotional and intelligent interaction between the user and the system. Recent generative-AI voice-cloning models combined with deep learning and large language models (LLMs) show the pinnacle of voice-cloning technology, perfectly imitating a specific speaker's intonation, breathing pattern, and even subtle emotional state with only a small amount of reference audio (zero-shot or few-shot) [8], [10].

At the center of this innovation is the GPT-SoVITS model, which improves the VITS structure and extracts semantic tokens of sound based on a self-supervised learning model (HuBERT). According to prior research, GPT-SoVITS can imitate a speaker with just 5 seconds of reference audio, and when fine-tuning is performed with less than 1 minute of data, it exhibits emotional expressiveness and naturalness that surpass existing commercial models. In particular, studies on optimization techniques for benchmarking and serving such high-performance models with low latency on limited hardware resources—such as Apple Silicon models or consumer mobile-edge environments—are also being actively published, raising the possibility of incorporation into actual services.

The effect of introducing emotional TTS in the sports and medical-healthcare domains is even more clearly demonstrated in the literature. Research on a framework that analyzes sports-broadcast video, such as baseball, with deep learning to generate thrilling commentary in real time shows that dynamic audio feedback matched to the progress of exercise can maximize the user's immersion and situational-awareness ability. Another study in a medical environment proposed the concept of using AI to clone the voices of family members for patients admitted to an intensive care unit (ICU) who suffer from extreme stress. That study confirmed that using a familiar, intimate speaker's voiceprint, instead of mechanical and uniform guidance sounds, can stabilize the sympathetic nervous system, greatly reduce anxiety, and powerfully motivate rehabilitation [9].

In step with these research trends and psychological mechanisms, the proposed system introduces, for the first time in a running app, a GPT-SoVITS v4-based voice-cloning pipeline so that users do not feel isolated during exercise and can continuously receive positive stimulation to maintain pace. Furthermore, by building a background asynchronous-processing and self digital-filtering pipeline to overcome hardware constraints, it realizes the theoretical value of prior research at the level of an actual mobile service.

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VII July 2026

Proposed System

This chapter details the architecture and detailed components that RunMate designed to process user requests in real time and guarantee a stable service. The system is tightly woven into a four-tier microservices form: a React Native-based frontend, Spring Boot-centric backend business logic, an AI model-serving layer driven by Python FastAPI, and a database specialized for spatial-data processing.

