ENHANCING AUTONOMOUS VEHICLE SECURITY

24-25J-140

SLIIT
FACULTY OF COMPUTING

OUR TEAM

Mr. Kavinga Abeywardena
Supervisor

Ms. Hansika Mahaadikara
Co-Supervisor

Wickramaarachchi J.C.
IT21369810

Albalushi O.T.M.G
IT21099472

Jayasinghe K.A.C.T
IT21146442

Wanigasekara W.M.I.W
IT21249648

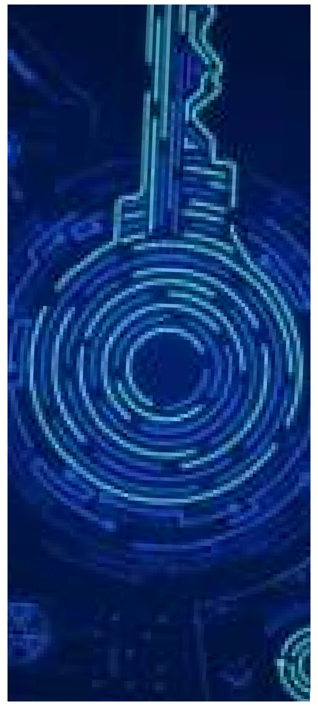
INTRODUCTION

- **Smart Key System:** We are developing a smart key system using an Android app to replace traditional vehicle key fobs, enhancing security and convenience.

- **Lightweight mechanism to mitigate Black-Hole Attack:** Our research includes implementing lightweight ECC for secure Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications, protecting against network attacks.

- **Physical Unclonable Functions (PUFs):** We are utilizing PUFs to create a robust challenge-response mechanism, enhancing authentication and guarding against side-channel attacks.

- **Mitigate GPS Spoofing:** A machine learning-based anomaly detection system is being developed to identify and counter GPS spoofing, ensuring reliable navigation for autonomous vehicles.



OBJECTIVES

To enhance the overall security of autonomous vehicles by developing the following components:

- Developing a Smart Key Vehicle Entry System
- Implement ECC-based authentication for V2V/V2I communications
- Implement a PUF based challenge response mechanism for autonomous vehicles.
- Develop comprehensive anomaly-based GPS spoofing detection framework.

SLIIT FACULTY OF COMPUTING

Research Questions

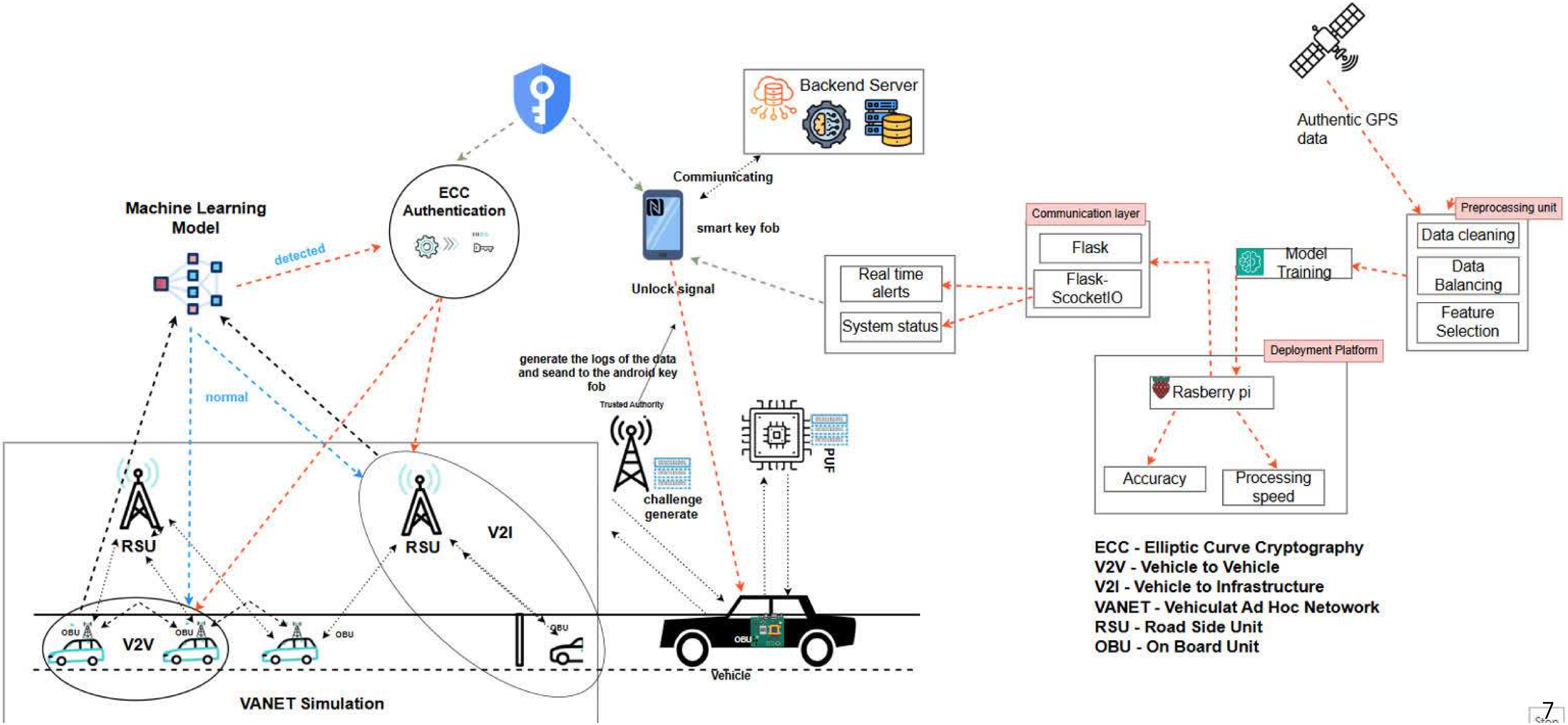
- How to enhance security by introducing a android application instead of traditional key fobs?
- How can lighweight ECC improve V2V and V2I security and efficiency?
- How effective is ECC-based authentication in mitigating black hole attacks compared to PKI?
- How can PUFs provide secure challenge-response mechanisms for autonomous vehicles?
- How can machine learning detect and mitigate GPS spoofing in autonomous vehicles?

Bridging the Gaps

Implementing Panel-Suggested Enhancements

- Expanded the system's compatibility to support all vehicle models, not just autonomous vehicles.
- Addressed feedback from the RP meeting regarding component disconnection by conducting in-depth research and refining the system architecture to ensure seamless integration.
- Implemented an offline unlocking mechanism based on the panel's recommendation.
- Replaced GPS with Bluetooth Low Energy (BLE) to improve accuracy in detecting whether the user is within the vehicle's proximity.
- Successfully integrated all hardware and software components with the mobile application for a more cohesive user experience.

SYSTEM DIAGRAM



ECC - Elliptic Curve Cryptography
V2V - Vehicle to Vehicle
V2I - Vehicle to Infrastructure
VANET - Vehiculat Ad Hoc Netowork
RSU - Road Side Unit
OBU - On Board Unit

IT21369810

Wickramaarachchi J.C.

Cyber Security

SLIIT
FACULTY OF COMPUTING

BACKGROUND & RESEARCH PROBLEM

- Traditional vehicle entry systems with basic RF chip key fobs are vulnerable to attacks like replay, roll jam, and rollback due to limited encryption and power constraints, making them easy targets for attackers.
- In the current automotive world most of the vehicles such as Honda Fit 2022, Honda Civic 2022, Honda VE-1 2022, Honda Breeze 2022 are vulnerable to Rolling-PWN attack.
- Current Android key fobs are often designed specifically for each manufacturer, limiting interoperability and flexibility across different vehicle brands.

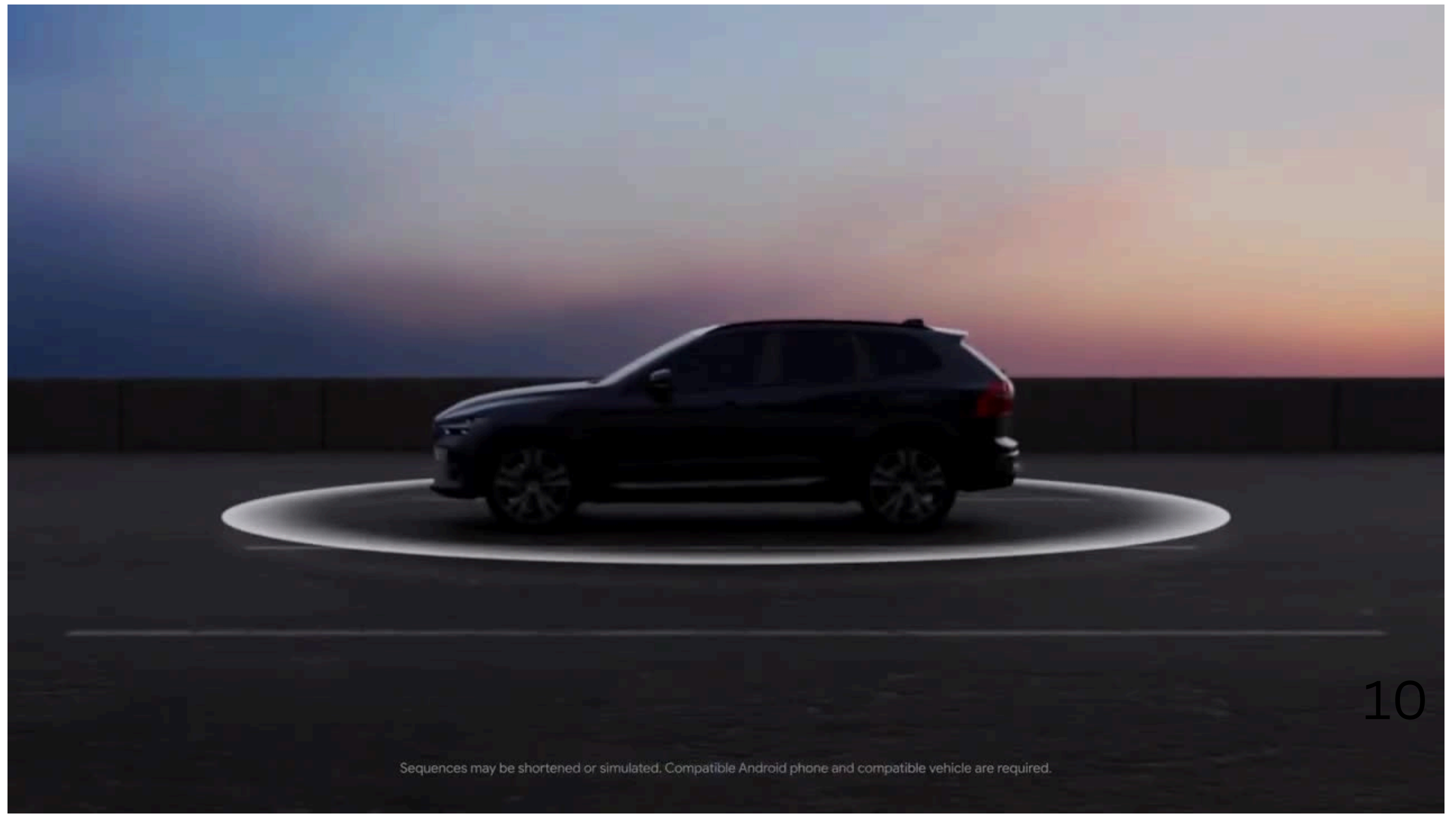
OBJECTIVES

Main Objectives

- To develop an Android application that replaces traditional key fobs by leveraging smartphones' computational power to generate longer and more secure encryption keys, encrypt signals to prevent man-in-the-middle attacks, and incorporate user authentication with Role-Based Access Control (RBAC) and time-based permissions for granting temporary access to authorized individuals.

Sub Objectives

- Design and Development of the Android Application
- Implement Enhanced Encryption Method
- Incorporate User Authentication and Access Control
- Establish Secure Communication Protocols



EXISTING RESEARCH

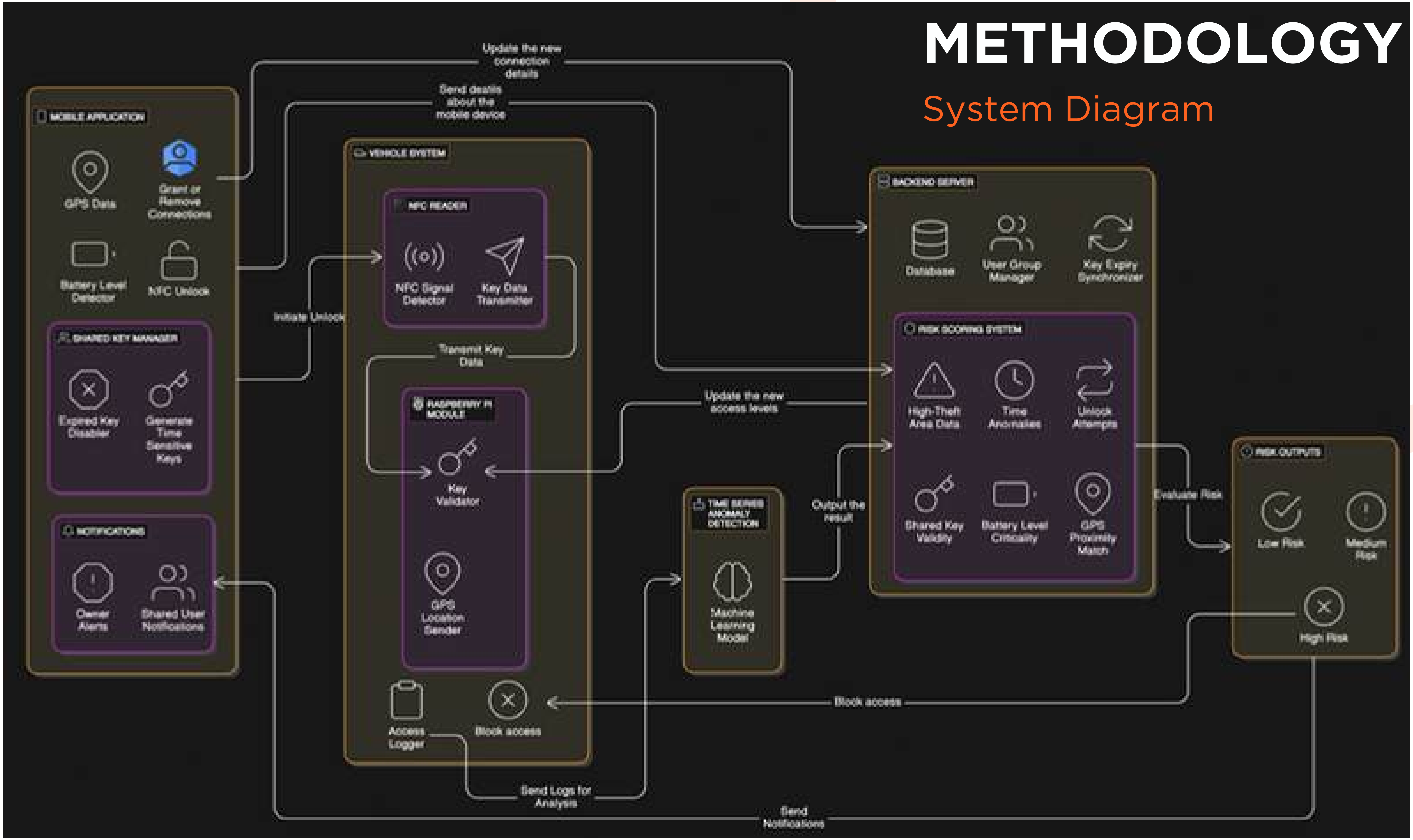
Title	Authors	Published Year
[1] An Android-Based Multifactor Authentication for Securing Passive Keyless Access System	• Aditya D Naik • Ritvik Vibhu • Udbhav P Saboji • Vanisha R.M • Nagasundari S • Prasad B Honnavalli	2022
[2] Enhancing Connected Vehicle Security: Innovations in Two-Factor Authentication	• Huseyin Karacali • Efecan Cebel • Nevzat Donum	2024
[3] PRESTvO: PRivacy Enabled Smartphone Based Access to Vehicle On-Board Units	• B. Groza • T. Andreica • A. Berdich • P. S. Murvay • E. H. Gurban	2024

RESEARCH GAP

Research / Review Paper / Article	Mobile Application	Access control for the USERS	Communication using NFC	Key Fob Anomaly detection & Risk Calculation	VIN Number Theft Protection
Research [1]	✓	✓	✗	✗	✗
Research [2]	✓	✗	✗	✗	✗
Research [3]	✓	✓	✗	✗	✗
Proposed Solution	✓	✓	✗	✗	✗

METHODOLOGY

System Diagram



SECURITY MEASURES

list of security measures implemented in NexLock

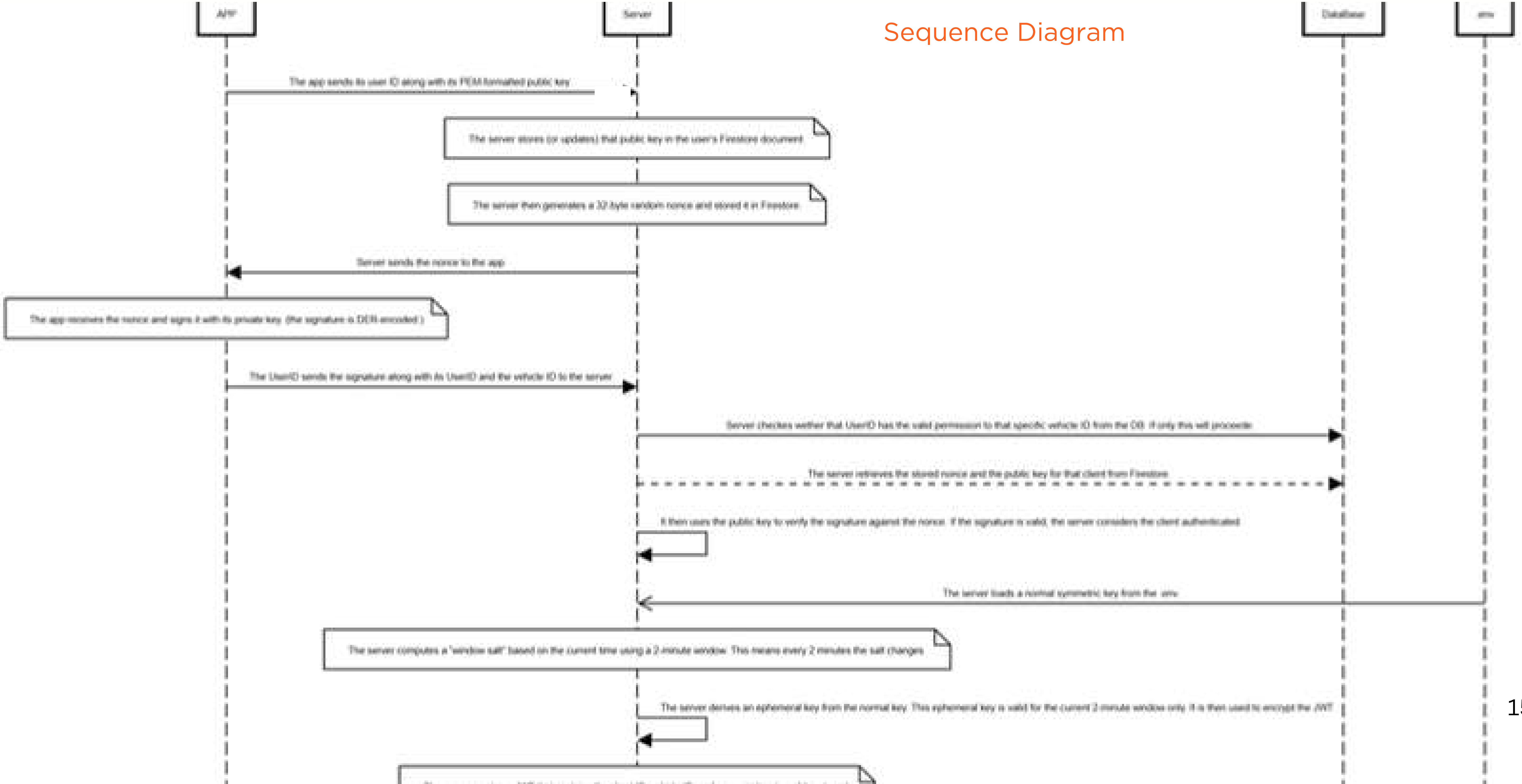
- Ephemeral Key Derivation (HKDF)
- One-Time key usage
- JWT with 1-Hour Expiry
- Role-Based Access Control
- AES-256-GCM Encryption
- Mutual BLE Challenge-Response
- Encrypted NFC Transmissions
- Multi-Layed authentication
- Stolen Vehicle Check
- Biometric Authentication
- Unlock Attempt Logging

