Automation in Construction 174 (2025) 106176

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Data-driven framework for pothole repair automation using unmanned ground vehicle fleets

Shripal Mehta, Abiodun B. Yusuf, Sepehr Ghafari *

School of Built Environment, Engineering and Computing, Leeds Beckett University, United Kingdom

A R T I C L E I N F O

Keywords:
Pothole repair
Automation
UGV
Fleet scheduling
Fault tolerance
Data analysis and visualisation

A B S T R A C T

Traditional pavement repair techniques are time-consuming, labour-intensive, prone to errors, and expose manpower to high-risk road traffic conditions. This paper proposes a data-driven solution for planning and automating the repair process for road potholes using a fleet of unmanned ground vehicles (UGVs). The project encompasses data mining, developing software tailored for fleet management, and enhanced fault tolerance. Additionally, it incorporates the integration of digital twins for advanced simulation purposes. The methodologies involve cross-industry standard processes for data mining (CRISP-DM) and preparation combined with rapid application development (RAD). To optimise repair schedules, the system takes parameters like fleet size, payload capacity, and material requirements based on pothole dimensions. This data-driven project concludes from simulations that a neighbourhood can be patched about 40 % faster and optimised to achieve a 12.5 % reduction in robot inter-travel time using three UGVs per defined residential area of 100,000 m^2 instead of two UGVs in the fleet.

1. Introduction

Despite the world transitioning into the era of smart cities, road surface defects are still a persistent challenge that needs significant attention. Potholes are a nuisance for drivers, posing a considerable threat to their vehicles and pedestrians. According to an annual independent survey conducted by The Asphalt Industry Alliance (AIA), the number of pothole-related breakdowns attended by Royal Automobile Club (RAC) Ltd. patrols reached a three-year peak in 2021 with 10,123 incidents (an average of 27 per day) [1]. The RAC Pothole Index reports a staggering six potholes on average per mile on council-controlled roads in England and Wales [2]. Vehicles can lose control upon sudden impact with potholes, leading to accidents. Even small potholes pose great danger for cyclists and motorcyclists since they potentially cause them to fall, swerve manoeuvres, or suffer tyre blowouts that increase the risk of collisions. The UK government has estimated 4775 accidents in 2021 because of unseen potholes [1]. Beyond safety, they adversely impact the driving experience and increase the wear and tear on the vehicle's suspension system [1]. The financial repercussions of potholes are considerable. The RAC reported increased pothole-related breakdowns by a third from 2022 to 2023 [3]. Decades of under-investment in local road maintenance have created a substantial backlog of repairs. These breakdowns have resulted in high repair costs for drivers and loss of productivity [3]. Local authorities, often facing budget constraints, struggle to keep pace with the constant formation of new potholes due to harsh weather, heavy traffic volume, and inadequate drainage systems [4]. Furthermore, if not dealt with in time, potholes contribute to the overall deterioration of road surfaces, demanding more extensive and expensive roadworks in the future [4,5].

Nevertheless, despite the numerous attempts to curb this problem, it has yet to be solved. The recent spike in road traffic volumes has further increased estimated costs for fixing potholes across nations. The UK government and state councils have allocated a significant amount of £5 billion for road maintenance and repairs until 2025 [2,6]. Traditional pothole repair methods in the UK are often surrounded by limitations and inefficiencies of varying complexity, as highlighted in [7]. The manual human-on-ground method, the most common approach, has limitations. It is usually time-consuming, disrupts the ongoing traffic and puts the labourers in high-risk areas on the roads. These limitations highlight the need for innovative solutions in this domain. This project, therefore, seeks to explore a novel solution by harnessing automation technology. The proposed solution – A fleet of automated pothole repair bots – can be deployed to address the limitations of current methods and potentially revolutionise pothole repair across diverse road networks

* Corresponding author.
E-mail addresses: a.b.yusuf@leedsbeckett.ac.uk (A.B. Yusuf), s.ghafari@leedsbeckett.ac.uk (S. Ghafari).

https://doi.org/10.1016/j.autcon.2025.106176
Received 29 October 2024; Received in revised form 22 March 2025; Accepted 29 March 2025
Available online 3 April 2025

with varying extents. It aims to establish a more efficient, cost-effective, and ultimately safer approach to maintaining the road infrastructure.

1.1. Aim and objectives

This project aims to develop a data-driven framework for automating the pothole repair process using a fleet of UGVs and visualise the route planning and workload scheduling for effective management. The key objectives include relevant data collection and modelling, developing a pothole repair planning system to plan the routes for UGVs, analysing and visualising the different KPIs of the planned routes, and incorporating a fault tolerance mechanism to focus on accomplishing the repair jobs despite any bot failures.

