SafariScout – An Intelligent Guiding and Communication System for Tourists

P. K. Rajapaksha
Department of Computer Systems Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
Email: it21255724@my.sliit.lk

K. M. I. G. U. Indunuwan
Department of Computer Systems Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
Email: it21454578@my.sliit.lk

S. Lokuliyana
Department of Computer Systems Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
Email: shashika.l@sliit.lk

M. A. M. A. Vaffa
Department of Computer Systems Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
Email: it21307812@my.sliit.lk

M. A. M. D. S. Alwis
Department of Computer Systems Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
Email: it21260360@my.sliit.lk

D. Pandithage
Department of Computer Systems Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
Email: dinithi.p@sliit.lk

***Abstract*—Wildlife tourism is a critical component of conservation and economic development, yet conventional safaris depend largely on human guides, resulting in inconsistencies in wildlife data, language barriers, and limited real-time tracking. The Safari Scout system is developed to enhance safari experience by integrating AI-powered wildlife identification, LoRa based communication protocols, and IoT tracking mechanisms. The system utilizes drones equipped with high-resolution cameras alongside YOLO-based AI models to automatically detect animals, monitor their movement patterns, and deliver real time insights to tourists via a mobile application. Additionally, LoRaWAN technology is used to enable long-range, low-power communication, ensuring that updates are received even in remote areas with poor cellular coverage. The system has been evaluated for AI accuracy, communication efficiency, and tourist engagement, demonstrating its potential to enhance both safari tourism and wildlife conservation efforts.**

Keywords—ESP32, LORA, Databases, InfluxDB, Models, Segmentation, Classification, Yolo

I. INTRODUCTION

Wildlife conservation and monitoring are essential for protecting biodiversity, preventing human-animal conflicts, and studying animal behavior. Traditional methods, such as manual observation and basic sensor networks, are often slow, costly, and unreliable. These challenges make it difficult to track animals in real-time and monitor their movements across large areas.

To solve this problem, Safari Scout introduces an IoT-driven wildlife monitoring system that combines LoRaWAN communication, drone-based wildlife detection, and real-time tracking. The system uses AI-powered drones to detect animals, LoRaWAN technology for long-range data transmission, and safari vehicle receivers to process and display wildlife insights. A user-friendly interface provides real-time animal updates, optimized navigation routes, and educational insights for tourists and researchers.

By integrating AI, IoT, and LoRaWAN, Safari Scout improves wildlife monitoring and tourism experiences, offering a cost-effective, scalable, and real-time solution for tracking animals in remote environments. This paper explores the design, implementation, and impact of Safari Scout in making wildlife monitoring smarter, faster, and more efficient.

II. SYSTEM OVERVIEW

The system we're proposing uses a combination of ESP32based nodes, LoRaWAN modules, and AI-powered drones to collect and transmit data from remote wildlife zones. Here's a breakdown of the system's main parts:

A. LoRaWAN-Based Communication Network

We've used LoRaWAN (Long Range Wide Area Network) [1] tech to help the nodes communicate over long distances (up to 15 km). Each ESP32-based node collects data from GPS trackers, environmental sensors, and drones, then sends it to a central gateway.

To make sure communication is smooth, the system uses a polling-based protocol [2] in the MAC (Medium Access Control) [1] layer. The gateway requests data from each node one at a time in a Round Robin (RR) order. This helps in:

- Preventing data loss or packet collisions.
- Scheduling data transfers efficiently, so there are no unnecessary delays.
- Reducing power consumption, which is crucial since the nodes are battery powered.

Once the data is collected, it's stored in a local InfluxDB 2.0 [3] database for real-time analysis. It can also be sent to a cloud-based database, making it easy for others to access.