NEX LOCK

Where Innovation meets Automation

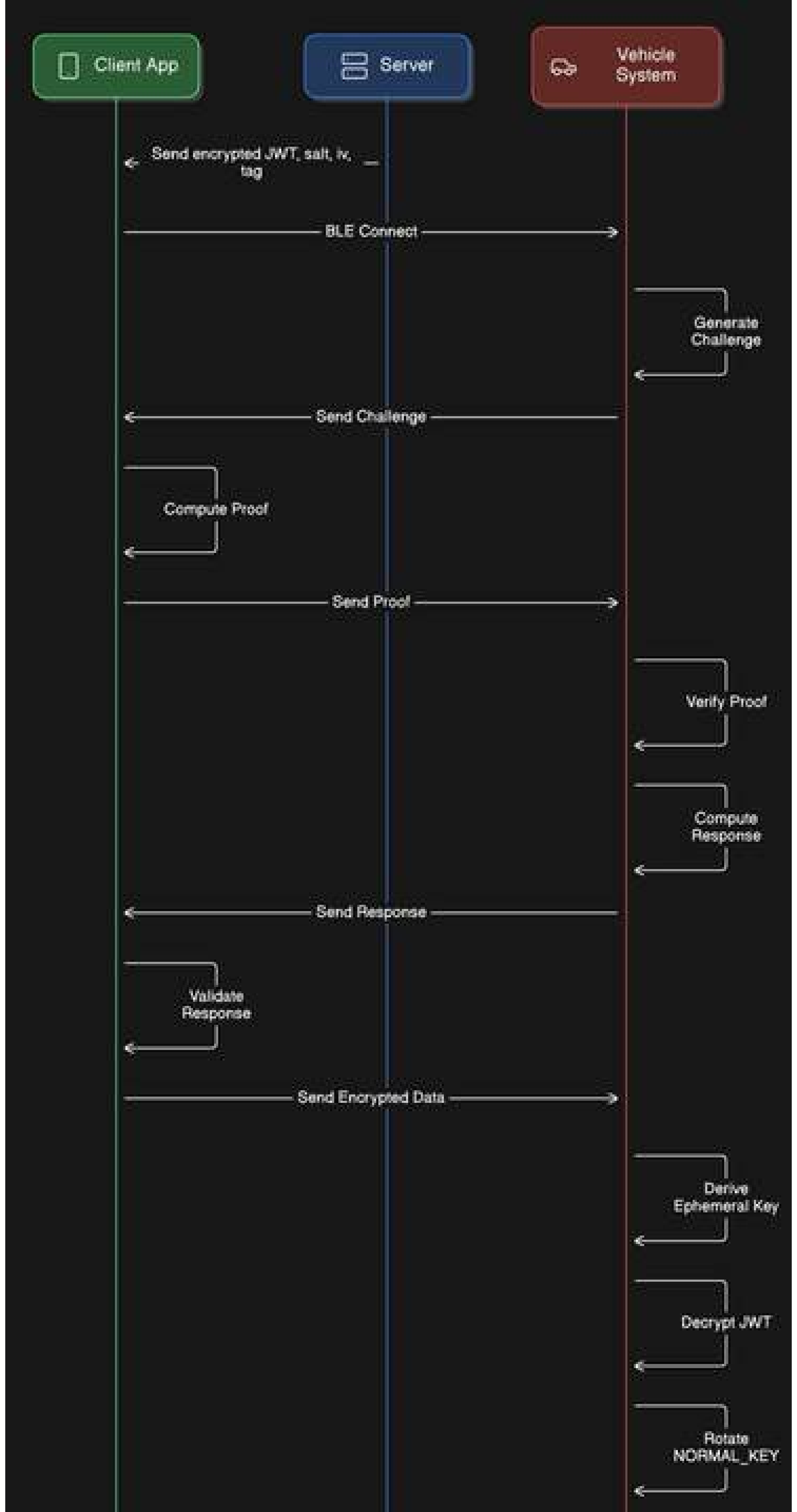
Key Management & Authentication

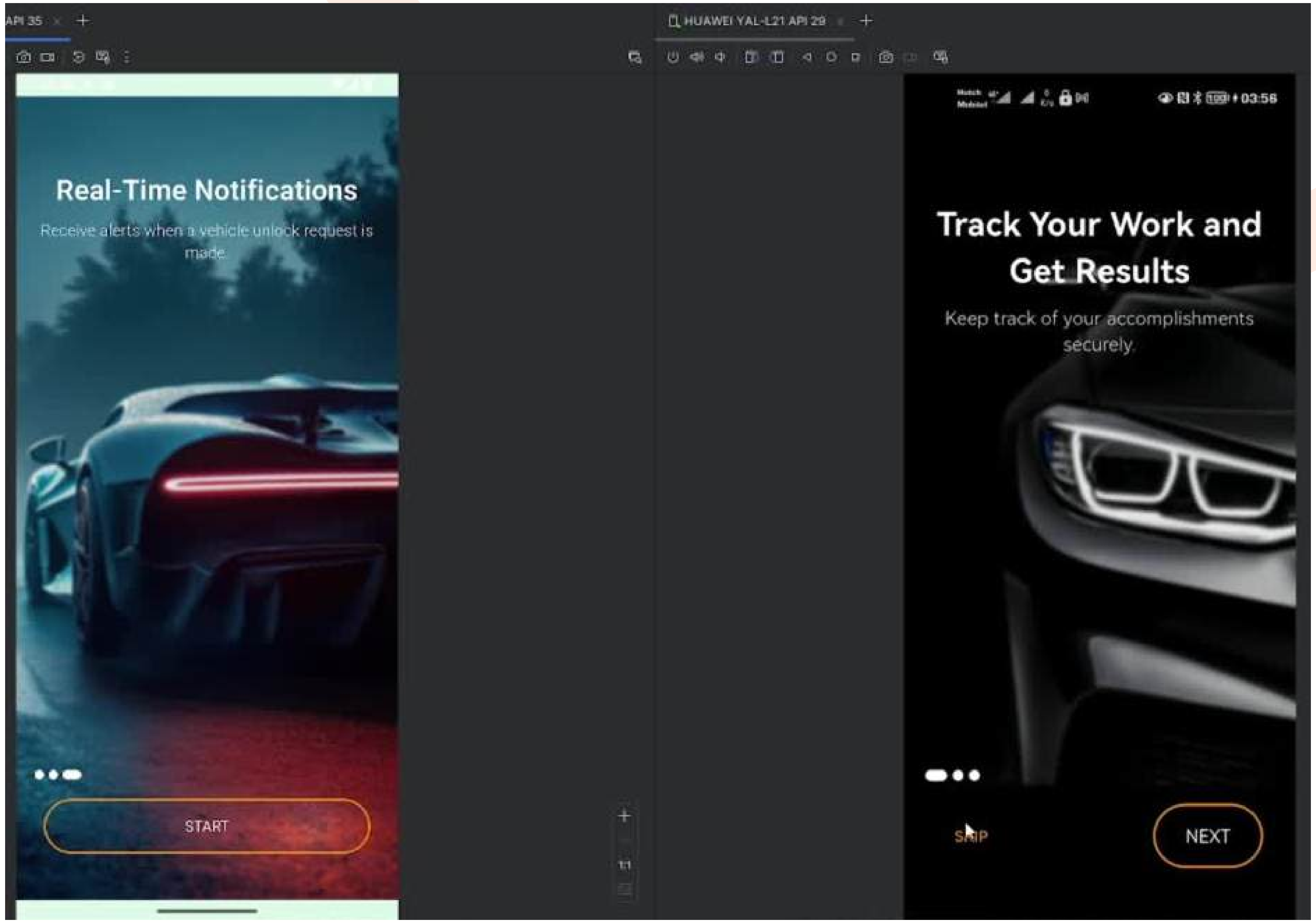
Sequence Diagram



Key Management & Authentication

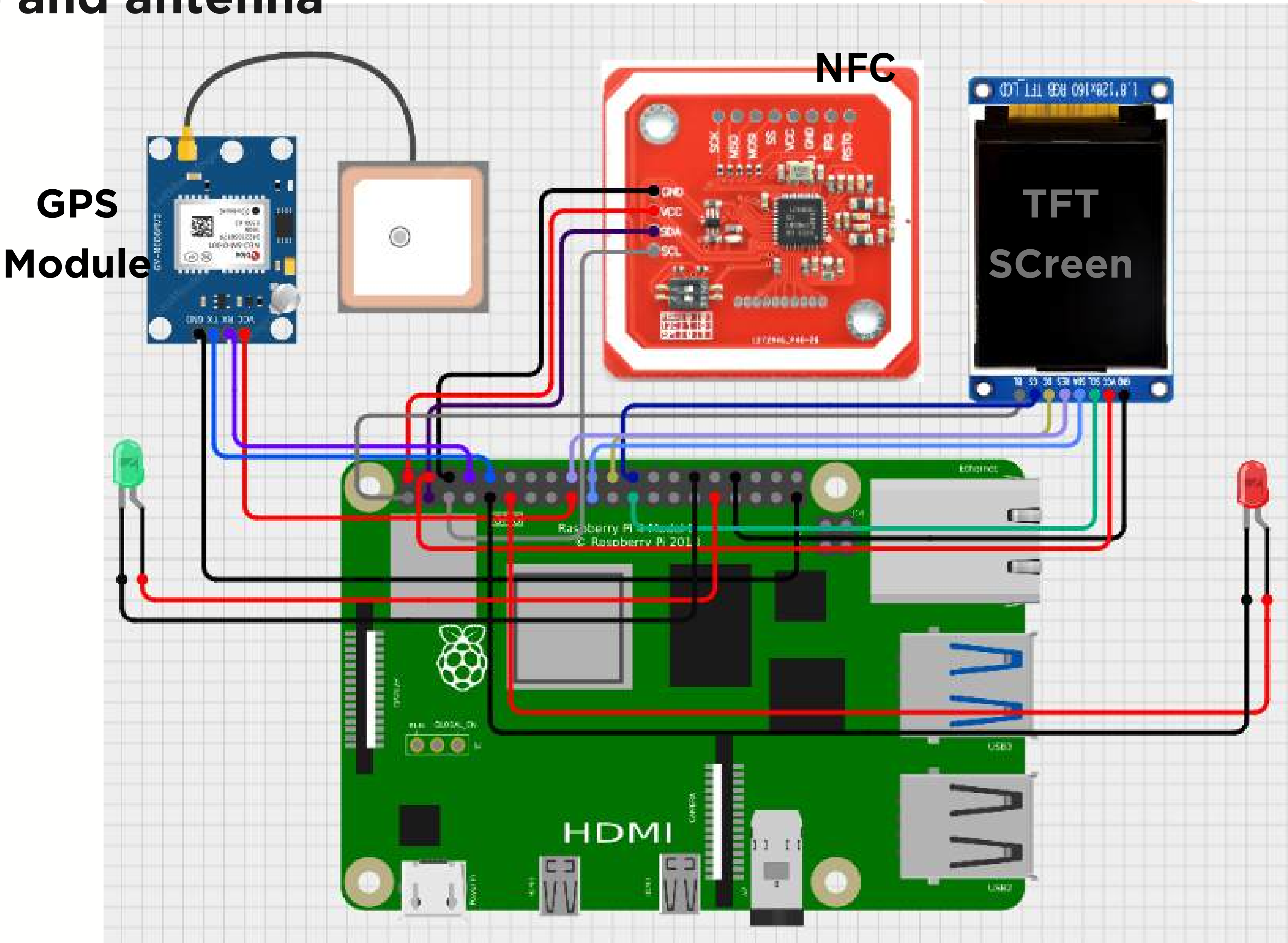
Sequence Diagram





Used Hardware Components

- Raspberry Pi 4 model B 4GB
- PN532 NFC RFID Module V3 Kits Reader Writer
- DS3231 Precision RTC Real time Clock Memory Module
- JBtek U-BLOX NEO-6M GPS Module and antenna



REQUIREMENTS

Functional Requirements:

- Authenticate users using passwords, biometrics, or multi-factor authentication (MFA).
- Implement Role-Based Access Control (RBAC) for temporary access permissions.
- Generate secure encryption keys and encrypt communications to prevent interception.
- Allow users to unlock, lock, and start the vehicle through the app.

Non- Functional Requirements:

- Security
- Performance
- Reliability
- Usability
- Scalability

Technical Requirements:

- Develop for Android, compatible with various smartphone models.
- Operate both online and offline, using secure network protocols.
- Use Kotlin and encryption libraries for development.

TOOLS & TECHNOLOGIES

Technologies

- Flutter(Android App Development) -
- Firebase
- NFC (BLE)
- Python
- Supabase
- Raspberry Pi

Algorithms & Architectures

- AES (Advanced Encryption Standard)
- Role-Based Access Control (RBAC)
- Multi-Factor Authentication (MFA)
- Elliptic Curve Cryptography (ECC)

Techniques

- End-to-End Encryption
- Time Stamped Ephemeral keys

Current Progress

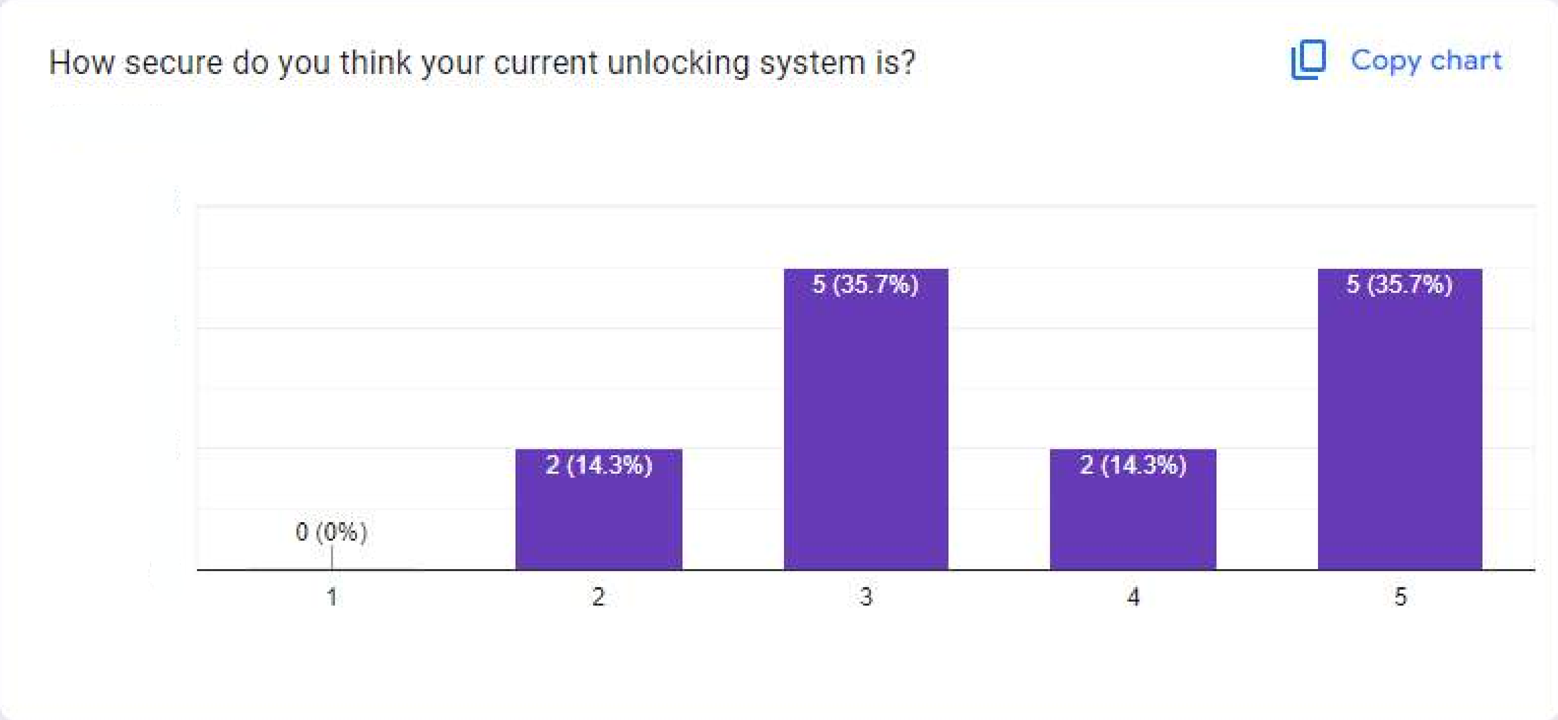
- Train ML Model to identify time series access anomalies.
- Use a dynamic risk matrix to calculate risk values.
- Developed the mobile application.
- Integrated the backend server for communication with application and the hardware.
- Setup the Raspberry pi and other modules and connected with the application.
- Used a strong cryptographic key management in the communication.
- Used Roles based user groups for vehicle access management.
- Integrated with the PUF solution.

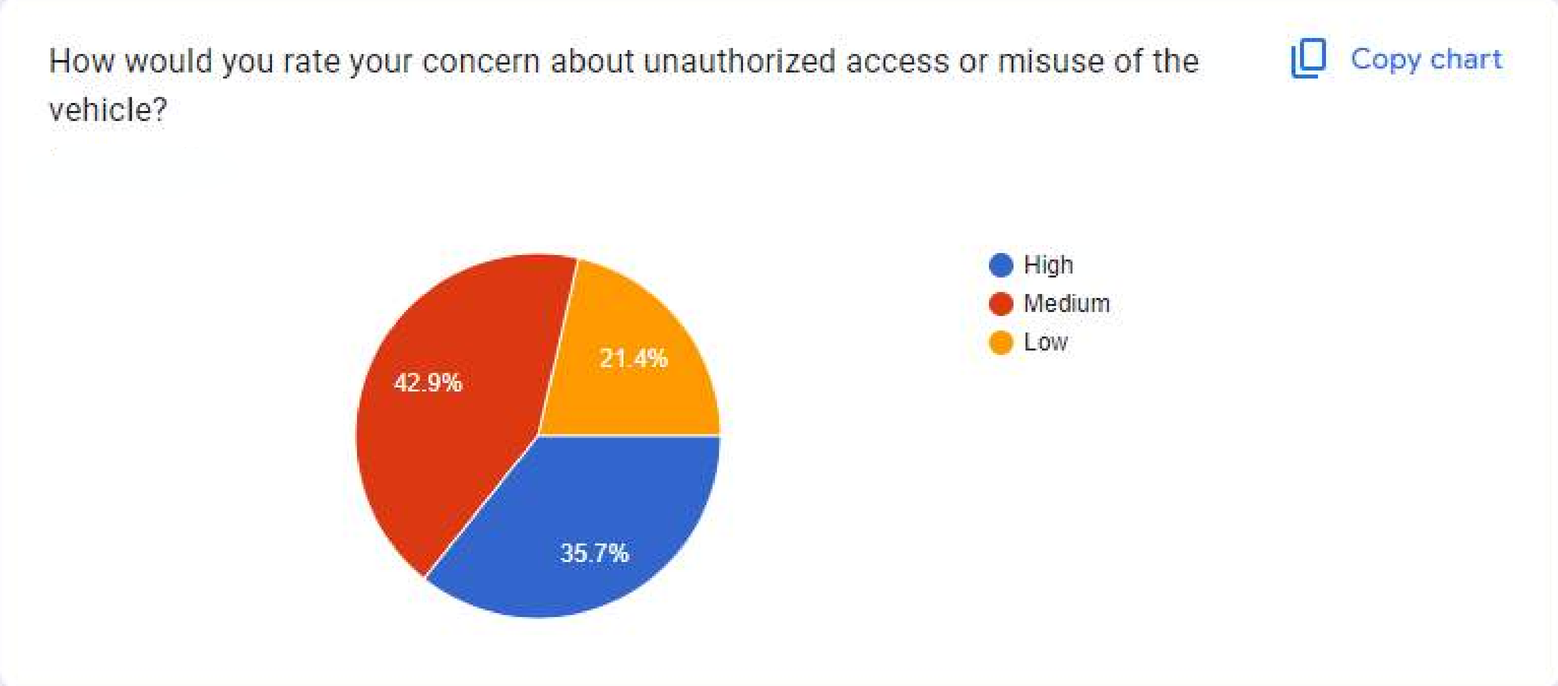
Future Step

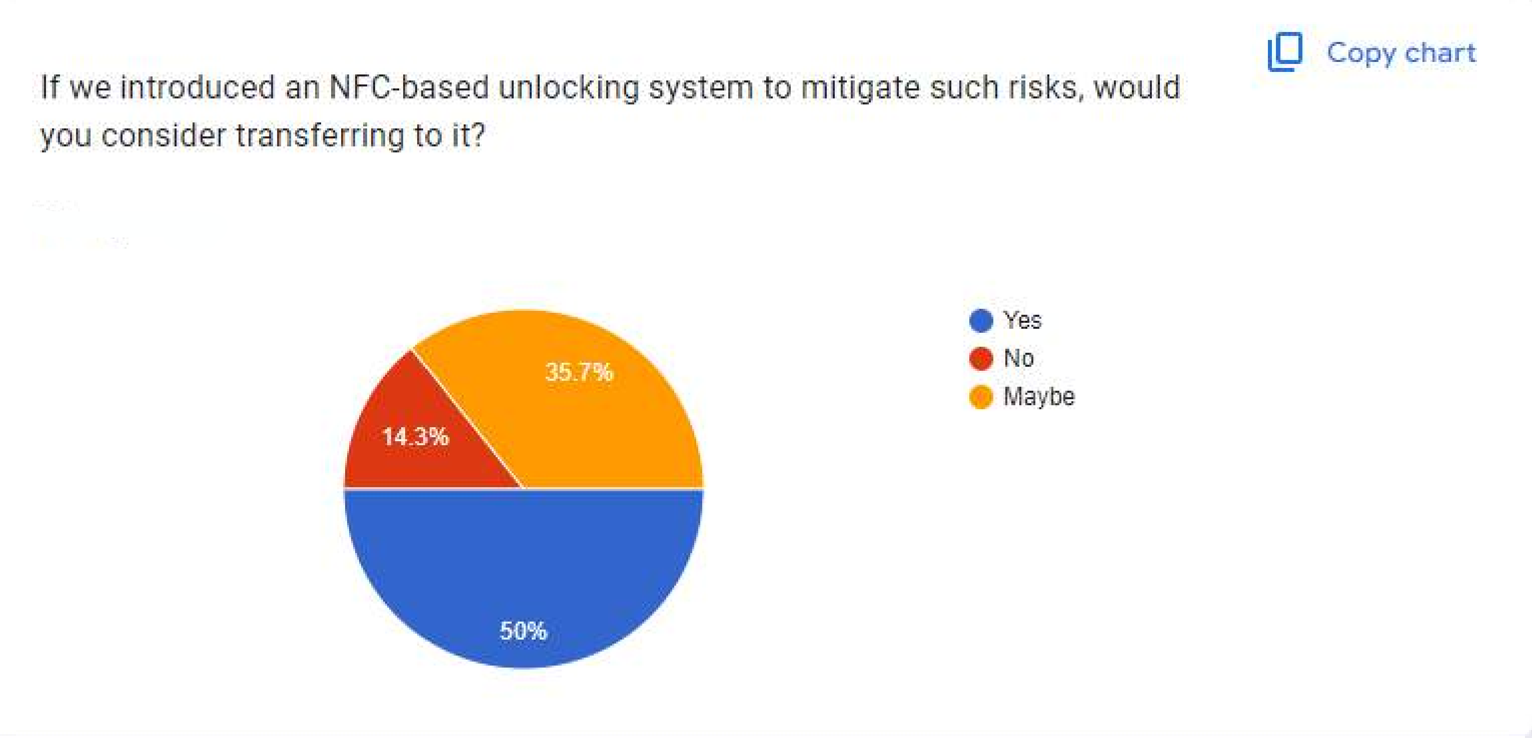
- Finetune the mobile application and integrate with other components.
- Integrate and adjust the previously trained ML model with the backend server to detect time anomalies.