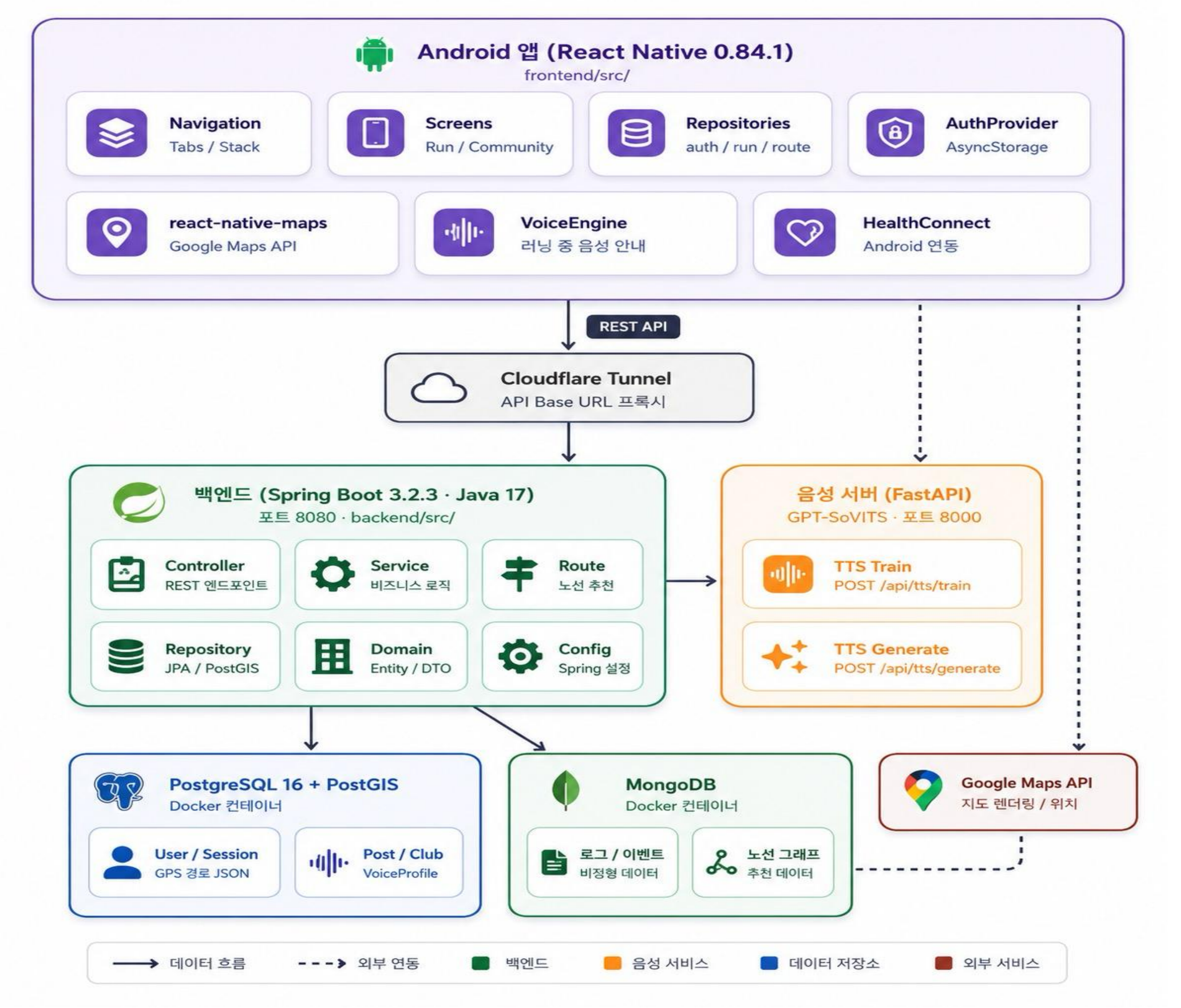


Fig. 1 RunMate overall system architecture (4-tier: React Native, Spring Boot, FastAPI voice server, dual database)

Hybrid Architecture and Dual Database System Design

The server infrastructure that forms the foundation of this system was devised as a Dual Database Architecture so as to accommodate all the essential characteristics of mobile healthcare. A mobile-healthcare platform must simultaneously and faultlessly process transactional “structured data” such as user accounts and community posts, and high-volume “unstructured time-series data” such as sensor logs and GPS-coordinate arrays that pour in continuously every second during a run. Processing these with a single database would cause lock phenomena due to I/O contention, leading to system-wide latency.

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VII July 2026

TABLE I **Dual Database (Polyglot Persistence) Design**

Database	**Separation Structure and Stored Data Type**	**Selection Rationale**	**Fault-Tolerance Mechanism**
PostgreSQL 16	Member information, running-session statistical aggregates (time, pace), personalized route-recommendation history, community posts and like records	Strong transactional consistency based on ACID principles; efficiency of join operations; adopted as the primary store for full compatibility with PostGIS 3.4, which supports spatial extension.	Invokes the framework's Synchronous Persistence Commit to finalize the disk commit with top priority within the main thread.
MongoDB 7.x	Raw GPS-trajectory array data transmitted per second (latitude, longitude, altitude), hardware sensor logs (acceleration, gyro), and meta logs needed for AI training	Flexible document-type schema structure, insert-operation optimization for pushing in tens of thousands of time-series records simultaneously, and ease of a scale-out structure.	Applies an Asynchronous Exception Isolation Wrapper structure for complete logical isolation. Designed so that even if a MongoDB insert operation fails due to a timeout, etc., the data of the main DB is not rolled back.

Through the Polyglot Persistence pattern described in the table above, this server system avoided bottlenecks even under heavy traffic. The backend system was implemented with the Java 17 and Spring Boot 3.2.3 framework. In addition, to overcome the constraints of a development server running on a local network and to allow multiple testers to connect from the outside using real devices, the deployment infrastructure was completed with a tunneling relay.

User Interface and Healthcare Frontend Layer

The frontend controlled the resolution and interface-fragmentation problem of the two major mobile operating systems with responsive size-calculation logic.

The view of the user application maximizes the visibility of major buttons by using an intuitive gray-card UI on a white background and orange (#FF7840) as the core point color to emphasize the dynamism of running.

On the main screen, the user's current location is centered on a full-screen map based on a commercial map-rendering API, and the user can freely switch among three running modes (free running, recommended-route search, ghost running).

Among these, “Ghost Running” is a core feature that provides an environment for competing with the user's past running records in real time. Based on speed and pace-change data from past runs, it renders a virtual runner (ghost) trajectory and gauge together on the map on which the user is currently running, strongly driving the psychological desire for achievement and immersion, similar to a dynamic-difficulty-adjustment (DDA) system. After exercise ends, the record is immediately delivered to the social community so the user can interact with other runners.

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VII July 2026

Fig. 2 RunMate main screen with three running modes (Free Run / Route / Ghost)

Self-built Road Network and Dynamic Personalized Safe-Route Recommendation Engine

The most advanced technical achievement of this system is that, rather than relying on the static routes provided by commercial routing APIs, it performs user-customized Dijkstra search on a complete in-house road graph loaded with physical-environment geographic information. To this end, the engine directly built a multi-stage spatial-processing pipeline.

Road-Network Graph Construction (2-Pass Scan Algorithm): About 1.57 million road-segment data extracted from public data and OpenStreetMap (OSM) were refined and bulk-loaded into the spatial database's Road Network Table. Because a single scan has an omission limit—it cannot split the intersection point of a new road that crosses the midpoint of a very long road network—a 2-Pass scan algorithm that finds intersection points was introduced. Using the intersection points thus identified as references, existing road objects were forcibly split, constructing a complete undirected/directed node-edge graph.

