IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⸎ Peer-reviewed & Refereed journal ⸎ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15427

An Intelligent AI-Based Vehicle Breakdown Assistance System with Network-Aware Mobile Deployment

KARTHIKEYAN M[1], Dr C. KARPAGAVALLI[2], Dr E. MARIAPPAN[3], Dr M. KALIAPPAN[4]

Student, Artificial Intelligence and Data Science,

Ramco Institute of Technology, Rajapalayam, Tamil Nadu, India[1]

Assistant Professor, Artificial Intelligence and Data Science,

Ramco Institute of Technology, Rajapalayam, Tamil Nadu, India[2]

Associate Professor, Artificial Intelligence and Data Science,

Ramco Institute of Technology, Rajapalayam, Tamil Nadu, India[3]

Professor, Artificial Intelligence and Data Science,

Ramco Institute of Technology, Rajapalayam, Tamil Nadu, India[4]

Abstract: When a vehicle breaks down in the wilderness, finding a reliable connection to get help or assistance is often difficult or sporadic. For this reason, this study proposes an AI-based intelligent assistant, IntelliFix, to help users with broken-down vehicles through a Progressive Web Application (PWA)-based mobile system. The classification and semantic matching functions of the app are based on a transformer-based NLP. This enables accurate interpretation of the user's breakdown by gathering both data and integrating the user's account within the context of the app, regardless of whether or not the user was connected to the network. An additional aspect of the proposed Progressive Web Application (PWA) is its use of GPS and OpenStreetMap to geocode the location of broken-down vehicles and rank them in proximity to nearby repair shops. The application will enable users to report repairs via WhatsApp as well as make sequential voice calls to various mechanics. The classification process produced an accuracy of 91.05%, and the semantic matching process had an accuracy of 99.47%. In summary, the breakdown assistance system presents an increase in mobilization and efficiency of providing intelligent roadside assistance.

Keywords: Vehicle Breakdown Assistance, Artificial Intelligence, Natural Language Processing, Progressive Web Application, Network-Aware Systems, Semantic Matching, OpenStreetMap, Intelligent Transportation Systems

I. INTRODUCTION

Road transport expansion has increased vehicle breakdown across all areas. Prompt assistance will help for safety, traffic and user convenience. Conventional systems for breakdown assistance are reliant on manual communication as they depend heavily on availability of network; this is difficult to successfully achieve, particularly in rural and highway regions due to their unpredictable connectivity. Advancements in artificial intelligence (AI) and mobile computing now provide a good opportunity to create intelligent assistance systems.

These systems can interpret breakdown situations and provide correspondingly appropriate assistance to users; this would improve response times in the event of vehicle breakdown. ASIS could also potentially reduce traffic issues caused by breakdowns, increase road safety and ultimately improve user's reliability on vehicles and breakdown assistance. Nonetheless, the majority of current support solutions cannot provide support without an internet connection, do not understand natural language well, and do not communicate with high reliability. Therefore, despite recent improvements in intelligent support systems, there is still a gap in research concerning a cohesive vehicle breakdown assistance framework that can properly evaluate unstructured natural language breakdown issues and perform efficiently whilst utilizing unreliable network connections. The majority of current vehicle breakdown assistance systems treat the identification of problems, filtering of location, and coordination of servicing as three distinct components and do not join them together for optimal support delivery. As a result, current vehicle breakdown assistance systems are highly

DOI: 10.17148/IJARCCE.2026.15427

reliant on matching by keyword and continual access to the internet, making their use impractical, in real-time on highways and rural roadways. Therefore, it is important that an AI-based, network-aware, and scalable vehicle breakdown assistance system be created as soon as possible.

In this paper, we present IntelliFix—an intelligent breakdown assistance service for vehicles. The goal of IntelliFix is to provide seamless service in both offline and online contexts. We accomplish this through AI transformer models, semantic similarity ranking, GPS proximity filtering, and automated communication.

It is built as a Progressive Web Application (PWA), giving it app-like qualities, offline support, and performance benefits across devices without requiring native install.
This paper has five main contributions:

- Vehicle issue classification through transformer model-based AI and DistilBERT.
- A semantic similarity-based vehicle mechanic recommendation model created from MiniLM sentence embeddings.
- A ranking model that fuses contextual and semantic relevance and geographic proximity through cosine similarity and Haversine distance.
- An online/offline switching mechanism that is network-aware.
- An ONNX model that is used for client-side semantic inference execution.

II. RELATED WORKS

Multiple studies on different aspects of ITS research into vehicle tracking and mobility analysis have been published. One study focused on how GPS-based vehicle tracking systems utilizing IoT technologies are improving vehicle tracking and transportation systems [5]. Another study analyzed human mobility patterns using large-scale mobility data with the goal of improving ITS [7] through advanced vehicle management systems and optimization techniques for vehicle trajectories for transportation under uncertain road conditions [6].

Context-aware computing has also contributed to making adaptive intelligent systems (through incorporating situational awareness into decision-making processes) more advanced than they would have been without context-aware computing [8]. Further, mobile cloud computing frameworks have provided the necessary scalable and reliable platform to deploy intelligent mobile systems in a cloud environment that supports real-time synchronization and service coordination within a cloud environment [9]. Last, edge computing paradigms have been suggested to overcome latency issues and reliance on networks when using adaptive system behaviors when connectivity is intermittent [34].

Intelligent fault detection and diagnosis systems have evolved from rule-based to data-driven methods over the years. Many existing machine learning approaches have been used to assist with classification and prediction problems through traditional machine learning techniques (Decision Trees, Support Vector Machines, Logistic Regression, and Neural Networks) [26], [27]. In particular, data-driven fault diagnosis approaches have recently emphasized the importance of developing strong and flexible fault diagnostic systems to deal with complex situations that commonly occur in the process industries [11].

The use of transformer models within language models has significantly improved the performance of intelligent systems in the area of natural language understanding. For example, BERT has improved performance on both contextual language modeling tasks and text classification tasks [21]. A new lightweight architecture based on DistilBERT has been proposed, which demonstrates similar performance as compared to heavyweight architectures while reducing compute complexity [22]. Sentence embedding methods (e.g., Sentence-BERT [23] and Minilm [24]) have provided significant improvements in computing semantic similarity between sentences, allowing for better context-aware matching of services in a recommendation system.

Models for emergency service distribution and location-based administration have been created with the intent to create the most efficient response time and allocation of services when factoring in changing travel conditions based on time [13]. Additional location-aware recommendation system models that incorporate contextual and location-related information into the conventional collaborative filtering method [18], [19] were developed in order to produce the most applicable service possible to each customer [14],[15]. However, most current mobile automotive application service delivery systems heavily depend on internet connectivity for real-time processing and data communication.

The proposed IntelliFix System leverages what has already been researched and developed by integrating contextual interpretation of language requirements, semantic ranking for service delivery, geospatial filtering, mobile cloud

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.15427

computing technology, and dynamic decision support based on the characteristics of the mobile network all within one mobile computing platform.

III. BACKGROUND

Vehicle-monitoring capabilities as well as roadside assistance have been dramatically improved with the advancements of mobile computing and intelligent transport systems (ITS). The collection of large quantities of vehicle-related data has been made possible by urban computing and location-aware systems and includes vehicle-related geospatial data, service interaction data, and vehicle breakdown reports created by users [16]. While a lot of progress has been made, a major obstacle to developing a better understanding of meaningful information from the heterogeneous and unstructured data associated with vehicle breakdowns will require the combination of mobile computing systems capable of functioning effectively under the constraints of the real world, machine learning methodologies, and geospatial intelligence.

A. Challenges in Vehicle Data and Breakdown Assistance

Vehicle breakdown data presents many challenges to be analyzed due to the large variety of mechanical and electrical fault types, as well as the diverse driving conditions of the vehicles when a fault occurs. Also, faults are described by the user in varying formats with fault codes not being used as in computer systems and human users describing their fault by using natural language which has many abbreviations and/or spelling errors, thus making it somewhat difficult to handle via rule-based systems.

In addition, the vehicle may also have been in an area with poor connectivity such as a highway, a remote country road, or a busy city street, therefore, poor connectivity makes coordination and communication between the vehicle user and service staff virtually impossible.

B. Role of Geospatial and Network-Aware Technologies

Geospatial technology is highly influential on vehicle assistance systems today; it gives users the potential for real-time vehicle location tracking and proximity-based service coordination. For example, devices equipped with GPS technologies can automatically determine the user's location, while open-source mapping systems (like OpenStreetMap [32]) provide interactive geographic visualizations of that user location in mobile applications. When the vehicle assistance system is using proximity to determine the distance between users and the nearest mechanic, distance calculation techniques such as the Haversine method [31] can be used.