2. Research background

Urban town planning is evolving with technological advancements, leading to enhanced societies and increased infrastructure demands. Road maintenance is crucial for smart cities; however, traditional pothole detection and repair are time-consuming, labour-intensive, and prone to errors [8]. These inefficiencies lead to increased road and vehicle damage, higher insurance claims, and potential accidents, emphasising the need for better road maintenance [9]. Reactive maintenance is economically burdensome and disrupts daily activities, businesses, and the flow of commerce. Smart cities offer solutions to these challenges by leveraging technology to improve efficiency and sustainability [10]. Wallin notes that digital twins — virtual replicas of physical infrastructure — can proactively monitor road conditions, predict problems, and optimise maintenance [10]. Simondetti et al. [11] highlight the potential of autonomous robotics in the architecture, engineering, and construction (AEC) industry, citing Robotiz3d's use of AI and robotics for pothole detection and repair. Recent research has designed pothole repair robots using image processing, neural networks and artificial intelligence [12–14]. However, as demonstrated by this project, a fleet of autonomous robots presents a highly promising solution, leveraging distributed parallel execution to achieve results within a single night, in the most literal sense.

This section provides a comprehensive overview of the research background pertinent to the study. In Section 2.1, Road surface defects, various types of road surface defects are examined, emphasising their causes and implications. Section 2.2, Conventional repair techniques, reviews traditional road repair methods, with their methodologies and limitations outlined. Section 2.3, Special purpose vehicles (SPVs), explores the role and functionality of SPVs in road maintenance. Section 2.4, a comparison of existing automation pothole repair solutions, presents a comparative analysis of current automated pothole repair technologies, with their effectiveness and technological frameworks assessed. Section 2.5, digital twin integration, discusses incorporating digital twin technology in road maintenance, illustrating its potential benefits and applications. Finally, in Section 2.6, Research gaps, the existing research gaps are identified, setting the stage for the contributions of this study.

2.1. Road surface defects

Potholes are basin-shaped road defects with a depth of 0.04 m and a width of 0.3 m [1,15]. Hafezzadeh et al. [15] define a pothole as "a localised deterioration of road pavement resulting from a loss of material or depression in the pavement surface." Other defects include cracks, dents, fissures, and ruts, varying in structure, impact and severity. All small or large defects should be repaired promptly [5]. Moisture and surface water seep through even the smallest of cracks to the lower layers of the road pavement structure. This, added to the temperature fluctuations, leads to freeze-thaw cycles, which negatively impact the strength of the foundation layers, eventually leading to pavement cracks and, ultimately, potholes [5,15,16]. With thinner surfaces and less stringent standards, rural roads and city streets are particularly vulnerable to deterioration. Ignored potholes worsen rapidly [5]. Therefore, timely intervention is crucial to restoring pavement condition and extending pavement life, avoiding recurrent rehabilitation operations and resulting in significant cost savings [17].

2.2. Conventional repair techniques

Dense-graded asphalt concrete is a key material in road construction and maintenance due to its excellent binding properties, lead-bearing capacity and durability [7]. Patching mixtures are classified into Cold Mix Patching (CMP) and Hot Mix Asphalt (HMA). CMP applied at cold temperatures (suitable for countries with cold climates) has less environmental impact and requires minimal rolling and compaction time before service. Still, it is less durable and has lower load-bearing capacity and crack resistance [7]. HMA, requiring temperatures up to 175 °C, requires longer rolling and compaction time before service but is used as a robust structural layer in new pavement construction and rehabilitation operations [7,15]. Common pothole filling techniques include Throw and Go, Throw and Roll, Edge Seal, Spray-Injection, Semi-Permanent, and Permanent [7,15]. Hafezzadeh et al. [15] elaborately compare the effort, time, equipment, and labour required and the life span of the resultant repaired pothole from each of these methods. However, these methods rely heavily on the human task force, putting them at risk of accidents. In addition to the risks, the precision and consistency of the repair can be prone to errors and inaccuracies [7,15]. This has led to several attempts to utilise robots capable of performing specific tasks with high precision while mitigating associated risks. These robots may operate autonomously or may also be controlled remotely.

2.3. Special purpose vehicles (SPVs)

Automated pothole repair has been attempted several times in the past, with successful examples including (see Fig. 1):

(a) Intelligent Pothole Repair Vehicle (IPRV)
(b) ARRES Prevent
(c) Pothole Pro
(d) Pothole Patrol
(e) Automated Pothole Filling Machine
(f) Autonomous 3D Printing Robot

2.3.1. Intelligent pothole repair vehicle (IPRV)

The concept of an autonomous pothole repair vehicle was first proposed by Homji [13]. This unmanned ground vehicle (UGV) was designed to detect and fill potholes without operator assistance, aiming for minimal cost and complexity. See Fig. 1 (a) for reference. It used a feedback mechanism for position and direction control, a TCP-based network interface for remote operation, and software to control the hardware. Safety measures were included to halt operations in case of communication loss or network lags. The SPV could detect potholes deeper than 0.02 m.

The SPV was built on an electric wheelchair, using its wiring schema for position and path control. It featured a filler tank for repair material and infrared distance sensors for accurate pothole detection and filling. When the material level matched the road surface, the infrared sensor triggered the motor to close the filler tank valve, completing the repair.

2.3.2. ARRES prevent

Robotiz3d Ltd. developed the world's first AI-based autonomous pothole repair vehicle, ARRES Prevent, starting in 2020. ARRES, short for Autonomous Road Repair System, completed its first test run outside the lab on March 6, 2024 [18,20–22]. Fig. 1 (b) is a representational image of the ARRES Prevent.