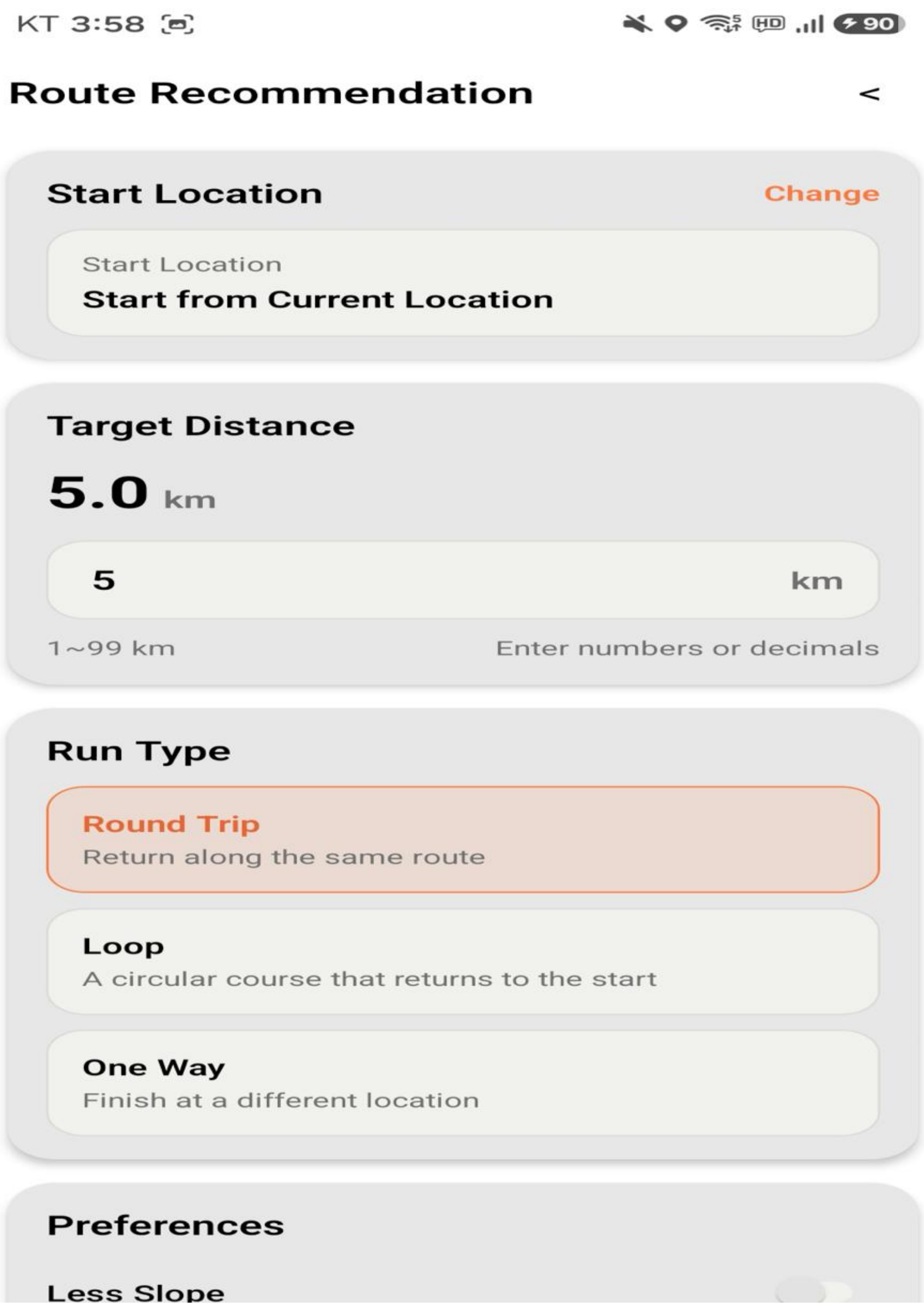
Safety-Data Attachment and Grid-Aggregation Optimization (Resolving the N+1 Spatial-Query Bottleneck): Querying streetlights, elevation points, and crosswalks individually for every traversal of millions of edges induces an enormous disk-I/O bottleneck. To block this at the source, at system startup the local space is forcibly divided into a specified grid scale and memory-based pre-aggregation is performed.

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VII July 2026

Lazy-Evaluation Multi-Weight Cost Formula: A lazy-evaluation scheme that computes the weight in real time only when the search pointer visits a particular edge was adopted. The cost-modeling formula directly substitutes the user's preference conditions into physical distance as follows.

By the mechanism of this formula, an uphill that rises 1 m in elevation is treated as a punitive distance cost equal to running 50 m more on flat ground, forcing the algorithm to find a detour. Conversely, a bright road with a high security-light density has its route cost reduced by multiplying a correction coefficient, naturally inducing the path to extend toward safe light.

Buffer-Avoidance Mechanism for Ensuring Diversity: A polygon buffer 20 m wide is generated around the trajectories of the most recent five runs, and the ratio at which a new recommended-route candidate intersects this buffer area is computed. Edges with high similarity have their pass-through cost forcibly amplified, endowing the system with the intelligence to actively explore entirely new detour routes.