To provide reliable roadside assistance, in addition to having geospatial intelligence, roadside assistance systems must also have network-aware operational mechanisms. There are several applications available today that require an internet connection to function. Unfortunately, during emergencies, the internet may not always be available. A network-aware system will respond according to real-time network availability. When there is no network connection available, the applications will switch from online mode to offline mode and save service request data locally until network connectivity becomes available again.

C. Machine Learning Techniques in Vehicle Breakdown Analysis

Vehicle fault detection and classification use many machine learning techniques today; the most well-known are Decision Trees, Support Vector Machines, Logistic Regression, and Neural Networks, where each of those Techniques analyzes the history of a vehicle to discover fault patterns based on how similar faults have been classified [10]. Overall, these techniques are typically more accurate than traditional (rules-based) systems; however, they tend to work well only if they are fed well-structured input data. In contrast, they generally do not perform well with text data.

With the newer approaches to natural language (NL) understanding through the use of advanced deep learning techniques such as transformer models, the potential for improving the natural language processing (NLP) capabilities of vehicles has also improved significantly. The transformer model, as described by Vaswani et al. [20], enables the learning of contextual representations based on self-attention mechanisms. Additionally, language models like BERT [21] (and the smaller version, DistilBERT [22]) have successfully used these models for text classification purposes, enabling them to interpret user-provided descriptions of faults more effectively. Further, Sentence-BERT [23] and MiniLM [24] models can be employed to compute semantic similarities quickly.

Using these capabilities in conjunction with other technologies was needed to create intelligent help systems that offer more advanced contextual-based recommendations and ranking that go beyond simple keyword matching to account for context-rich breakdown descriptions.

D. Need for Intelligent and Scalable Vehicle Assistance Systems

As AI systems improve their capabilities, the need for reliable, scalable, and deployable AI systems is becoming increasingly important. Intelligent vehicle breakdown assistance systems provide reliable and accurate predictions as well as low latency and reliability while functioning within the constraints of mobile computing. Research on

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁝⁝ Peer-reviewed & Refereed journal ⁝⁝ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.15427

recommender systems has shifted from traditional collaborative filtering approaches [18], [19] to pursue approaches that are context aware and location based [13] [15] - this demonstrates the importance of integrating semantic relevance with geographic proximity when developing recommender systems.
Currently, roadside assistance systems are being designed and developed to be used in rural areas and on busy roads. This is accomplished through the use of cloud based data systems and distributed systems; thus allowing for the scalability of AI-assisted workflows. Additionally, having the ability to manage connectivity and communicate properly is a critical part of any successful roadside assistance system regardless of whether or not there is a connectivity issue [33].
With the progression towards new transportation technologies and transportation innovation in the form of next generation mobility solutions, there is now a greater demand for vehicle breakdown assistance solutions that utilize transformer based issue identification and semantic recommendations, as well as geospatial filtering and network awareness; all within a cohesive mobile solution.

IV. DATASET USED

The IntelliFix system is built upon a Hybrid DataSet Framework that includes both synthetic text-based (for AI training) as well as real-time operational (saved through the Firebase FireStore Database) datasets. All of the real-time operational data will allow for the complete management of the authentication, service coordination, as well as the request life cycle; therefore, every operational data value affects how the user is classified, what recommendations are made based on the semantics, and how the user-mechanic interaction in terms of geospatial filtering will be developed in an automatic manner through the data set configuration.

A. User and Mechanic Authentication Dataset

For Firebase Firestore, there are two different but well-structured data sets for authenticating users. Users can use either their Google Account or email to sign in. Once signed in, their user profile is created with user ID, email address, phone number, role, and date of creation.
Additional mechanic data like work related to the mechanic(s) is stored separately within the database to differentiate between users and mechanics. Examples of mechanic data stored include: shop name, types of work performed by mechanic(s), address of workshop, geographic location of workshop, and contact details for the mechanic(s). Overall, the two separate datasets are well-organized and allow for role-based authentication use across the IntelliFix Ecosystem.
Currently, IntelliFix, in its prototype form, can tap into a sample dataset provided by a set of mechanics in Virudhunagar district in the state of Tamil Nadu, India. The dataset currently includes basic information about each of these mechanics, such as their workshop names, types of services provided, contact information, and exact coordinates of each workshop's location. In the future, it is envisioned that IntelliFix will be able to tap into actual data about mechanics in the state of Tamil Nadu, and then expand its reach to encompass the entire country.

B. Vehicle Issue Classification Dataset

In order to understand the notes provided by the user about the breakdown of the vehicle, we have developed a synthetic dataset of 2,500 vehicle issue classification data. The dataset is designed in a way that it can be used to classify different types of issues, including those with the engine, batteries, brakes, steering, electrical, etc. The dataset is divided into two parts, one for training and one for testing, in order to understand how well the model is performing. The dataset is used to fine-tune the model, allowing it to classify the breakdown notes.

C. Semantic Issue–Service Matching Dataset

We designed a specific synthetic dataset to support context-aware mechanic recommendations. The dataset comprises 5,000 pairs, with each pair including a normal car issue description and a specific service description that a mechanic offers.
To obtain dense vector representations of these descriptions, we fine-tuned the sentence-transformer model 'all-MiniLM-L6-v2' on our dataset. At runtime, we use these dense vector representations to rank mechanics based on how semantically relevant they are, by comparing these vectors using cosine similarity. The dataset was divided into a training set and a validation set.

Dataset Distribution Analysis

In order to ensure that the model is trained on an equally balanced set of data and that there is no bias in classification, we have carefully structured the division of the data across various classes of vehicle issues. For the synthetic dataset used in classifying vehicle issues, there are 2,500 samples in total across the five main classes of breakdowns:

DOI: 10.17148/IJARCCE.2026.15427

TABLE I. DATASET DISTRIBUTION ACROSS VEHICLE ISSUE CATEGORIES

Issue Category	Number of Samples
Engine Issues	520
Battery Failures	480
Brake Problems	500
Steering Issues	480
Electrical Faults	520

This even distribution of classes also assists in controlling overfitting for a given class while promoting generalization for various class breakdowns. In the field of semantic matching, the issue-service dataset comprises 5,000 pairs of issue and service descriptions that have been split into:

TABLE II. SEMANTIC MATCHING DATASET SPLIT

Training Set	4,000 pairs (80%)
Validation Set	1,000 pairs (20%)

This division is helpful if you want to adjust the embeddings while not sacrificing enough data to effectively test the model. Essentially, the dataset is designed to be robust, well-structured, and well-balanced and is therefore good practice for learning with transformers and applicable in real-world scenarios.

D. Real-Time Service Request Dataset

Tickets are stored in the Firebase Firestore database in the "tickets" collection. When the "Raise Repair Request" button is clicked in the application, a new ticket is created in the database. The database contains information like the time of creation, identity of the user, description of the problem, type and specification of the problem, coordinates, media if any is attached, and the status of the problem, among others.

The status of the problem is the main driving force behind the dynamic life cycle of the ticket, with statuses like searching, assigned, and completed, among others. After the creation of the ticket, artificial intelligence is used to identify the problem, match the problem with the profile of the mechanic, and use semantic similarity to narrow down the options based on the coordinates of the problem.

E. Geospatial Data Representation and Processing

User requests and mechanic profiles contain latitude-longitude pairs, which are saved in Firestore. The latitude-longitude pairs are used to calculate the distance between two points on a map using the Haversine formula. The mobile app uses OpenStreetMap for geographic visualization and tracking of user location in real time. The geographic data is used to match stranded users with mechanics efficiently, utilizing both semantic and spatial relationships.

F. Data Privacy and System Integrity

This ensures that all the operational data is kept within the confines of the Firebase cloud environment with proper authenticated access. Additionally, the personal information is kept within the confines of the operational coordination at all times, never spilling over into the synthetic data sets used in training the AI model. This integrated dataset ensures that the scalable, safe, and reliable deployment of the IntelliFix vehicle breakdown assistance system is achieved.

V. EXISTING WORK

The research in the detection of vehicle breakdowns and providing assistance has shifted away from traditional rule-based systems and towards smarter and more adaptive systems. Despite significant advancements in the field of intelligent transportation systems and mobile computing, there exist limitations in the methods that we currently employ.

A. Traditional Rule-Based Vehicle Assistance Systems

The traditional roadside assistance systems that help stranded drivers have been based on logic that follows sets of rules. Such systems have been based on sets of codes and mapping that help identify potential issues in the cars and guide the stranded drivers towards repair facilities. Such systems are easy to deploy and operate but don't fit well in dynamic and

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.15427

changing situations. They perform better when they receive structured data but struggle when they receive unstructured data from the user. Moreover, the traditional systems that involve manual intervention, such as hotlines, make the system less agile.

