A Mini Project Report on

DIAL DETECTIVE

Submitted in partial fulfillment for the degree of Bachelor of Technology in Computer Science and Technology

Submitted by

Madhuri Patil (45)

Durva Patkar (47)

Vaishnavi Rawate (53)

Under the guidance of

Prof. Sumedh Pundkar

USHA MITTAL INSTITUTE OF TECHNOLOGY

S.N.D.T WOMEN'S UNIVERSITY

MUMBAI – 400049

2024 – 2025

CERTIFICATE

This is to certify that **Madhuri Patil,Durva Patkar,Vaishnavi Rawate** has successfully completed the Mini-project phase 2 work on **DialDetective: Phone Number Details Retrieval System** in partial fulfillment for the Bachelor's Degree in **Computer Sciene and Technology** during the year 2024-25 as prescribed by SNDT Women's University.

GUIDE
Prof. SUMEDH PUNDKAR

HEAD OF THE DEPARTMENT
Prof. KUMUD WASNIK

PRINCIPAL
Dr. YOGESH NERKAR

EXAMINER 1

EXAMINER 2

ACKNOWLEDGEMENT

We would like to express our sincere appreciation to everyone who gave their invaluable assistance and support in this project. We are highly indebted to Prof. Sumedh Pundkar for his guidance and constant supervision as well as for providing necessary information regarding the project and for his support in this project. Without his help, this project would not have been possible.

We would also like to thank our Head of Department, Prof. Kumud Wasnik, for her insights in shaping the direction and content of this report. We would also like to extend our gratitude to our classmates who provided valuable feedback and encouragement throughout the process.

Finally, we would like to express our appreciation to our families for their unwavering support and understanding during the long hours and late nights spent working on this project.

Thank you all for your contributions, support, and encouragement.

Contents

List of Figures

1. Abstract

Dial Detective is an advanced system designed for phone number validation and data extraction, offering essential details such as location, network provider, time zone, and spam detection. It plays a crucial role in fraud detection, security, and verification, helping organizations and individuals identify potential threats, prevent scams, and ensure reliable communication.

The project Dial Detective utilizes Python's powerful phonenumbers library to validate, parse, and extract comprehensive details from phone numbers. By integrating advanced algorithms, it provides key insights such as:

- **Location**: Identifies the country, state, or city associated with the number.
- **Network Information**: Determines the carrier or service provider.
- **Time Zone**: Retrieves the local time based on the number's region.
- **Spam Detection**: Incorporates AI-based classification to detect potential spam or fraudulent numbers.

Dial Detective is a fast, reliable, and secure solution that simplifies phone number analysis, making it accessible for businesses, law enforcement, and cybersecurity applications.

2. Introduction

Dial Detective is an advanced phone number validation and data extraction system designed to enhance fraud detection, security, and verification. It provides essential details such as location, network provider, time zone, and spam detection, helping individuals and organizations identify potential threats and prevent scams. The system utilizes Python's **phonenumbers** library to extract phone details and applies machine learning techniques like **K-Nearest Neighbors (KNN)** and to classify numbers as spam or legitimate.

With the increasing number of fraudulent calls and scams, Dial Detective serves as a crucial tool for ensuring safe and reliable communication. It efficiently processes user input, validates numbers, and provides accurate insights while addressing challenges like privacy concerns and data accuracy. Future enhancements include improved detection accuracy, a mobile application version, and more advanced machine learning models for better fraud prevention.