INTERNATIONAL JOURNAL OF RESEARCH AND INNOVATION IN SOCIAL SCIENCE (IJRISS)
ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VII July 2026

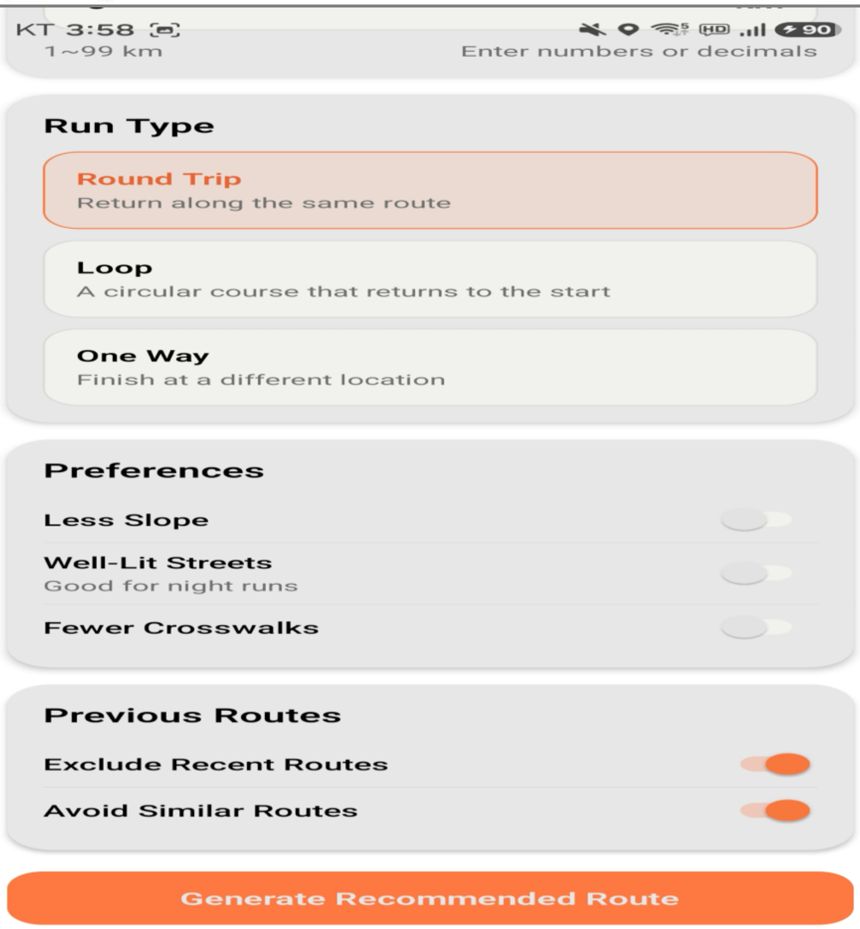


Fig. 3 User route-preference input (distance, run type, and safety preferences)