B. Machine Learning-Based Fault Detection Systems
With an increase in data availability, machine learning techniques have come to be used to improve the classification of faults. Decision Trees, Support Vector Machines, Logistic Regression, and Neural Networks use historical data to predict what kind of faults may occur in the future. These techniques have shown promising results in comparison to rule-based models.
However, traditional models require well-structured features. They may not perform well when they encounter unstructured data in the form of natural language. When people face a roadside fault, they may not express it in a well-structured manner. Therefore, these models may not be able to understand the situation.

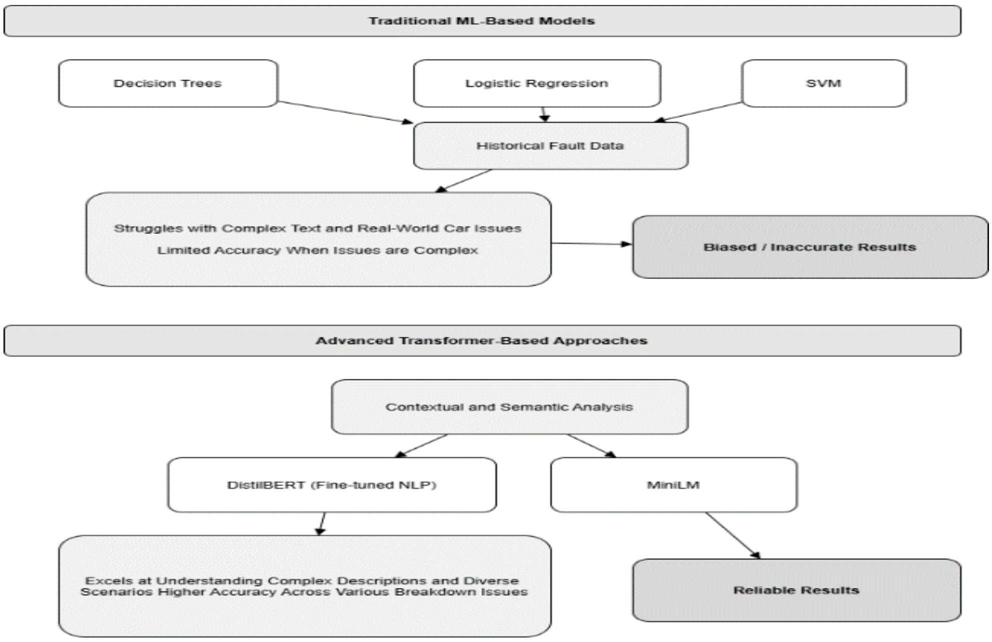


Fig. 1. Comparison of Traditional ML-Based Models and Advanced Transformer-Based Approaches for Vehicle Fault Classification.

C. Mobile-Based and Location-Aware Assistance Platforms
However, recent advances in mobile computing technology have enabled the development of location-aware vehicle help apps. The apps use GPS technology to identify your location and connect you with service providers. The geographic proximity filter reduces response times for your requests.
Despite these advances, most mobile help apps require a stable Internet connection to transmit your requests, process them, and connect you with service providers. If you live in an area with unreliable Internet access, your requests may not be sent or may take longer to receive a response. Most mobile help apps also use simple keyword recognition to understand your requests, without a deeper understanding of the context of your requests.

D. Limitations of Existing Approaches
Even though there are individual research papers on the ideas of tracking vehicles, fault classification, and providing recommendations based on the location, there is no integrated system that combines these ideas. The current system lacks the following features:

1. An effective natural language understanding system using the transformer architecture for unstructured fault descriptions.
2. A semantic similarity-based system for measuring the level of expertise of the mechanic.
3. The system must have the provision for switching between online and offline modes for dealing with network issues.
4. The system must be cloud-based for real-time synchronization.

All these features indicate the necessity for an intelligent, scalable, and network-aware system for assisting in vehicle breakdowns.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15427

VI. THE PROPOSED METHODOLOGY

IntelliFix is an intelligent solution that is a modern interpretation of fixes that go beyond the traditional rule-based scripts and standard machine learning techniques. It is an amalgamation of transformer-based natural language processing, semantic similarity ranking, geospatial proximity filtering, and a hybrid online/offline decision framework, all in a mobile-first design.

The architecture is based on a modular processing pipeline that begins with user input from the mobile application. Users can describe the breakdown in plain language, with the option to include media as an optional input. They can also choose to share their real-time location. It is capable of adapting in real time, changing behavior based on network availability, allowing it to function even in offline scenarios.

Using PWA architecture enables features like offline caching, background sync, and fast loading. This improves the system's ability to work well even in low or no network conditions.

The IntelliFix application is a Progressive Web App and will function efficiently regardless of whether there is an internet connection or not. The application features a network-dependent module that senses your connection status and switches to the relevant mode of operation based on the result. In the case of offline activity, the requests are temporarily stored locally and uploaded to the cloud database once a network connection becomes available.

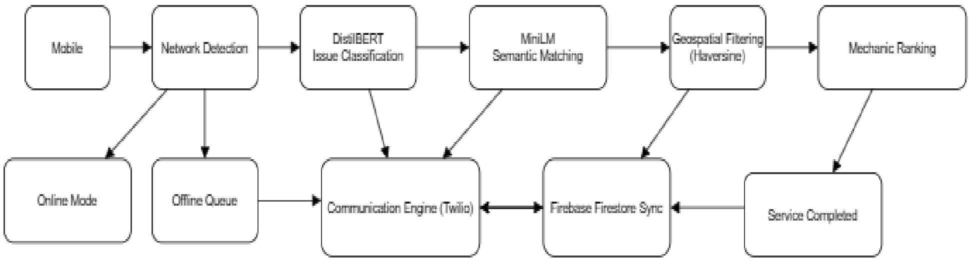


Fig. 2. End-to-End Processing Pipeline of the Proposed IntelliFix Methodology.

A. Network Detection and Operational Control

A network detection module remains vigilant about the connectivity status. It switches between online and offline modes depending on the connectivity status. If the network is online, real-time AI inference and service coordination are performed instantaneously. If the network is offline, the requests are stored locally in a structured queue. As soon as the network is back online, the requests are synchronized with the cloud database.

Apart from the basic offline user request storage facility, further enhancements have been made in the system with respect to more sophisticated offline handling facilities. A structured local queue management system is implemented in the system for effective management of user requests during offline conditions. Moreover, real-time user feedback facilities have also been implemented in the system for indicating offline conditions. Optimum synchronization strategies have also been implemented in the system for ensuring that all user requests are sent to the cloud during connected conditions.

B. AI-Based Issue Classification

The first intelligence layer in the system is responsible for the classification of vehicle issues using a fine-tuned model called the DistilBERT model. When a user enters a description of the vehicle's breakdown, the entered text is first processed and then fed into the model for classification.

The model can then identify the exact category of the vehicle breakdown, i.e., whether the problem is with the engine, battery, brakes, steering, or electrical system of the vehicle. By using the transformer architecture, the model can develop a better understanding of the input provided, thereby being more reliable compared to other machine learning techniques.

C. Semantic Matching and Recommendation Engine

Once classification is finished, it activates its semantic similarity module, which employs a fine-tuned MiniLM model. The module produces dense vector embeddings of the user's problem description as well as the profiles of the mechanic services.

Through the assessment of the similarity of these embeddings using cosine similarity, the system can rank the mechanics based on their level of relevance rather than the keywords. This ensures that the recommendations are accurate and that the selected mechanics are suitable for the breakdown scenario.

ISSN (O) 2278-1021, ISSN (P) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.15427

D. Geospatial Proximity Filtering

To provide prompt help, the system utilizes the latitude and longitude of the device for filtering. It computes the distance of the user from the mechanic using the Haversine formula and then filters the list based on the service radius. The mechanics are ranked according to how relevant they are and how close they are. The mobile interface is made better by using OpenStreetMap.

Mathematical Formulation of Core Components

To formally describe the way the IntelliFix architecture works, we first describe the essential mathematical models that the IntelliFix architecture uses for semantic similarity evaluation, distance computation, and hybrid ranking.

1) Cosine Similarity for Semantic Matching:

The similarity of the user-reported issue with respect to the mechanic's service profile is determined by calculating the cosine similarity of their vectors.

$$Sim(U,M) = (U \cdot M) / (||U||\ ||M||)$$

- U = embedding vector of user issue
- M = embedding vector of mechanic profile
- U · M = dot product of vectors
- ||U|| = magnitude of vector U
- ||M|| = magnitude of vector M

The similarity score is in the range of -1 to 1, with values close to 1 indicating strong semantic relevance.