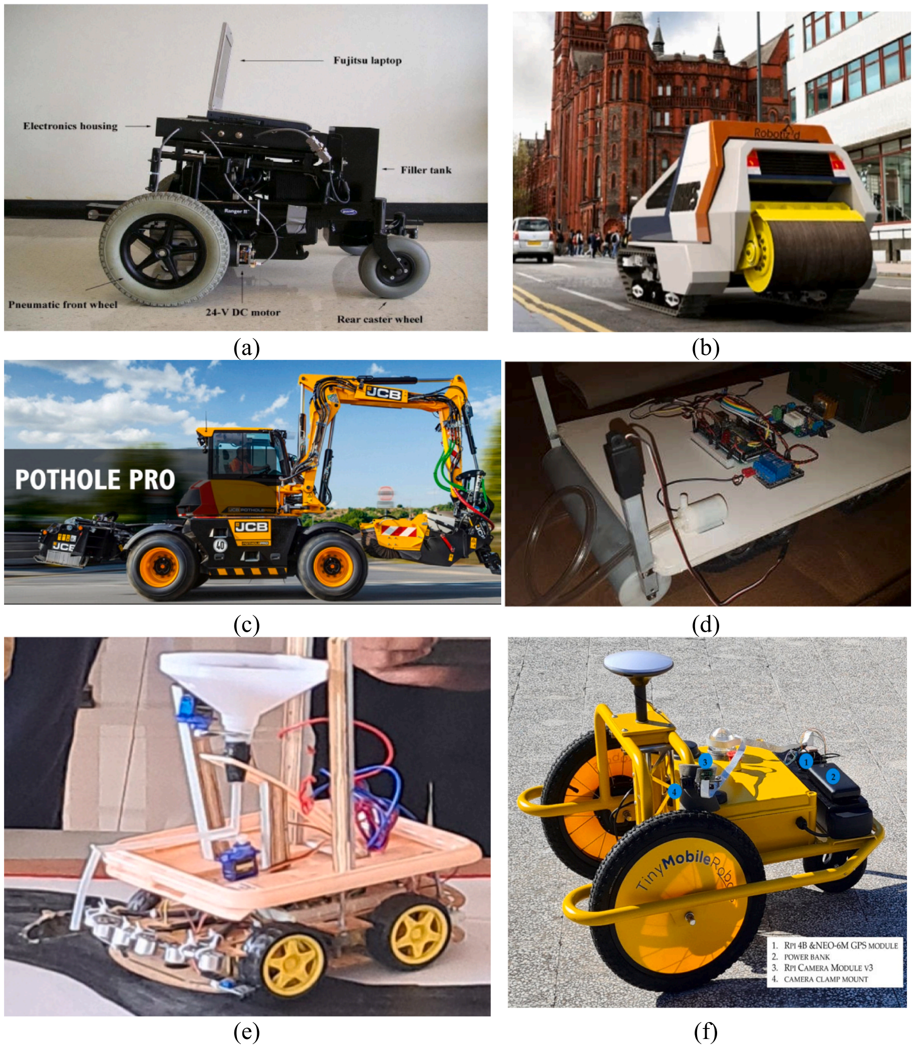


Fig. 1. Examples of automated pothole repair vehicles: (a) Side view of the IPRV [13], (b) ARRES Prevent [18], (c) JCB's Pothole Pro [19], (d) Side view of Prototype of Pothole Patrol [12]. (e) Automated Pothole Filling Machine [24] (f) Autonomous 3D Printing Robot [25,26].

Using advanced detection and repair technologies, ARRES can analyse and predict road defects' severity with laser scanners, LiDAR cameras, and RGB + D cameras. Using AI, it characterises, identifies, and detects cracks, potholes, and other defects [23]. The system fuses data from various sensors and applies over 30 algorithms for image processing, machine learning, and deep learning to measure crack depths accurately, irrespective of the viewing angle [23]. This SPV fills identified cracks to prevent water seepage and pothole formation, aiding local authorities in preventative road maintenance [14]. This extends the lifespan and safety of asphalt roads, reduces repair time and costs, and minimises carbon emissions and material waste, making it a sustainable solution for future smart cities [14,18].

2.3.3. *Pothole pro*

J C Bamford Excavators Ltd., one of the top three global construction equipment manufacturers, introduced the JCB Pothole Pro in 2021 [19]. See Fig. 1 (c) for reference. This 3-in-1 machine repairs potholes efficiently and economically, eliminating the need for extra manpower. It completes pothole repairs in 8 min and can cover 250 m^2 per day [19]. The Pothole Pro [19], operated by one person from a cab, features a 0.6 m planer with hydraulic depth control and self-levelling for consistent performance. It uses a 0.4 m cropping tool with a 360° tilt rotator for uniform hole profiles and can work near ironworks. The machine also has a 1.2 m wide sweeper collector with dust suppression for cleaning and recycling materials.

The Pothole Pro can also be fitted with various attachments for other tasks. It has 360° visibility and 4-wheel steering, which enhances safety by eliminating manual labour and Hand-Arm Vibration Syndrome (HAVS) risks [19].

2.3.4. *Pothole patrol*

The Pothole Patrol system, developed by Ali et al. [12], is an innovative road maintenance solution using autonomous vehicles with advanced sensors. It succeeds Leo Mara's Rapid Road Repair Vehicle system and addresses the limitations of manual inspections by providing continuous road surface monitoring and real-time data collection [12]. Fig. 1 (d) shows a side view of the prototype for the Pothole Patrol SPV.