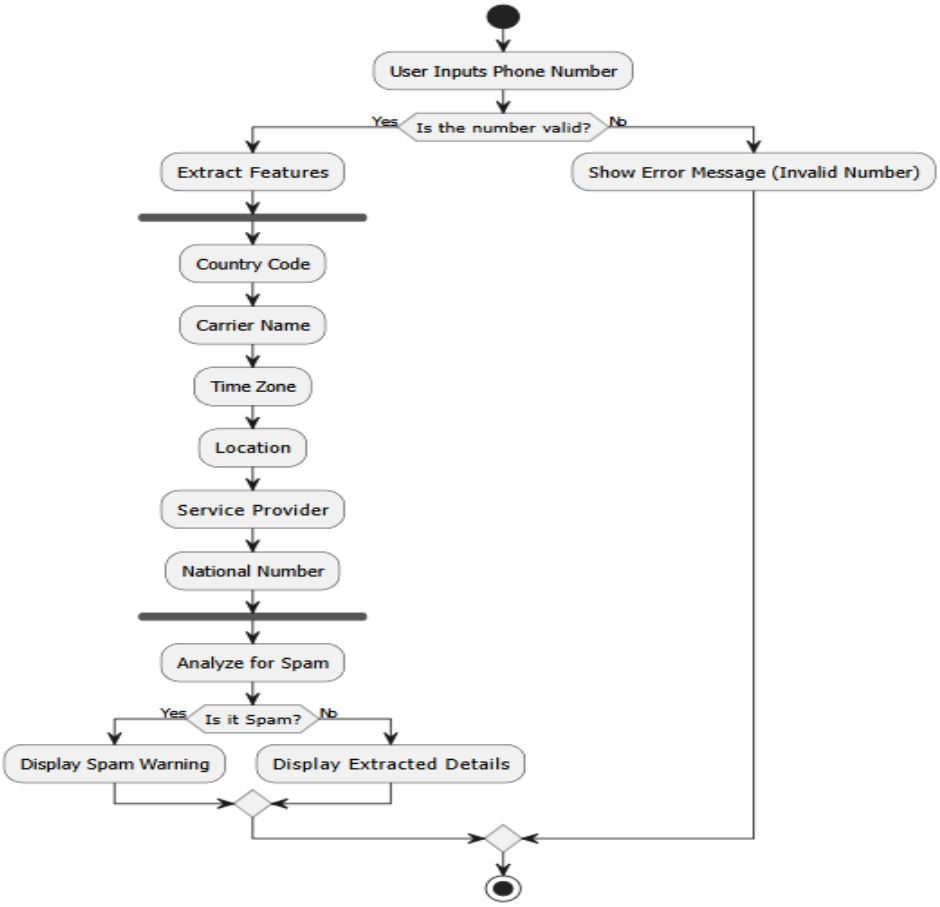

Figure 2.1: Flowchart

3. Problem Statement

In today's digital era, the increasing number of fraudulent calls, spam messages, and scam attempts pose a significant threat to individuals and organizations. Cybercriminals exploit phone numbers for phishing, identity theft, financial fraud, and unauthorized access, making it difficult to distinguish between legitimate and malicious calls.Existing phone number verification systems often lack accuracy, privacy concerns.

To address these challenges, Dial Detective provides an intelligent solution that validates phone numbers, extracts essential details (such as location, network provider, and time zone), and detects spam using machine learning algorithms (KNN). The system enhances security, prevents scams, and ensures trustworthy communication while continuously improving its accuracy through real-time data updates and advanced AI-based filtering.

4. Literature Survey

4.1 Technical Papers

A literature survey, also known as a literature review, involves analyzing and synthesizing previously published research to identify knowledge gaps, assess current advancements, and establish a foundation for further development. It plays a crucial role in shaping the direction of a project by providing insights into existing solutions and methodologies.

During the research phase of Dial Detective, we explored various research papers and articles related to phone number validation, fraud detection, and spam classification. After careful evaluation, we selected three key papers that contributed significantly to the development of this system. Among these, Research Paper No. 3 served as the foundation of our project, as it utilized machine learning techniques such as K-Nearest Neighbors (KNN) and for spam classification.

These research papers provided valuable insights into phone number tracking, machine learning-based spam detection, and validation techniques. They helped in addressing challenges such as privacy concerns, real-time data accuracy, and the need for an efficient number verification system. By leveraging the knowledge from these studies, we designed the architecture of Dial Detective, implemented AI-driven spam detection, and enhanced the system's accuracy.

Author(s) & Year	Abstract	Techniques	Technology	Limitations	Conclusion
Tejas Kulkarni, Aditya Siddhu, Durgesh Chauhan, Bharati Dhomne, Abhishek Patil (2022)	The Phone Locator system tracks mobile numbers, providing location, network details, and regional information while ensuring privacy. It operates nationwide and identifies active numbers or complaints.	• Network Tracking – Identifies provider & location. • Address Tracking – Estimates registered address. • KNN Algorithm – Finds closest match. • Real-time Retrieval – Updates tracking data.	• OpenCage API – Converts location to coordinates. • Phonenumbers – Validates phone details. • Flask – Web interface. • Google Geocode API – Improves location accuracy.	• Signal Issues – Affects accuracy. • High Battery Use – GPS drains power. • Internet Needed – For updates. • Public Data Dependent – Limits accuracy.	Efficient mobile tracking without costly GPS devices. Relies on GPS, internet, and network. Future focus: better accuracy, lower power use.
Satvik Rana , Ayush (2023)	Mobile tracking supports security and navigation. This project explores GPS, triangulation, and Wi-Fi, addressing privacy and accuracy challenges. AI can improve future precision.	• GPS – Satellite-based tracking. • Triangulation – Uses network towers. • Wi-Fi – Enhances accuracy. • IP Geolocation – Tracks via internet. • Hybrid – Combines methods.	• GPS/GNSS – Outdoor tracking. • Cellular – Uses tower signals. • Wi-Fi/Bluetooth – Indoor tracking. • AI/ML – Enhances accuracy. • Big Data – Processes location data.	• Accuracy – Affected by signals. • Privacy – Risk of unauthorized access. • Legal – Strict tracking regulations. • Connectivity – Requires GPS/internet. • Battery – High power usage.	Mobile tracking aids security but has accuracy and privacy issues. AI and data security will improve efficiency.
Allison McDonald, Tamy Guberek, Carlo Sugatan, Florian Schaub (2021)	Phone numbers enable authentication but risk privacy and fraud. This study examines threats and security solutions.	• Validation – Confirms active numbers. • Encryption – Secures data. • MFA – Adds OTP security. • Fraud Detection – Identifies threats.	• AI/ML – Detects fraud. • Blockchain – Prevents unauthorized access. • Authentication – Secures accounts with OTPs. • Mobile Data/GPS – Verifies ownership & location.	• Privacy – Spam & tracking. • Security – Fraud risk. • Recycling – Data leaks. • Network – Needs stable connection.	Phone numbers pose risks; AI & encryption improve security.