2) Haversine Distance for Geospatial Filtering:

The Haversine formula is used to determine the geographic distance between the user and the mechanic:

$$d = 2R \arcsin\left(\mathrm{sqrt}\left(\sin^2(\Delta\varphi/2) + \cos(\varphi_1)\cos(\varphi_2) \sin^2(\Delta\lambda/2) \right) \right)$$

Where R is the Earth's radius (approximately 6371 km), φ_1, φ_2 represent latitude values, $\Delta\varphi$ is the difference in latitude, $\Delta\lambda$ is the difference in longitude, and d is the great-circle distance between user and mechanic.

This guarantees that real-time service allocation is supported by precise proximity filtering.

3) Hybrid Mechanic Ranking Score:

A weighted hybrid scoring function is presented to combine geographic proximity and semantic relevance:

$$Score = \alpha \times Sim(U,M) - \beta \times d$$

Where α, β are weighting coefficients, Sim(U,M) is the cosine similarity score between user issue and mechanic profile, and d is the geographic distance calculated using the Haversine formula.

Mechanics that are closer in geographic distance and highly relevant are ranked highest.

E. Communication and Request Lifecycle Management

Once a suitable mechanic is identified, the system starts running its communication workflows. The system sends notifications using messaging services, and in case the initial contact does not yield a response, it goes into a sequential contact. Every service request goes through a number of lifecycle phases, including assigned, completed, and searched. Firebase is used by the system to synchronize in real time.

F. System Integration Overview

The following layers make up the suggested methodology:

1. Layer of User Interface
2. Module for Network Detection
3. AI Processing Layer (MiniLM + DistilBERT)
4. Module for Geospatial Filtering
5. Engine for Communication
6. Data Synchronization via the Cloud

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁝ Peer-reviewed & Refereed journal ⁝ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15427

By combining transformer-based contextual analysis, network-aware operational logic, and geospatial intelligence, the proposed IntelliFix system guarantees an effective, accurate, and dependable vehicle breakdown assistance service that can be used in practical situations.

VII. SYSTEM ARCHITECTURE AND APPLICATION WORKFLOW

The IntelliFix system is designed around a modular, cloud-based architecture, focusing on intelligent, dependable, and effective support for breakdowns in a way that is aware of practical constraints. The IntelliFix components include a mobile client, AI-based processing modules, geospatial filtering, communication, and a cloud-based database. The IntelliFix workflow is straightforward, including user engagement, mechanic assignment, and service completion.

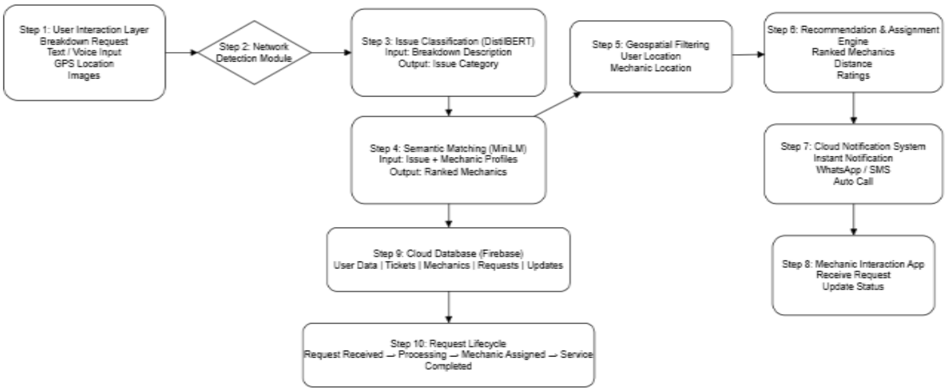


Fig. 3. Overall System Architecture and Application Workflow of IntelliFix.

The architecture that is discussed is a modular and layered pipeline architecture. It includes network detection, issue classification with the help of transformers, ranking based on semantic similarity, geospatial filtering, and synchronization with the cloud in real time. The architecture is a hybrid architecture and is expected to perform well in both online and offline scenarios.

A. User Interaction Layer

The first step in the process is the interface of the mobile application, where the authenticated user sends a breakdown request. The user describes the problem with the vehicle in simple terms, while the application logs the GPS location in real time. The user can add media, for example, pictures of the vehicle, to be sent with the request.

Apart from this, the pictures that the users share of their cars are another source of information for the mechanics. These help you understand the condition of the car even before you start touching the vehicle. The pictures provide visual representations of the problem, which may be difficult to explain in words. These visual representations of the problem help everyone understand the problem better, and this speeds up the process, reducing the time required by the mechanics to get the help rolling.

After the submission of the problem, the first step is the network detection module, followed by the AI processing.

B. Network Detection and Adaptive Mode Switching

The network detection module always checks for the availability of the internet. Depending on this availability, the system can be in one of the following modes:

- Online Mode: The communication with the cloud and AI happens in real-time.
- Offline Mode: The requests are stored locally until they can be processed.

When offline, it syncs the stored requests with the Firebase cloud database. Such an offline process has also been improved with enhanced state management and smooth transitions between offline and online states. Any offline operations performed by the user are securely saved and executed without delay. Once the connection is restored, the system synchronizes smoothly without losing or duplicating data. All these improvements enhance the reliability and seamless performance of the system.

C. AI Processing Layer

The AI processing layer consists of two primary aspects: sorting out the issue type and then meaningfully matching with the right mechanic. First off, the issue type is identified by running the user's breakdown through a fine-tuned DistilBERT model. The model is capable of recognizing different types of issues: engine issues, battery issues, brake issues, steering issues, electrical issues, and so on.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15427

Once this is done, the semantic matching module is triggered. The module uses a fine-tuned MiniLM to generate dense representations for the issue description and the existing profiles of the mechanics. The context relevance is then determined by calculating the cosine similarity between the two representations and then ranking the mechanics accordingly.

D. Geospatial Filtering Module

To accelerate our response time and optimize the assignment of individuals to a task, we utilize a geographic proximity filter. This is achieved by determining the distance between the user and each individual mechanic using the great-circle distance formula, known as the Haversine formula. Those within our service radius are then ranked using a hybrid scoring system that takes into account:

- Semantic Similarity
- Geographic Distance
- Availability Status

OpenStreetMap is integrated into the application for map visuals and easy confirmation of locations.

To address emergencies where connectivity is intermittent, IntelliFix proposes a hybrid approach for offline-online mapping. In this prototype, we have prepared offline geospatial data for the Virudhunagar district, which is locally stored in the app. The map data was obtained through the BBBike extract service for "Virudhunagar, Tamil Nadu" in shapefile format (shp.zip). In offline mode, when no internet is available, the app will display the locally stored map of the Virudhunagar district, and geospatial filtering will be done based on these coordinates. Once an internet connection is restored, the system will automatically go into online mode, where it will fetch map data in real-time through OpenStreetMap, allowing normal functionality anywhere.

This is because these functions are implemented using the Progressive Web App (PWA) technology. PWA uses service workers that cache data in order to allow users to still have access even if there is no network.

E. Communication and Assignment Workflow

Once the right mechanic has been selected, the communication engine of the system sends a message to the candidates using the messaging service. If the first mechanic fails to respond within a certain time frame, fallbacks occur, sending sequential notifications to the candidates. Once a mechanic has responded, they receive the details of the request and update the status of the service using the mechanic interface. The request goes through a certain sequence of steps:

- Request Received
- Processing
- Mechanic Assigned
- Service Completed

The status is updated in real-time using Firebase Firestore.

In an actual environment along the roadside, the service coordination might break down if the selected mechanic is not available or fails to respond to the service request within the expected time. To ensure the service coordination provided by IntelliFix is always reliable, we have introduced an automated system for the recovery from failure and retries for the communication flow. Once the hybrid semantic geospatial ranking approach selects the service provider, the system immediately notifies the service provider and begins the response timeout.

If the service provider fails to acknowledge the assistance request within the set time window, it is recorded as an unacknowledged assistance request. The coordination process then proceeds to the next service provider in line, selecting the service provider with the next highest ranking within the service provider radius.

This process of retrying happens on its own without any intervention by the user, hence there is no possibility of holdups in case of emergencies. Each of these processes of reassigning the request is recorded in the request lifecycle dataset, ensuring that the operation is transparent, as well as ensuring that the real-time status is consistent in both the user interface as well as the mechanic interface. Even in cases when there is no response from mechanics after several attempts at retrying, the request remains in an active searching state.

This proposed process for handling failures improves the robustness of the system by ensuring that requests are not left pending and by providing a reliable response to requests, ensuring that everything runs smoothly even when availability is inconsistent.