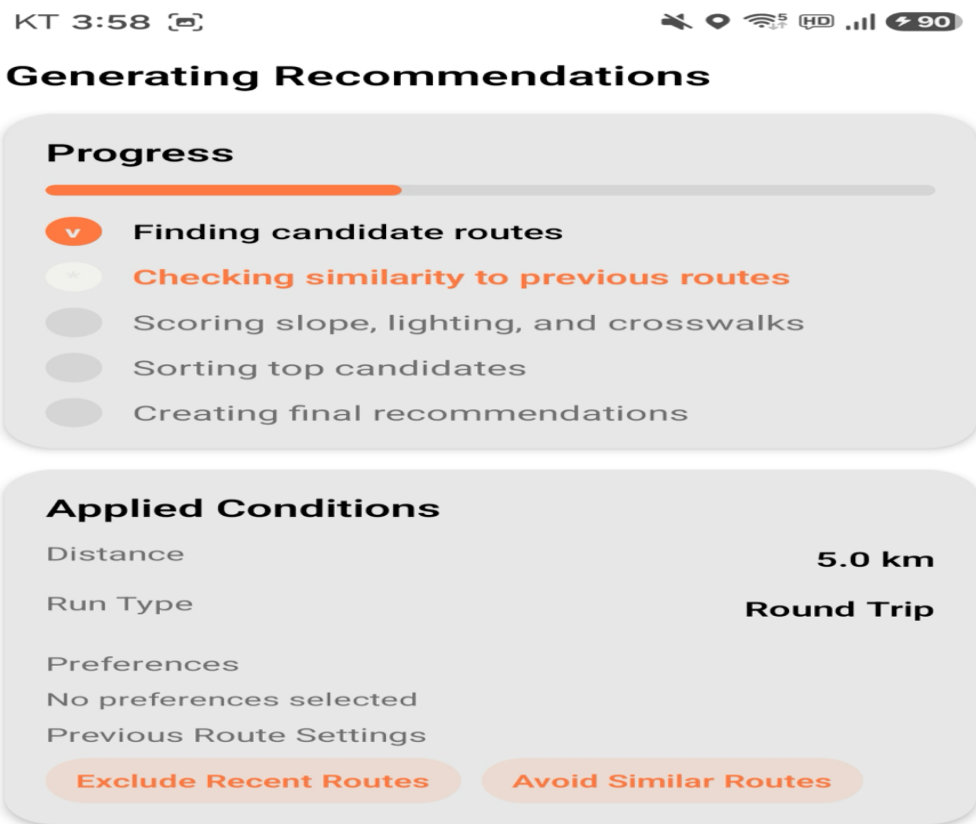


Fig. 4 Route generation process screen

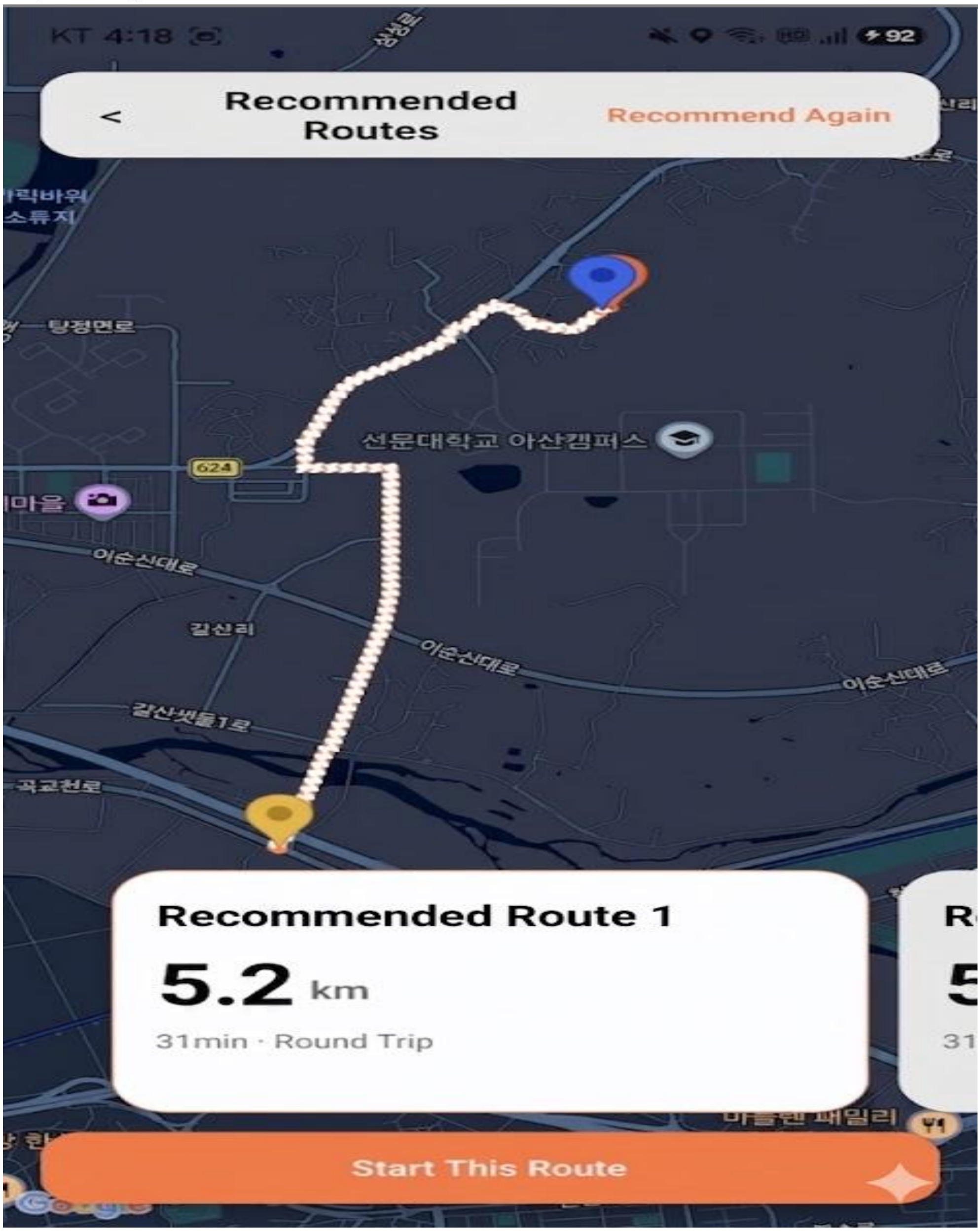


Fig. 5 Route generation result screen

GPT-SoVITS-based Intelligent Personalized AI Voice-Cloning Pipeline

The technology of synthesizing the voiceprint of the user themselves or a familiar speaker to provide guidance, rather than a simple notification beep, acts as a powerful psychological driver that increases attachment to the system and dramatically lowers the probability of giving up on exercise. RunMate overcame the limitations of mobile devices by introducing a background asynchronous AI-pipeline architecture that combines the FastAPI framework with the GPT-SoVITS v4 model.

Native Audio Collection and High-Precision Digital Preprocessing: When a script is recorded on the mobile client side, the native API captures the audio in a high-quality format and delivers it to the AI serving layer via a proxy. To remove the noise of an everyday recording environment, the server passes the audio through a high-order high-pass filter (cutoff frequency 80 Hz) within the User Voice Training Pipeline. In addition, through a Frequency Domain Noise Reduction Module, only background noise is eliminated while the speaker's inherent voiceprint characteristics are perfectly preserved.

RMS-based Optimal Reference Selection: In few-shot learning, among the preprocessed voices, the root-mean-square (RMS) amplitude value is back-computed to automatically designate the audio index with no waveform

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VII July 2026

clipping and the best volume as the Optimal Reference Audio file, and the structured text metadata of the reference data is persistently stored to prevent generation distortion caused by prompt-text mismatch.

Self-Healing Inference Asynchronous Loop: The array of Guide Script Metadata needed for the running state is batch-inferred by an Asynchronous Background Worker. To prevent hallucination or silence phenomena that occur when inferring many files, a self-healing mechanism was applied in which, if defective data is identified, a monitoring loop within the Speech Inference Engine detects it and forces re-inference of only that sentence up to a maximum of five times.