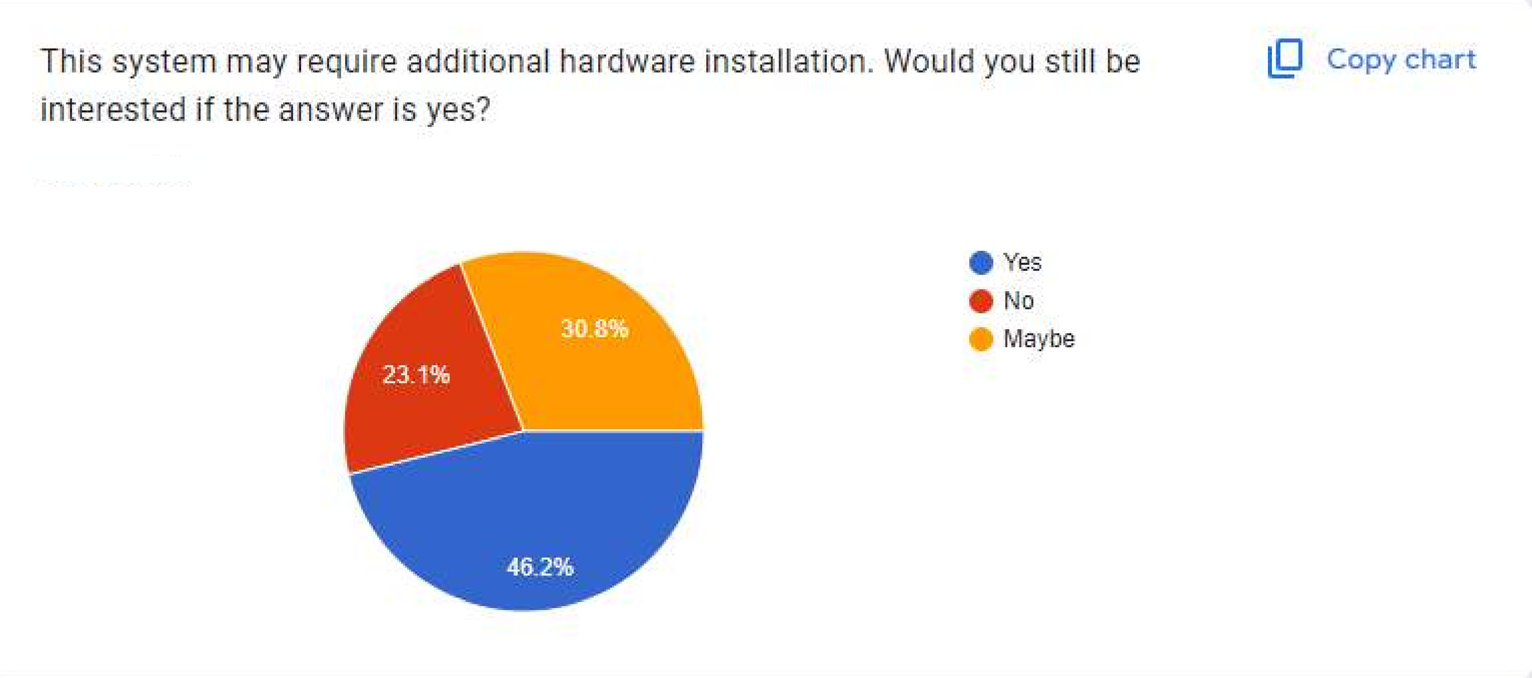
Current **Progress**

Survey to identify Sri Lankan user perspective and security knowledge



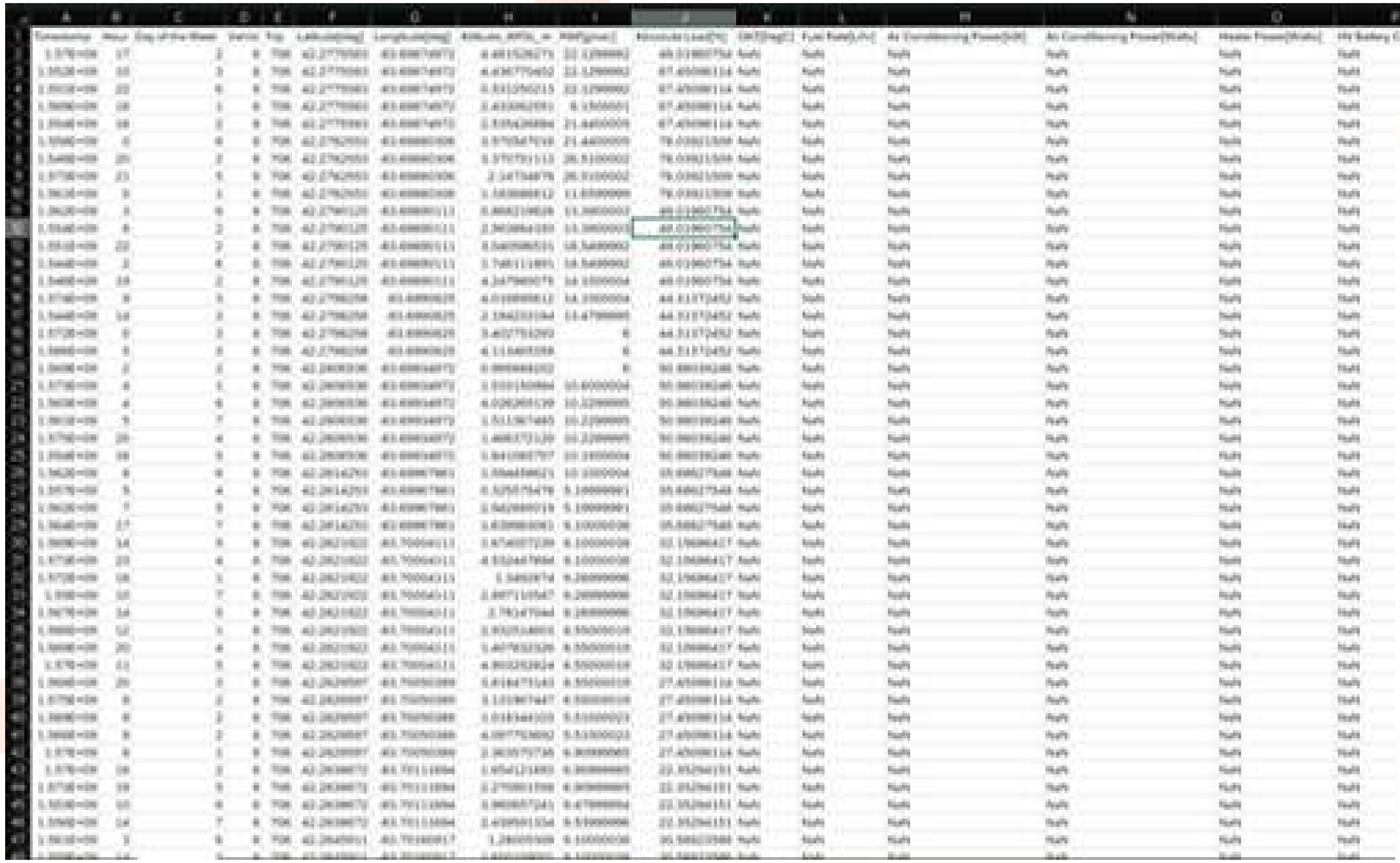


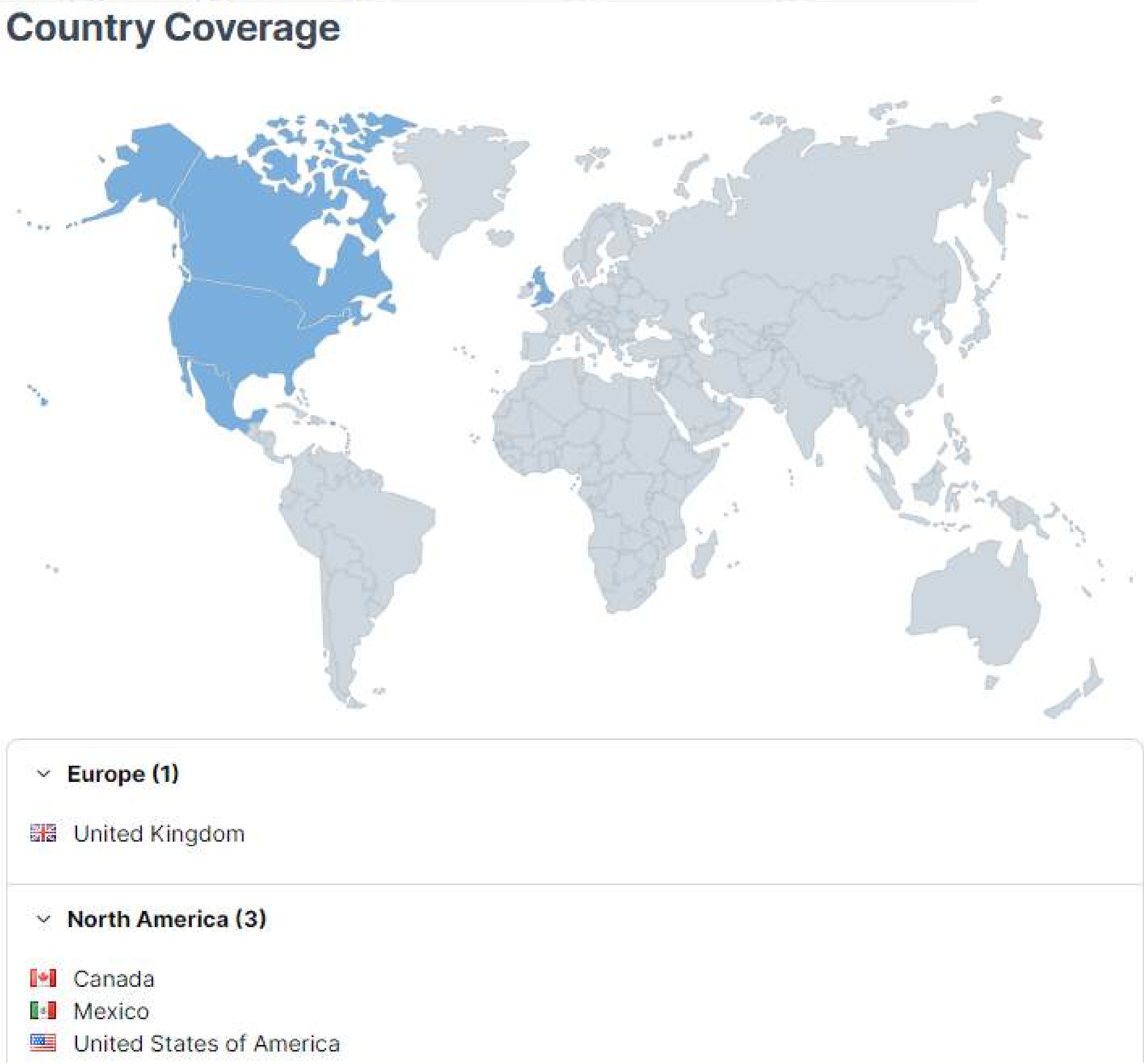




Current Progress

ML and -Data set





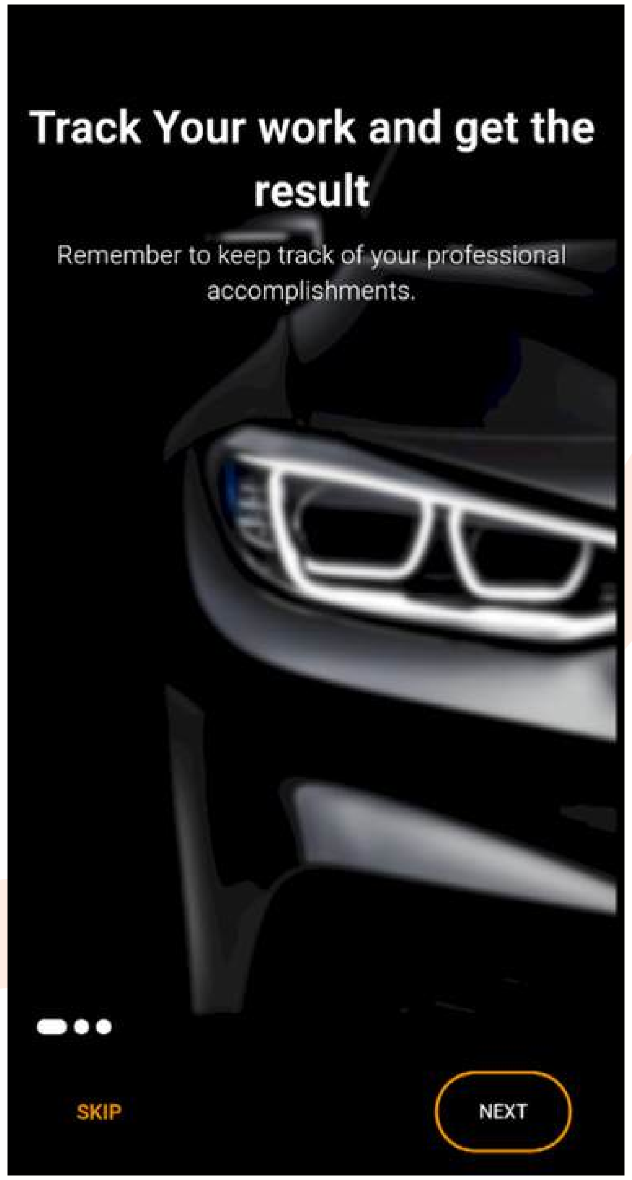
Current Progress

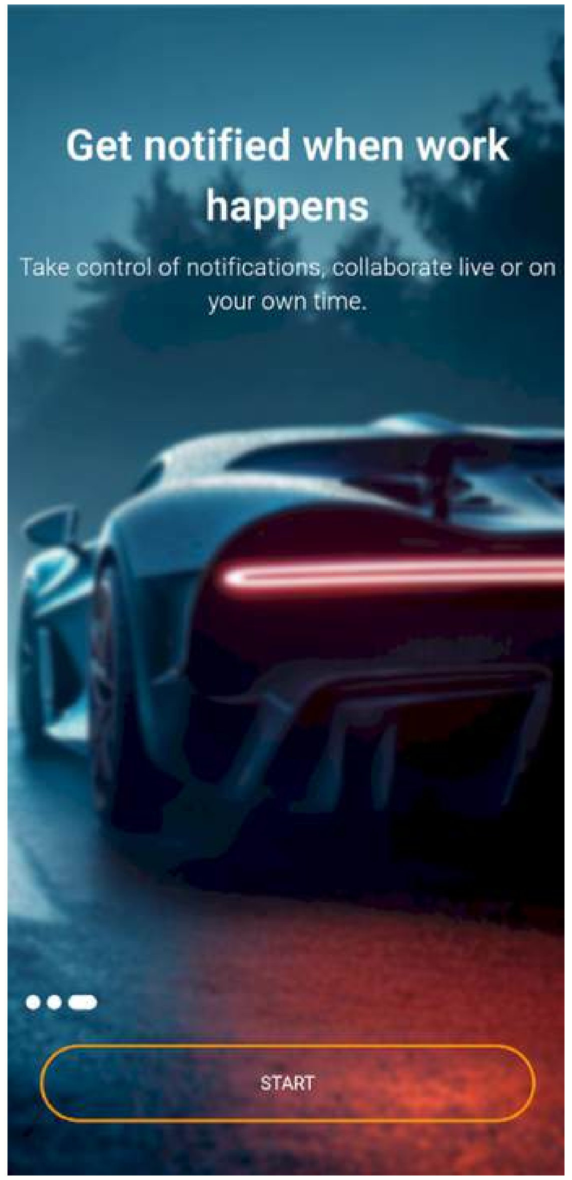
Accuracy - Autoencoders

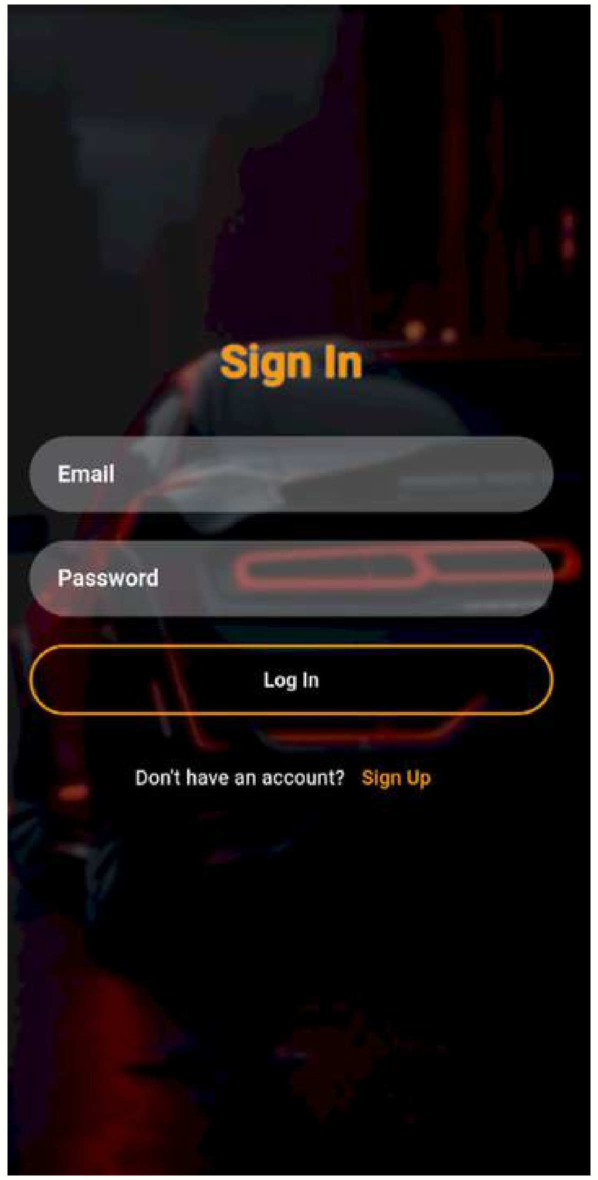

```
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
Normal Data Metrics: Accuracy=0.9436619718309859, Precision=0.0, Recall=0.0, F1=0.0
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: UndefinedMetricWarning: Recall is ill-defined and being
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

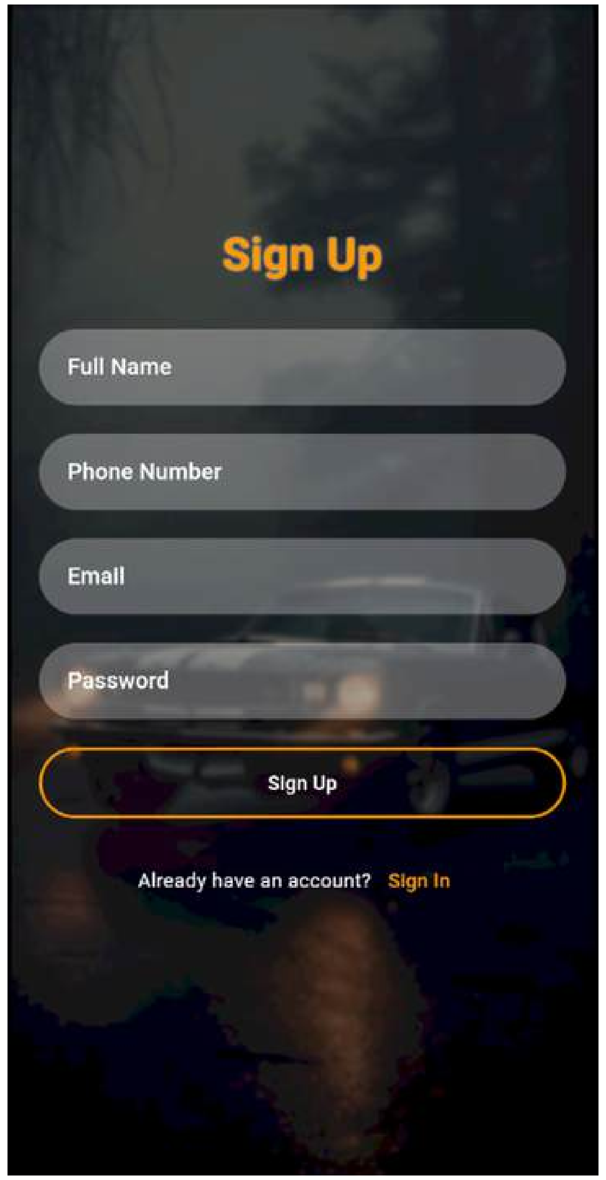
Current Progress

Designed UI



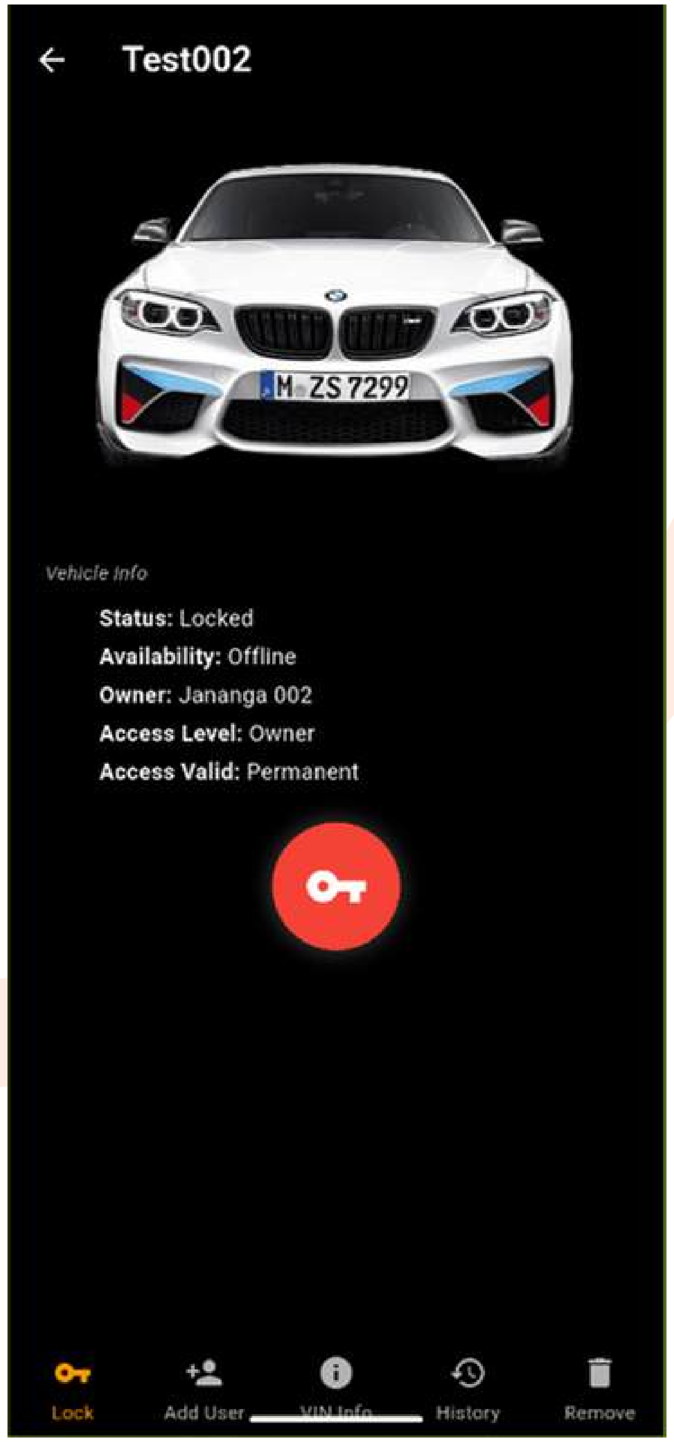


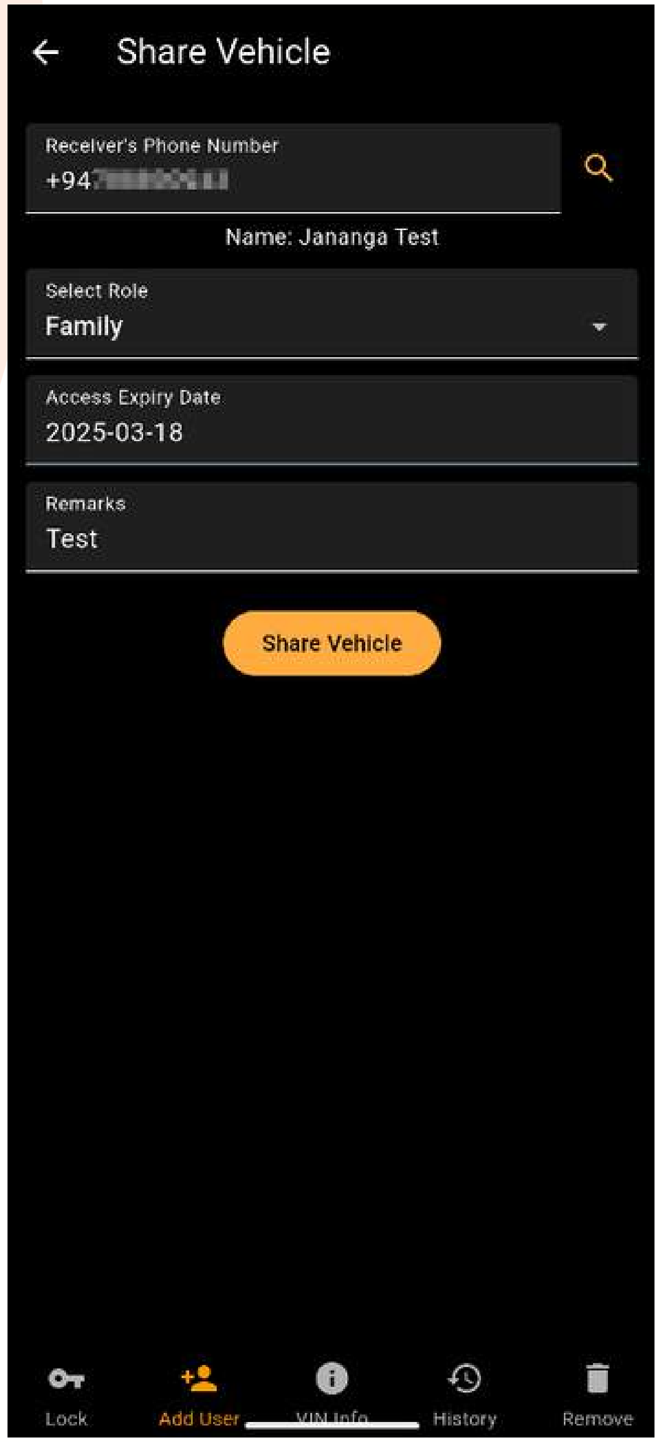


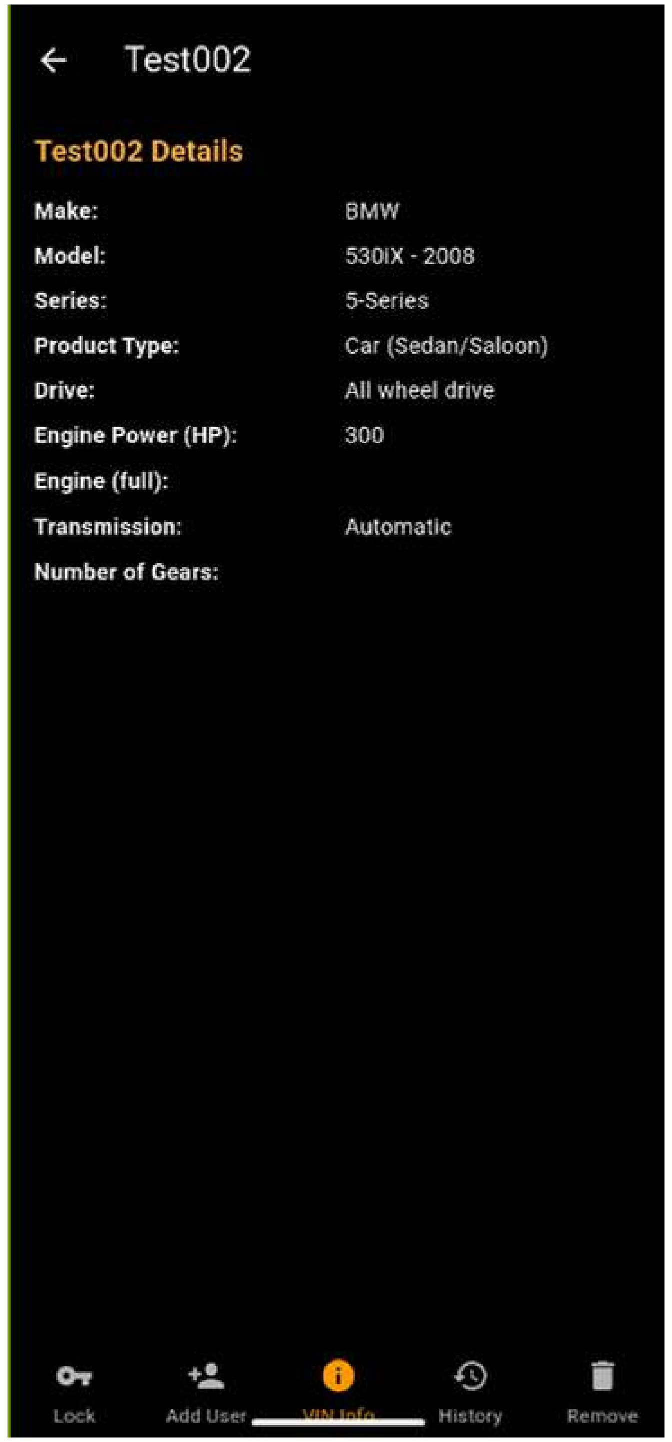


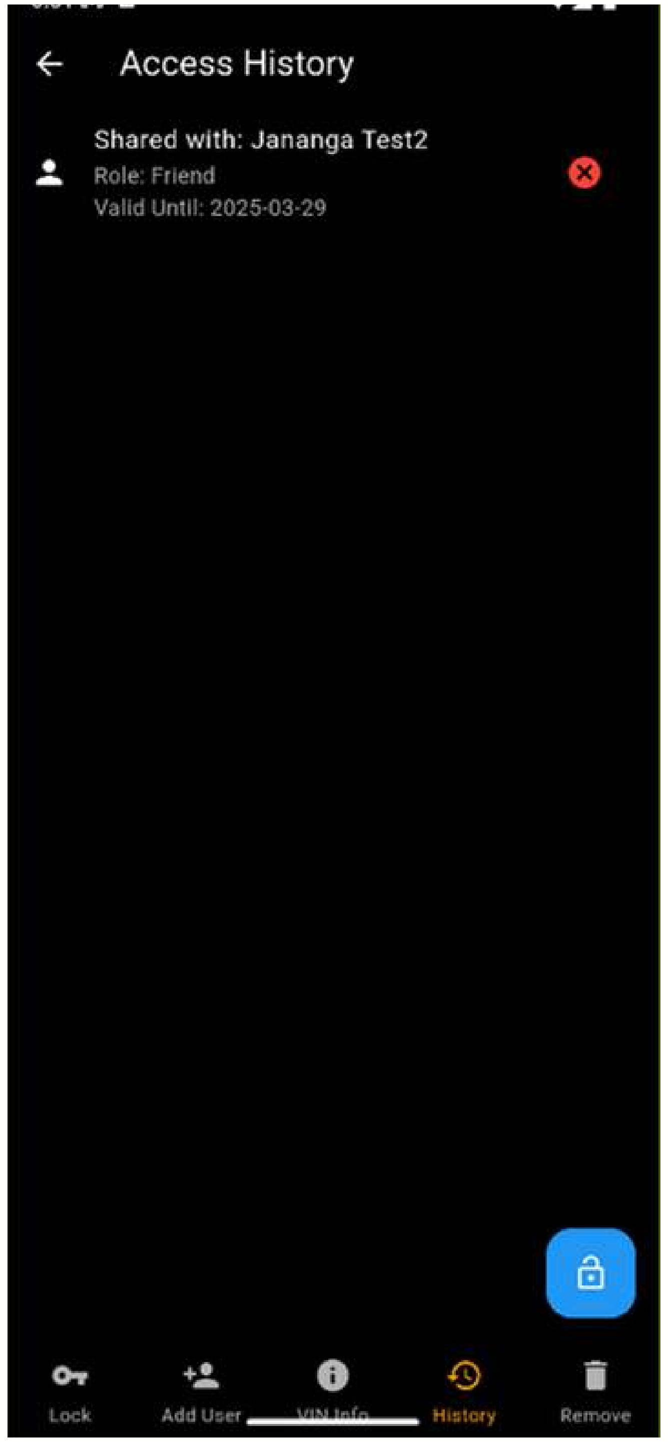
Current Progress

Designed UI



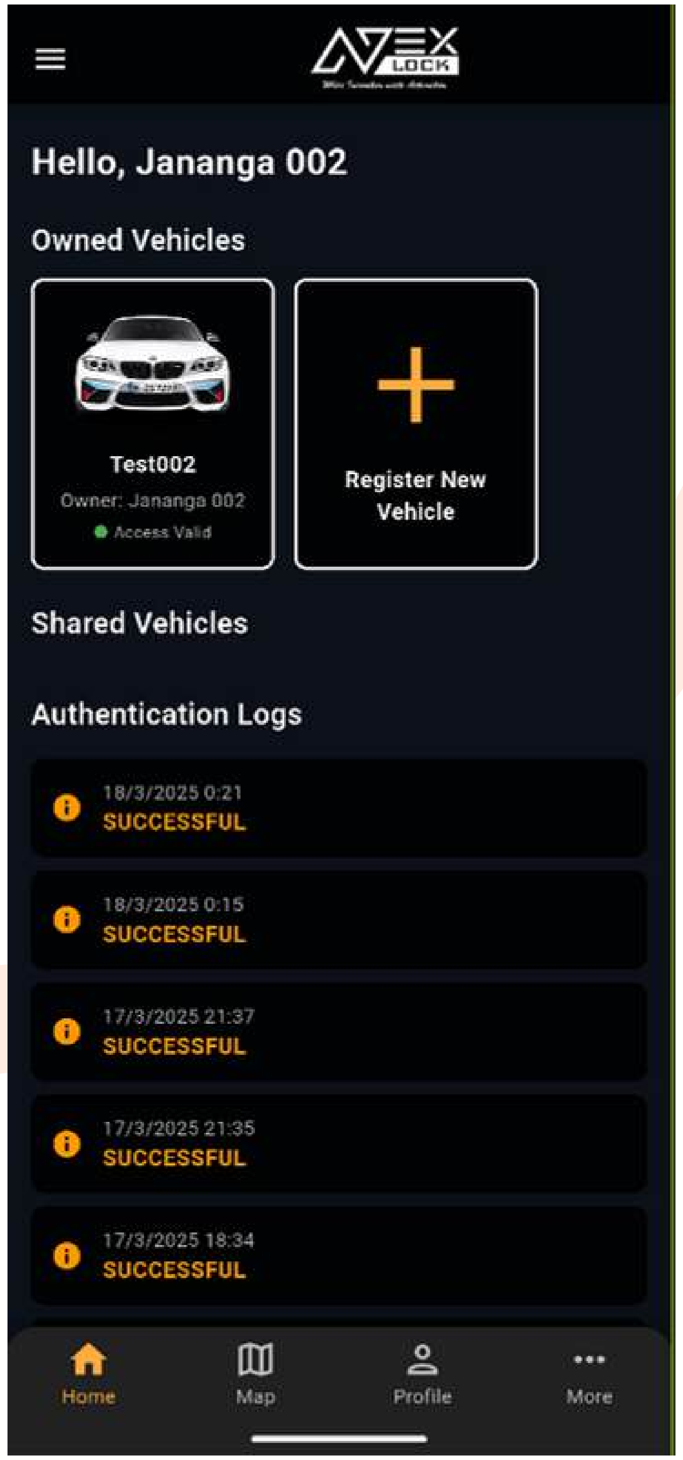


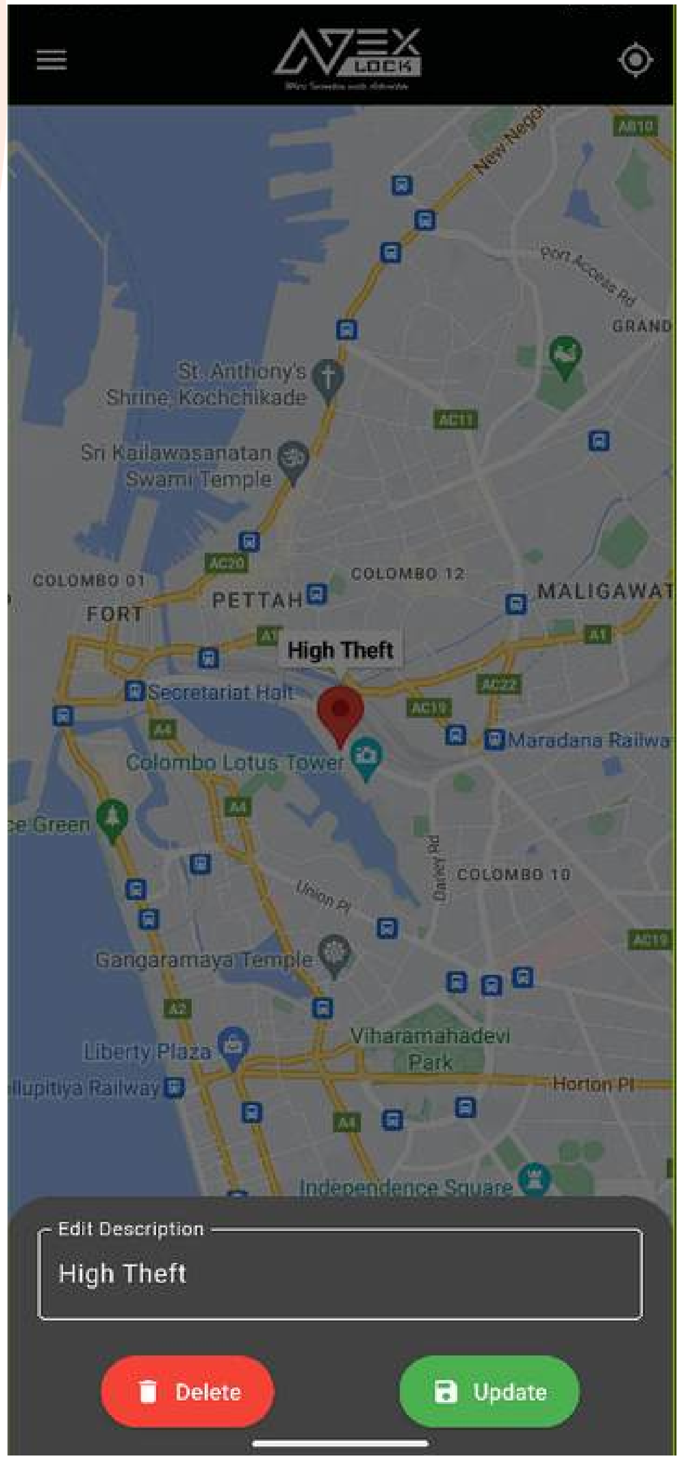


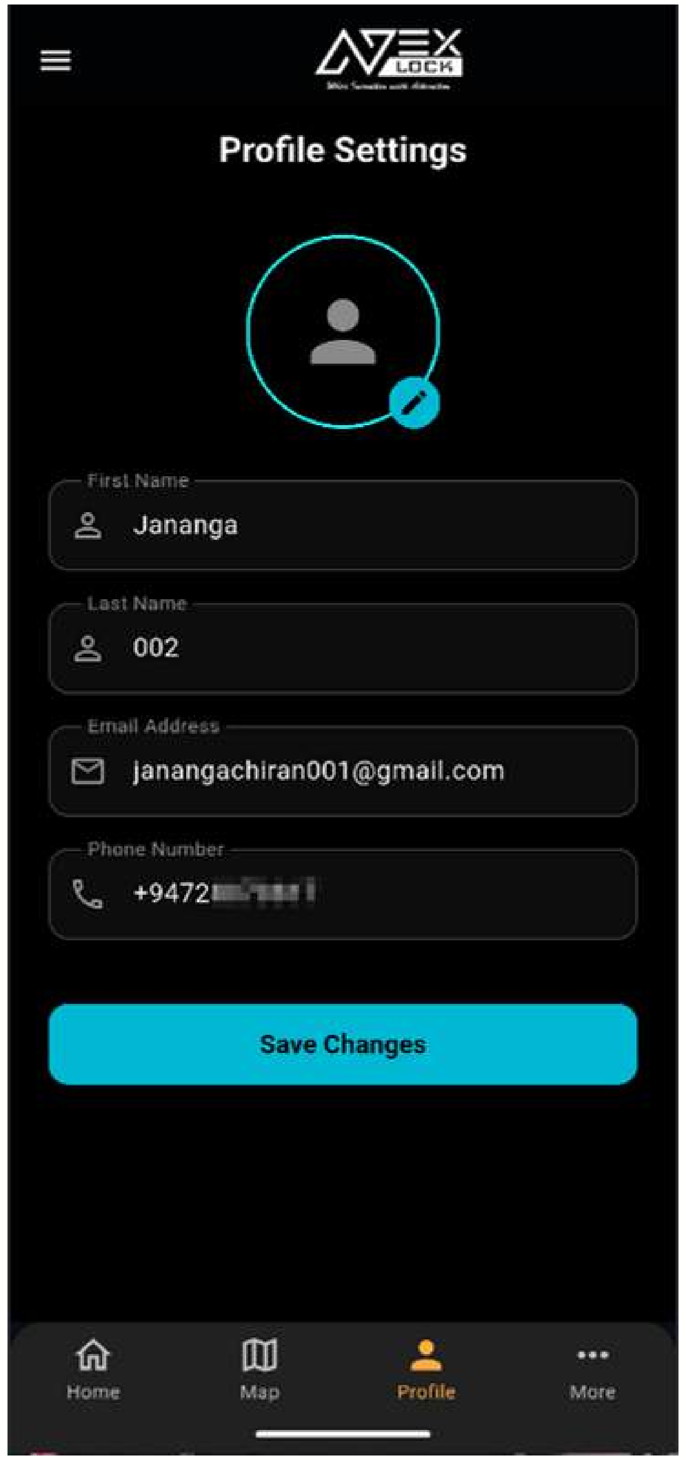


Current Progress

Designed UI







Current Progress

VIN Number Decoder

VINCARIO
Global VIN decoder

VIN DECODER API PRICING LOGIN

API Response

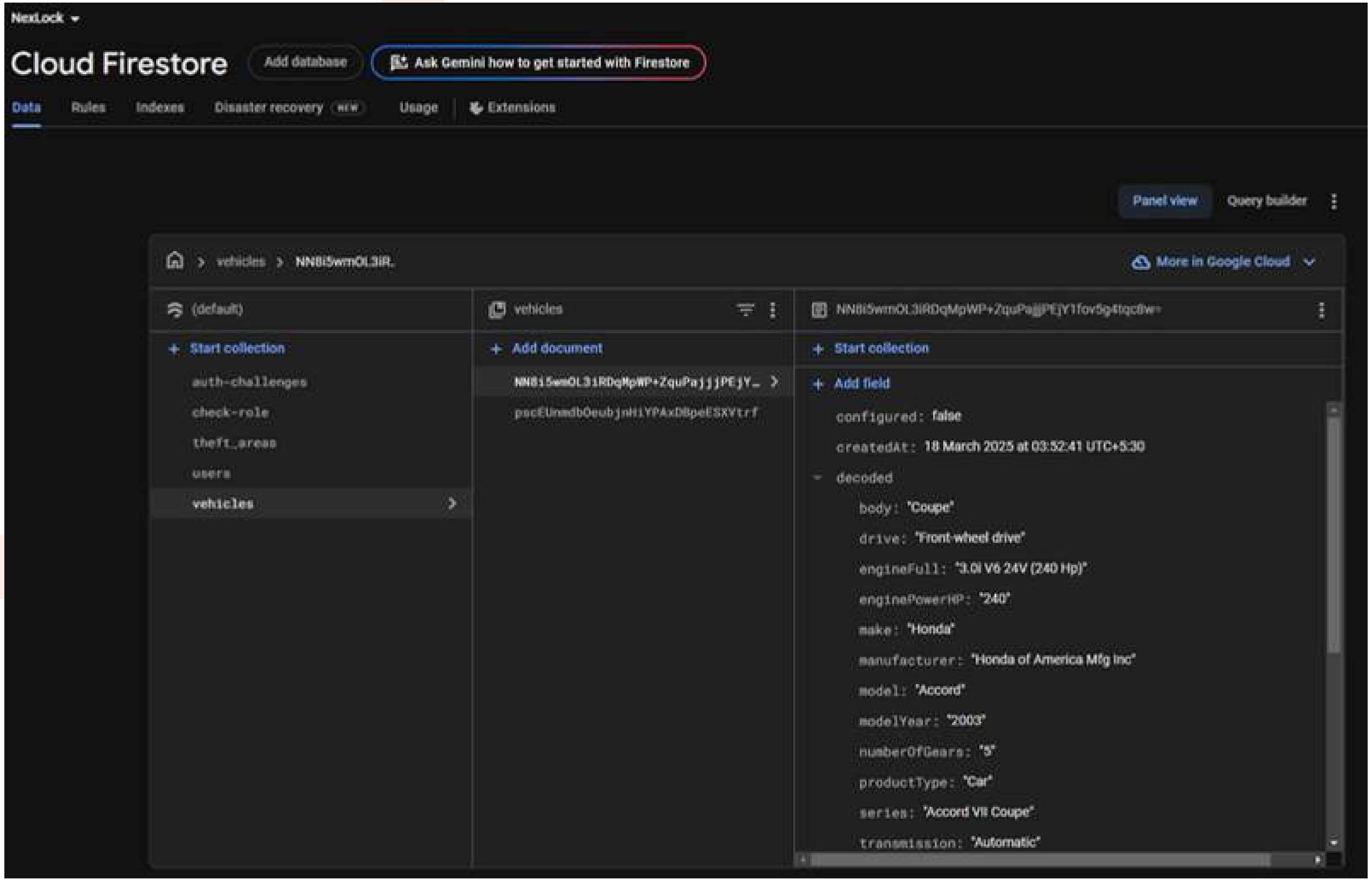
- Price: 0
- Price Currency: USD
- Balance:
  - API Decode: 18
  - API Stolen Check: 3
  - API Vehicle Market Value: 5
  - API OEM VIN Lookup: 0
- Decode:
  - VIN: 1HGCM82633A123456
  - Vehicle ID: 1951
  - Make: Honda
  - Make ID: 43
  - Model: Accord
  - Model ID: 17382
  - Model Year: 2003
  - Product Type: Car
  - Product Type ID: 4
  - Body: Coupe
  - Body ID: 4
  - Trim: EX-V6
  - Series: Accord VII Coupe
  - Drive: Front-wheel drive
  - Drive ID: 1
  - Engine Displacement (ccm): 2999
  - Engine Power (kW): 179
  - Engine Power (HP): 240
  - Fuel Type - Primary: Gasoline
  - Fuel Type - Primary ID: 8
  - Engine (full): 3.0i V6 24V (240 Hp)
  - Engine Code: J30A4
  - Transmission: Automatic
  - Transmission ID: 2
  - Number of Gears: 5
  - Manufacturer: Honda of America Mfg Inc

Stolen Checker

- Stolen:
  - Code: cz
    Status: not-stolen
  - Code: hu
    Status: not-stolen
  - Code: ro
    Status: not-stolen
  - Code: si
    Status: not-stolen
  - Code: vincario
    Status: not-stolen

Current Progress

Firebase FireStore Database



CHALLENGES

Challenges

- One Raspberry Pi was damaged due to an unstable power supply.
- Traditional NFC readers can only retrieve UIDs and cannot directly access NFC data.
- Users should be able to unlock their vehicle even when it is offline.
- Faced challenges with deprecated libraries and difficulties in app development.
- Difficulty connecting Raspberry Pi with other modules.
- Inconsistent location data affecting geofencing features.

References

[1] A. D. Naik, R. Vibhu, U. P. Saboji, V. R. M, N. S and P. B. Honnavalli, "An Android-Based Multifactor Authentication for Securing Passive Keyless Access System," 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai, India, 2022, pp. 1-8, doi: 10.1109/I2CT54291.2022.9824254.

[2] H. Karacali, E. Cebel and N.Donum, "Enhancing Connected Vehicle Security: Innovations in Two-Factor Authentication," International Conference on Technology (IConTech), May 02-05, 2024, Alanya/Turkey, pp. 108-121

[3] B. Groza, T. Andreica, A. Berdich, P. -S. Murvay and E. H. Gurban, "PRESTvO: PRivacy Enabled Smartphone Based Access to Vehicle On-Board Units," in IEEE Access, vol. 8, pp. 119105-119122, 2020, doi: 10.1109/ACCESS.2020.3003574.

[4] S.Hamdare, O.Kaiwartya, M. Aljaidi, M. Jugran, Y. Cao, S. Kumar, M. Mahmud, D. Brown and J. Lloret “Cybersecurity Risk Analysis of Electric Vehicles Charging Stations”. Sensors 2023, 23, 6716. https://doi.org/10.3390/s23156716

IT21099472

Al balushi O.T.M.G.

Cyber Security

SLIIT
FACULTY OF COMPUTING

BACKGROUND & RESEARCH PROBLEM

Current Authentication Mechanisms: Existing V2V and V2I communication systems primarily use traditional cryptographic methods,

Black Hole Attacks: V2V and V2I communications are vulnerable to black hole attacks, where malicious nodes drop packets, disrupting network reliability and safety.

Advantages of ECC: Elliptic Curve Cryptography (ECC) offers stronger security with smaller key sizes, making it suitable for resource-constrained environments like vehicular networks.

OBJECTIVES

Completed Objectives

Design and implement a lightweight authentication mechanism using ECC for V2V communication

Mitigate blackhole attacks in vehicular networks through secure authentication

Evaluate performance metrics including authentication delay, throughput, jitter, and packet loss

Quantify the impact of blackhole attacks on network performance

Ongoing Objectives

Adding trust level security among vehicle node and blacklist attacking vehicles.

EXISTING RESEARCH

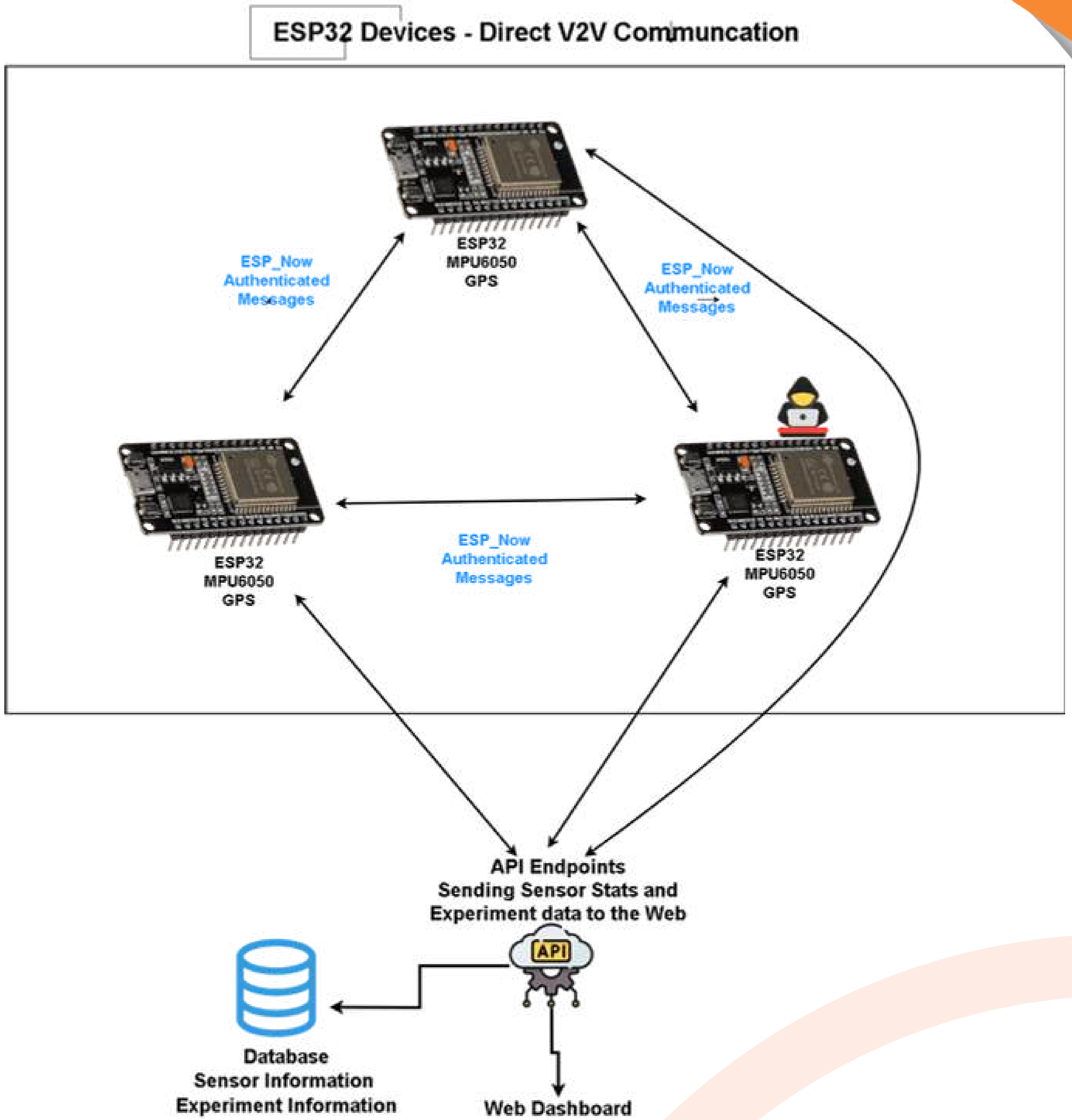
Title	Published Year
[1] An ECC-Based Conditional Privacy-Preserving Authentication Scheme for V2V Communication in VANETs.	2022