F. Cloud Database and Data Synchronization

At the center of all this is Firebase, the central coordination layer, the cloud database that contains structured sets of data:

- Users
- Mechanics
- Tickets (Service Requests)
- Status Updates

DOI: 10.17148/IJARCCE.2026.15427

Because of the real-time synchronization, changes are trickled down to the devices with little delay. The cloud backbone provides secure authentication of the users, storage of the data, and the delivery of the service networks in many regional locations.

Besides that, the system has various security and privacy features for safe deployment. The Firebase Authentication service provides safe authentication using tokens and time-based ID tokens for verified users. Also, the Firestore database has security rules that allow role-based access control, limiting database access based on whether the user is a normal user or a mechanic.

All communication between the client application, backend services, and Firebase is done over HTTPS, providing encrypted security for the data in transit. Sensitive config values like the Firebase credentials and Twilio API keys are stored in the backend as encrypted environment variables, inaccessible to unauthorized parties. For the flow to use Twilio, it does so in a safe manner by using the Twilio API endpoints, thus providing secure communication for the WhatsApp messages and voice calls.

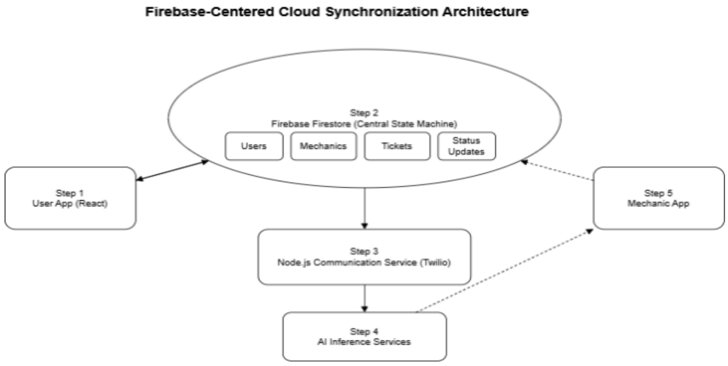


Fig. 4. Firebase-Centered Cloud Synchronization Architecture of the IntelliFix System.

The proposed IntelliFix architecture aims to scale to various regions and increase its service network. For example, the cloud-based architecture of the proposed IntelliFix architecture allows for smooth deployment in multiple cities, where technicians and users from various cities can be dynamically registered using distributed Firestore collections.

As more mechanics enter the system, the proximity filtering and ranking process works seamlessly with the use of indexed geospatial queries as well as fast comparison of embeddings. The stateless backend microservice architecture enables horizontal scaling in the cloud, coping with the high volume of requests during roadside emergencies without compromising speed.

This hybrid deployment model facilitates the distributed execution, where the semantic similarity is computed on the client devices and the classification services run in the cloud. This model enables the sustainable operation of the IntelliFix system in different transportation environments.

G. End-to-End Workflow Summary

The whole process takes place in the following manner:

1. The user makes a breakdown request.
2. Network detection determines the mode of operation.
3. DistilBERT determines the type of problem.
4. The system uses MINI LM for semantic match.
5. The system uses geospatial filtering for the location of the mechanics.
6. The communication engine allocates the mechanic.
7. The status of the service is constantly updated until the service is done.

IntelliFix combines AI-driven context understanding, semantic ranking, geospatial information, and network intelligence to provide a reliable solution for breakdown support.

In addition to the technical process, it is important to understand the relationship between various stakeholders in the system in order to successfully implement roadside assistance. This IntelliFix system is different from others in that it not only crunches numbers, it also enables interaction among all those involved. The players in this case include the user,

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⧖ Peer-reviewed & Refereed journal ⧖ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.15427

intelligent assistance, and registered mechanics. The process starts when a stranded user contacts IntelliFix, stating that their vehicle has broken down, describing in natural language how it happened, as well as stating its exact location.

Once the user makes the request, the system processes the request through various stages such as network status, AI classification, semantic similarity, and geospatial proximity filtering. After the computation of the request, the system uses a hybrid ranking score for the selection of the appropriate mechanic.

The chosen mechanic receives an instant message via communication tools built into the system, where they receive both a message and a series of calls. The mechanic has the opportunity to view information about each service request, such as what type of problem it is, where it is, and what images may be associated with it. And then, as everything happens in real time, the system keeps the user and mechanic in constant communication via the cloud network.

Throughout the service lifecycle, IntelliFix is an intelligent coordination layer that facilitates service requests, communication, and tracking of progress towards completion of the task. This model promotes seamless, transparent collaboration between users and service providers, reducing the need for manual interventions in emergency roadside services.

VIII. MODEL DEPLOYMENT AND PERFORMANCE

IntelliFix is a system built as a mobile web application with client-side intelligence and cloud-coordinated services. On the client side, the system is built with the React and Vite stack, providing a responsive and lightweight UI for emergency roadside use cases.

The app is a Progressive Web App (PWA) by design, which means that users can install it on their devices, use it offline, and experience app-like performance. It uses service workers and smart caching to enable offline support.
The current prototype deployment focuses on the Virudhunagar district, which is used to validate the offline geospatial capabilities as well as the network-aware coordination workflow before scaling the solution to a larger geographical domain.

This client-side inference ability also helps in partial offline intelligence by reducing the need for constant backend connectivity, thus improving responsiveness in areas with poor connectivity.

A. On-Device Model Deployment

To obtain fast, low-latency semantic matching, we use the all-MiniLM-L6-v2 model as a quantized ONNX model running directly in the browser. The @xenova/transformers library is used to perform model inference via Web Assembly (WASM) for native-like performance without needing constant backend requests.
The model runs through a Content Delivery Network and is cached using the browser's Cache API. This provides better user privacy and eliminates server-side computation. The performance of the model on the validation data set resulted in:

TABLE III. MINILM SEMANTIC MATCHING MODEL PERFORMANCE ON VALIDATION SET

Metric	Score
Accuracy	99.47%
Precision	98.91%
Recall	98.76%
F1-Score	98.83%

B. Issue Classification Model Deployment

The distilbert-base-uncased model is used to perform real-time vehicle issue classification in an online setting. Unlike the MiniLM model, the classification inference is done in the backend inference service. The performance results obtained from the test dataset are:

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁑ Peer-reviewed & Refereed journal ⁑ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15427

TABLE IV. DistilBERT ISSUE CLASSIFICATION MODEL PERFORMANCE ON TEST SET

Metric	Score
Accuracy	91.05%
Precision	90.12%
Recall	89.74%
F1-Score	89.93%

These results clearly show that transformer models can successfully perform the interpretation of unstructured descriptions of vehicle breakdowns.

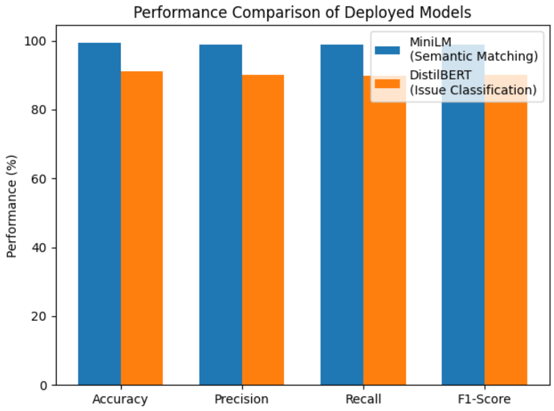


Fig. 5. Performance Comparison of MiniLM (Semantic Matching) and DistilBERT (Issue Classification) Models.

Baseline Model Comparison

To validate the proposed transformer-based classification model, we conducted a comparative analysis against traditional machine learning techniques as well as rule-based approaches. As a baseline, we tested keyword matching, logistic regression, and support vector machine (SVM)-based classification models. All these models were trained and tested on the same dataset.

TABLE V. PERFORMANCE COMPARISON OF CLASSIFICATION MODELS

Model	Accuracy	Precision	Recall	F1-Score
Keyword Matching	65.34%	62.11%	60.47%	61.28%
Logistic Regression	78.42%	76.85%	75.90%	76.37%
Support Vector Machine	84.16%	83.22%	82.75%	82.98%
DistilBERT (Proposed)	91.05%	90.12%	89.74%	89.93%

As shown in the table, the performance of the proposed method based on the transformer model is significantly better than other traditional methods in the interpretation of unstructured vehicle breakdown descriptions.

Confusion Matrix Analysis

To have a better understanding of the performance of the classifier, Table VI shows the confusion matrix of the proposed DistilBERT classifier. It provides a clear overview of the predicted and actual issue categories, which have been classified into five major types of vehicle breakdowns.