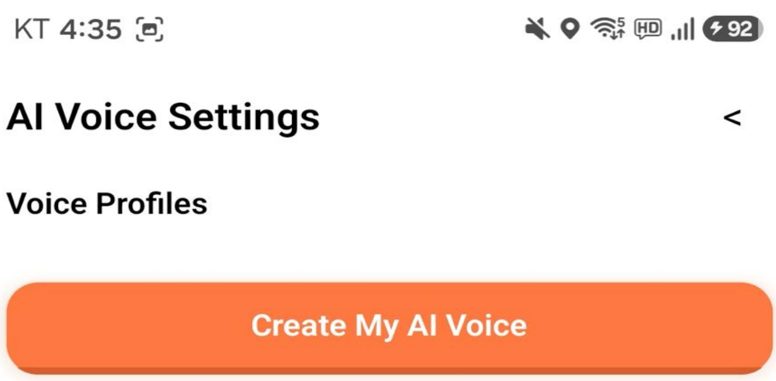


Fig. 6 AI voice settings (voice-profile creation)

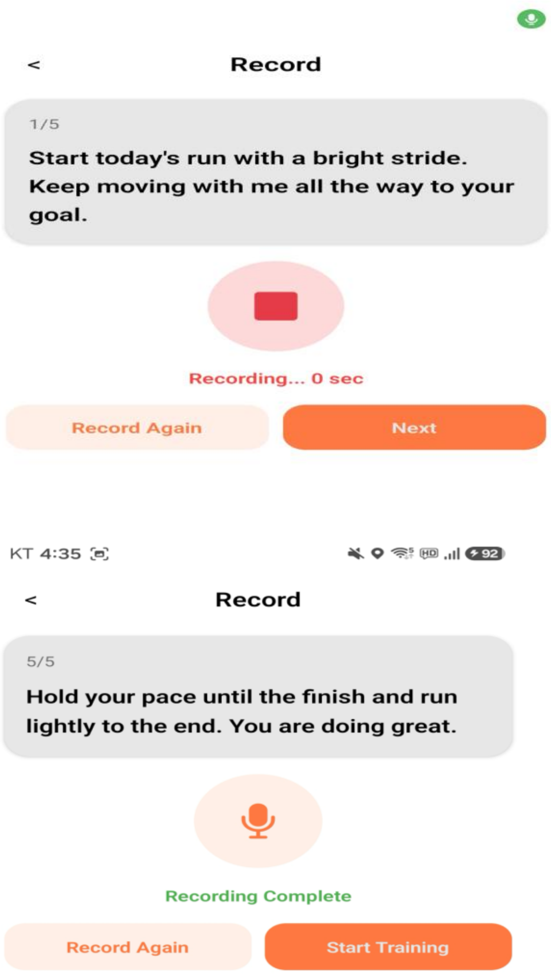


Fig. 7 Training-voice recording (script 1/5–5/5)

INTERNATIONAL JOURNAL OF RESEARCH AND INNOVATION IN SOCIAL SCIENCE (IJRISS)
ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VII July 2026

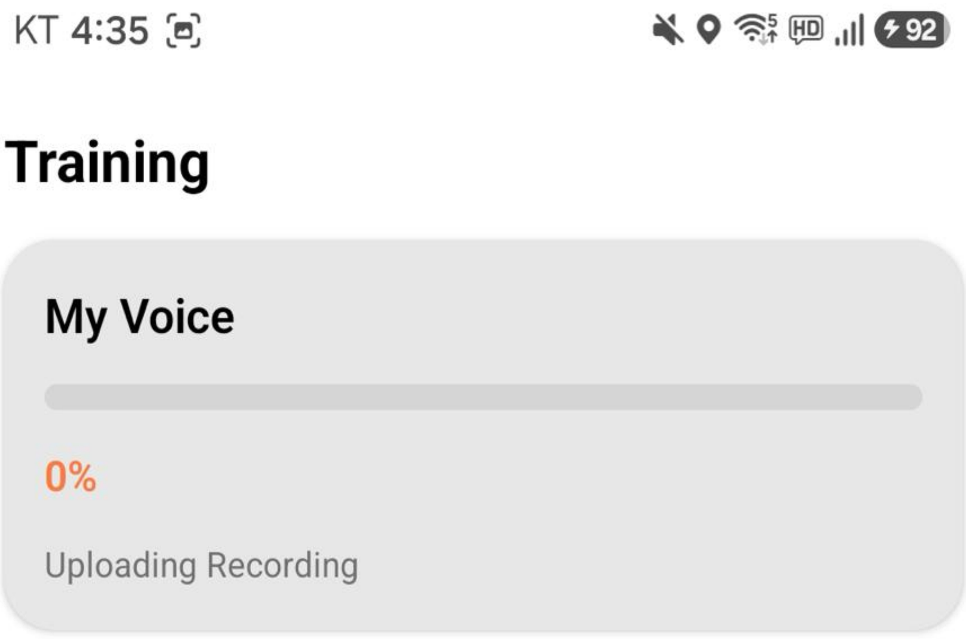


Fig. 8 Voice training (recording upload and model training)

Real-time Running-Trajectory Tracking and Multidimensional Position-Correction Filtering Engine

On the mobile client, a Native Geolocation Tracking Module is introduced to collect coordinates based on a sensor-fusion location provider (FusedLocationProvider). To control multipath errors caused by radio-wave reflection that occurs in an urban forest of buildings, a powerful six-stage geometric multi-filter was logically implemented.