[2] An Efficient Dynamic Solution for the Detection and Prevention of Black Hole Attack in VANETs	2022
[3] Cyber Security Challenges and Solutions for V2X Communications	2019

RESEARCH GAP

Research / Review Paper / Article	Lightweight ECC based Authentication	Blackhole Attack Mitigation	Trust based Mechanism	Scalable Solution for V2V and V2I	ML-Based Detection
Research [1]	✓	✗	✓	✗	✗
Research [2]	✗	✓	✗	✗	✗
Research [3]	✗	✓	✗	✗	✗
Proposed Solution	✓	✓	✓	✓	✓

METHODOLOGY

SYSTEM DIAGRAM



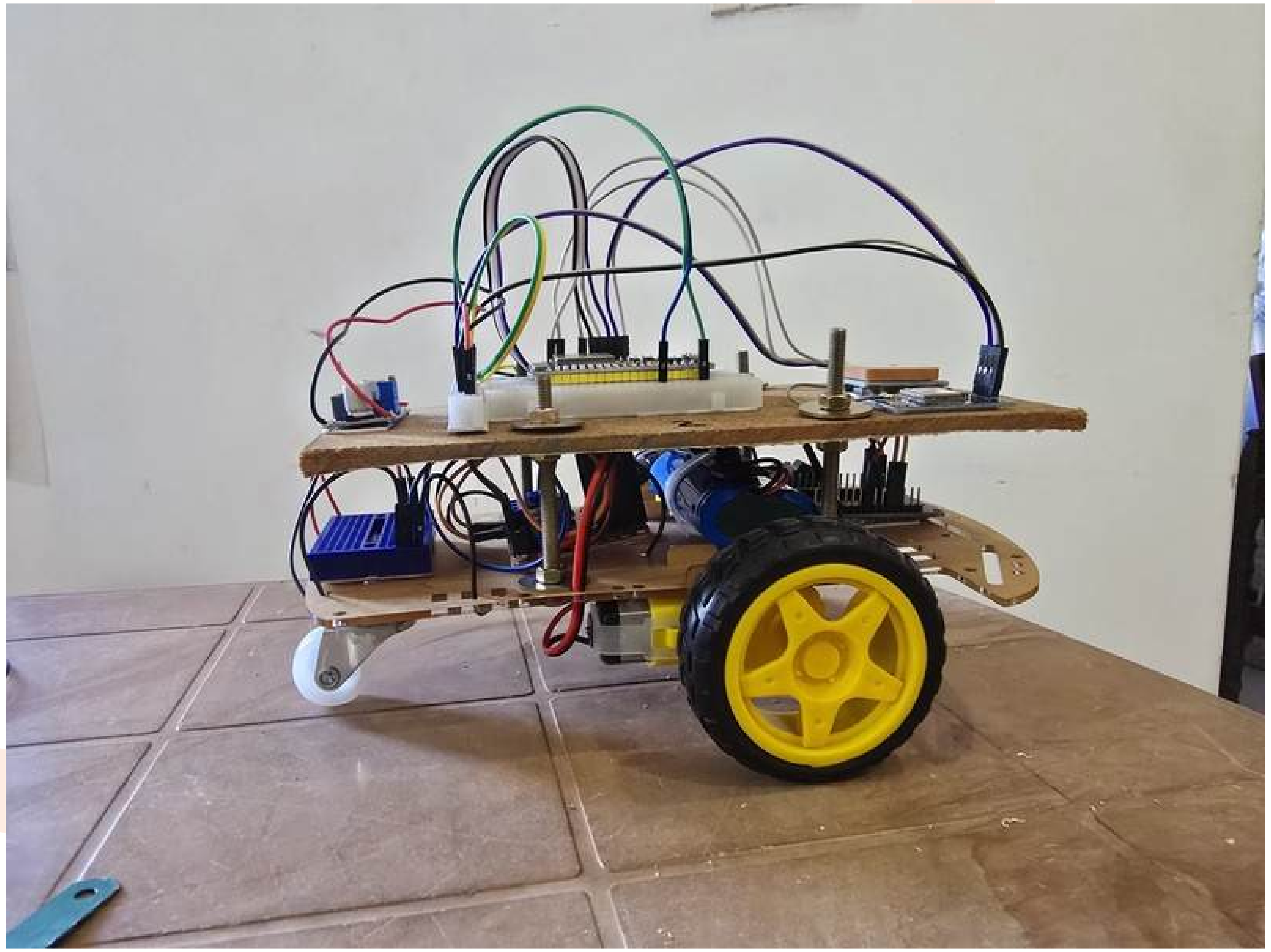
Authentication Metrics

- **Authentication Delay (authDelay):**The average time taken to sign or verify a message using the cryptographic algorithm (ECC or shared key with SHA-256).
- **Maximum Authentication Time (maxAuthTime):** The longest time taken to sign or verify any single message during the experiment
- **Minimum Authentication Time (minAuthTime):** The shortest time taken to sign or verify any single message during the experiment

Network Performance Metrics

- **Throughput:**The rate at which messages are successfully received, measured in messages per second.
- **Jitter:** The average variation in inter-arrival time between consecutive messages.
- **Packet Loss:** The percentage of messages that were sent but not receive;
- **Attack Impact:** The percentage of messages that were dropped due to the blackhole attack compared to the total messages that should have been received.

Use of HARDWARE



ESP32 - WROOM 32

MPU-6050

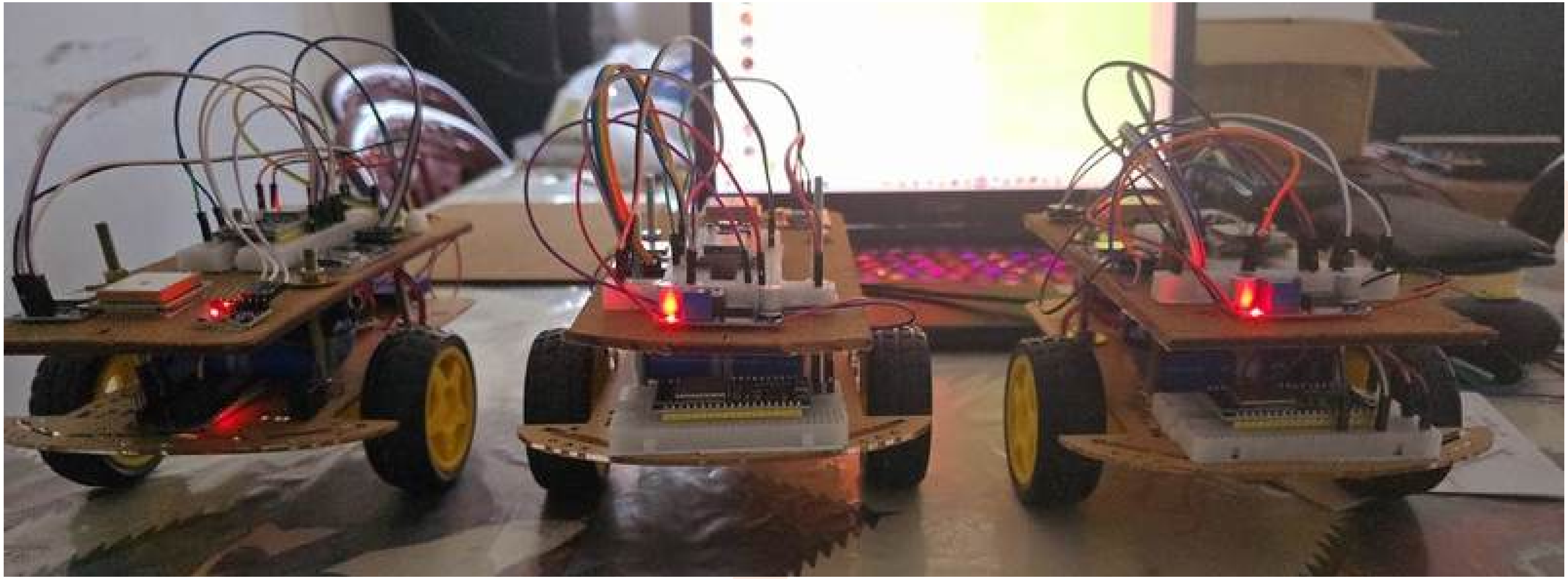
GPS Sensor

SD card Module

Volatge stepdown module

Jumper Wires

CREATED 3 CARS



ESP32 Sensor Dashboard

ESP32 Sensor Dashboard

Implementing Lightweight ECC based authentication in V2V to Mitigate Black Hole Attack

Status: Fetched 0 readings
Connection: Disconnected
Devices: 0
Last Updated: 9:41:39 AM
Refresh Rate: 1.0s

Manual Refresh

Disable Auto-Refresh

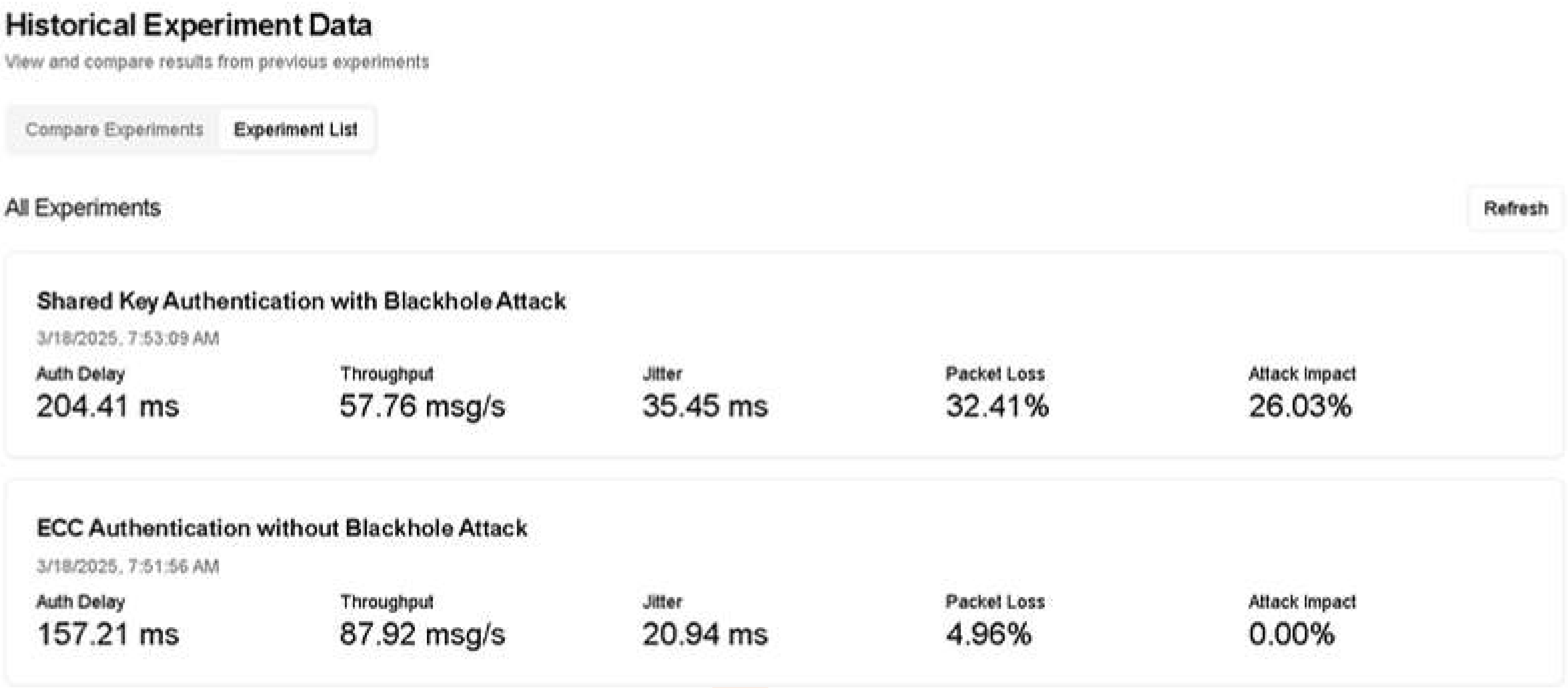
Disconnected 0 devices connected

Sensor Overview | Sensor History | Experiment Control | Experiment Results

Waiting for devices to connect...

Make sure your ESP32 devices are powered on and configured correctly.

Authentication Results



REQUIREMENTS

Functional Requirements:

- Secure vehicle authentication.
- Real-time data exchange between authenticated vehicles
- Blackhole attack detection and mitigation
- Performance monitoring and metrics collection
- Scalability to support multiple vehicle nodes
- Fault tolerance and recovery mechanisms

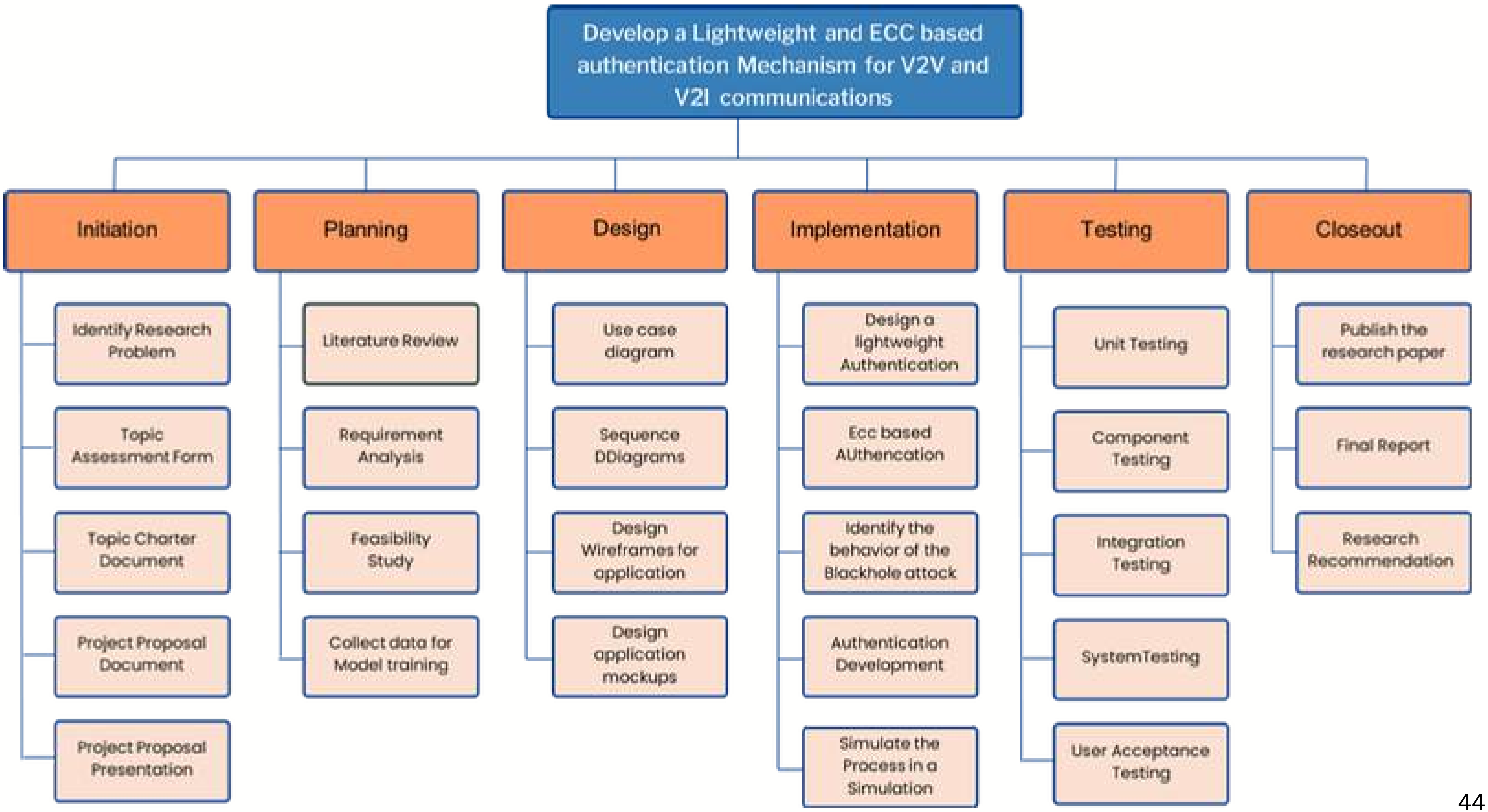
Technical Requirements:

- ECC implementation using Curve25519 or secp256r1
- Secure key storage on ESP32
- Efficient cryptographic operations optimization
- WebSocket server with secure connection handling
- Data visualization and analysis capabilities
- Logging and monitoring infrastructure

TOOLS & TECHNOLOGIES

- Arduino IDE/PlatformIO for ESP32 development
- ECC libraries optimized for embedded systems
- WebSocket protocol for real-time communication
- Data analysis tools for performance evaluation

Statistical analysis for experimental validation


Develop a Lightweight and ECC based authentication Mechanism for V2V and V2I communications
Initiation
Planning
Design
Implementation
Testing
Closeout
Identify Research Problem
Topic Assessment Form
Topic Charter Document
Project Proposal Document
Project Proposal Presentation
Literature Review
Requirement Analysis
Feasibility Study
Collect data for Model training
Use case diagram
Sequence DDiagrams
Design Wireframes for application
Design application mockups
Design a lightweight Authentication
Ecc based AUthencation
Identify the behavior of the Blackhole attack
Authentication Development
Simulate the Process in a Simulation
Unit Testing
Component Testing
Integration Testing
SystemTesting
User Acceptance Testing
Publish the research paper
Final Report
Research Recommendation

References

T. Ali, X. Li, H. Zhang, and J. Pan, "An ECC-Based Conditional Privacy-Preserving Authentication Scheme for V2V Communication in VANETs," in *Proc. IEEE International Conference on Communications (ICC)*, pp. 1-6, 2022. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-16-8586-6_6

"An Efficient Dynamic Solution for the Detection and Prevention of Black Hole Attack in VANET," *Sensors*, vol. 22, no. 5, pp. 1897, Mar. 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/5/1897

"Cyber Security Challenges and Solutions for V2X Communications," *arXiv preprint arXiv:1901.01053*, Jan. 2019. [Online]. Available: https://arxiv.org/pdf/1901.01053

IT21146442

Jayasinghe K.A.C.T

Cyber Security

BACKGROUND & RESEARCH PROBLEM

- Traditional cryptographic methods are becoming insufficient because they are vulnerable to sophisticated side-channel attacks, cloning attempts, and tampering threats.
- Physical Unclonable Functions (PUFs) offer a promising solution due to their inherent uniqueness and resistance to cloning.
- The challenge lies in integrating PUFs into a comprehensive challenge-response mechanism that ensures the security and efficiency required for Autonomous Vehicles (AVs).