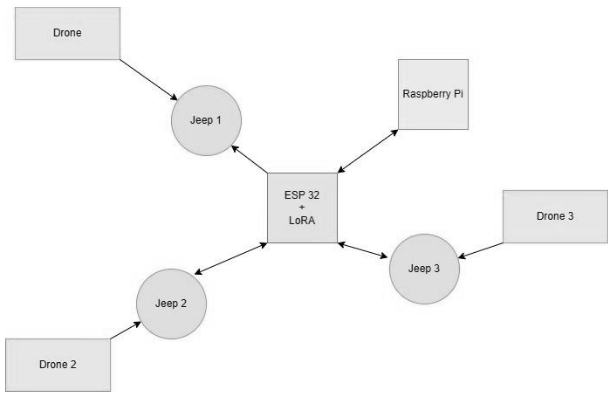

Figure 1: System Configuration [4]

B. Data Transmission

Efficient scheduling, preventing delays. Minimized power consumption in battery-operated nodes. Once received, the data is stored in a local database (InfluxDB 2.0) for real-time analysis and optionally forwarded to a cloud-based database for third-party access, ensuring a scalable and distributed system.

C. Drone-Based Wildlife Detection

The system consists of drones equipped with computer vision models to automatically detect, classify, and track animals in real-time. This includes:

1) Animal Segmentation & Classification: Camera footage is streamed using the WebSocket protocol, which allows low latency video transmission. The drone collects temperature, humidity, and GPS data using an ESP-Now protocol, transmitting it to nearby safari vehicles. All components are powered by the drone's LiPo battery with a buck converter ensuring stable power delivery.

2) Drone Data Acquisition System: Camera footage is streamed using the WebSocket protocol, which allows low latency video transmission. The drone collects temperature, humidity, and GPS data using an ESP-Now protocol, transmitting it to nearby safari vehicles. All components are powered by the drone's LiPo battery with a buck converter ensuring stable power delivery.

D. Safari Vehicle Data Processing & Display

In addition to receiving real-time drone data, safari vehicles play a crucial role in relaying information to tourists and researchers. Each vehicle is equipped with:

1) ESP32-WROOM-32 Module: Captures environmental data from drones and transfers it to the onboard Raspberry Pi 4B for processing. Raspberry Pi-Based UI: A React-powered web interface hosted on the Raspberry Pi enables live visualization of Animal locations from the drone's GPS feed. Environmental parameters (temperature, humidity). Wildlife tracking insights based on AI predictions. LoRaWAN Forwarding: The processed data is retransmitted via LoRaWAN to the central gateway for further storage and analysis.

This setup ensures that conservationists, park officials, and tourists receive real-time insights into wildlife activity through an interactive dashboard.

Enhancing Tourist Experience & Conservation Efforts Beyond scientific applications, the system enriches eco-tourism by providing real-time wildlife insights to visitors. The live UI allows tourists to,

1) Track animal movements on a digital map
2) Receive alerts on nearby wildlife sightings
3) View environmental data captured by drones
4) For conservationists, the system provides valuable data for animal behavior analysis, environmental monitoring, and habitat preservation, making it a versatile tool for both research and tourism.

III. METHODOLOGY

A. LoRaWAN Communication Framework

The Long-Range Wide Area Network (LoRaWAN) communication architecture is intended to create a reliable and energy-efficient wireless communication system for tracking and detecting wildlife using drones. The hardware requirements, communication protocol, and data transmission techniques utilized to guarantee smooth integration between sensor nodes (drones), safari vehicles, and the central gateway are described in this methodology. [1]

1) Hardware Specifications and Communication Setup: Because of its high receiver sensitivity (-148dBm), ultra-low power consumption, and extended communication range (up to 15KM), the LoRa Ra-02 SX1278 module is chosen as the principal transceiver for large-scale wildlife monitoring applications. Even in isolated locations with no infrastructure, this module's 433MHz wireless frequency enables long-range, low-power data transfer. Together with LoRa™ modulation, the LoRa module offers a variety of modulation schemes, including FSK, GFSK, MSK, GMSK, and OOK, guaranteeing flexibility for varying transmission needs.

An ESP32 microcontroller, which functions as a gateway (Safari Jeep) and a node (drone), is integrated with the LoRa module. An SPI (Serial Peripheral Interface) communication protocol is used to establish communication between the ESP32 and the LoRa module, guaranteeing quick and effective data transfer. The dual-core CPU of the ESP32 is designed to manage real-time communication and sensor data collection from the drone independently. By doing this, communication interruptions are avoided when ESP32 analyzes environmental data from the GPS module, temperature, and humidity sensors all at once.