DOI: 10.17148/IJARCCE.2026.15427

TABLE VI. CONFUSION MATRIX OF DistilBERT MODEL'S CLASSIFICATION PERFORMANCE

Actual \ Pred.	Eng	Bat	Brk	Str	Elec
Eng	210	5	3	2	4
Bat	6	195	4	3	2
Brk	4	6	202	5	3
Str	3	2	7	198	4
Elec	5	3	4	6	200

Abbreviations: Eng – Engine; Bat – Battery; Brk – Brake; Str – Steering; Elec – Electrical.
Essentially, the confusion matrix just sits on a strong diagonal, indicating that the model is good at classifying correctly. There is a little bit of misclassification, but it is mostly between classes that are mechanically linked, like engine problems versus electrical problems, as their descriptions overlap.

C. Communication and Backend Deployment

A Node.js & Express microservice is used as a communication orchestration layer. It is implemented as a stateless containerized service, enabling horizontal scaling to cope with peaks in request volumes. Twilio is integrated for automated voice calls, and WhatsApp/SMS notifications are used for acknowledging mechanics' service events. Environment variables are maintained securely, ensuring the integrity of Firebase & Twilio credentials for all communications.

D. Real-Time Database Persistence

The Firebase Firestore database will serve as our central NoSQL data store and backbone for keeping the state in sync. It is schema-less and will be able to handle a variety of data types, such as text descriptions, image attachments, geographic coordinates, and status updates on the request. Real-time listeners will be used to ensure that everyone is in sync. With the distributed cloud infrastructure, we will have high availability and low-latency updates for everyone, from the user to the mechanic.

Runtime Performance Analysis

To evaluate the ability of the IntelliFix system to run in real time, the system's performance was measured from both the client and backend sides. From the backend side, the performance of the DistilBERT classifier for each request was approximately 148 milliseconds, including the tokenization and prediction phases, for prompt issue categorization. Quantized MiniLM ONNX running in the browser using WebAssembly measures semantic similarity at an average of 42 milliseconds per inference, reducing the load on the servers in urgent situations. DistilBERT weighs in at approximately 256 MB, whereas the quantized MiniLM ONNX variant reduces to 22 MB. This makes caching in a browser much more efficient and speeds up load times when using a Content Delivery Network (CDN) for deployment. The entire processing cycle, including classification, semantic matching, geospatial filtering, and syncing with the database, excluding network latency, took an average of 310 milliseconds.

The results have reaffirmed that the hybrid deployment setting strikes an appropriate balance between efficiency and accuracy. The IntelliFix prototype was tested in simulated roadside breakdown scenarios with multiple users and registered mechanics interacting at the same time via the mobile app. The testing was conducted under different network conditions: with a steady network connection, with an intermittent network connection, and with no network connection at all.

The proposed framework was shown to have the capability to perform issue classification, semantic ranking of mechanics, proximity-based filtering, and task assignment within real-time constraints. The experiments have shown that the requests could be synchronized, mechanics assigned, and client devices could be managed throughout their lifecycle. The testing of the prototype has shown the viability of deploying the IntelliFix framework in emergency roadside scenarios.

IX. SYSTEM OUTPUT

In this section, actual results from the IntelliFix system, as implemented, are presented. The images demonstrate how both the front and back ends of the system work together, thereby validating the actual workflow of the proposed vehicle

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15427

breakdown assistance system. Each image shows an actual module of the system, demonstrating how it works in actual scenarios. The results validate that IntelliFix actually works in real-time.

A. Backend Execution Output

The backend is running as expected, which means the server-side components are initialized and functioning correctly. The backend is built using Node.js with Express, and the terminal logs indicate that the application is integrated with Firebase and Twilio correctly. The backend connects with the Firebase Firestore service during startup, which is responsible for real-time data syncing, as well as loading the AI models that are used during the process of classifying the issues and performing semantic matching.

In addition, ngrok is used to securely share the local server with other networks. This is done using the public URL that is created. This ensures that there is communication with the backend using webhooks, which is essential in ensuring that the backend works in real time. Moreover, the logs have been able to capture events such as response management by the mechanic, ticket management, and messaging. This is an indication that the backend works in real time and is able to handle emergency services with efficiency, as shown in Fig. 6.

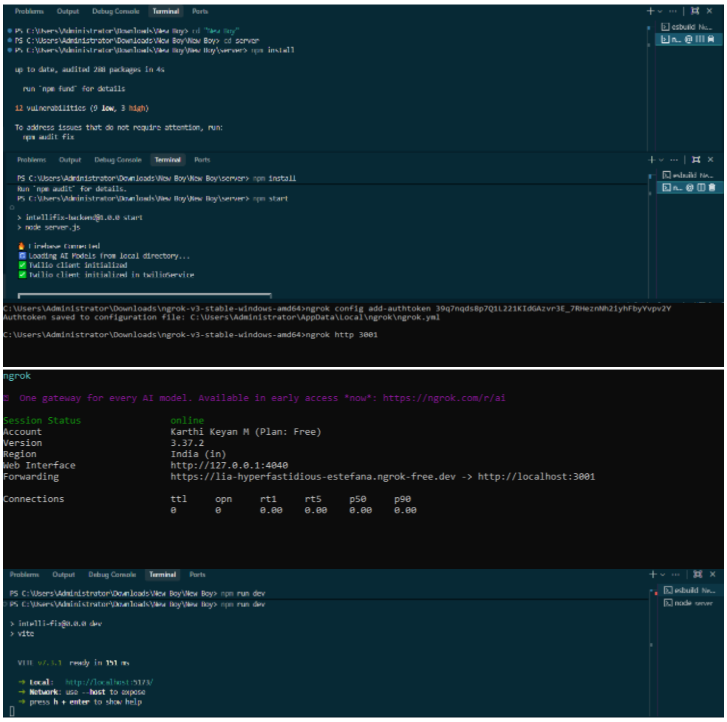
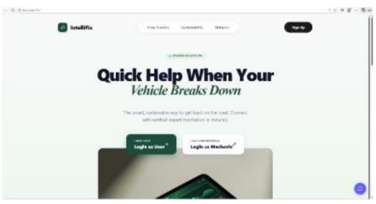


Fig. 6. Backend Initialization and Communication Workflow of IntelliFix System.

B. User Interface Screens

The user interface screens act as a gateway to the IntelliFix system. The landing page is a clean interface that allows users to access the application. The user can go to the login or signup page to get authenticated. The login interface is designed to ensure secure authentication by using user credentials. Once authenticated, users can access various features through the dashboard. The dashboard is a clean interface that shows users their status, location on a map, and actions such as raising a repair request. The interface is user-friendly, ensuring ease of use in emergency situations, as shown in Fig. 7.



IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15427

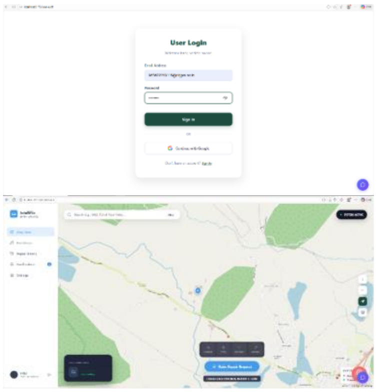


Fig. 7. User Interface Screens of IntelliFix Application.

C. Location and Request Processing

The following is a walkthrough on how your location is determined in real-time, as well as how your repair request is processed. It begins by asking for your permission to use your device's GPS to determine your location. It then identifies your location on the map using OpenStreet Map. With your location identified, you can then submit your repair request by describing your vehicle's problem. The system is able to take this natural language input. By combining your location with your problem description, you now have all that is required for the next steps, as shown in Fig. 8.

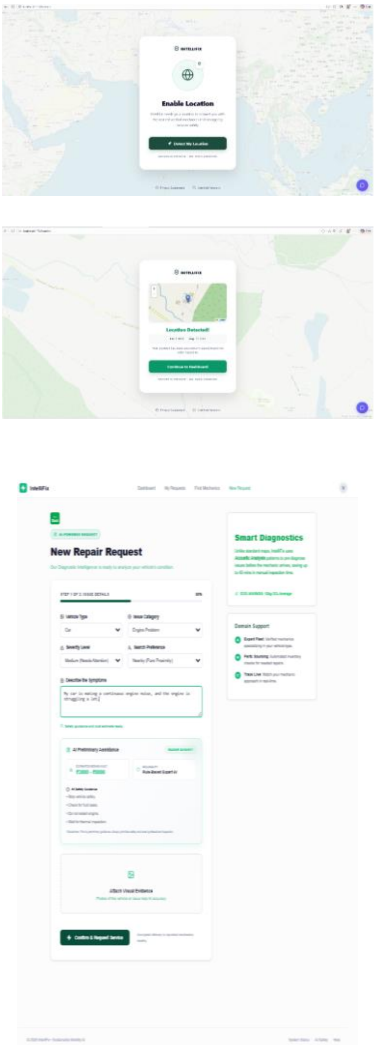


Fig. 8. Location Detection and Repair Request Processing in IntelliFix.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15427

D. Mechanic Recommendation and Filtering

The recommendation module uses your location to find the service providers near you. It uses geospatial filtering techniques such as the Haversine formula to calculate the distance between the service providers. In addition to the distance factor, the system also checks the similarity between the description of the problem you are having and the services provided by the mechanics.