Figure 4.1: Literature Survey

5. Existing System

5.1 Existing System: Truecaller

The current system for phone number validation and spam detection is largely represented by applications like **Truecaller**. Truecaller is a widely used caller identification service that allows users to **identify callers, fetch spam reports, and search for phone numbers**. However, the system has notable limitations in terms of **accuracy, privacy, and dependency on the internet**.

5.1.1 Components of the Existing System (Truecaller)

- **Truecaller Module**
  - Provides two main functionalities:
    - **identifyCaller(phoneNumber: PhoneNumber): String** – Identifies the caller's name.
    - **fetchSpamReports(phoneNumber: PhoneNumber): Integer** – Retrieves the number of spam reports for a phone number.
- **User Database**
  - Maintains a list of stored numbers and spam reports.
  - Functions:
    - **storedNumbers: List PhoneNumber** – Stores previously identified numbers.
    - **spamReports: Map PhoneNumber, Integer** – Maps phone numbers to their spam report count.
    - **searchNumber(phoneNumber: PhoneNumber): String** – Searches for details of a given phone number.
- **PhoneNumber Class**
  - Stores essential phone number attributes:
    - **number: String** – Actual phone number.
    - **countryCode: String** – Country code of the number.
    - **nationalNumber: String** – Localized phone number format.
    - **networkProvider: String** – Carrier information.
- **Limitations of the Existing System**

– **Accuracy Issues** – Spam detection is based on **user reports**, which may lead to false positives or negatives.

– **Privacy Concerns** – Truecaller stores **user data in its database**, raising potential privacy risks.

– **Requires Internet** – The system relies on **internet connectivity** to fetch caller details and spam reports.