- PUF-based authentication mechanisms face challenges in resisting predictive attacks and lack robust frameworks for real-world adversarial testing.

OBJECTIVES

MAIN OBJECTIVES

Develop a PUF-based challenge-response mechanism to ensure robust vehicle authentication and protection against physical attacks.

SUB OBJECTIVES

- Research current Physical Unclonable Function (PUF) technologies and their use cases in security systems.
- Analyze the benefits of different PUF types (e.g., SRAM, Ring Oscillator) for vehicle authentication.
- Develop a challenge-response mechanism utilizing PUF technology that is tailored for vehicle authentication.
- Conduct rigorous testing of the PUF-based authentication mechanism under various Environmental scenarios.

PUF-Physical Unclonable Function

- The Physical Unclonable Function (PUF) is a hardware-based security technology that uses the unique physical characteristics of devices to generate identifiers. This helps protect the device from being copied or tampered with.

On-Board Units (OBUs)

- On-Board Units (OBUs) are electronic devices installed in vehicles to enable communication with other vehicles, infrastructure, or systems.

Trusted Authority (TA)

- A Trusted Authority (TA) is a reliable organization or system that verifies identities, manages security credentials, and ensures trust in digital communications, such as in authentication or encryption processes.

Challenge Response Mechanism

- The Challenge-Response Mechanism is a security method where a system sends a random question (challenge) to a user or device, and the user/device must provide the correct answer (response) to prove their identity.

Challenge Response Authentication

Enrollment Phase

- Multiple challenges are sent to the PUF.
- The corresponding responses are collected.
- These CRPs (challenge + response) are stored in the server/database for future use.

Verification Phase

- The server selects a stored challenge from the database.
- This challenge is sent again to the PUF to generate a new response.
- The new response is compared with the stored response.
  - If they match (within acceptable range) → Authentication successful.
  - If not → Authentication failed.

OBJECTIVES

Completed Objectives

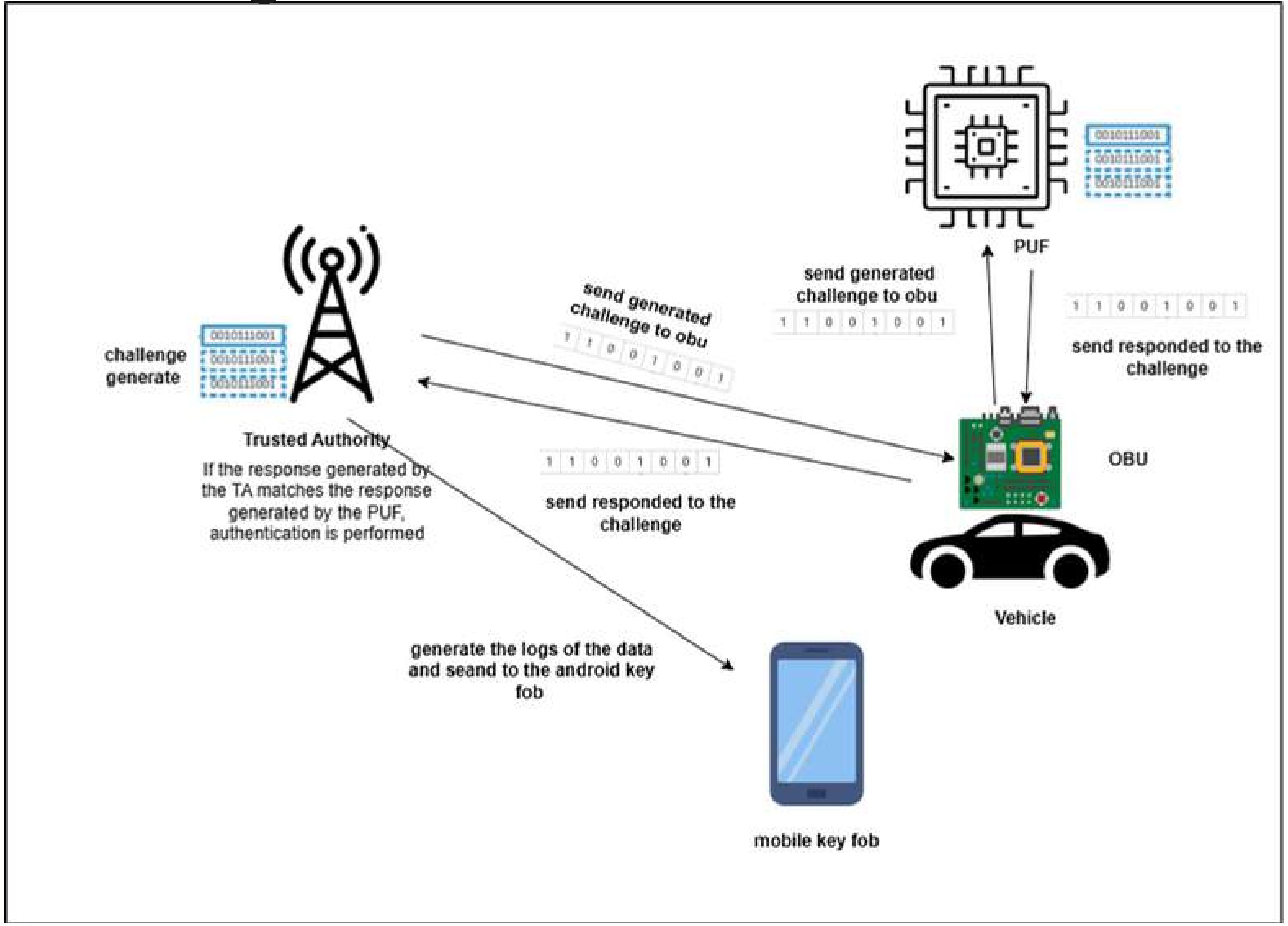
- implement the Challenge Response Mechanism using the Cryptographically Secure Pseudorandom Number Generators(CSPNG).
- Developed logging mechanism when genrate logs when authentication happen and save the challenges and the response.

Ongoing Objectives

- Developed the Physical Unclonable Function (PUF) technologies and their use cases in security systems.
- Conduct rigorous testing of the PUF-based authentication mechanism under various Environmental scenarios.

METHODOLOGY

System Diagram



REQUIREMENTS

Functional Requirements:

- The PUF must be implemented in such a way that it can generate unique, unpredictable responses based on physical hardware characteristics.
- The system must support a challenge-response protocol where a vehicle can generate a response to a given challenge using the PUF.
- The system must verify the authenticity of vehicles based on their challenge-response pairs

Non- Functional Requirements:

- Security
- Performance
- Reliability
- Usability
- Scalability

Technical Requirements:

- Implement a secure random number generator (RNG) to produce unique and unpredictable challenges.
- Test PUF responses under various environmental conditions (temperature, voltage) to ensure stability.
- Implement a system to periodically regenerate challenges to ensure they are unpredictable.

TOOLS & TECHNOLOGIES

Technologies

- C++
- Python
- PyCrypto
- HDL

Algorythm & Architechtures

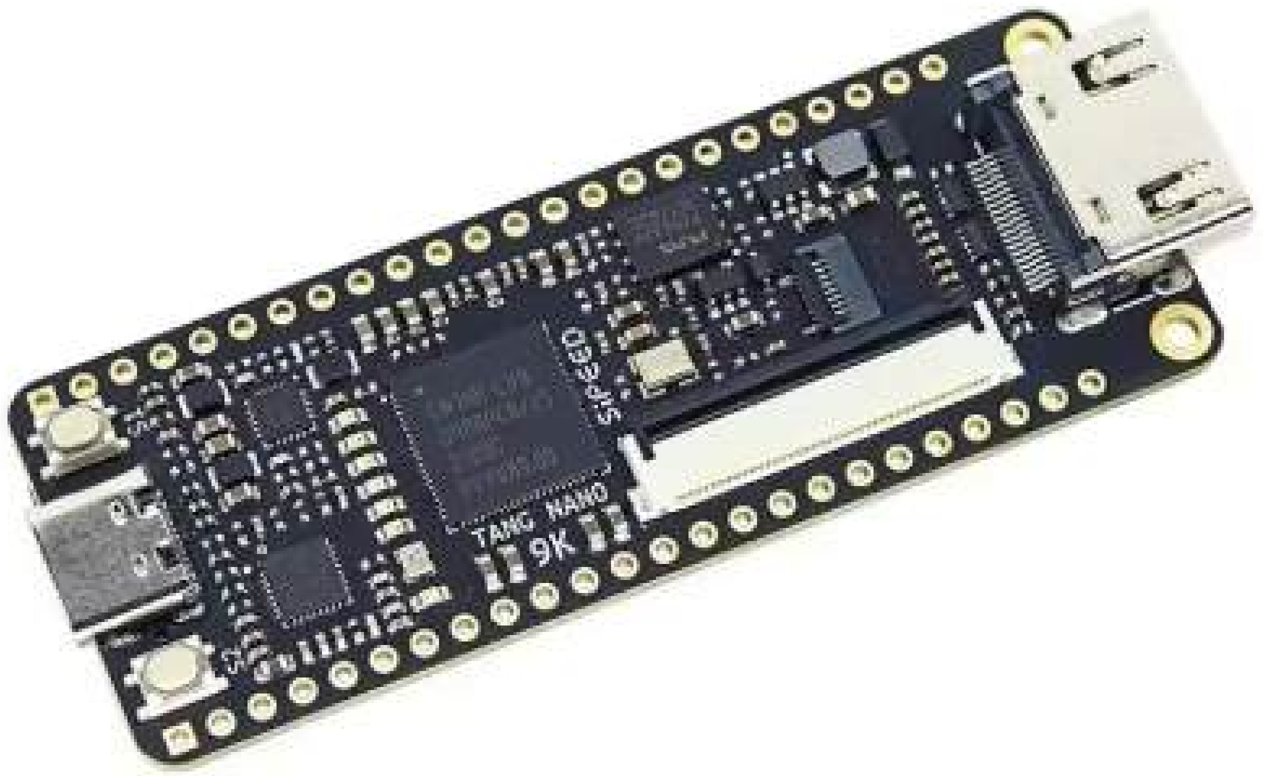
- challenge-response mechanism
- Physical Unclonable Functions

Techniques

- Performance Evaluation
- Data Encryption and Decryption

Hardware Components Needed

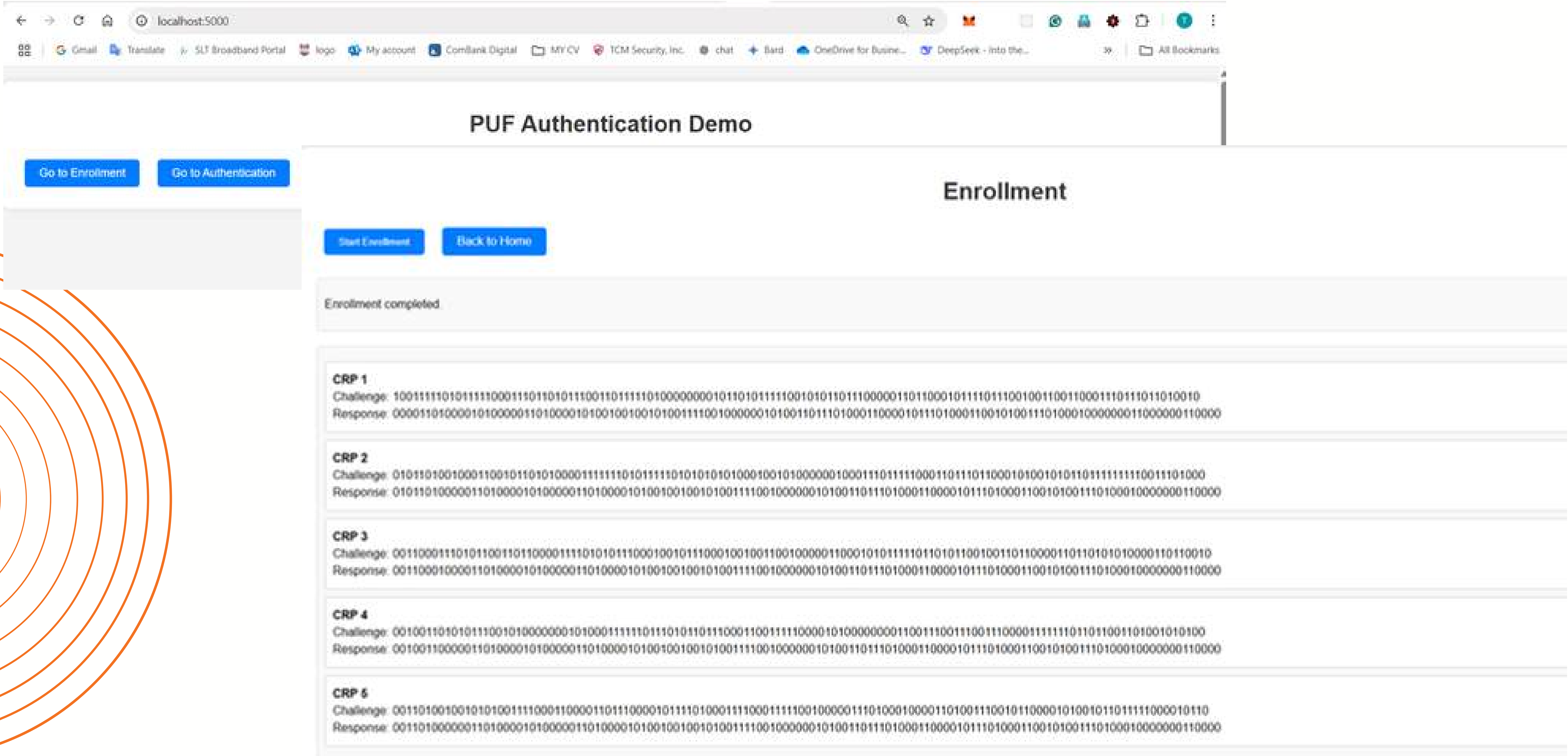
- Field-Programmable Gate Array (FPGA)



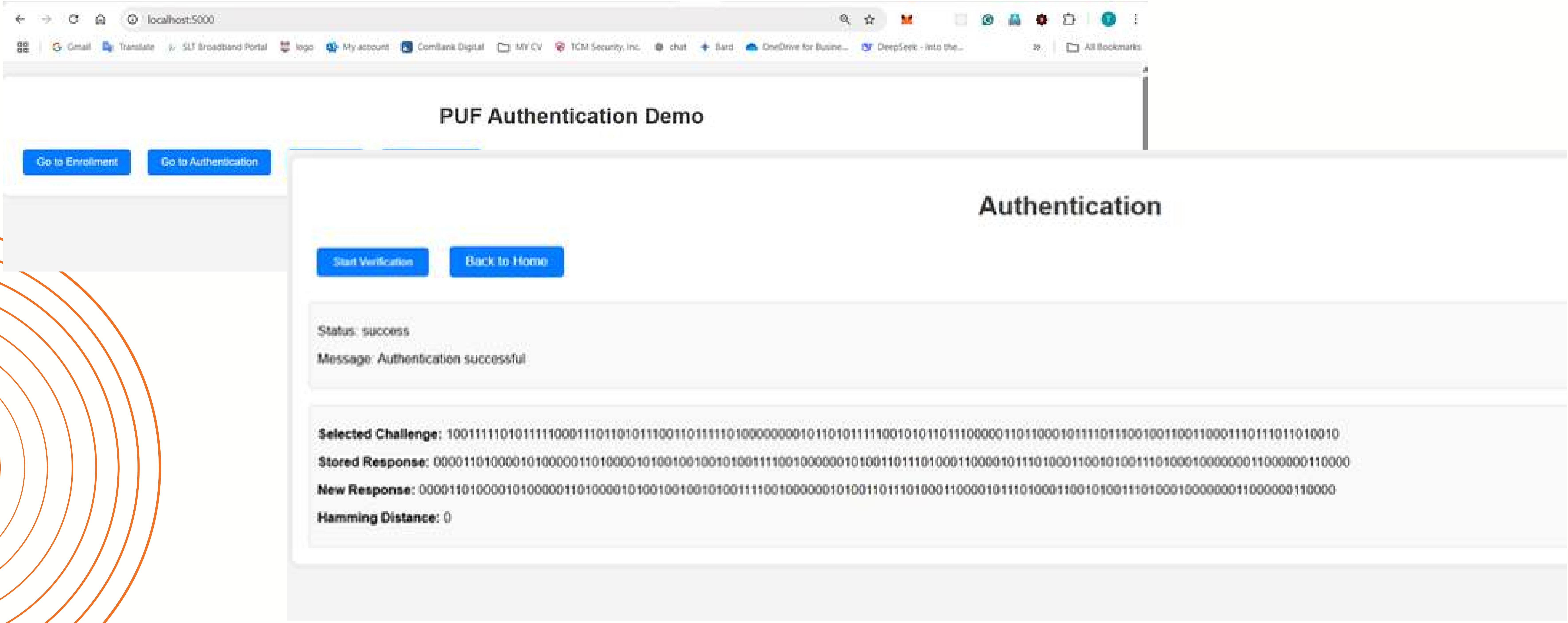
Current Progress

- Challenge Generation & Transmission
  - Implemented the challenge generation module.
  - Transmitting challenges via a Python Flask server.
- FPGA Implementation
  - Programmed the FPGA using HDL to generate responses.
  - Implemented a UART-USB interface for challenge transmission and response reception.
- Database Integration & Authentication Process
  - Stored generated challenges and responses in a database.
  - Verification Process:
    - The server selects a stored challenge and exchanges it with the PUF.
    - The generated response is compared with the stored response in the database.

Hardware Components Needed



Hardware Components Needed



Challenge-Response Mechanism

Challenge generate

- CSPRNG is used to generate secure and unpredictable challenges.
- The challenge is a 128-bit random string that ensures high entropy and unpredictability.
- This challenge is created using Python’s secrets.token_bytes() method.
- The challenge is converted into 16 bytes for efficient communication with the PUF.