Equipped with a chassis like an RC tank, custom roller, Arduino board, IR sensors, DC motors, accelerometers, GPS sensors, mini water pump, and RC linear actuator, the Pothole Patrol uses Visual Basic and Visual Studio software [12]. Accelerometers collect vibration data, which machine learning models analyse to detect potholes and other road defects. The system includes two software interfaces: one for users to report potholes and another for administrators to update the sensor-detected pothole database. Upon assignment, the SPV autonomously locates the pothole, dispenses repair material, stops when the surface is level, and runs the roller for compaction. The system was developed for under $650 [12].

2.3.5. Automated pothole filling machine

Chavan et al. [24] developed an automated pothole-filling machine using the line-following principle for navigation. Pothole detection utilises infrared and ultrasonic sensors, measuring distances between 0.2 m and 0.8 m with an accuracy of 0.003 m. Mounted at the vehicle's front, these sensors detect potholes by identifying increased distances from the default value, triggering a light signal and halting the car via an Arduino microcontroller. Fig. 1 (e) shows a prototype of this SPV.

A servo motor then activates the filling mechanism, dispensing material from a container through tubes controlled by the servo motor. Once the pothole is filled to the expected level, the valve closes, and the vehicle moves forward. A flap at the vehicle's tail levels the filled pothole. The authors highlight advantages such as improved efficiency, reduced labour, fewer errors, and potentially more durable repairs, though details on dispense time and repair duration are unspecified [24].

2.3.6. Autonomous 3D printing robot

Bruno et al. [25,26] developed a small autonomous robot to detect and repair road potholes using photogrammetric techniques and 3D printing. This SPV was created as part of the European Commission's Horizon 2020 InfraROB project (Grant Agreement N. 955,337) [25,26]. This small (0.828 m in length, 0.688 m in width, and 0.491 m in height) robot SPV is adapted to accommodate necessary sensors and the material extrusion system. It has an integrated 3D printer and a tank to hold the filling material with a capacity of 5 kg.

The robot moves in circles (at 0.4 m/s) around the pothole, capturing 2D images from different angles using a low-cost camera and recording the respective location details through a GPS sensor [25]. Using Raspberry Pi hardware and Bentley ContextCapture software, it employs photogrammetry and aero-triangulation techniques to reconstruct the 3D geometry of potholes. At the end of the 3D reconstruction process, after about 40 min per pothole, it outputs an STL file, which is required for 3D printing. This information calculates the volume and, thus, the backfill material needed to repair the pothole. The integrated 3D printer then dispenses material from the extruder set on the rear end of the SPV. See Fig. 1 (f) for reference.

This SPV utilises a specific cold-asphalt mixture composed of 100 % reclaimed asphalt pavement (RAP) and recycling agents [25,26]. A comparative study of various potential mixtures elaborately discusses the benefits of recycling agents and the potential for reuse in road construction [27]. In addition to the several laboratory and in-situ tests in favour of the chosen cold-asphalt mix, a few major reasons include — (i) compatibility and suitability for 3D printing, (ii) to avoid heating systems and relevant complexities in the robot design, (iii) RAP is low-cost and require less labour and equipment, (iv) environmentally beneficial since it promotes recycling, reduces CO^2 emissions, and conserves resources, and (v) stability under traffic load and any weather conditions. Table 1 compares these IPRVs mentioned above against various parameters.

2.4. Research gaps

So far, various attempts have been made to automate the pothole repair process. They have typically used a single vehicle at a time, focusing on individual repairs [12,13,18,19,24]. The next step is using a fleet of autonomous special-purpose vehicles for parallel execution. While fleets are common in cargo logistics and parcel delivery [28,29], this project explores integrating autonomous pothole repair with fleet scheduling and management. This approach allows for workload redistribution in case of failures, enhancing efficiency and precision while mitigating risks and errors.

2.5. Digital twin integration

As defined by IBM [30], a digital twin is a virtual representation of a physical object or system, updated with real-time data and using simulation, machine learning, and reasoning for decision-making.

Digital twin technology is extensively researched for buildings, bridges, and cities to support sustainable smart cities [31,32]. It allows for virtual experimentation and visualisation before practical implementation. Integrating pothole-repair planning with a digital twin offers several advantages by simulating repair operations while changing various parameters and conditions, such as fleet size, UGV capacities

Table 1
Comparison of existing automated pothole repair solutions.