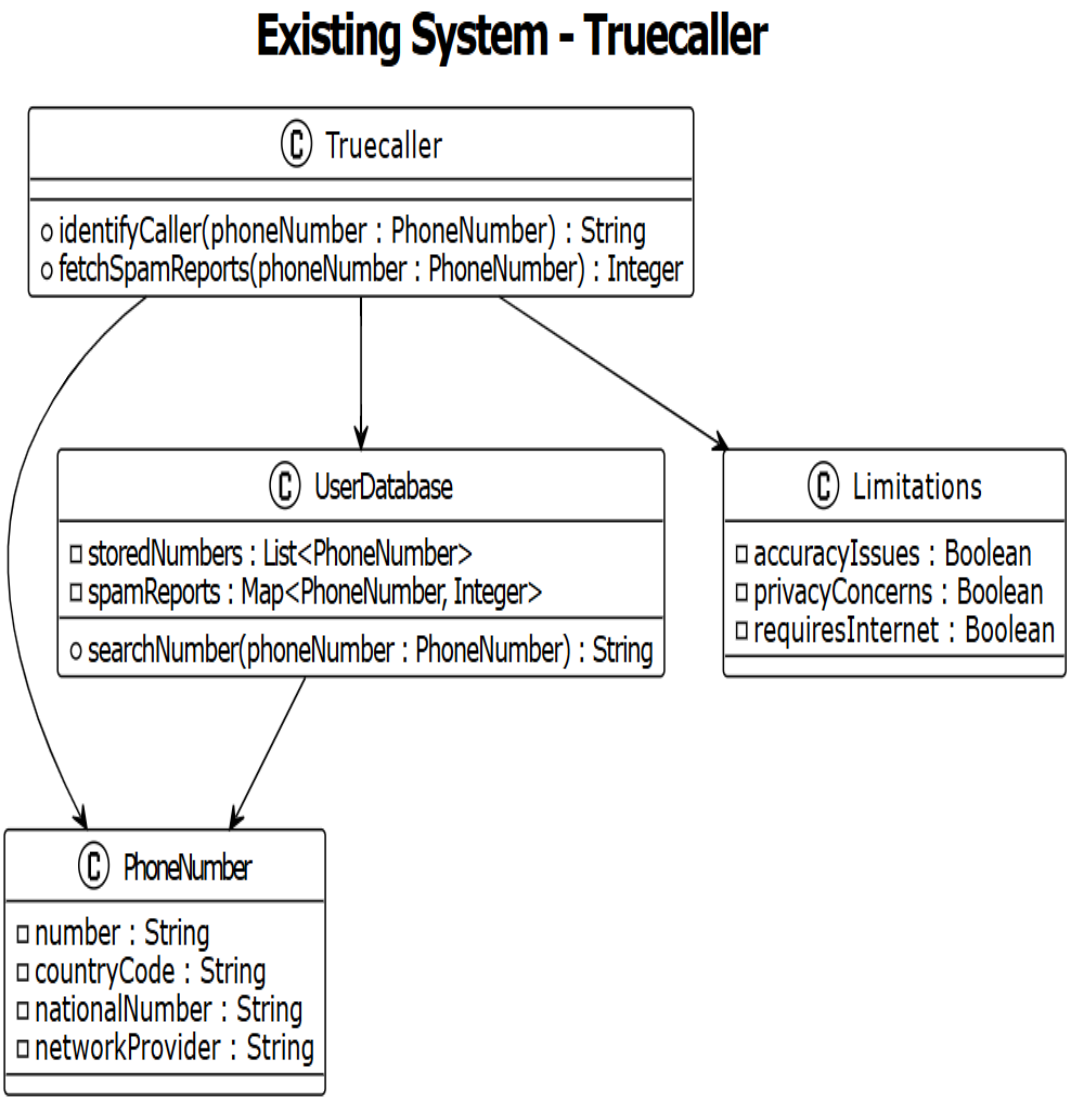

Figure 5.1: Existing System: Truecaller Architecture

6. Proposed System

6.1 Proposed System - Dial Detective

The **Dial Detective System** is designed to analyze phone numbers, validate them, and provide geolocation-based insights. The system consists of multiple interconnected components, each responsible for specific tasks.

6.1.1 Key Components

- **User Interface (UI)**
  - Takes user input (phone number).
  - Displays the analysis result.
- **Dial Detective System**
  - Main logic component of the system.
  - Receives the input from UI and processes the phone number.
  - Generates an **AnalysisResult** using:
    - **PhoneValidator**
    - **PhoneGeoLocator**
    - **SpamDetector**
- **Phone Validator**
  - Checks if the given phone number is valid.
  - Returns a **Boolean** value indicating validity.
- **Phone Geolocator**
  - Fetches the geographical location of the phone number.
  - Returns details such as **city, region, and country**.
- **Phone Number**
  - Stores details of the phone number:
    - **Number**
    - **Country Code**
    - **Network Provider**

- **Location**
  - Stores geolocation data of the phone number:
    - **City**
    - **Region**
    - **Country**
- **Analysis Result**
  - Generated after phone number analysis.
  - Contains:
    - **isValid:** Whether the number is valid or not.
    - **Location:** Geographical data of the phone number.
    - **Risk Level:** Indicates if the number is suspicious or safe.
    - **spamProbability:** Checks if the number is spam or not spam.
- **SpamDetector**
  - Identifies potential spam or fraudulent numbers.

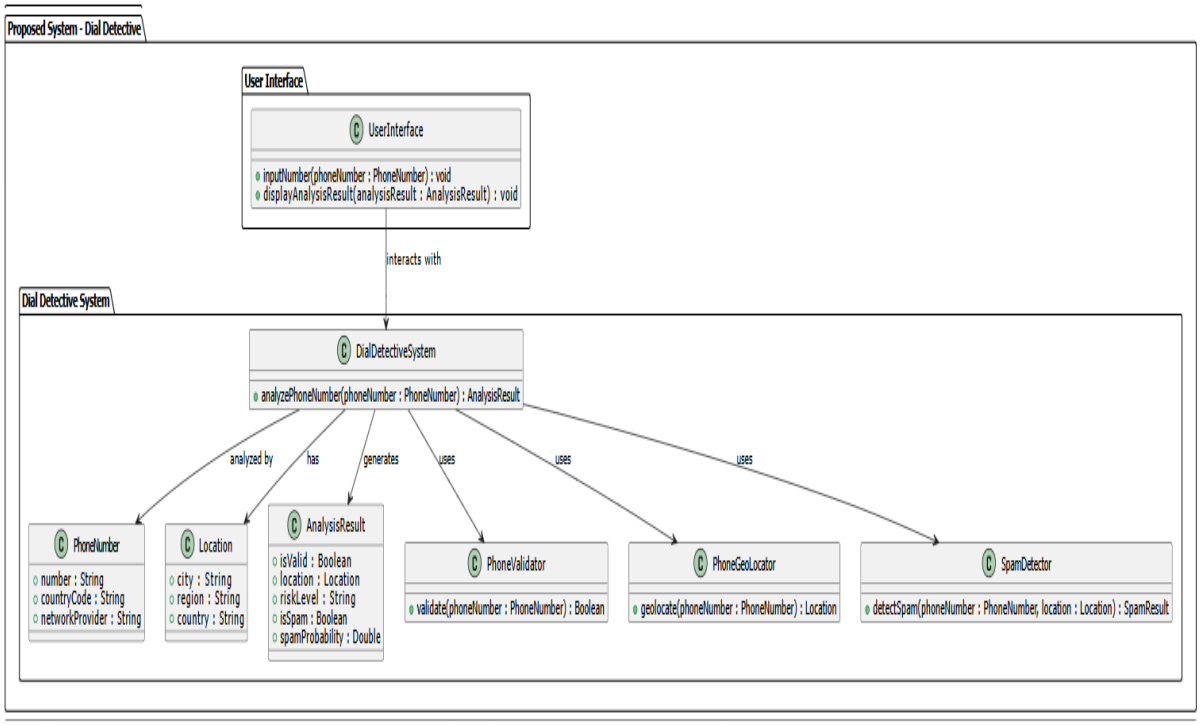

Figure 6.1: Proposed System - Dial Detective

This **proposed system** improves upon existing solutions (like Truecaller) by providing **better validation, geolocation tracking, and risk assessment** for phone numbers.