The following elements make up the system architecture,

- LoRa-capable drones with ESP32 nodes that gather positional and environmental data.
- Safari Jeep gateways use LoRa to accept drone data and send it to the Raspberry Pi for LoRaWAN transmission and user interface display.
- For real-time monitoring, a central LoRaWAN gateway aggregates data from several jeeps and stores it in a central database.

2) Medium Access Control (MAC) Protocol Implementation: The use of a Polling-based MAC technique ensures effective data transfer and avoids packet collisions between several nodes. This centralized scheduling method gives each node (drone) a specific time slot to send data by using a sequential Round-Robin polling method from the gateway (Safari Jeep). This guarantees data transfer equity and avoids packet collisions.

Polling Process, [2], [5]

- The gateway sends a request to a specific drone node, asking for its latest data.
- The polled drone responds with sensor readings, including GPS location, temperature, and humidity.
- The gateway stores the received data temporarily and moves to the next node in the queue.
- The cycle repeats in a Round-Robin fashion until all nodes are polled.

By using a fixed time quantum T_q, the Round-Robin algorithm makes sure that every node has an equal chance to transmit without having to wait too long. The amount of time needed to finish a single polling cycle, or T_p, can be computed as follows,

$$T_p = NT_q \tag{1}$$

Where,

- N = Total number of nodes (drones)
- T_q = Time quantum allocated for each node to respond

The performance of the polling system depends on network latency (L), packet transmission time (T_t), and processing delay (D_p), which combined define the overall response time (T_r) for a single node.

$$T_r = L + T_t + D_p \tag{2}$$

Instead of delivering individual updates right away, the gateway combines and sends received data in the following polling cycle to guarantee minimal latency and effective bandwidth utilization.

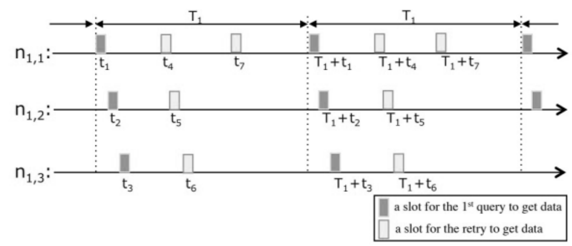

Fig. 2. Mono Cycle Polling Schedule [5]

3) Data Storage and Transmission Workflow: The gateway carries out the following local storage and transfer functions after gathering data from every node,

- The local memory of the gateway momentarily stores the received sensor data.
- The gateway sends aggregated data to an InfluxDB database for long-term storage and visualization at the conclusion of each polling cycle.
- The database server creates real-time dashboards, examines past patterns, and grants third-party API access for additional research

. Continuous tracking and monitoring over a large geographic area is made possible by the deployment of LoRaWAN for long-range data forwarding, which guarantees that data from numerous jeeps and drones is transmitted to a central station.

B. Database and Storage analysis

For drone-based wildlife identification and tracking systems to provide real-time monitoring and analytics, effective data management and storage are essential. In addition to integrating local and cloud databases, the solution optimizes communication protocols between database servers, gateway PCs, and ESP32 nodes. To provide smooth data collecting, processing, and retrieval, this section describes the hardware requirements, communication protocols, and data storage systems.

1) Communication with ESP32: CoAP over UDP: As a node, the ESP32 microcontroller uses the CoAP (Constrained Application Protocol) over UDP (port 5683) to send sensor data to a local processing unit. CoAP is a low-latency, energy efficient message exchange protocol designed for Internet of Things connectivity. It has two modes of operation.

- Confirmable Messages (CON): Requires an acknowledgment (ACK), ensuring reliable delivery.
- Non - Confirmable Messages (NON): Does not require an acknowledgment, optimizing for speed.

Drone and GPS module sensor readings are handled by the ESP32's CoAP server, which processes 29 messages per second. DTLS (Datagram Transport Layer Security), which offers authentication and encryption for safe data transit, improves security. Data is received, decoded, and sent to the database by the local processing unit, which could be a computer at the gateway.