	Intelligent Pothole Repair Vehicle (IPRV)	Robotiz3d (ARRES Prevent & ARRES Ultra)	Pothole Pro	Pothole Patrol	Automated Pothole Filling Machine	Autonomous 3D Printing Robot
Year	2005	2020	2021	2022	2023	2023
Key Developers	Homji, R.A.M.	Robotiz3d Ltd.	J C Bamford Excavators Ltd.	Ali et al.	Chavan et al.	Bruno et al. (as part of the Horizon 2020 InfraROB project)
Technology Used	Infrared distance sensors, TCP network	AI, LiDAR, RGB + D cameras, laser scanners	3-in-1 design (Cut, Crop, Clean)	RC tank Arduino board, IR sensors, GPS, accelerometers	Line following principle, IR and ultrasonic sensors	Raspberry Pi, 3D printing, Photogrammetry, GPS
Autonomous Operation	Yes	Yes	one operator required	Yes	Yes	Yes
Detection Mechanism	Infrared sensors	Laser scanning, LiDAR, AI	Manual operation by an operator	IR sensors, Accelerometer	Ultrasonic sensors, IR sensors	Photogrammetry
Repair Mechanism	Filler tank with motorised valve	AI-based detection and repair process	Cut, Crop, Clean mechanism	Dispense repair material, compact with roller	Dispense repair material, level with flap	Detect, assess, 3D reconstruction & printing, extrusion, and monitor
Detection Capability	0.02 m	Not specified	NA	0.003 m	From 0.03 m away.	1.0 m - 1.4 m
Estimated Repair Time	Not specified	Not specified	8 min per pothole	Not specified	Not specified	2 min per pothole; immediate reopening of roads
Repair Efficiency	Minimal cost and complexity	Preventative maintenance to reduce cost and emissions	High efficiency, 250 m^2/day	Real-time data collection	Improved efficiency, reduced labour	High; monitored for ~150 days
Cost of Development	Not specified	Not specified	Not specified	Under $650	Not specified	Not specified
Additional Features	Remote operability, safety measures	Prevents pothole formation, sustainable	Multiple attachments, 360° visibility, 4-wheel steer	User reporting interface, administrative interface	Arduino microcontroller, servo motor control	Improves safety by avoiding human involvement on roads.

(including payload, speed, fuel consumption or battery life, and other operational constraints.), and weather conditions to name a few. The reader is notified that the capabilities of the framework have been simplified. The scope of this work does not include the battery or fuel consumption of UGVs or weather conditions due to the complexity of integrating weather APIs and also to enable ease of reproducibility of implementing the framework by the wider research community, as demonstrated in Fig. 5. Digital twins can be advantageous for experimenting with and validating specific scenarios, such as bot malfunctions. Additionally, iterative testing, while changing different parameters in each test, and data analysis can help optimise bot efficiency. Through simulations, it is possible to identify potential risks and develop effective solutions, mitigating the impact of unexpected issues.

3. Method

3.1. Data mining

Several frameworks are widely used for data mining projects. Table 2 compares the steps involved in each of the below frameworks:

1. Knowledge Discovery from Databases (KDD) [33]
2. Sample, Explore, Modify, Model, Assess (SEMMA)
3. Cross Industry Standard Process for Data Mining (CRISP-DM) [34]

This project employs the CRISP-DM methodology for data mining activities. Introduced in 2000, it has become the de facto standard and a domain-independent process model for data mining projects [34].

Fig. 2 shows these steps and the flow of execution.

While incorporating data mining, software development, and simulation in a digital twin environment, this project will follow the principles of these methodologies but remain flexible to adapt as needed. The goal is to leverage the best practices and customise them according to the project requirements. Inspired by those mentioned above, the following steps are followed as part of this project.

3.2. Data acquisition and staging

This stage involves gathering data. Many councils provide datasets on pothole repairs, enquiries, and expenditures. If available, download datasets with pothole locations (latitude and longitude), dimensions, and severity. If not, collect primary data from a neighbourhood with these attributes – location coordinates (latitude, longitude), surface defect type (pothole, crack, dent, etc.), and dimensions (length, width, depth) of the road surface defect. Finally, place the files in the project's directory for further use.

3.3. Feature selection and data preprocessing

After staging the data, only the necessary attributes were selected. Remove irrelevant, redundant, or noisy features to reduce storage and computational costs [36,37].

The data preprocessing stage involves preparing the data for effective use by the project modules, focusing on data cleaning, transformation, and reduction. The data cleaning step ensures that the data is free from discrepancies. It includes removing duplicates to avoid skewed results and overrepresentation [38], handling missing values by either dropping records or imputing missing values with mean, median, mode, or those generated by employing machine learning algorithms [39,40], and validating the data to ensure that the data adheres to the defined rules for accuracy and reliability [38,41]. Data transformation transforms data into a format that simplifies the next steps, like data mining and analysis [40]. This includes creating new, calculated fields, converting data types, standardising formats and even joining datasets and filtering or aggregating attributes [38]. The data reduction step is used to reduce the size of the transformed data for effective storage and management [40].

Table 2
Comparison between KDD, SEMMA, and CRISP-DM [35].

KDD	SEMMA	CRISP-DM
–	–	Business Understanding
Selection Preprocessing	Sample Explore	Data Understanding
Transformation	Modify	Data Preparation
Data Mining	Model	Modelling
Interpretation/Evaluation	Assessment	Evaluation
–	–	Deployment

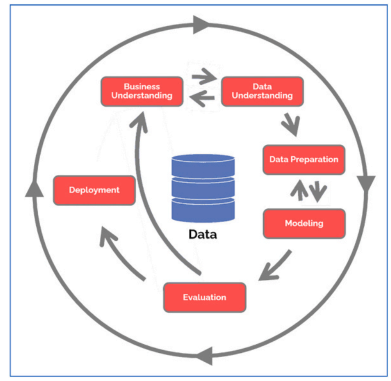


Fig. 2. CRISP-DM methodology [35].