7. Architectural Overview

The **Dial Detective System** is structured into multiple interconnected modules that process a phone number entered by the user and extract relevant details. The architecture consists of the following components:

7.1 System Components

7.1.1 User Device

- The user interacts with the system through a **User Interface**.
- The user inputs a phone number for analysis.

7.1.2 Dial Detective System

The core system is divided into several modules:

Validation Module

- Validates the entered phone number.
- If the phone number is invalid, an **error is shown**, and the process is terminated.

Processing Engine

- If the number is valid, the **Processing Engine** extracts the phone number details.

Feature Extraction Module

- Extracts specific details about the phone number.
- Sends the extracted information for further analysis.

Spam Detection Module

- Analyzes the extracted information to detect potential spam or fraud.
- Determines the spam risk level based on known spam patterns and indicators.

Output Module

- Displays the extracted details or an error message (if the number is invalid).
- Also shows the result of the spam detection (e.g., **Spam**, **Not spam**).

7.2 Feature Extraction Details

The extracted phone number details include:

- **Country Code**: The country associated with the phone number.
- **Carrier Name**: The network provider of the phone number.
- **Time Zone**: The time zone of the phone number's location.
- **Location Data**: The geographical location of the phone number.
- **Service Provider**: The telecom service provider.
- **National Number**: The local/national version of the phone number.

7.3 Architecture Diagram

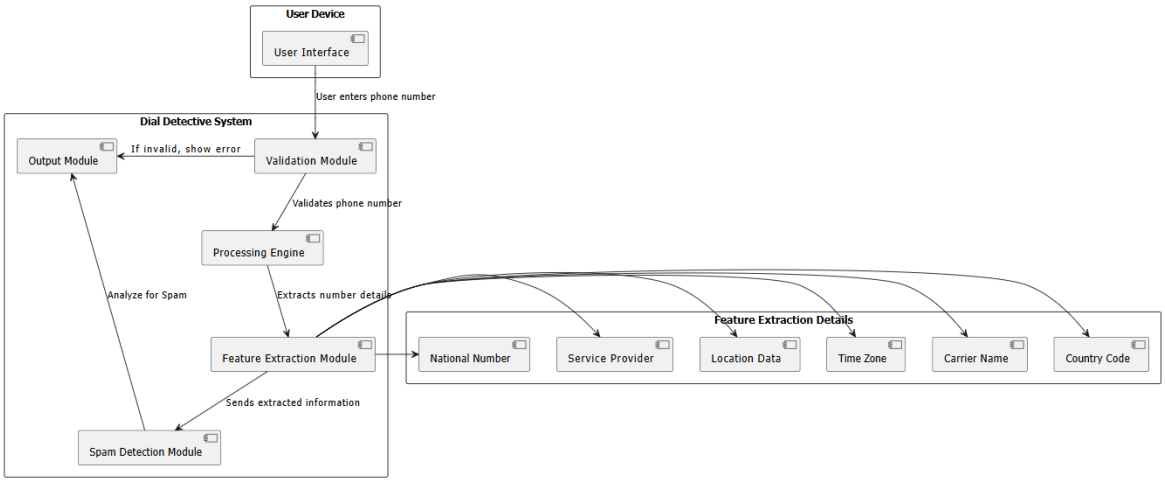

Figure 7.1: Dial Detective System - Architecture Overview

This architecture ensures a **structured approach** to phone number validation and feature extraction, making the **Dial Detective System** efficient and reliable.

8. Hardware and Software Requirements

8.1 Hardware Requirements

- **System Processor**: Intel Core i3 (or equivalent)
- **RAM**: Minimum 64 MB primary memory
- **Storage**: 10 GB free space
- **Internet Connection**: Required for API access

8.2 Software Requirements

- **Operating System**: Windows 7 or above
- **Programming Language**: Python
- **Python Libraries**: Phonenumbers
- **Frontend**: HTML, CSS
- **Tool**: Jupyter Notebook, VS Code

9. Implementation Details

The implementation of the **Dial Detective** system involves various modules that work together to retrieve and analyze phone number details. The system is developed using **Python** and utilizes the **Phonenumbers** library for extracting information. The following are the key components of the implementation:

1. **User Interface (Frontend):** The DialDetective system features an interactive frontend where the user inputs a phone number to trace its details.