```python
def generate_csprng_challenge():
    """Generate a 128-bit challenge using a CSPRNG with secrets.token_bytes."""
    challenge_bytes = secrets.token_bytes(16)
    challenge = ''.join(format(byte, '08b') for byte in challenge_bytes)
    log_message("INFO", f"Generated CSPRNG challenge: {challenge}")
    return challenge
```

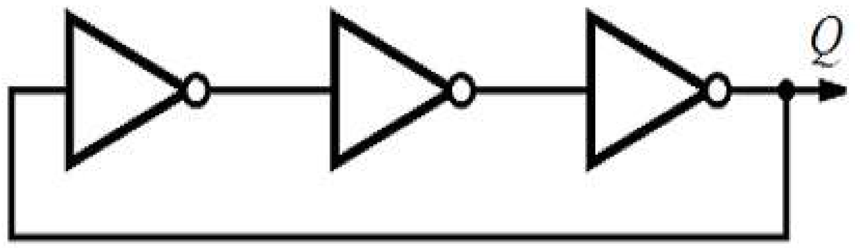
Challenge-Response Mechanism

Sending the Challenge to FPGA

- A random 128-bit challenge is generated in binary (e.g., 00100011...).
- It is converted into 16 bytes (e.g., 23 ee e2 8a...) because hardware and serial ports work with data in bytes.
- The 16-byte challenge is sent to the FPGA using UART (serial communication).

How the PUF Works in FPGA

- The FPGA has 32 Ring Oscillators (ROs). These are tiny circuits that toggle on/off very fast.
- Due to tiny manufacturing differences, each RO toggles at a unique speed.
- All ROs are turned on, and their toggles are counted for a short time (1000 cycles) using 8-bit counters.



Ring Oscillators

Challenge-Response Mechanism

Generating the 128-bit Response

- For each bit in the 128-bit response:
- The challenge selects two ROs using simple math.
- The counts of these two ROs are compared.
  - If RO_A > RO_B, the response bit is 1.
  - If RO_A ≤ RO_B, the response bit is 0.
- This creates a unique 128-bit response based on the physical chip.

Sending the Response Back

- The 128-bit response is converted into 16 bytes.
- Sent back to the computer through UART.

Unit	File	Register	LUT	ALU	BSRAM	SSRAM
ro_puf_...	**src\ro_puf_top.v**	685(1)	7175(2)	256(0)	0 (0)	0 (0)
uart_rx(u...	src\uart_rx.v	156(156)	77(77)	0 (0)	0 (0)	0 (0)
uart_tx(u...	src\uart_tx.v	29(29)	236(236)	0 (0)	0 (0)	0 (0)
ro_puf_c...	src\ro_puf_core.v	499(499)	6860(6860)	256(256)	0 (0)	0 (0)

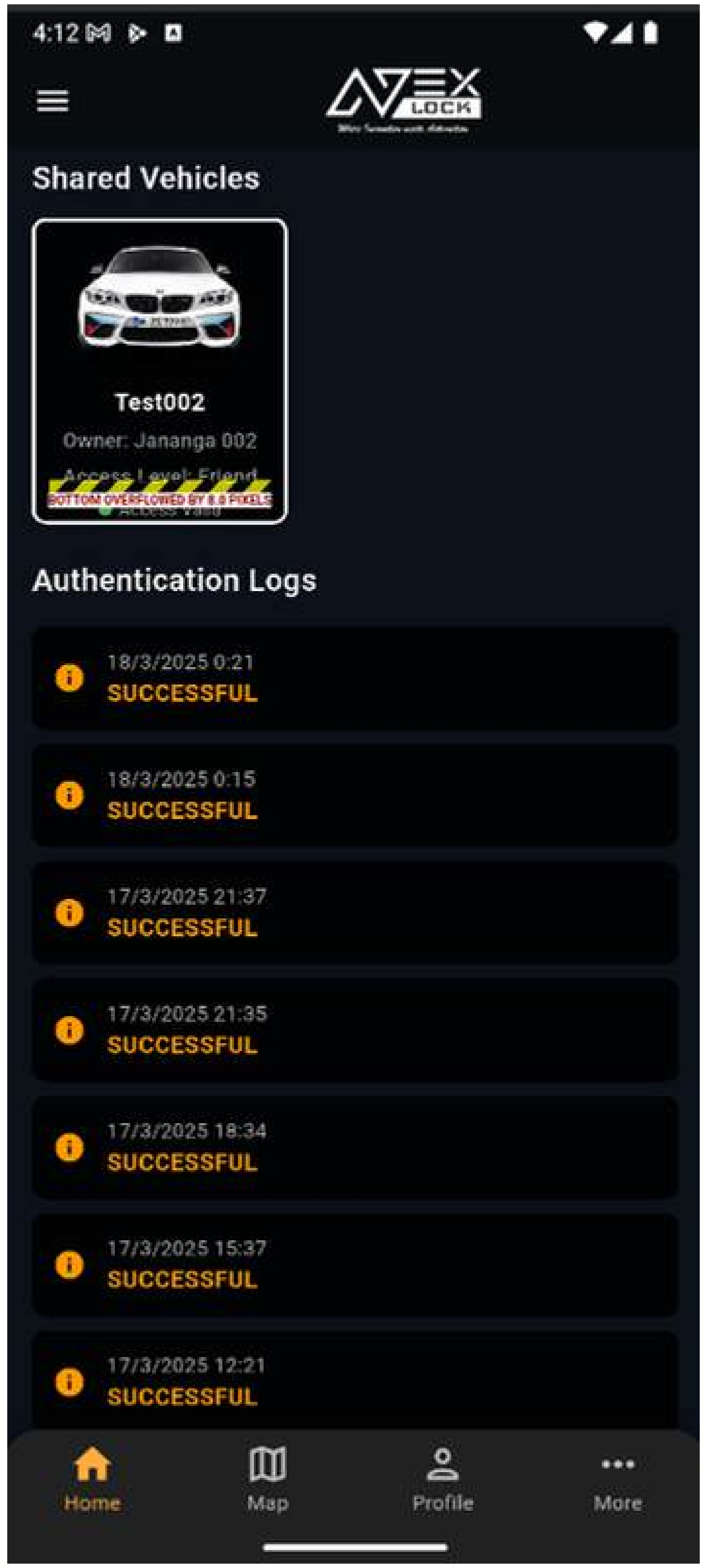
Why This Is Secure

- The response is unique to each FPGA because of unclonable physical differences.
- It’s very hard to guess or fake the response.
- Small changes (noise) are allowed we check Hamming Distance to confirm if the response is close enough for success.

Log interface on the Mobile App

Purpose of Integrating Authentication Logs into the Mobile App

- Enhances security visibility by providing real-time monitoring.
- Allows users to detect suspicious activities or failed authentication attempts.
- Increases transparency and improves security awareness.



References

- Cyber Security Protocol for Secure Traffic Monitoring Systems using PUF-based Key Management https://ieeexplore.ieee.org/abstract/document/9426088

- Chaotic map-based authentication scheme using physical unclonable function for Internet of autonomous vehiclehttps://ieeexplore.ieee.org/document/9994238

- Two-Factor Authentication Protocol Using Physical Unclonable Function for IoV doi:https://ieeexplore.ieee.org/document/8855828

IT21249648

Wanigasekara W.M.I.W

Cyber Security

BACKGROUND & RESEARCH PROBLEM

- GPS provides real-time location data, enabling autonomous vehicles to determine their precise position on a map
- By providing location data, GPS helps the vehicle maintain safe distances from other objects and vehicles, contributing to collision avoidance systems.
- GPS spoofing involves transmitting falsified signals to mislead receivers, causing autonomous vehicles to misinterpret their location.
- This can result in erroneous positioning data, leading to misrouting of vehicles, accidents, or unauthorized access to restricted areas.

OBJECTIVES

Main Objectives

- The objective is to design and deploy a machine learning-based GPS spoofing detection system that shall be capable of real-time analysis and immediate threat notification.

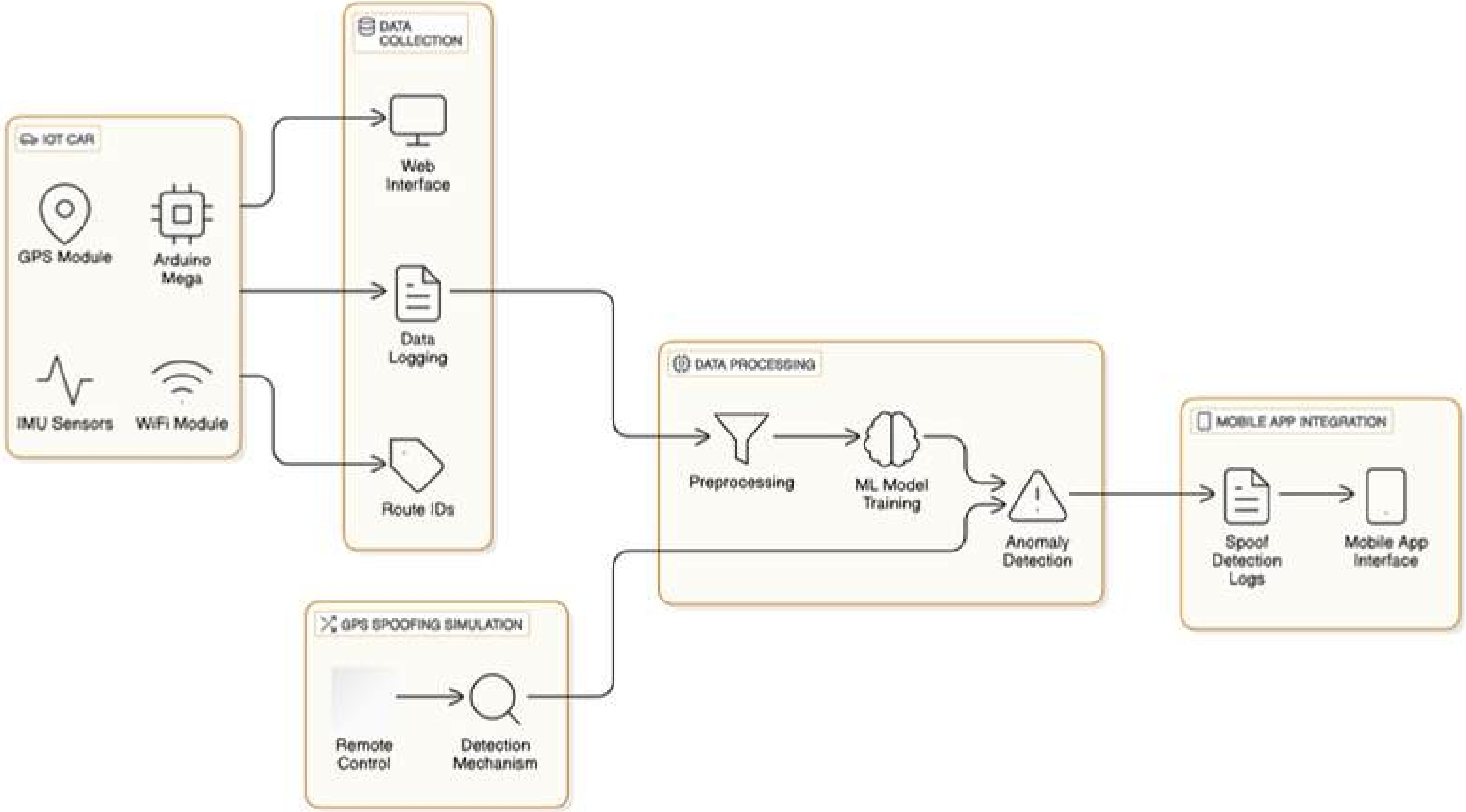
Sub Objectives

- Collect high-quality GPS data (authentic gps data) for model training and validation.
- Evaluate and select suitable machine learning/deep learning models (e.g., RF, FCNN, KNN, SVM, XGBoost) for GPS anomaly detection.
- Implement the detection system on an embedded device (e.g., Raspberry Pi) for real-world deployment.
- Sending real-time alerts and system status to the Android app to enhance user interaction.

Methology

- To simulate a GPS spoofing attack, a HackRF tool is typically required.
- However, due to legal restrictions for transmitting fake signals and high costs, an alternative approach is used.
- The experiment begins with data collection using an IoT-based rover equipped with an Arduino Mega, GPS module, IMU sensors, and WiFi module.
- The rover is driven along predefined routes, such as A → B and B → C, under various conditions.
- GPS coordinates, speed, and trajectory data are collected and used to train a machine learning model.
- Once trained, the model learns the expected paths and movement patterns of the rover.
- During real-time operation, the rover autonomously follows the trained routes.
- If it encounters GPS data that does not match the known paths or detects an unknown trajectory, the model classifies it as an anomaly.
- Instead of following the spoofed route, the rover prevents itself from moving along the incorrect path and generates an alert.
- To enhance security, the spoof detection logs are integrated into a mobile application. (component 01)