3.4. Exploratory data analysis

After validating data quality, Exploratory Data Analysis (EDA) was performed to identify patterns and anomalies [42]. This step helps us better understand the dataset before its use. EDA includes calculating mean, median, mode, standard deviation, and variance for numerical attributes. It also involves visualisations like correlation heatmaps, histograms, and scatterplots to explore relationships between attributes [42].

3.5. Software design and development

Once data is staged and understood, the software solution will be designed and developed. Despite Agile methodologies being common in software development, this project employs the rapid application development model due to time constraints (short timelines) and the need for iterative prototyping cycles of "build – test – refine" through minimal planning [43]. This methodology focuses on rapid prototyping. It allows for the simultaneous development of several modules while incorporating changes to refine them. The RAD model is shown in Fig. 3.

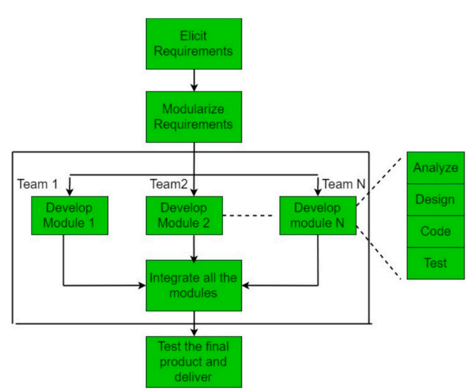


Fig. 3. Rapid application development methodology [44].

(See Fig. 4.)

3.6. Data modelling

The data model structures and organises data used by the software application. Given the project's geographical nature, a Geographic Information System (GIS) approach will manage spatial data related to potholes and bot operations by effectively utilising the APIs mentioned in Section 4.2.1. The commonly used Geographic Coordinate System (GCS) is EPSG:4326, also known as WGS84, which utilises a datum, prime meridian, and angular unit of measure [45].

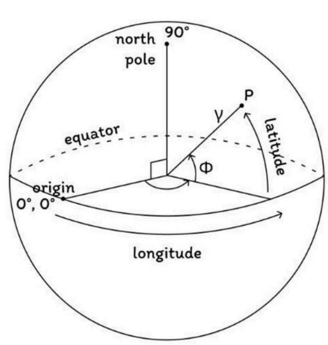
Cartographers use various projection methods to map the geoid shape of the Earth accurately, each with benefits and limitations [46]. Today's widely used projected coordinate system (PCS) is EPSG:3857, or Web Mercator, employed by ArcGIS, Google Maps, and OpenStreetMap for accurate angles and distances and easy zoom configurations [45].

GIS data will be stored in formats like GeoJSON, adhering to GIS standards [45]. JSON will be used for NoSQL geographic features, and CSV will be used for relational data. A specific map projection system, such as EPSG:4326, will be chosen for consistent visualisation and analysis, ensuring accurate geographical representation [45,47].

3.7. Planning phase and system modules

The system functionalities are divided into manageable modules such as fleet management, route planning, fault tolerance, and data visualisation. The fleet management module creates and manages (assigns repair jobs and monitors) a fleet of UGVs. The route planning module is responsible for planning optimised routes for each UGV in the fleet. The fault tolerance module is only invoked in case of bot failures. In a scenario where a UGV becomes non-operational during trips or while carrying out repair jobs, this module computes the pending work at the time of failure and redistributes the workload among the remaining UGVs in the fleet. The data visualisation module visualises the pothole repair jobs, UGV assigned paths, and other data analytics computed on top of it.

Unified Modelling Language (UML) was used to create a flowchart representing the project structure and behaviour. The flowchart, shown in Fig. 5, clarifies the system's design and aids communication between developers and other stakeholders.



GCS uses spherical coordinates (λ, Φ, γ) to represent Point P where:
λ = +/- degrees longitude
Φ = +/- degrees latitude
γ = radial distance from centre

Fig. 4. Geographic coordinate system explained.

3.8. Data visualisation

Effective data visualisation is crucial for transforming data into actionable insights that stakeholders can easily understand during and after the execution of the software solution. By leveraging various visualisation techniques such as bar charts, pie charts, plotting on geographic maps, etc., it can be possible to visualise, analyse and monitor pothole repair operations with greater efficiency and optimisation. The various visualisations implemented to achieve this objective are as follows:

1. Potholes plotted on the map.
2. Route Visualisation for each UGV.
3. Distance vs Time for each UGV.
4. Workload distribution (number of repair jobs) per UGV.
5. Total Distance travelled by each UGV.
6. Repair material deployed per UGV.
7. Payload versus time per UGV.

Each of these KPIs is visualised as part of the data visualisation phase.

3.9. Digital twin integration

This stage involves creating a digital twin of the project's road network. Tools and techniques such as Python, Webots and SUMO for this purpose were evaluated to model the road network, including potholes, the neighbourhood, and unmanned ground vehicles (UGVs). It was discovered that Webots worked best for the purpose of generating a feasible virtual representation of the area of interest incorporating pothole data such as GIS information (geo-location coordinates), pothole dimensions, and neighbourhood specifics such as road width, permissible height, junctions, radius of curvature on turns, etc. Once the digital environment is set up with the bots and potholes, the next step is to define controller logic for the bots and simulate scenarios while recording real-time metrics using sensors such as GPS, accelerometer, LiDAR, and RGB cameras.