    - **Input Form :**Users can enter a phone number in the international format (e.g., +91 for India) through the web interface. Once submitted, the number is sent to the backend for processing.
    - **Brand Header and Tagline :**The page displays the system's logo and tagline, e.g., "Find the location of any phone by its number."
    - **Features:**Users can access buttons like **Help**, **Settings**, and **Report**. The results display the number's location, service provider, and the spam status.
    - **Technologies Used:**
        - HTML, CSS, JavaScript
        - Flask (for communication with the backend)
        - Bootstrap (optional for styling)

2. **Model Training (KNN Classifier):** The core of the DialDetective system uses the K-Nearest Neighbors (KNN) algorithm to classify phone numbers as either spam or non-spam based on extracted numerical features.

    - **Dataset:**The system uses a CSV file with labeled numbers, containing features like:
        - Sum of digits
        - Unique digit count
        - Frequency of the first digit
    - **Features Extraction:** For each phone number, it extracts:
        - Country code
        - Carrier name
        - Number format (e.g., national, international)
        - Computed features like sum of digits and frequency-based metrics
    - **Model Training:** The KNN model is trained with scikit-learn using the extracted features and is saved with the joblib library for later use.

- **Technologies Used:**
  - Python
  - Pandas, NumPy
  - Scikit-learn (for training the KNN classifier)
  - Joblib (for saving the trained model)

3. **Main Program Execution (Backend with Flask) :**The backend of the DialDetective system is implemented using Flask, which handles the processing of phone numbers, including validation, feature extraction, and spam detection.

   - **Input Handling:**
     - The backend accepts the phone number via a POST request in JSON format. The phone number is extracted and cleaned of any spaces.
     - The system ensures the phone number is in the correct international format by checking for a leading + symbol.
   - **Phone Number Parsing and Feature Extraction**
     - The `phonenumbers` library parses the phone number, extracting details like location, service provider, carrier, and time zone.
       Feature extraction is performed on the national number:
       * **Sum of digits**: The sum of all digits in the phone number.
       * **Unique digit count**: The number of distinct digits in the phone number.
       * **First digit frequency**: The frequency of the first digit in the number, normalized by the total length of the number.
   - **Spam Detection**
     - If the phone number is listed as "not spam" in (spam_dataset.csv), the predefined dataset the system bypasses spam detection and returns a "not_spam" status.
     - Otherwise, the extracted features are passed through the pre-trained KNN model to predict whether the number is "spam" or "not_spam". The KNN classifier was trained on a dataset containing features like the sum of digits, unique digit count, and first digit frequency.
   - **Output Response :** The backend returns a JSON response containing:
     - **Country Code**: The international dialing code of the number.
     - **National Number**: The phone number without the country code.
     - **Timezone**: The timezone associated with the phone number.
     - **Location**: The geographical location of the phone number.

- **Service Provider**: The carrier of the phone number.
- **Spam Status**: A prediction of whether the phone number is "spam" or "not_spam".

If the phone number is invalid or there's any error in processing, appropriate error messages are returned.

- **Technologies Used**
  - Python (Flask for backend processing)
  - Phonenumbers (for parsing and extracting phone number details)
  - Pandas, NumPy (for dataset manipulation and feature extraction)
  - Scikit-learn (for KNN model training and inference)
  - LabelEncoder (for encoding labels in the dataset)