This interface enables users to search for and filter available mechanics. The dashboard for the mechanics enables them to view incoming requests, along with information such as location and issue description, in order for them to respond as required. This module ensures the efficient and intelligent allocation of service providers, as shown in Fig. 9.

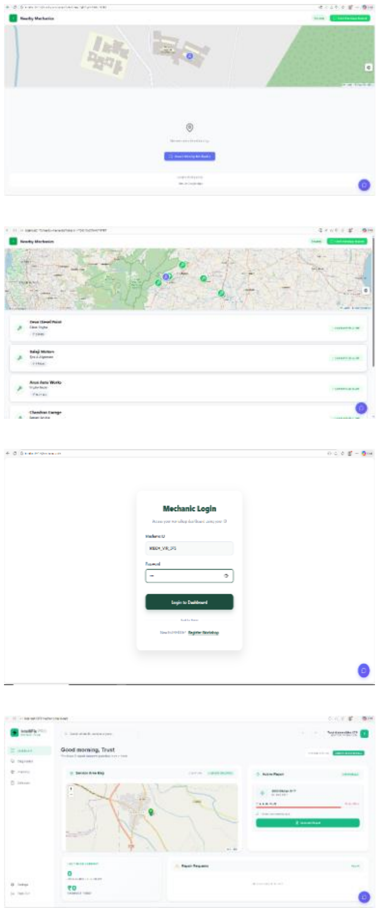


Fig. 9. Mechanic Search, Filtering, and Dashboard Interaction.

E. Communication and Confirmation

The communication module also demonstrates how users and mechanics interact through these automated messages. Once the user has selected their mechanic, the system will alert the mechanic through WhatsApp using Twilio's APIs. The screenshots demonstrate how the message has been successfully sent, including request information and prompts. The mechanic also has the ability to reply to these messages, for example, through an ACCEPT message.

The system displays confirmation screens after a request is sent out, and the service is allocated. Each message has a unique ID. This module evaluates how effectively IntelliFix facilitates communication in real-time, which is a significant part of the emergency roadside service, as shown in Fig. 10.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15427

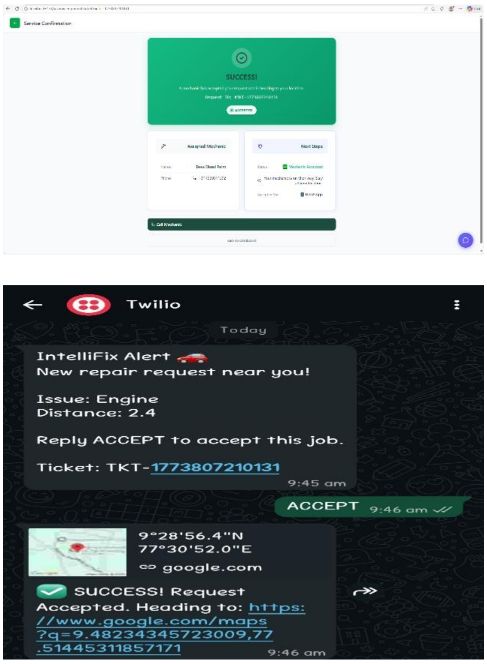


Fig. 10. WhatsApp-Based Communication and Service Confirmation.

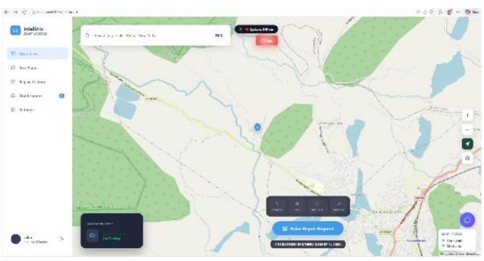
F. Offline Operation Output

The IntelliFix application is able to work offline seamlessly, which proves that it is capable of providing responses to user queries in absence of an internet connection. It continuously monitors the internet connectivity status, and in case of any change in status, it switches to offline mode, retaining all its primary functionalities.

If a user requests a repair while offline, then an alert appears informing the user that there is no Internet connection and their request has been saved locally. The system is able to save user requests when they don't have Internet access. User requests will contain crucial information such as the type of vehicle, description of the problem, priority, and location. The process of submitting a request to repair remains completely functional even when there is no internet connection available. The user will be able to fill out his/her car data and explain the issue in common words, receiving some basic assistance from the AI system, such as an approximate repair cost.

The application is able to render maps even without a connection to the Internet thanks to geospatial information stored on the device. In the map view, one will be able to view their current location, including system status icons like "System Offline," and even emergency buttons like the SOS function. Once connectivity is restored, your local requests will be synced automatically with the cloud database. This is where the backend process comes into play, processing your request and performing semantic matching before assigning the appropriate mechanic to attend to your request. You'll be notified through the service confirmation portal, where the selected mechanic's information, together with the status of your request, will be made available.

Such results illustrate how IntelliFix efficiently supports the offline-online approach to ensure uninterrupted services, reliable work, and coordinated operations in the practical fieldwork of roadside assistance emergencies, as shown in Fig. 11.



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15427

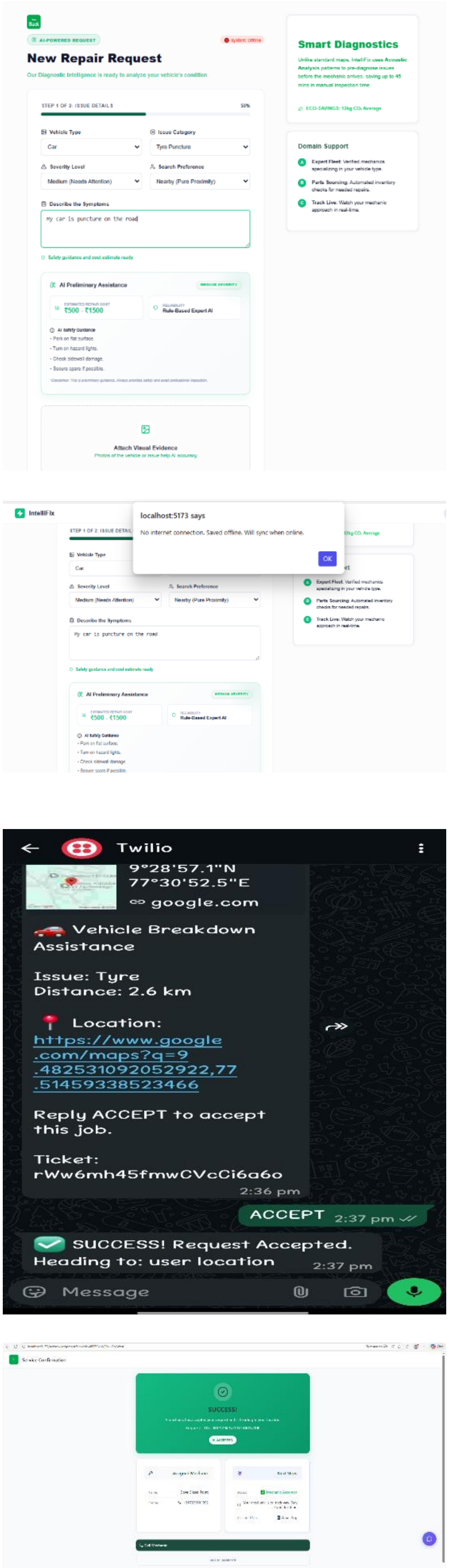


Fig. 11. Offline Request Processing, Data Storage, and Synchronization Process in the IntelliFix System.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15427

X. CONCLUSION

The paper proposes an intelligent vehicle breakdown assistance system called IntelliFix. The proposed system is intended to overcome the limitations faced by traditional intelligent systems and machine learning approaches. The proposed system consists of different modules: natural language understanding with the help of the transformer architecture, semantic similarity-based mechanic recommendation, geospatial proximity filtering, and adaptive online/offline operation logic.

Unlike other systems that rely on strict inputs and internet connectivity, IntelliFix makes use of contextual transformer models like DistilBERT and MiniLM in understanding unstructured and user-provided breakdown descriptions. With the help of semantic embedding, the ranking of the mechanics is improved. Moreover, the network detection module and request queue help in service delivery despite the unreliable internet.