The test scenarios include: *1. Happy scenario*: A base scenario involving pothole repairs using 'n' UGVs in the fleet, with no bot failures. *2. Bot Failure Scenario*: The base scenario, but one UGV fails during repair or transit. Thus, the fault tolerance module shall re-compute the pending workload and re-assign tasks among the remaining bots in the fleet. If there are any more failures, the same steps shall repeat unless there are no more active bots in the fleet.

4. Design and implementation

4.1. Project design

The flowchart shown in Fig. 5 describes the high-level flow of the project. The initial part of the project deals with the data handling and its understanding. The second part deals with the logical implementation of the project use case. The final part deals with integrating it with a digital twin of the road network.

4.2. Project implementation

The project is developed in a Jupyter Notebook (it can also be executed in a docker instance) using Python with relevant libraries and dependencies downloaded and installed from the internet.

4.2.1. APIs and library dependencies used

The project utilises several APIs such as OpenStreetMap (OSM) – which provides geographical information such as map objects, latitudes and longitudes, and address search; Folium – a Python library using Leaflet to display OSM maps and its objects in Jupyter notebooks; and OpenRouteService (ORS) – offers endpoints for Pelias Geocoder API and

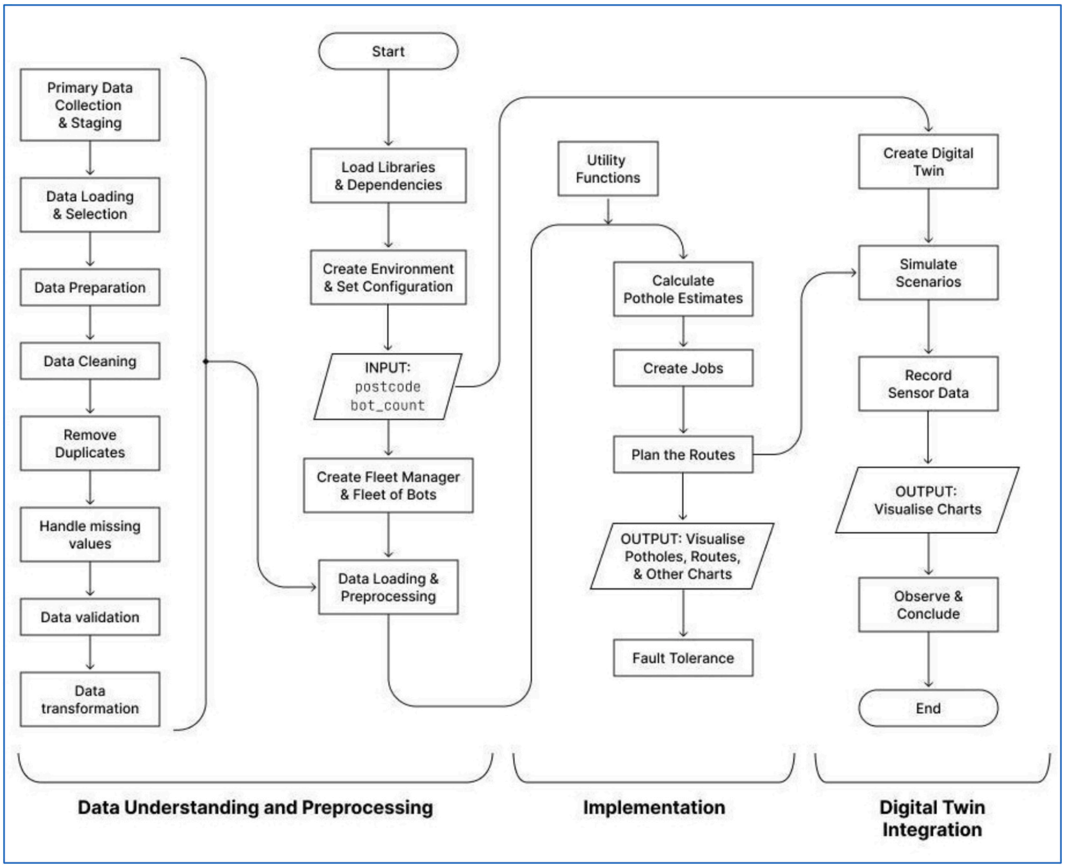


Fig. 5. Project flowchart.

VROOM API. Pelias geocoding converts geographic coordinates (latitude and longitude) to text addresses and vice versa. VROOM optimises vehicle routing using Open-Source Routing Machine (OSRM), featuring customisable vehicle and job attributes [48]. VROOM employs Kruskal's Algorithm for path-planning and route optimisation and Munkres algorithm for job assignment problems [48]. ORS is also responsible for enabling fleet scheduling for this project. A custom logic is crafted for the fault tolerance module, and the matplotlib Python library is used to generate various visualisations for observation and analysis. Lastly, Webots (an open-source software) and Python are used to create the digital twin environment.