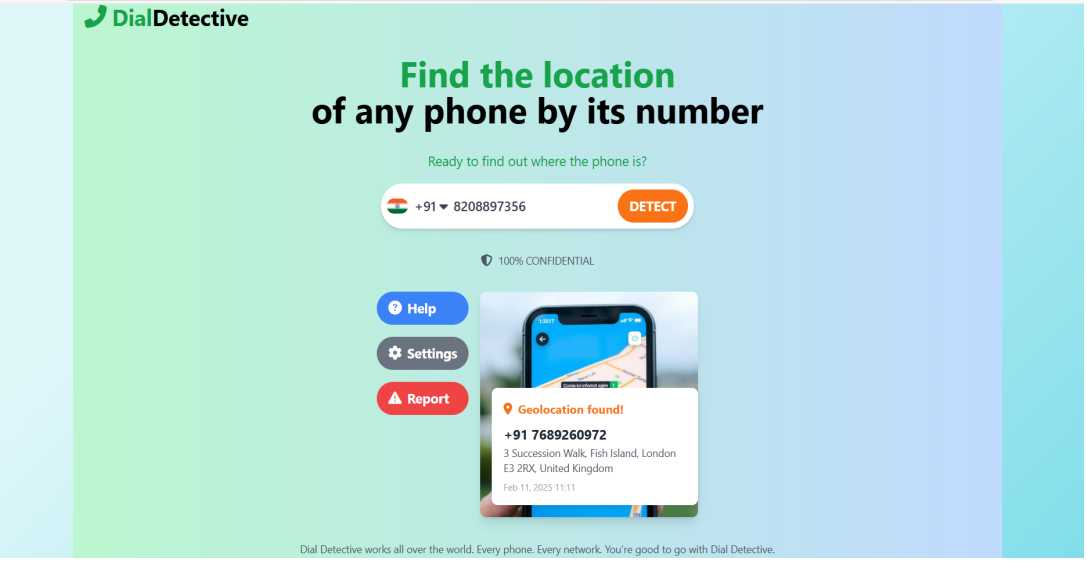

Figure 9.1: User Interface

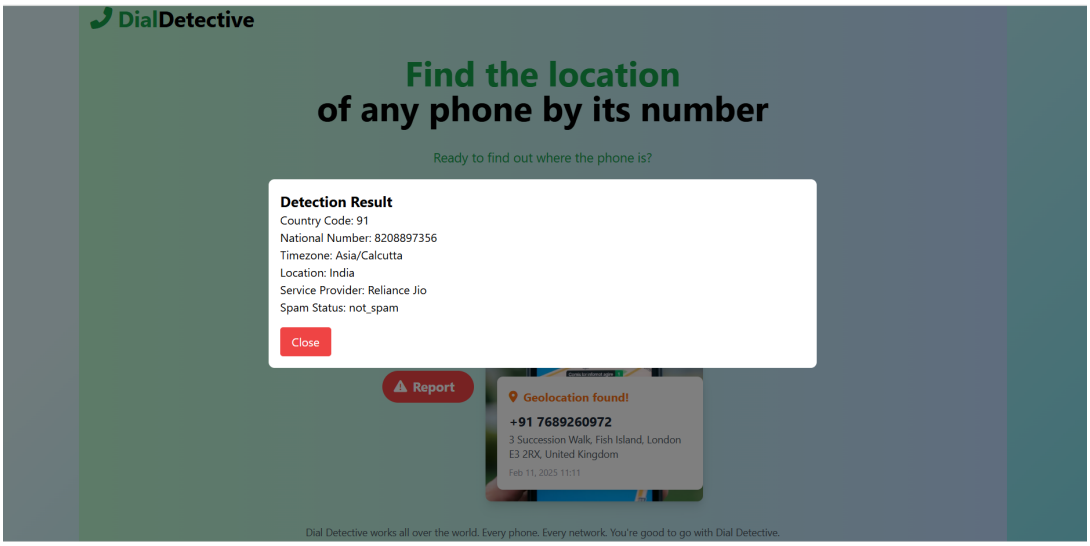

Figure 9.2: Detection Result 1

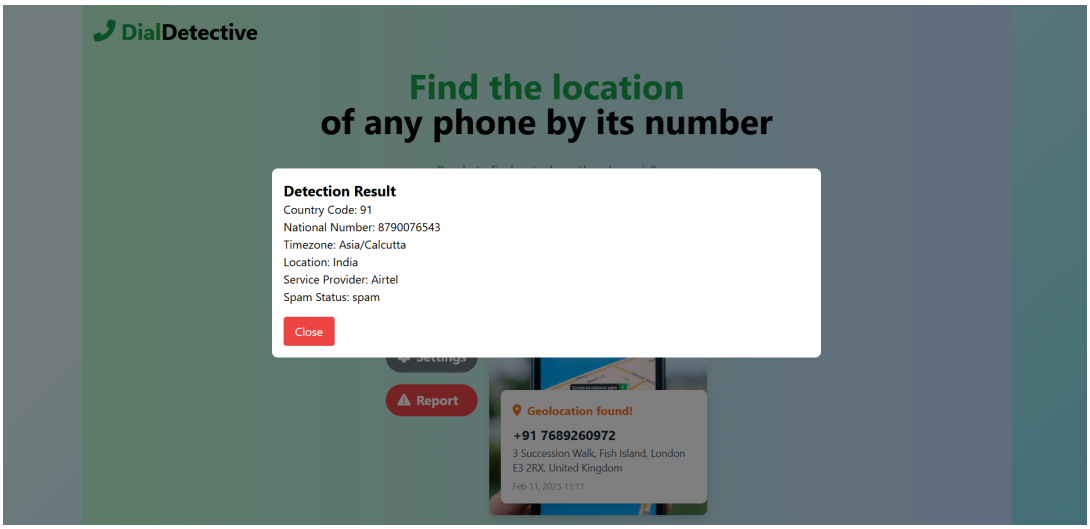

Figure 9.3: Detection Result 2

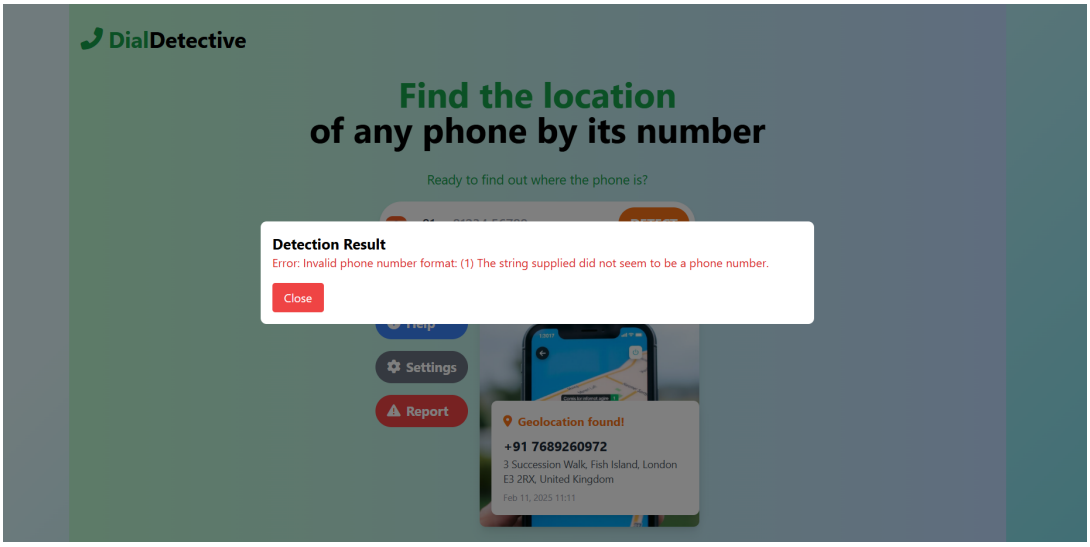

Figure 9.4: Detection Result 3

10. Applications

(a) **Detect Spam Calls**

- Detects and warns about spam or fraud calls.
- Helps users avoid telemarketing and scam calls.

(b) **Checking Phone Number Validity**

- Verifies if a phone number is real and active.
- Ensures accurate phone number details for communication.

(c) **Finding Caller Location**

- Identifies the country and region of the phone number.
- Helps users recognize international or unknown calls.

(d) **Helping Businesses Verify Customers**

- Ensures that customers provide real phone numbers.
- Reduces fake registrations and improves security.

(e) **Supporting Police Investigations**

- Can be used to track scam or fraud numbers.
- Assists in forensic analysis of suspicious callers.

(f) **Detecting Fake or Spoofed Numbers**

- Identifies whether a number is being used for fraudulent purposes.
- Prevents identity theft and impersonation scams.

(g) **Enhancing Online Registrations**

- Helps websites and apps verify phone numbers during sign-ups.
- Reduces the risk of bots and fake accounts.

11. Conclusion and Future Scope

11.1 Future Scope of Dial Detective

(a) **Improved Spam Detection**

- Implement advanced machine learning algorithms to enhance spam number detection.
- Use deep learning for better accuracy in fraud detection.

(b) **Real-Time Caller Identification**

- Develop a system to analyze incoming calls and provide real-time alerts for spam or fraud.
- Improve response time for detecting suspicious numbers.

(c) **Global Phone Number Validation**

- Expand the system to support international number validation.
- Integrate with global telecom databases for real-time information.

(d) **Offline Mode**

- Develop an offline version that can validate and detect spam numbers without an internet connection.
- Store essential data locally for quick access.

(e) **Enhanced Privacy and Security**

- Implement end-to-end encryption for protecting user data.
- Ensure GDPR and cybersecurity compliance for data safety.