- The mobile app provides real-time notifications if a spoofed path is detected.

Diagram


DATA COLLECTION
IOT CAR
GPS Module
Arduino Mega
IMU Sensors
WiFi Module
Web Interface
Data Logging
Route IDs
DATA PROCESSING
Preprocessing
ML Model Training
Anomaly Detection
MOBILE APP INTEGRATION
Spoof Detection Logs
Mobile App Interface
GPS SPOOFING SIMULATION
Remote Control
Detection Mechanism

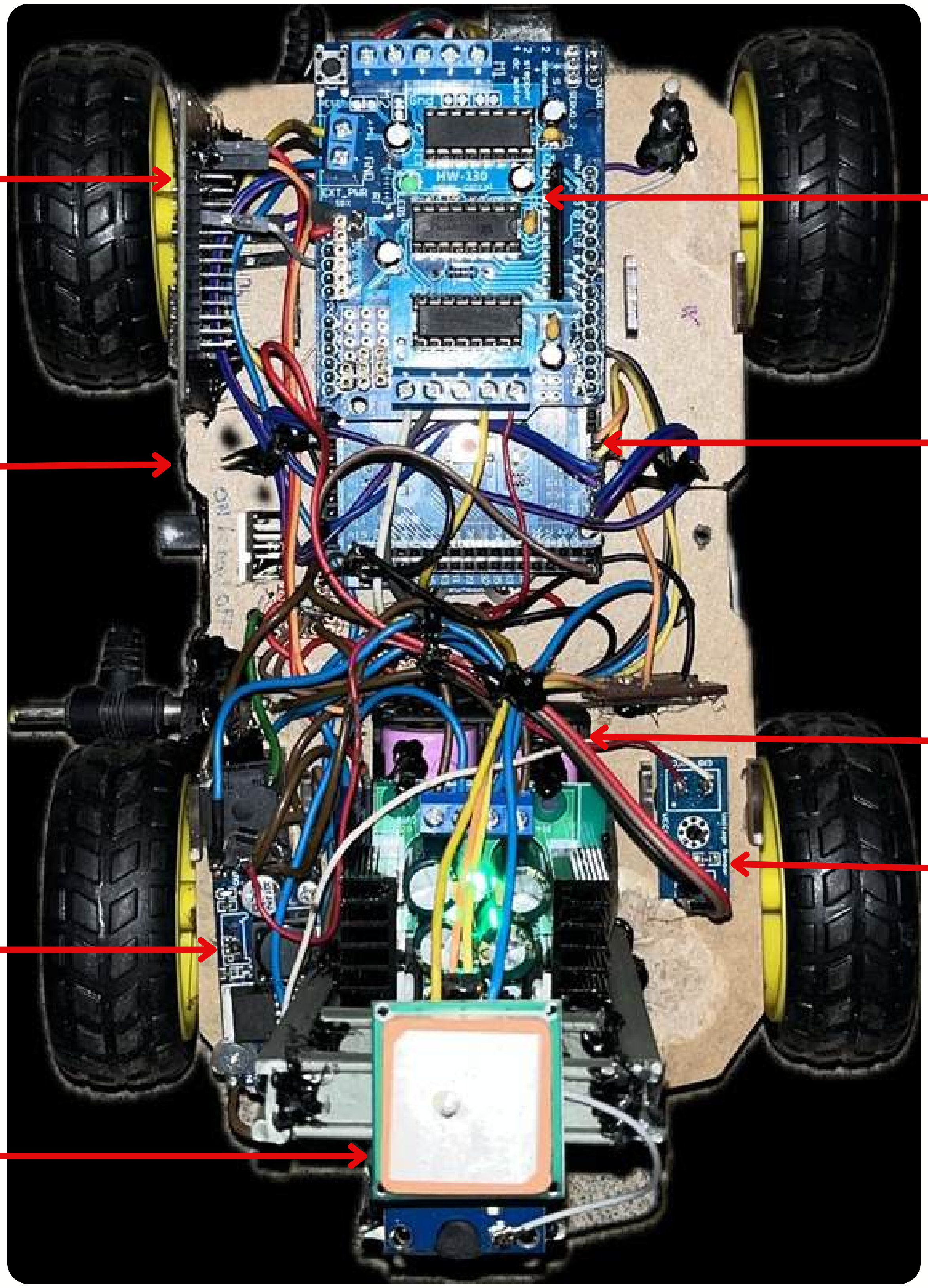
About The Rover

NodeMCU ESP8266 Board
Handles WIFI communication

Accelerometer Gyroscope Module
Monitors motion and orientation

BMS Charger Protection
Ensures safe battery management

GPS Module
Provides real-time location tracking



Motor Driver
Controls wheel movement

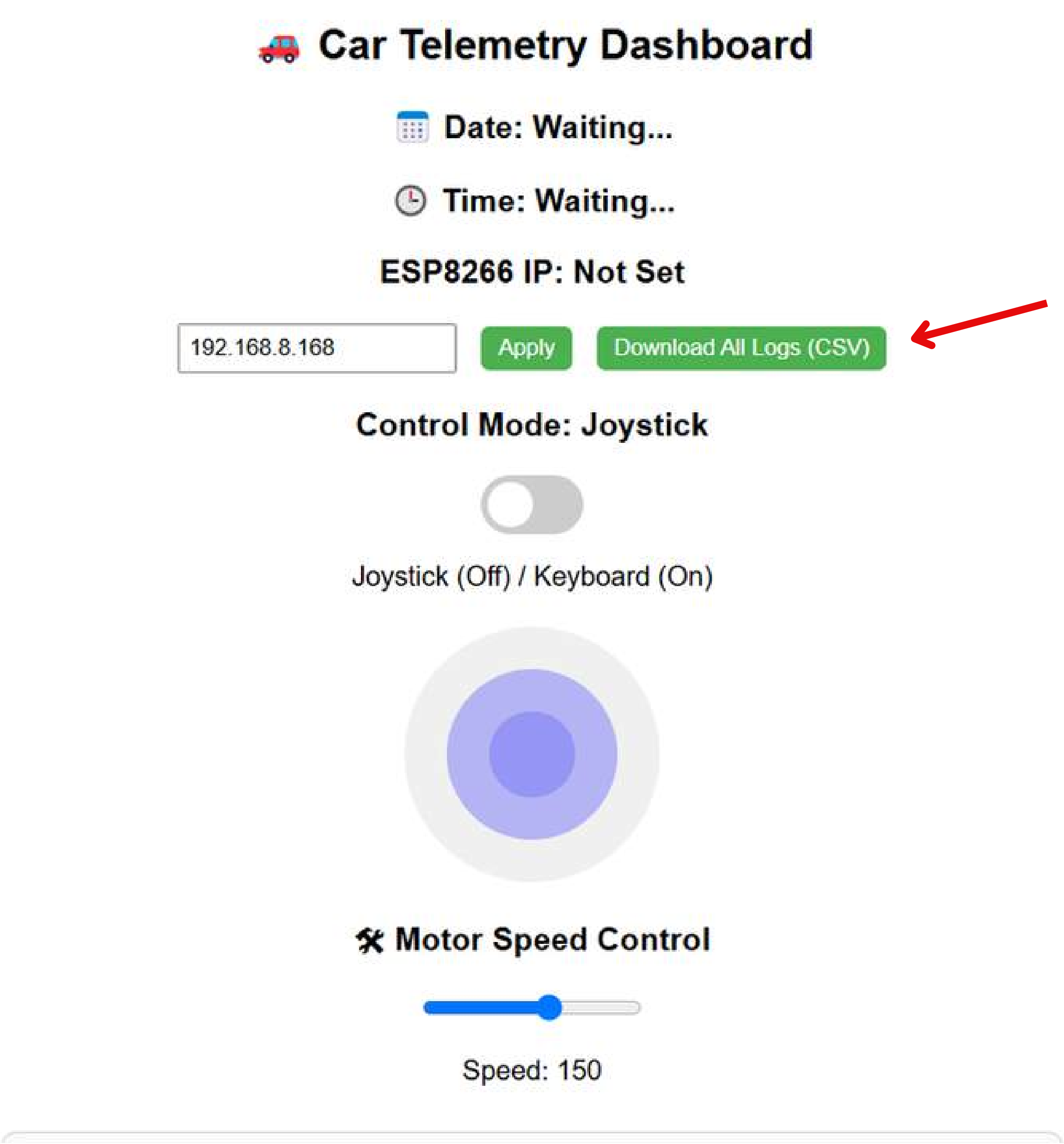
Arduino MEGA Board
Main microcontroller for processing

Batteries

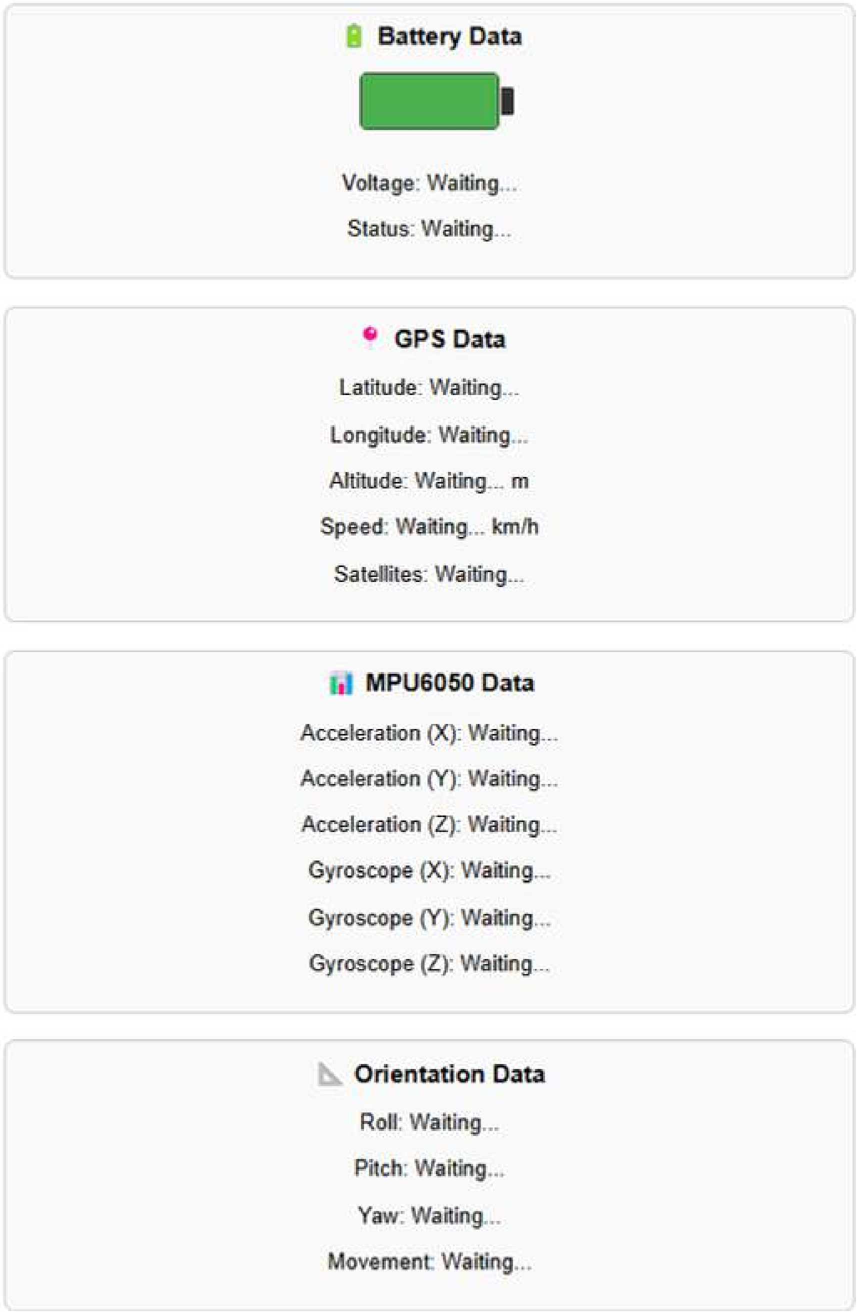
Voltage Sensor
Measures power levels

About The Controller

- The rover is controlled via a webpage and operated using the arrow keys on a laptop.



Downloaded the data logs in to .csv file



About the Rover

Core functionalities (Nodemcu)

- WiFi Connectivity: Manages WiFi connection using the ESP8266 capabilities
- WebSocket Server: Sets up a WebSocket server on port 81 for real-time communication
- Serial Communication: Maintains communication with the Arduino via SoftwareSerial
- EEPROM Management: Stores WiFi credentials persistently in EEPROM memory

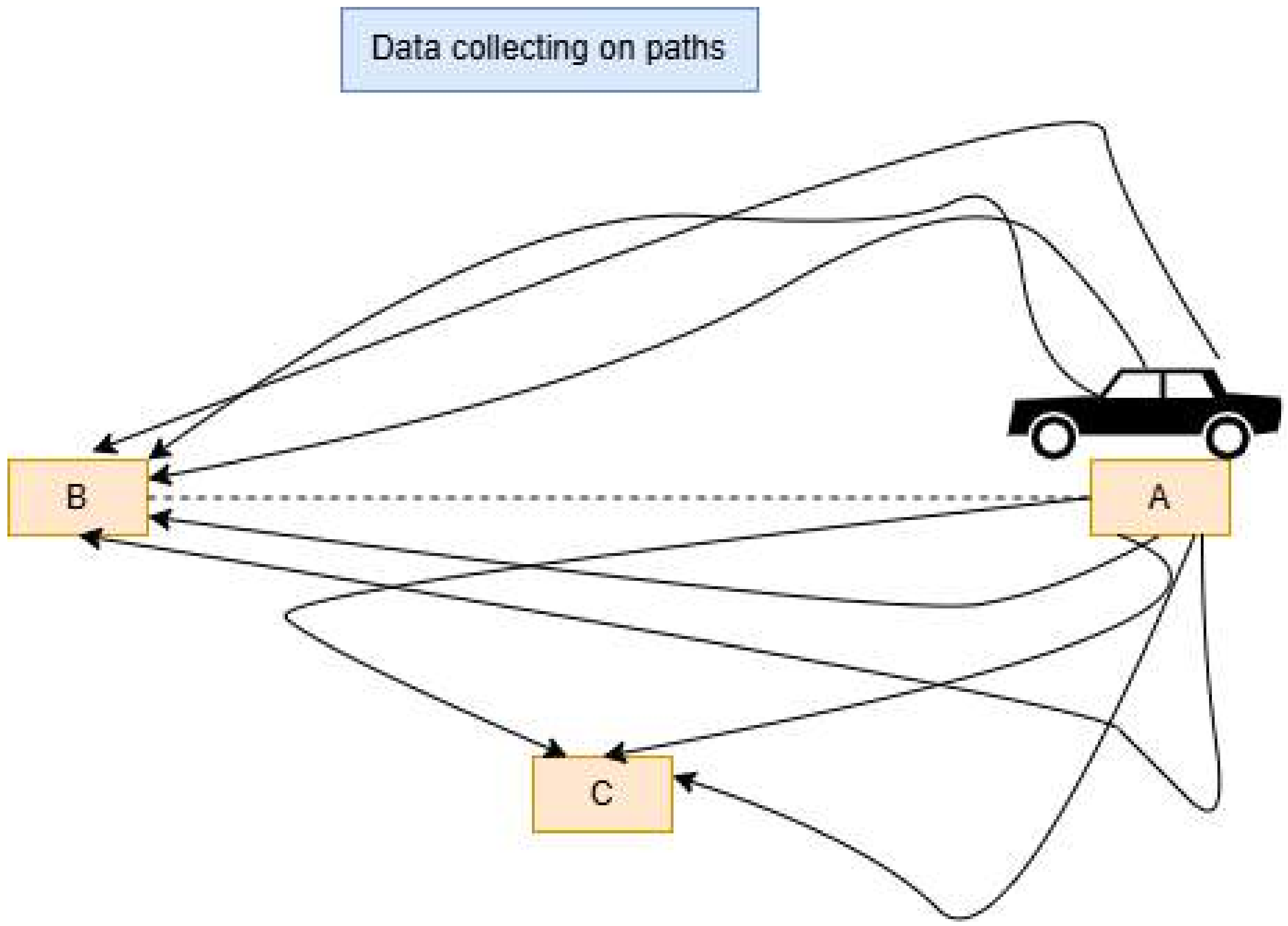
Core functionalities (Arduino MEGA)

- Motor Control: Controls four DC motors using the AFMotor library
- GPS Tracking: Uses a GPS module with TinyGPSPlus to get location data
- Motion Sensing: Implements an MPU6050 accelerometer/gyroscope to detect movement
- Battery Monitoring: Monitors battery voltage through an analog pin
- WiFi Connectivity: Communicates with an ESP8266 WiFi module

Current Progress

Data Collection Phase

Pre-Defined Area



- Drive the rover on all the paths in pre defined road.
- Collect all GPS, accelerometer, and gyroscope data in different scenarios and behaviors.
    - straight paths
    - slow Turns
    - sharp Turns
    - speed Variations
    - intersections

Current **Progress**

Collecting data

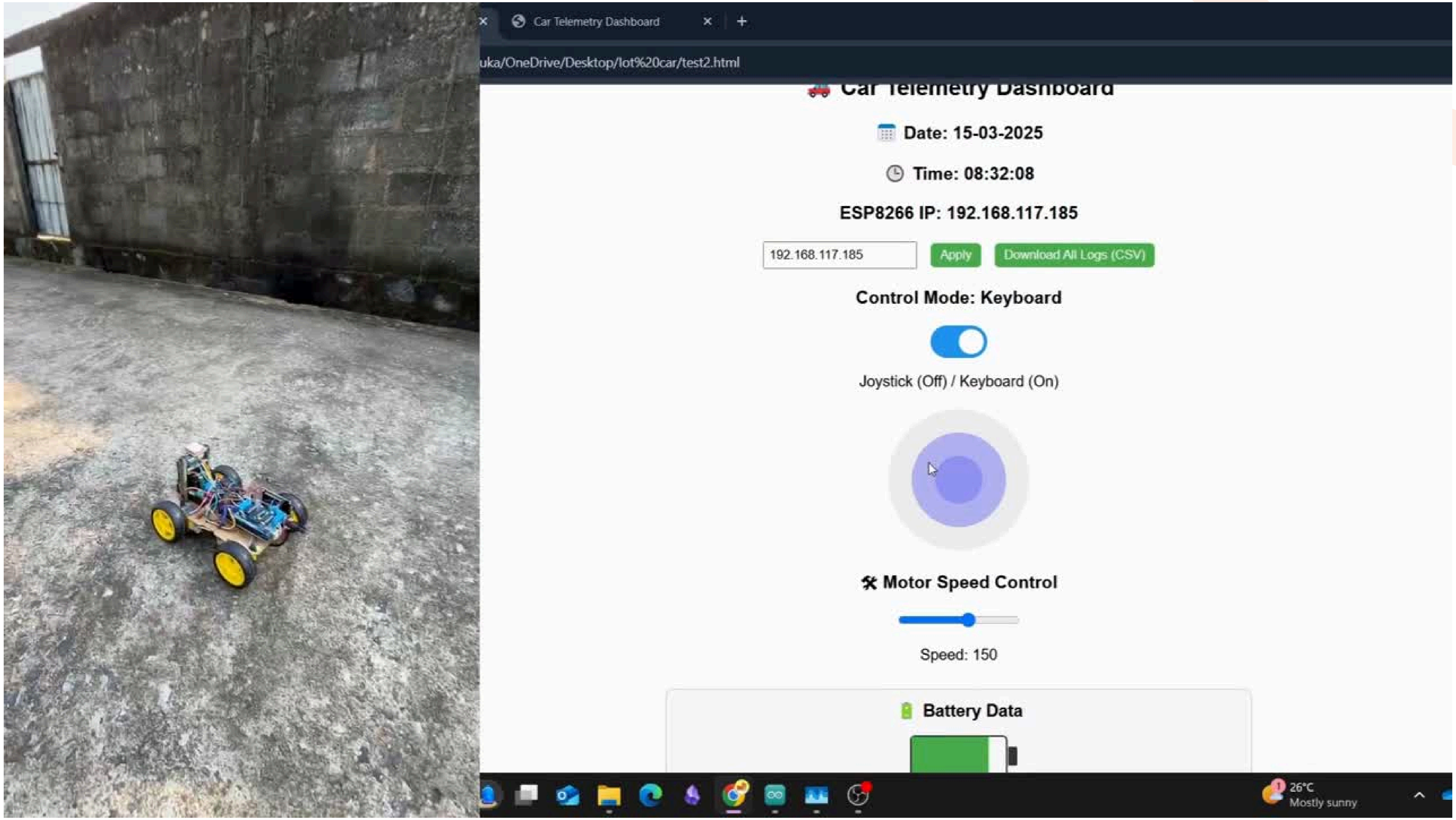
- Drives the rover to get sample data logs from the sensors.
- It includes timestamp, GPS_Latitude, GPS_Longitude, GPS_Altitude, GPS_Speed GPS_Satellites, Accel_X, Accel_Y, Accel_Z, Gyro_X, Gyro_Y, Gyro_Z, Roll, Pitch, Yaw

Timestamp	GPS_Latitude	GPS_Longitude	GPS_Altitude	GPS_Speed	GPS_Satellites	Accel_X	Accel_Y	Accel_Z	Gyro_X	Gyro_Y	Gyro_Z	Roll	Pitch	Yaw
2025-03-15T02:49:08.731Z	6.723295	80.094475	23	2.07	5	160	-584	14364	8	203	256	-2.33	-0.64	272.35
2025-03-15T02:49:10.221Z	6.723294	80.094482	22.9	1.19	5	292	-472	14436	-18	172	305	-1.87	-1.16	275.68
2025-03-15T02:49:11.474Z	6.723291	80.094475	22.2	1.48	5	252	-476	14444	-34	163	252	-1.89	-1	278.09
2025-03-15T02:49:12.647Z	6.72329	80.094475	22.5	1.48	5	276	-444	14564	-16	165	282	-1.75	-1.09	280.62
2025-03-15T02:49:13.910Z	6.723281	80.094475	23.7	1.7	5	360	-504	14344	-17	171	284	-2.01	-1.44	283.36
2025-03-15T02:49:15.076Z	6.723279	80.094475	23.4	1.87	5	344	-528	14500	N/A	206	307	-2.09	-1.36	286.09
2025-03-15T02:49:16.323Z	6.723279	80.094475	23.9	1.43	5	180	-512	14268	-34	110	235	-2.06	-0.72	288.33
2025-03-15T02:49:17.661Z	6.72327	80.094467	23.3	2.46	5	228	-448	14372	-6	162	233	-1.79	-0.91	290.71
2025-03-15T02:49:18.721Z	6.723258	80.09446	23.3	2.52	5	388	-552	14520	-31	216	293	-2.18	-1.53	293.08
2025-03-15T02:49:19.949Z	6.723258	80.094467	24.8	2.13	5	324	-492	14436	40	225	271	-1.95	-1.28	295.62
2025-03-15T02:49:21.153Z	6.723253	80.09446	25.3	2.35	5	264	-620	14524	4	204	253	-2.44	-1.04	297.94
2025-03-15T02:49:22.473Z	6.723254	80.09446	25.2	1.83	5	376	-500	14424	-2	182	293	-1.99	-1.49	300.9
2025-03-15T02:49:23.699Z	6.723258	80.09446	25.2	1.31	4	216	-556	14372	-4	203	261	-2.22	-0.86	303.34
2025-03-15T02:49:24.757Z	6.723258	80.094467	25	0.93	4	236	-424	14476	-20	164	227	-1.68	-0.93	305.17
2025-03-15T02:49:26.009Z	6.723262	80.094467	25.4	0.35	4	296	-480	14288	-3	149	248	-1.92	-1.19	307.54
2025-03-15T02:49:27.181Z	6.723257	80.09446	24.6	0.91	4	288	-500	14560	8	186	307	-1.97	-1.13	310.29
2025-03-15T02:49:28.395Z	6.723261	80.094467	24.6	0.61	4	272	-480	14444	-23	157	219	-1.9	-1.08	312.32
2025-03-15T02:49:29.608Z	6.723254	80.094467	24.3	1.2	5	324	-444	14364	3	165	255	-1.77	-1.29	314.68
2025-03-15T02:49:30.820Z	6.723242	80.09446	22.3	1.52	5	276	-564	14368	-35	187	285	-2.25	-1.1	317.32

SLIIT FACULTY OF COMPUTING

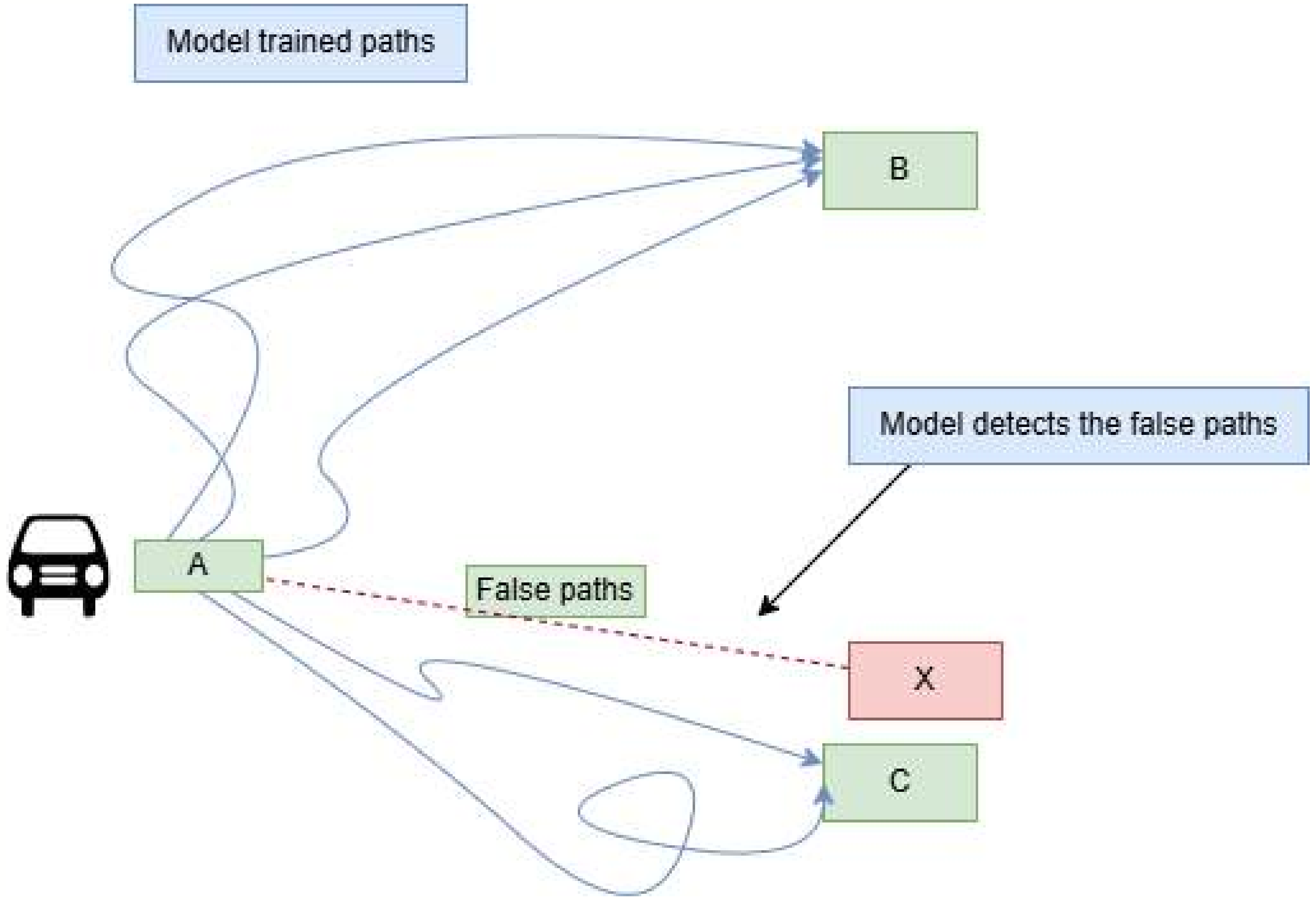
Current Progress

Data Collection Phase



Current **Progress**

Detection Phase



- The rover follows predefined paths (e.g., A → B and A → C) based on trained models.
- If the rover attempts to deviate from the expected path (e.g., moving toward X instead of C), the model detects it as a false path.
- The system identifies GPS anomalies in real time and prevents the rover from following untrained or spoofed routes.

Current Progress

Data Preprocessing Phase

Adding RouteID for Datasets

- After the data collection phase, the data is saved in a CSV file.
- The dataset contains 600+ rows of recorded GPS data.
- A RouteID column was added to each dataset using Google Colab for better identification and analysis.