TABLE II **Six-Stage Geometric GPS Correction Filter**

Filter Stage	Logical Control and Threshold Design	Effect and Verification
① Accuracy-limit control	Discard bad coordinates whose error-radius range, returned by the OS hardware itself, exceeds 70 m	Block fatal initial noise at the source
② Geometric speed validation	If the physical travel speed between two consecutive coordinates exceeds 10 m/s, regard it as a mechanical jump and ignore it	Suppress abnormal spatial-warp phenomena
③ Smoothing	Compute a 3-point moving average over the most recent 3 valid coordinates to cancel noise	Mitigate fine jitter of the trajectory
④ Initial-stabilization exclusion	Forcibly ignore the first 2 dummy coordinates that come in before the sensor stably locks the GPS satellite signal	Perfectly defend against distance inflation at the start point
⑤ Jump and recovery filter	Initially ignore abrupt movement exceeding 30 m, but if coordinates are received consecutively in that direction, recognize it as movement and recover the distance	Secure resilience for signal recovery after passing through a tunnel between buildings
⑥ Stationary micro-drift control	Exclude vibration movement of 4 m or less occurring while the user is stationary from the measured-accumulation pipeline	Control the increase of phantom exercise distance while waiting at a crosswalk

Based on the filtered coordinates, the user's real-time coordinate is orthogonally projected onto the many polylines of the recommended route to derive the nearest perpendicular intersection (snap coordinate) [7]. In the backend computation, only the "measured coordinates" that have passed filtering are summed independently, thereby completely blocking the inflation phenomenon in which distance is shrunk or exaggerated due to screen matching.

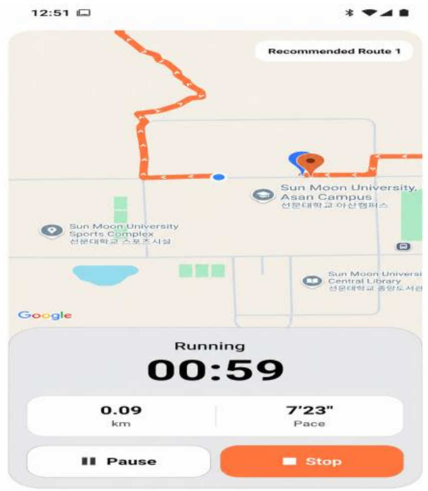


Fig. 9 In-run map screen — route-polyline tracking and current-position snap (live distance, time, pace)

IV. Research Results and Troubleshooting

Benchmarking Verification of the Dynamic Route-Recommendation Server and Technology Stack

As a result of introducing spatial-data pre-aggregation and the Dijkstra lazy-evaluation algorithm, the response latency of the Route Generation Interface achieved an optimization result, being shortened to within an average of 1–3 seconds, compared with an average of 15 seconds in the previous scheme that combined external commercial APIs and applied post-filtering. This made it possible to respond in real time to high-dimensional exercise courses, such as circular routes that weave together multiple waypoints by splitting the bearing of the departure direction.

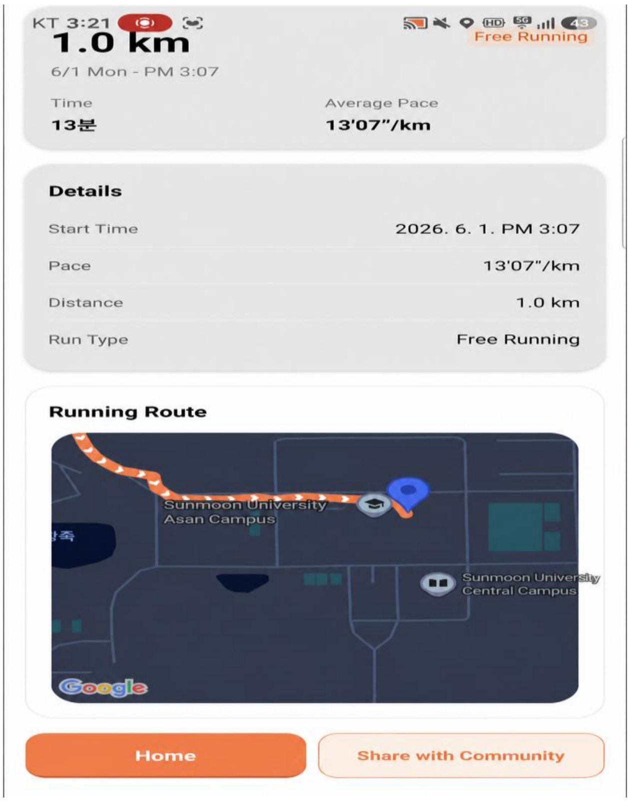


Fig. 10 Post-run result summary — distance, time, average pace, and traced route

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VII July 2026

B. Verification of the Geometric Turn-Guide and Ghost-Running Interpolation Algorithms

By applying spherical trigonometry, only the points at which the absolute bearing difference between the vector the user has just traversed and the vector to be entered exceeds 30 degrees were detected in real time as valid turn sections. In particular, to prevent sounds from overlapping in bursts in complex alleys, an Adjacent Corner Merging Algorithm clustered them into a single corner with the highest curvature. In ghost-running mode, the Ghost Trajectory Comparison Algorithm applied binary-search-based linear interpolation to precisely restore the ghost's time axis down to the nanosecond, making the real-time competition environment operate without error.