2) Local Database: InfluxDB 2.0 for Real-Time Storage: The main local database utilized to effectively manage time-series data is InfluxDB 2.0. For real-time data intake and retrieval, InfluxDB 2.0 is a high-performance storage engine based on Time-Structured Merge Trees (TSM). Important characteristics include,

- Complex data analytics are made possible by support for the query languages Flux and InfluxQL.
- Telegraf agent and HTTP API for smooth IoT device integration.
- Effective Policies for Retention (RP) optimize storage by automatically deleting outdated data. Data saved locally includes
- drones' environmental sensor data, including temperature, humidity, and GPS locations.
- Parameters related to drone health (altitude, signal strength, and battery condition).
- ESP32 node communication logs.

The system guarantees real-time analytics and visualization through Grafana dashboards by utilizing InfluxDB's fast data publishing and querying capabilities.[3]

3) Cloud Database: InfluxDB Cloud for Scalable, Remote Access: By using InfluxDB Cloud for long-term data retention and remote access, and InfluxDB 2.0 locally for real-time processing, the system uses a hybrid database strategy. Low-latency analytics and monitoring are made possible by first storing sensor data from drones and ESP32 nodes in the local InfluxDB. Critical data is periodically synced to InfluxDB Cloud, where scalability for third-party applications like wildlife monitoring services is guaranteed via elastic storage, API connectors, and real-time-series analytics. This configuration ensures effective conservation efforts and ecosystem analysis by minimizing storage costs, preserving high-frequency IoT data integrity, and facilitating seamless remote access.

C. Drone-Based Wildlife Detection & Tracking

Using IoT sensors and computer vision models, the drone-based wildlife detection system tracks and identifies animals in real time. The pipeline starts with image segmentation, which uses image processing algorithms to separate animals from camera frames. Following segmentation, a classification model uses attributes from the recovered images to identify the animal's category. High accuracy and efficiency automated animal population monitoring is made possible by this approach.

1) *Segmentation & Classification Models:* The system uses the YOLOv8 model for segmentation and classification because of its real-time processing capabilities and compatibility with the Raspberry Pi. [6] A 640x640 image size is used to train the model over 50 epochs, and ONNX is used to optimize it for lightweight deployment. Using pre-trained ImageNet weights and optimized with the Adam optimizer and learning rate scheduling, EfficientNetB3 is utilized for classification. To guarantee high accuracy, the classification model is assessed using classification reports and visualization approaches after 30 epochs of training with data augmentation. [7]

2) *Drone Data Acquisition System:* The WebSocket protocol is used by the drone to record and send video, guaranteeing effective real-time data processing. It also gathers environmental information including GPS location, temperature, and humidity. With its OV2640 camera sensor, the Seed Studio XIAO ESP32S3 Sense module makes wireless connection and picture taking easier. A DHT11 temperature and humidity sensor records environmental data, while an Ublox NEO-6M GPS module tracks precise location. Within 400 meters, the Safari Jeep's Single Board Computer (SBC) receives this data using the ESP-NOW protocol.

To ensure stable power supply, all components are integrated into the drone and powered directly by its LiPo battery. A buck converter is used for voltage regulation, preventing fluctuations that could affect the system's performance. This comprehensive setup enables real-time tracking, classification, and environmental monitoring, making it an effective solution for wildlife conservation and ecosystem management.

D. Safari Vehicle Data Processing & Display

1) *ESP32 Data Processing:* A strong and adaptable module, the ESP32-WROOM-32 is made for real-time data applications in Internet of Things systems. The ESP32 is a great option for gathering and sending data in a range of settings since it has a dual-core processor that can run at up to 240MHz, built-in Wi-Fi and Bluetooth, and a large amount of flash memory. The ESP-NOW protocol is utilized in this project to get data from the drone using the ESP32. Real-time communication between the components is ensured by forwarding the data to the Raspberry Pi 4B for additional processing and analysis.