(f) **Voice and Call Pattern Analysis**

- Introduce AI-based voice recognition to detect robocalls and scam attempts.
- Analyze call patterns to identify suspicious behavior.

(g) **Automated Reporting and Community Feedback**

- Allow users to report spam numbers, improving detection accuracy.
- Build a crowdsourced database for real-time updates on fraud numbers.

11.2 Conclusion

The **Dial Detective** system provides an efficient and reliable solution for phone number validation, spam detection, and fraud prevention. By utilizing **Python's Phonenumbers**

library along with advanced **machine learning techniques**, the system ensures accurate identification of phone numbers, their carrier details, location, and risk level.

The implementation of **real-time validation, feature extraction, and automated spam classification** makes **Dial Detective** a valuable tool for individuals, businesses, and law enforcement agencies. Additionally, the system enhances **security and privacy** by reducing exposure to spam calls, phishing attempts, and fraudulent activities.

In the future, **Dial Detective** can be improved by incorporating **deep learning models, real-time caller monitoring, mobile app development, and global database integration**. These enhancements will further increase its accuracy and usability, making it a **powerful tool for secure and efficient communication**.

This project successfully demonstrates the **importance of phone number intelligence** in today's digital world and lays the foundation for future advancements in **telecom security and fraud prevention**.

12. References

(a) IRJMETS, “Phone Number Tracking Using Python,” *International Research Journal of Modernization in Engineering, Technology and Science*, vol. 6, IRJMETS Publications, 2022.

(b) Quest Journals, “Phone Number Tracking System Using Python,” *Journal of Software Engineering and Simulation*, vol. 9, no. 5, Quest Journals, 2022.

(c) A. McDonald, “Phonenumbers: Parsing, Formatting, and Validating International Phone Numbers,” *ACM CHI Conference on Human Factors in Computing Systems*, ACM Publications, 2021.

(d) Scribd, “Phone Tracking Using Python,” *Scribd Digital Library*, Scribd Publications, 2022.