```python
# Import pandas library
import pandas as pd

# Read the CSV file
# Replace 'your_file.csv' with your actual file name
df = pd.read_csv('log1.csv')

# Add a new column 'Route_ID' with value 'A-B' for all rows
df['Route_ID'] = 'A-B'

# Display the first few rows to verify
print(df.head())

# Save the modified dataset to a new CSV file
df.to_csv('output_file_with_route.csv', index=False)
print("File saved as 'output_file_with_route.csv'")
```

```
                  Timestamp  GPS_Latitude  GPS_Longitude  GPS_Altitude  \
0  2025-03-15T02:49:08.731Z      6.723295      80.094475          23.0
1  2025-03-15T02:49:10.221Z      6.723294      80.094482          22.9
2  2025-03-15T02:49:11.474Z      6.723291      80.094475          22.2
3  2025-03-15T02:49:12.647Z      6.723290      80.094475          22.5
4  2025-03-15T02:49:13.910Z      6.723281      80.094475          23.7

   GPS_Speed  GPS_Satellites  Accel_X  Accel_Y  Accel_Z  Gyro_X  Gyro_Y  \
0       2.07               5      160   -584.0    14364     8.0     203
1       1.19               5      292   -472.0    14436   -18.0     172
2       1.48               5      252   -476.0    14444   -34.0     163
3       1.48               5      276   -444.0    14564   -16.0     165
4       1.70               5      360   -504.0    14344   -17.0     171

   Gyro_Z  Roll  Pitch    Yaw Route_ID
0     256 -2.33  -0.64  272.35      A-B
1     305 -1.87  -1.16  275.68      A-B
2     252 -1.89  -1.00  278.09      A-B
3     282 -1.75  -1.09  280.62      A-B
4     284 -2.01  -1.44  283.36      A-B
File saved as 'output_file_with_route.csv'
```

Timestamp	GPS_Latitude	GPS_Longitude	GPS_Altitude	GPS_Speed	GPS_Satellites	Accel_X	Accel_Y	Accel_Z	Gyro_X	Gyro_Y	Gyro_Z	Roll	Pitch	Yaw	Route_ID
2025-03-15T02:49:08.731Z	6.723295	80.094475	23	2.07	5	160	-584	14364	8	203	256	-2.33	-0.64	272.35	A-B
2025-03-15T02:49:10.221Z	6.723294	80.094482	22.9	1.19	5	292	-472	14436	-18	172	305	-1.87	-1.16	275.68	A-B
2025-03-15T02:49:11.474Z	6.723291	80.094475	22.2	1.48	5	252	-476	14444	-34	163	252	-1.89	-1	278.09	A-B
2025-03-15T02:49:12.647Z	6.72329	80.094475	22.5	1.48	5	276	-444	14564	-16	165	282	-1.75	-1.09	280.62	A-B
2025-03-15T02:49:13.910Z	6.723281	80.094475	23.7	1.7	5	360	-504	14344	-17	171	284	-2.01	-1.44	283.36	A-B
2025-03-15T02:49:15.076Z	6.723279	80.094475	23.4	1.87	5	344	-528	14500		206	307	-2.09	-1.36	286.09	A-B
2025-03-15T02:49:16.323Z	6.723279	80.094475	23.9	1.43	5	180	-512	14268	-34	110	235	-2.06	-0.72	288.33	A-B
2025-03-15T02:49:17.661Z	6.72327	80.094467	23.3	2.46	5	228	-448	14372	-6	162	233	-1.79	-0.91	290.71	A-B
2025-03-15T02:49:18.721Z	6.723258	80.09446	23.3	2.52	5	388	-552	14520	-31	216	293	-2.18	-1.53	293.08	A-B
2025-03-15T02:49:19.949Z	6.723258	80.094467	24.8	2.13	5	324	-492	14436	40	225	271	-1.95	-1.28	295.62	A-B
2025-03-15T02:49:21.153Z	6.723253	80.09446	25.3	2.35	5	264	-620	14524	4	204	253	-2.44	-1.04	297.94	A-B
2025-03-15T02:49:22.473Z	6.723254	80.09446	25.2	1.83	5	376	-500	14424	-2	182	293	-1.99	-1.49	300.9	A-B
2025-03-15T02:49:23.699Z	6.723258	80.09446	25.2	1.31	4	216	-556	14372	-4	203	261	-2.22	-0.86	303.34	A-B
2025-03-15T02:49:24.757Z	6.723258	80.094467	25	0.93	4	236	-424	14476	-20	164	227	-1.68	-0.93	305.17	A-B
2025-03-15T02:49:26.009Z	6.723262	80.094467	25.4	0.35	4	296	-480	14288	-3	149	248	-1.92	-1.19	307.54	A-B

Current **Progress**

Data cleaning process

```python
# Step 1: Handle missing values
combined_df.replace('N/A', np.nan, inplace=True)
numeric_columns = ['GPS_Latitude', 'GPS_Longitude', 'GPS_Altitude', 'GPS_Speed',
                   'Accel_X', 'Accel_Y', 'Accel_Z', 'Gyro_X', 'Gyro_Y', 'Gyro_Z',
                   'Roll', 'Pitch', 'Yaw']
for col in numeric_columns:
    combined_df[col].fillna(combined_df[col].median(), inplace=True)

# Step 2: Convert data types
combined_df['Timestamp'] = pd.to_datetime(combined_df['Timestamp'], errors='coerce')
for col in numeric_columns:
    combined_df[col] = pd.to_numeric(combined_df[col], errors='coerce')
combined_df['GPS_Satellites'] = combined_df['GPS_Satellites'].astype(int)

# Step 3: Handle outliers
def remove_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return df[(df[column] >= lower_bound) & (df[column] <= upper_bound)]

columns_to_check = ['GPS_Speed', 'Accel_X', 'Accel_Y', 'Accel_Z']
for col in columns_to_check:
    combined_df = remove_outliers(combined_df, col)

# Step 4: Remove duplicates
combined_df.drop_duplicates(inplace=True)

# Step 5: Validate data consistency
combined_df = combined_df[
    (combined_df['GPS_Latitude'].between(-90, 90)) &
    (combined_df['GPS_Longitude'].between(-180, 180)) &
    (combined_df['GPS_Speed'] >= 0)
]
valid_routes = ['A-B', 'A-C']
combined_df = combined_df[combined_df['Route_ID'].isin(valid_routes)]
```

Timestamp	GPS_Latitu	GPS_Longi	GPS_Altitu	GPS_Spee	GPS_Satell	Accel_X	Accel_Y	Accel_Z	Gyro_X	Gyro_Y	Gyro_Z	Roll	Pitch	Yaw	Route_ID
2025-03-15 02:49:10.221000+00:00	6.723294	80.09448	22.9	1.19	5	292	-472	14436	-18	172	305	-1.87	-1.16	275.68	A-B
2025-03-15 02:49:11.474000+00:00	6.723291	80.09448	22.2	1.48	5	252	-476	14444	-34	163	252	-1.89	-1	278.09	A-B
2025-03-15 02:49:12.647000+00:00	6.72329	80.09448	22.5	1.48	5	276	-444	14564	-16	165	282	-1.75	-1.09	280.62	A-B
2025-03-15 02:49:13.910000+00:00	6.723281	80.09448	23.7	1.7	5	360	-504	14344	-17	171	284	-2.01	-1.44	283.36	A-B
2025-03-15 02:49:15.076000+00:00	6.723279	80.09448	23.4	1.87	5	344	-528	14500	-21	206	307	-2.09	-1.36	286.09	A-B
2025-03-15 02:49:16.323000+00:00	6.723279	80.09448	23.9	1.43	5	180	-512	14268	-34	110	235	-2.06	-0.72	288.33	A-B
2025-03-15 02:49:22.473000+00:00	6.723254	80.09446	25.2	1.83	5	376	-500	14424	-2	182	293	-1.99	-1.49	300.9	A-B
2025-03-15 02:49:23.699000+00:00	6.723258	80.09446	25.2	1.31	4	216	-556	14372	-4	203	261	-2.22	-0.86	303.34	A-B
2025-03-15 02:49:24.757000+00:00	6.723258	80.09447	25	0.93	4	236	-424	14476	-20	164	227	-1.68	-0.93	305.17	A-B
2025-03-15 02:49:26.009000+00:00	6.723262	80.09447	25.4	0.35	4	296	-480	14288	-3	149	248	-1.92	-1.19	307.54	A-B
2025-03-15 02:49:27.181000+00:00	6.723257	80.09446	24.6	0.91	4	288	-500	14560	8	186	307	-1.97	-1.13	310.29	A-B
2025-03-15 02:49:28.395000+00:00	6.723261	80.09447	24.6	0.61	4	272	-480	14444	-23	157	219	-1.9	-1.08	312.32	A-B
2025-03-15 02:49:29.608000+00:00	6.723254	80.09447	24.3	1.2	5	324	-444	14364	3	165	255	-1.77	-1.29	314.68	A-B
2025-03-15 02:49:30.820000+00:00	6.723242	80.09446	22.3	1.52	5	276	-564	14368	-35	187	285	-2.25	-1.1	317.32	A-B
2025-03-15 02:49:32.033000+00:00	6.723237	80.09446	22.1	1.67	5	160	-480	14504	5	189	270	-1.9	-0.63	319.82	A-B
2025-03-15 02:49:33.298000+00:00	6.723242	80.09447	21.9	0.98	5	224	-464	14376	-14	161	270	-1.85	-0.89	322.42	A-B
2025-03-15 02:49:34.618000+00:00	6.723248	80.09448	21.7	0.43	5	400	-492	14520	-21	158	254	-1.94	-1.58	324.98	A-B
2025-03-15 02:49:35.687000+00:00	6.723264	80.09448	21.7	1.06	5	380	-568	14396	4	190	294	-2.26	-1.51	327.38	A-B
2025-03-15 02:49:36.905000+00:00	6.723264	80.09448	21.1	0.81	4	216	-508	14528	-5	187	214	-2	-0.85	329.37	A-B
2025-03-15 02:49:38.137000+00:00	6.723268	80.09448	21.6	0.93	4	304	-612	14264	-48	193	280	-2.46	-1.22	332	A-B
2025-03-15 02:49:39.383000+00:00	6.723269	80.09449	22	0.87	4	276	-560	14548	-15	160	259	-2.2	-1.09	334.47	A-B
2025-03-15 02:49:40.698000+00:00	6.723271	80.09449	22.9	1.13	4	264	-460	14268	70	192	297	-1.85	-1.06	337.45	A-B
2025-03-15 02:49:41.826000+00:00	6.723274	80.0945	24.3	1.26	5	368	-528	14544	24	211	242	-2.08	-1.45	339.53	A-B
2025-03-15 02:49:43.060000+00:00	6.723275	80.09451	24.7	1.11	4	228	-456	14456	35	190	263	-1.81	-0.9	342.01	A-B
2025-03-15 02:49:45.512000+00:00	6.723278	80.09451	24.8	1.17	4	404	-368	14384	-60	186	290	-1.47	-1.61	347.29	A-B
2025-03-15 02:49:46.740000+00:00	6.723283	80.09451	23.5	1.24	5	256	-568	14416	-8	157	243	-2.26	-1.02	349.56	A-B

Current Progress

Combining the Datasets

- Combine the datasets that contain Route_IDs A-B and A-C into a single dataset.

```python
import pandas as pd

# Load the two datasets (replace with your file names)
df_ab = pd.read_csv('route1.csv')  # Dataset with Route_ID "A-B"
df_ac = pd.read_csv('route2.csv')  # Dataset with Route_ID "A-C"

# Ensure Route_ID is present and correct
df_ab['Route_ID'] = 'A-B'  # Already added, but confirming
df_ac['Route_ID'] = 'A-C'  # Already added, but confirming

# Combine the datasets
combined_df = pd.concat([df_ab, df_ac], ignore_index=True)

# Display the first few rows to verify
print(combined_df.head())

# Check the number of rows for each Route_ID
print(combined_df['Route_ID'].value_counts())

# Save the combined dataset to a new CSV file
combined_df.to_csv('combined_dataset.csv', index=False)
print("Combined dataset saved as 'combined_dataset.csv'")
```

```
                  Timestamp  GPS_Latitude  GPS_Longitude  GPS_Altitude  \
0  2025-03-15T02:49:08.731Z      6.723295      80.094475          23.0
1  2025-03-15T02:49:10.221Z      6.723294      80.094482          22.9
2  2025-03-15T02:49:11.474Z      6.723291      80.094475          22.2
3  2025-03-15T02:49:12.647Z      6.723290      80.094475          22.5
4  2025-03-15T02:49:13.910Z      6.723281      80.094475          23.7

   GPS_Speed  GPS_Satellites  Accel_X  Accel_Y  Accel_Z  Gyro_X  Gyro_Y  \
0       2.07               5      160   -584.0    14364     8.0     203
1       1.19               5      292   -472.0    14436   -18.0     172
2       1.48               5      252   -476.0    14444   -34.0     163
3       1.48               5      276   -444.0    14564   -16.0     165
4       1.70               5      360   -504.0    14344   -17.0     171

   Gyro_Z  Roll  Pitch     Yaw Route_ID
0     256 -2.33  -0.64  272.35      A-B
1     305 -1.87  -1.16  275.68      A-B
2     252 -1.89  -1.00  278.09      A-B
3     282 -1.75  -1.09  280.62      A-B
4     284 -2.01  -1.44  283.36      A-B
Route_ID
A-C    410
A-B    285
Name: count, dtype: int64
Combined dataset saved as 'combined_dataset.csv'
```

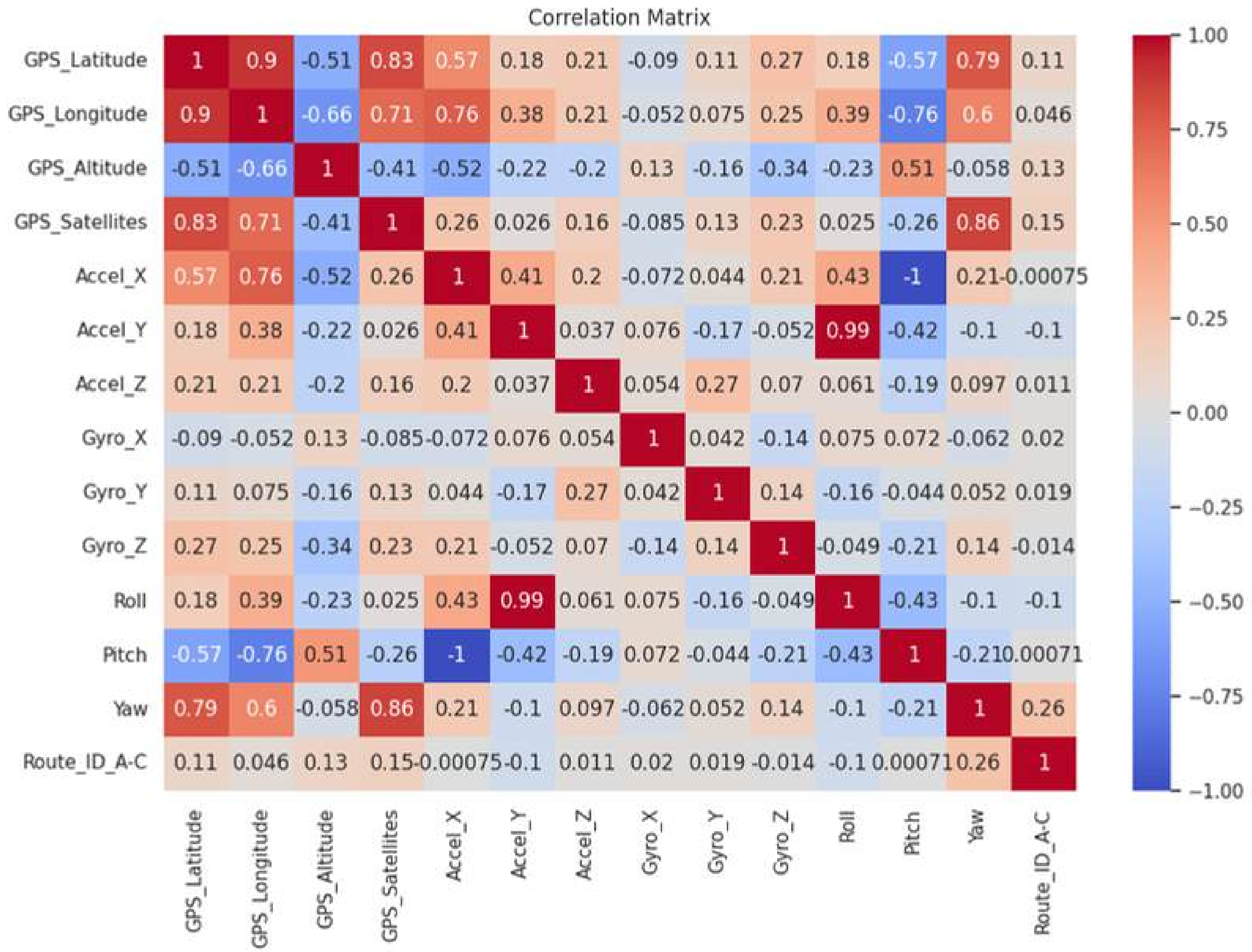
Current Progress

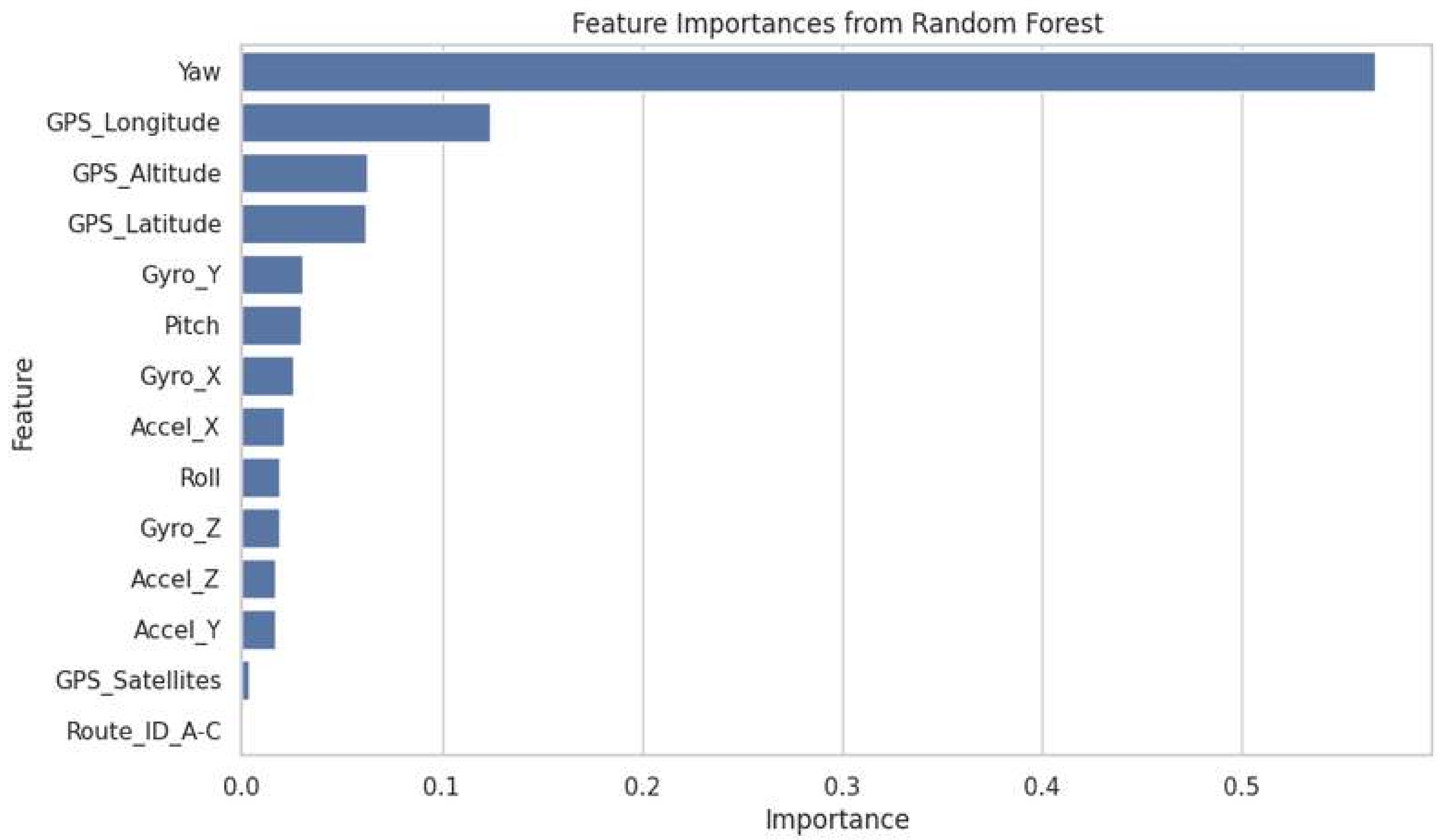
Data balancing

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
df = pd.read_csv('balanced_route_dataset.csv')

# Step 1: Check Route_ID balance
route_distribution = df['Route_ID'].value_counts()
print("Route_ID Distribution:")
print(route_distribution)

plt.figure(figsize=(8, 5))
sns.countplot(x='Route_ID', data=df, palette='viridis')
plt.title('Distribution of Route_ID')
plt.xlabel('Route_ID')
plt.ylabel('Count')
plt.show()

route_percentage = df['Route_ID'].value_counts(normalize=True) * 100
print("\nPercentage of each Route_ID:")
print(route_percentage)

# Step 2: Check GPS_Speed distribution
print("\nBasic Statistics of GPS_Speed:")
print(df['GPS_Speed'].describe())

plt.figure(figsize=(10, 6))
sns.histplot(df['GPS_Speed'], bins=30, kde=True, color='skyblue')
plt.title('Distribution of GPS_Speed')
plt.xlabel('GPS_Speed')
plt.ylabel('Frequency')
plt.show()

skewness = df['GPS_Speed'].skew()
print(f"\nSkewness of GPS_Speed: {skewness}")

plt.figure(figsize=(8, 5))
sns.boxplot(x=df['GPS_Speed'], color='lightgreen')
plt.title('Box Plot of GPS_Speed')
plt.xlabel('GPS_Speed')
plt.show()

# Step 3: Compare GPS_Speed by Route_ID
plt.figure(figsize=(10, 6))
sns.boxplot(x='Route_ID', y='GPS_Speed', data=df, palette='Set2')
plt.title('GPS_Speed Distribution by Route_ID')
plt.xlabel('Route_ID')
plt.ylabel('GPS_Speed')
plt.show()

plt.figure(figsize=(10, 6))
sns.histplot(data=df, x='GPS_Speed', hue='Route_ID', kde=True, palette='Set2', element='step')
plt.title('GPS_Speed Distribution by Route_ID')
plt.xlabel('GPS_Speed')
plt.ylabel('Frequency')
plt.show()
```

Current **Progress**

Feature extraction





Current Progress

Model training

- used KNN, Logistic Regression models to train the dataset.

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
import joblib
import time

# Load the balanced dataset
df = pd.read_csv('balanced_route_dataset.csv')

# Define features and target
selected_features = ['GPS_Latitude', 'GPS_Altitude', 'Accel_X', 'Accel_Z',
                     'Gyro_X', 'Gyro_Y', 'Roll', 'Yaw']
X = df[selected_features]
y = df['Route_ID_encoded']  # 0 ("A-B"), 1 ("A-C")

# Split the data
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

# Scale the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_val_scaled = scaler.transform(X_val)
```

```
Logistic Regression Performance:
Training Time: 0.0322 seconds
Accuracy: 0.6200

Classification Report:
              precision    recall  f1-score   support

         A-B       0.62      0.60      0.61        50
         A-C       0.62      0.64      0.63        50

    accuracy                           0.62       100
   macro avg       0.62      0.62      0.62       100
weighted avg       0.62      0.62      0.62       100


Confusion Matrix:
[[30 20]
 [18 32]]

k-Nearest Neighbors Performance:
Training Time: 0.0024 seconds
Accuracy: 0.5100

Classification Report:
              precision    recall  f1-score   support
...
k-NN Accuracy: 0.5100

Best model saved as 'logistic_regression_model.pkl'
Scaler saved as 'scaler.pkl'
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

REQUIREMENTS

Functional Requirements

- Collect & transmit telemetry (GPS, MPU6050, battery) in real time via WebSocket.
- Enable remote motor control (fwd, bwd, left, right, stop) via joystick/keyboard.
- Monitor battery & stop motors if voltage < 6.5V.
- Log telemetry & export as CSV.

Non- Functional Requirements

- Security
- Performance
- Reliability
- Usability
- Scalability

Technical Requirements

- Real-time data processing & visualization(JSON< 400 chars).
- Use C++ (Arduino/ESP8266), HTML/JS (dashboard), libraries: TinyGPSPlus,MPU6050, ESP8266WiFi,WebSocketsServer, nipplejs

TOOLS & TECHNOLOGIES

Technologies

- TensorFlow
- Python
- Sklearn
- Jupyter
- C++

Architectures & Algorithms

- Random forest
- KNN
- XGBoost
- SVM

Techniques

- Real-Time Data Transmission: JSON-formatted GPS and motion data via WebSocket.
- Spoofing Detection Model: Machine learning/statistical model to analyze GPS deviations and detect irregular patterns.
- Motor Control: Command-based control (e.g., FWD, BWD) with speed adjustment.
- Orientation Calculation: Roll, pitch, yaw from MPU6050 data.
- Battery Monitoring: Voltage thresholds (stop if < 6.5V).
- Data Logging & Analysis: Telemetry logged locally, exportable as CSV for model training/validation.

References

- Yang, Zhen, et al. “Anomaly Detection Against GPS Spoofing Attacks on Connected and Autonomous Vehicles Using Learning From Demonstration.” IEEE Transactions on Intelligent Transportation Systems (2023).

- Manesh, Mohsen Riahi, et al. “Detection of GPS spoofing attacks on unmanned aerial systems.” 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC). IEEE, 2019..

- D. G. Yang et al., “Intelligent and connected vehicles: Current status and future perspectives,” Sci. China-Technol. Sci., vol. 61, no. 10, pp. 1446–1471, Oct. 2018.

Thank you