C. Operation Results of the Dynamic Audio-Queuing and Self-Fallback System

To control hardware-channel-contention conflict phenomena when several guidance announcements occur simultaneously during a run, a five-stage priority-queue logic was developed. In particular, to prevent guidance sounds from pouring out, a background timeout queue based on a Delay Scheduler Algorithm was introduced so that they are scheduled with an ergonomic time gap. In a data-network-loss situation, an error code is detected and the system instantly falls back to the binding layer of a Local Default Audio Resource bundled at app installation, providing uninterrupted coaching guidance.

D. System Integration and Major Troubleshooting During Development

Table III summarizes the major technical problems faced during system integration and development, together with the engineering measures taken to resolve them.

TABLE III MAJOR TROUBLESHOOTING DURING SYSTEM INTEGRATION

Faced Problem and Failure Symptom	Solution and Architectural Measures
Transaction non-atomicity in the dual-DB structure (post-mapping error)	Apply Null Coalescing to tolerate session-load delay. Finalize the primary store first via a Synchronous Persistence Commit within the main thread, and separate sub-DB operations with an Asynchronous Exception Isolation Wrapper.
TTS tunnel timeout in the API deployment environment	Convert the existing Synchronous Blocking Invocation into an asynchronous worker. Release the connection immediately upon a client request, hand off the actual voice computation to an Asynchronous Task Queue, and then restructure into a webhook scheme that polls.
Road-network intersection-split omission defect	Introduce a 2-Pass scan architecture that collects intersection coordinates and re-splits existing road segments based on them, overcoming the limit of a 1-Pass single scan.
Premature-termination malfunction of the round-trip route algorithm	Advance the arrival-judgment conjunction so that the algorithm terminates only when the three defensive conditions—'destination proximity,' 'route progress ≥ 0.95,' and 'measured cumulative travel distance ≥ 0.85 of the total'—are simultaneously satisfied.
Engine lock-up due to hardware audio-session deprivation	Identified the phenomenon in which the Playback State Lock Flag is held indefinitely when a native error occurs from external-app intervention. Forcibly bind the native audio player's Exception Detection Callback to the engine to establish a recovery routine that instantly resets the flag when an exception occurs.

CONCLUSION AND FUTURE WORK

This paper proposed and designed a comprehensive Real-time Log-based AI Personalized Voice Coaching and Dynamic Safe Route Recommendation System to address the shortcomings of conventional static fitness applications. By integrating multi-modal telemetry log processing with advanced machine learning algorithms, the system successfully delivers real-time, individualized voice coaching tailored to the runner's physiological status and fatigue levels. Additionally, the incorporation of a dynamic safe routing algorithm ensures that users can navigate optimal paths while minimizing exposure to environmental and safety hazards.

Experimental evaluations and architectural validations indicate that the proposed system significantly improves workout efficiency, enhances user safety, and provides a highly engaging, personalized running experience. Future work will focus on scaling the system architecture, incorporating predictive analytics for long-term health tracking, and conducting extensive field tests across diverse urban environments.

ACKNOWLEDGMENTS

This research was supported by the MSIT(Ministry of Science ICT), Korea, under the National Program for Excellence in SW, supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation) in 2026 (No. 2024-0-00023).

REFERENCES

1. T. Shi and F. Gao, “Utilizing Multi-Source Geospatial Big Data to Examine How Environmental Factors Attract Outdoor Jogging Activities,” Remote Sensing, 2024.
2. W. Yang, J. Fei, Y. Li, H. Chen, and Y. Liu, “Unraveling nonlinear and interaction effects of multilevel built environment features on outdoor jogging with explainable machine learning,” Cities, 2024.
3. H. Issa, A. Guirguis, S. Beshara, S. Agne, and A. Dengel, “Preference based Filtering and Recommendations for Running Routes,” SciTePress, 2016.
4. A. Fitro, S. Suryono, and R. Kusumaningrum, “Shortest Route at Dynamic Location with Node Combination-Dijkstra Algorithm,” EECSI, 2018.
5. J. Zheng, Z. Zhang, B. Ciepłuch, A. C. Winstanley, P. Mooney, and R. Jacob, “A PostGIS-based pedestrian way finding module using OpenStreetMap data,” Geoinformatics, 2013.
6. R. O. Obe and L. S. Hsu, “pgRouting: A Practical Guide,” Locate Press, 2018.
7. W. Li, Y. Wang, D. Li, and X. Xu, “MCM: A Robust Map Matching Method by Tracking Multiple Road Candidates,” AAIM, 2022.
8. Y. Geng, J. Xu, Z. Liang, J. Yang, X. Shi, and X. Shen, “Scaling Under-Resourced TTS: A Data-Optimized Framework with Advanced Acoustic Modeling for Thai,” ACL, 2025.
9. H. Zhou, X. Wu, and L. Yu, “The comforting companion: using AI to bring loved one’s voices to newborns, infants, and unconscious patients in ICU,” Critical Care, 2023.
10. Y. Jiang, T. Wang, H. Wang, C. Gong, Q. Liu, Z. Huang, L. Wang, and J. Dang, “Expressive Text-to-Speech with Contextual Background for ICAGC 2024,” ISCSLP, 2024.