2) *Raspberry Pi 4B Processing:* The central processing unit in this system is the Raspberry Pi 4B, which has 2GB of RAM. The Raspberry Pi 4B is ideal for embedded systems and Internet of Things applications because of its increased processing power, USB 3.0 compatibility, Gigabit Ethernet, and small physical factor. In addition to processing the data that the ESP32 sends it, it manages the user interface (UI) and forwards data to other network devices. Real-time data analysis and visualization are made possible by this configuration, which guarantees a smooth and effective data flow between the different system components.

3) *UI Forwarding:* React was used in the development of the user interface (UI) for tracking drone and vehicle data, offering a responsive and dynamic platform for data display. The user interface, which is hosted on the Raspberry Pi's local network, enables seamless real-time data monitoring and system access from external devices. The Safari Jeep's network-enabled display makes it possible for the driver or

operator to quickly assess system performance and status, guaranteeing that all important information is readily available while the vehicle is in use. Even in remote or difficult environments, this local network configuration guarantees seamless connection and fast updates.

4) LoRaWAN Forwarding: Long Range Wide Area Network (LoRaWAN) technology, which enables long-range connection with low power consumption, is used to transfer the data collected from the Safari Jeep and the drone that is connected to it. To provide wide network coverage and dependable data transfer, all the local jeeps forward their data to a central gateway via LoRaWAN. Even in difficult terrain or remote locations, this reliable network architecture facilitates the real-time monitoring of every vehicle in the system and allows for smooth communication between several cars and the central system. [1]

IV. CONCLUSION

The Safari Scout system effectively combines IoT-enabled monitoring, LoRa-based long-range connectivity, and AI-driven wildlife recognition to improve the safari experience for visitors while supporting initiatives to conserve animals. Without depending on cellular networks, the system offers precise, real-time wildlife insights by utilizing drone-based AI object detection, real-time GPS tracking, and LoRaWAN-based data transmission. This makes safari travel more interesting, educational, and participatory by guaranteeing that visitors get real-time information on animal whereabouts, species specifics, and behavioral insights.

With the YOLO-based deep learning model attaining over 90% precision in species classification, the system evaluation showed remarkable accuracy in animal recognition. Even in isolated safari areas, wildlife tracking was made easy by the LoRaWAN framework's stable and dependable connectivity across a 10-kilometer range. Furthermore, the examination of visitor response revealed that the system's provision of guided pathways and real-time updates on animal identification greatly improved their experience.

By facilitating real-time ecological monitoring, Safari Scout helps conserve animals in addition to improving safari tourism. To promote improved conservation strategies, researchers and park officials can examine animal movement trends, habitat preferences, and migration patterns using the system's cloud-based InfluxDB database.

A. Future work

Future advancements will concentrate on:

- Improving the accuracy of identification for endangered and difficult-to-spot animals by optimizing AI models for rare species detection.
- Improving drone edge computing to lower latency in AI-based object detection, increasing the effectiveness of wildlife identification.

Through constant innovation in cutting-edge AI, IoT, and connectivity technologies, Safari Scout seeks to revolutionize wildlife tourism by turning conventional safaris into immersive, data-driven, and environmentally conscious experiences.

REFERENCES

[1] H. U. Rahman, M. Ahmad, H. Ahmad, and M. A. Habib, "Lorawan: State of the art, challenges, protocols and research issues," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, 2020, pp. 1–6.

[2] M. Boon, R. Mei, and E. Winands, "Applications of polling systems," *Journal of The American Society for Information Science and Technology - JASIS*, vol. 16, 03 2011.

[3] M. Nasar and M. Abu Kausar, "Suitability of influxdb database for iot applications," 08 2019.

[4] X. Liu, T. Yang, and B. Yan, "Research on the architecture of wildlife observation and communication system," in *2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, 2015, pp. 415–418.

[5] Y. Igarashi, R. Nakano, and N. Wakamiya, "A polling-based transmission scheme using a network traffic uniformity metric for industrial iot applications," *Sensors*, vol. 19, 01 2019.

[6] M. Sohan, T. Ram, and V. Ch, *A Review on YOLOv8 and Its Advancements*, 01 2024, pp. 529–545.

[7] A. Ray and M. Kolekar, *Image Segmentation and Classification Using Deep Learning*, 11 2022, pp. 19–36.