The project is developed as a Progressive Web Application (PWA) using React and Vite. For the client-side, it uses semantic inference with quantized ONNX models in WASM. For the backend, it uses Node.js, communication workflows with Twilio, and real-time synchronization with Firestore. For evaluating the performance, it achieved good accuracy, precision, recall, and F1 scores, indicating that the models used are effective.

The IntelliFix framework offers various innovations that are both practical and architectural in nature and help this framework differentiate itself from other vehicle breakdown assistance systems. Unlike other systems that require constant cloud connectivity, this system is built with network awareness and is capable of dynamically transitioning between online and offline modes. The client-side semantic inference is also available with this system and is carried out with the help of the MiniLM model and ONNX.

Another new and exciting innovation involves the concept of a hybrid mechanic recommendation methodology that incorporates the semantic understanding of user-reported issues related to the vehicle and the geospatial proximity of service. This effectively leverages the contextual relevance of the user request through the use of the cosine similarity measure and the Haversine distance formula. Additionally, the incorporation of the offline emergency handling methodology effectively ensures that the system remains operational even in the midst of rural highways. With this, the traditional service coordination setup has effectively been redefined into an intelligent and efficient AI-driven roadside assistance ecosystem.

However, there are some drawbacks in the proposed system. First, the AI model is trained on artificial data, not real-world data. The second drawback is that it may not work well when there is a mix of languages in the user's query. The third drawback is that, in offline mode, there is no immediate backend-based classification of issues unless there is an internet connection. The effectiveness of the proposed system is also dependent on the GPS of the user's device. It is not just the technical performance that is enhanced; there are also some positive social and environmental impacts that come with the IntelliFix system. The system allows for prompt assistance on the roadside and the dispatching of mechanics, thus reducing congestion on the roads and keeping the flow of traffic smooth. This, in turn, reduces the amount of fuel that is wasted when cars break down on the way. The system also allows for increased access, especially on the highway. In total, IntelliFix offers an entire architecture that combines the power of transformer models, geospatial distance calculation, and cloud-based strategies to provide an effective and efficient breakdown assistance service. Future plans and research directions involve the integration of real-time vehicle sensors for predictive diagnostics, expanding the support for multiple languages, and employing reinforcement learning to optimize the efficiency of the mechanics assignment.

XI. FUTURE WORK

The idea is that IntelliFix will expand its scope and capabilities. Currently, our prototype is based on a small set of mechanics from Virudhunagar, along with some geospatial data related to that area. In the future, we intend to validate data for each mechanic in the state of Tamil Nadu. The ultimate aim is to expand IntelliFix to cover all mechanics in India. Additionally, IntelliFix will also have an option for users to download maps offline, allowing them to use maps even when connectivity is bad. The next big step for the future is the implementation of multilingual support for natural languages. This will allow users to describe their car problems in local languages like Tamil or Hindi. This will make the system accessible to a wider audience. The next big step for the future is the implementation of the real-time availability of the mechanic. This will allow the system to not just rate the relevance and distance of the mechanic, but also their availability.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15427

The system could be improved by incorporating predictive vehicle diagnostics that utilize data from the vehicle's numerous sensors and On-Board Diagnostic II. This will allow the system to identify problems before they cause a vehicle breakdown. It could also be improved by incorporating reinforcement learning that refines the selection of a mechanic by utilizing feedback from previous servicing records, response rates, etc. As for the future, IntelliFix is planned to extend its scope to include more services like towing services, fuel delivery services, and emergency services. Even though it is currently running as a Progressive Web App, there is room for further development that could include making it a native Android or iOS application, which would allow for deeper integration with hardware components. This could be done in order to have better communication with the device.

REFERENCES

[1] M. Kaliappan, E. Mariappan, M. V. Prakash, and B. Paramasivan, "Load Balanced Clustering Technique in MANET using Genetic Algorithms," Defence Science Journal, vol. 66, no. 3, pp. 251–258.

[2] M. Sivaram, M. Kaliappan, S. J. Shobana, Prakash, and V. Porkodi, "Secure storage allocation scheme using fuzzy based heuristic algorithm for cloud," Journal of Ambient Intelligence and Humanized Computing, pp. 1–9.

[3] S. Vimal, Y. H. Robinson, M. Kaliappan, K. Vijayalakshmi, and S. Seo, "A method of progression detection for glaucoma using K-means and the GLCM algorithm toward smart medical prediction," The Journal of Supercomputing, vol. 77, no. 1, pp. 1–17, 2021.

[4] M. Kaliappan et al., "Analyzing Public Sentiment on Demonetization Using SVM: A Machine Learning Approach," Journal of Computer Science, pp. 2482–2487, Dec. 2025.

[5] S. Jeevan, J. Sharma, M. Moharir, and A. K. AR, "GPS based efficient real time vehicle tracking and monitoring system using two factor authentication and internet of things (IoT)," in Proc. 2nd Int. Conf. Intelligent Cyber Physical Systems and Internet of Things (ICoICI), 2024, pp. 473–478.

[6] A. K. Shafik and H. A. Rakha, "Integrated back of queue estimation and vehicle trajectory optimization considering uncertainty in traffic signal timings," IEEE Trans. Intell. Transp. Syst., vol. 25, no. 12, pp. 21393–21403, 2024.

[7] C. Chen, J. Ma, Y. Susilo, Y. Liu, and M. Wang, "The promises of big data and small data for travel behavior (aka human mobility) analysis," Transportation Research Part C: Emerging Technologies, vol. 68, pp. 285–299, 2016.

[8] A. K. Dey, "Understanding and using context," Personal and Ubiquitous Computing, vol. 5, no. 1, pp. 4–7, 2001.

[9] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," Future Generation Computer Systems, vol. 29, no. 1, pp. 84–106, 2013.

[10] H. Hartenstein and L. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," IEEE Communications Magazine, vol. 46, no. 6, pp. 164–171, 2008.

[11] A. Abid, M. T. Khan, and J. Iqbal, "A review on fault detection and diagnosis techniques: basics and beyond," Artificial Intelligence Review, vol. 54, no. 5, pp. 3639–3664, 2021.

[12] M. Amoozadeh et al., "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving," IEEE Communications Magazine, vol. 53, no. 6, pp. 126–132, 2015.

[13] V. Schmid and K. F. Doerner, "Ambulance location and relocation problems with time-dependent travel times," European Journal of Operational Research, vol. 207, no. 3, pp. 1293–1303, 2010.

[14] J. Bao, Y. Zheng, and M. F. Mokbel, "Location-based and preference-aware recommendation using sparse geo-social networking data," in Proc. 20th Int. Conf. Advances in Geographic Information Systems, 2012, pp. 199–208.

[15] M. D. C. Rodríguez-Hernández et al., "Location-aware recommendation systems: Where we are and where we recommend to go," in CEUR Workshop Proc., 2015.

[16] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: concepts, methodologies, and applications," ACM Trans. Intell. Syst. Technol., vol. 5, no. 3, pp. 1–55, 2014.

[17] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems," IEEE Trans. Knowl. Data Eng., vol. 17, no. 6, pp. 734–749, 2005.

[18] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in Proc. 10th Int. Conf. World Wide Web, 2001, pp. 285–295.

[19] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," Advances in Artificial Intelligence, vol. 2009, no. 1, 2009.

[20] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems, vol. 30, 2017.

[21] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, vol. 1, 2019, pp. 4171–4186.

[22] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," arXiv:1910.01108, 2019.

[23] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in Proc. EMNLP-IJCNLP, 2019, pp. 3982–3992.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.15427

[24] W. Wang et al., "MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 5776–5788.
[25] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in Proc. EMNLP: System Demonstrations, 2020, pp. 38–45.
[26] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, vol. 1. Cambridge: MIT Press, 2016.
[27] C. M. Bishop and N. M. Nasrabadi, Pattern Recognition and Machine Learning. New York: Springer, 2006.
[28] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," Trans. Association Computational Linguistics, vol. 5, pp. 135–146, 2017.
[29] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv:1301.3781, 2013.
[30] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in Proc. EMNLP, 2014, pp. 1532–1543.
[31] R. W. Sinnott, "Virtues of the Haversine," Sky and Telescope, vol. 68, no. 2, p. 158, 1984.
[32] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," IEEE Pervasive Computing, vol. 7, no. 4, pp. 12–18, 2008.
[33] M. Satyanarayanan, "Mobile computing: the next decade," ACM SIGMOBILE Mobile Computing and Communications Review, vol. 15, no. 2, pp. 2–10, 2011.
[34] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016.