4.2.2. User input

The project takes input for location postcode (to select a neighbourhood on the map), the number of bots in the fleet, and fleet homogeneity. Currently, the fleet of robots simulated in this study is assumed to be homogeneous, i.e. all UGVs have similar skills and capabilities for the repair jobs. In this case, all robots will attend to an existing pothole and should be able to complete a full repair from aggregate deployment to the patched surface's smoothing before moving on to the next algorithmically efficient pothole in the path-planning process. The utility of such abstraction is mainly to emphasise the feasibility of such an implementation, and indeed, homogeneity reduces computational complexity and model susceptibility to uncertainty. This can be extended (in the next versions) to accommodate scenarios where each UGV has custom skills and capabilities for granular control and dedicated task(s) implementation within the repair job scope of work (i. e. One robot is responsible for the aggregate deployment, another would be responsible for smoothing).

4.2.3. Data collection and staging

The pothole data can be crowd-sourced from web portals managed by government bodies. The dataset contains information about the pothole's geo-location, dimensions, defect type, and severity. It can be visualised using Mercator projection and OSM maps exported in GeoJSON format and saved in the <postcode>.geojson data directory. This dataset file is then loaded in the project using geopandas library.

4.2.4. Data preprocessing

The data imported (especially when crowd-sourced) needs to be preprocessed. This stage involves cleaning and transforming tasks such as removing duplicates, checking for missing values and transforming dimensions from "LxWxD" format to numeric columns "L", "W", and "D". These steps are very crucial because unclean data leads to unwanted discrepancies. If duplicates are not removed, it creates skewed data and incorrect analysis. If there is a missing value, it may lead to errors during execution. Transforming data into atomic values helps conduct mathematical computations, which would otherwise be impossible.

Fig. 6 shows how the pothole dimensions in the 'dim' attribute are transformed, as part of data preprocessing, into three columns, namely L (for length), W (width) and D (depth). These values are now stored as decimal values, making calculations for area and volume possible.

4.2.5. Resource utilisation statistics

As the data is pre-processed and cleaned, it is now ready for further computation of resource utilisation. The available data calculates the size (volume) of the road surface defects. For computational efficiency, this project treats all potholes to be cuboidal in shape. This can be replaced with advanced features such as integrating photogrammetry techniques to accurately determine the pothole volume from 2D images, as demonstrated by Bruno et al. for the interested reader [25,26]. Since the asphalt density is known to be ~2400 kg/m^3 [7], the model developed couples this value with the pothole volume to set an estimated amount of repair material required for a task run. Thus, the formula for

	defect	dim	geometry
0	pothole	1x0.5x0.2	POINT (-1.5202653014356144 53.7886094292194912)
1	pothole	0.5x1.2x0.2	POINT (-1.5185276122489597 53.7884573448203867)
2	pothole	0.5x1.2x0.2	POINT (-1.5181200061435902 53.7886537870654990)
3	pothole	1x0.5x0.2	POINT (-1.5165367499993749 53.7894171973833295)
4	pothole	1x0.5x0.2	POINT (-1.5158937220954840 53.7902010524229013)
5	pothole	2x0.5x0.2	POINT (-1.5157207769570675 53.7901240147059525)
6	pothole	1x0.5x0.2	POINT (-1.5150777490531766 53.7897200289646946)
7	pothole	1x0.5x0.2	POINT (-1.5165475271180355 53.7900878370366797)
8	pothole	1x0.5x0.1	POINT (-1.5178539965092739 53.7884356902426077)

(a)

	defect	dim	l	w	d	
0	pothole	1x0.5x0.2	1.0	0.5	0.2	POINT (-1.5202
1	pothole	0.5x1.2x0.2	0.5	1.2	0.2	POINT (-1.5185
2	pothole	0.5x1.2x0.2	0.5	1.2	0.2	POINT (-1.5181
3	pothole	1x0.5x0.2	1.0	0.5	0.2	POINT (-1.5165
4	pothole	1x0.5x0.2	1.0	0.5	0.2	POINT (-1.5158
5	pothole	2x0.5x0.2	2.0	0.5	0.2	POINT (-1.5157
6	pothole	1x0.5x0.2	1.0	0.5	0.2	POINT (-1.5150
7	pothole	1x0.5x0.2	1.0	0.5	0.2	POINT (-1.5165
8	pothole	1x0.5x0.1	1.0	0.5	0.1	POINT (-1.5178

(b)

Fig. 6. Data preprocessing: (a) Before (b) After.

this is:

$$Required\ Repair\ Material\ (Mass) = Density \times Volume \quad (1)$$

Estimated repair time can also be calculated using volume and surface area functions. These calculations are implemented in the Utils package of the project code. The repair time is treated as a function of the volume and surface area of the pothole, such that greater travelling times are required by the UGV proportionally to surface area fill and levelling of the surface-top throughout the pothole area, while more volume, naturally aligns with more material requirements and greater pouring time. It can be represented using the following equation:

$$Repair\ time\ T = (a \times v) + (b \times s) + c \quad (2)$$

where: v = volume, s = surface area, and a, b, c are proportionality constants.

4.2.6. Create repair jobs

With all the parameters captured – the defect type, defect geolocation, area and volume of the road surface defects, estimated repair material required, the estimated time to fix it, and the fleet size, the next step was to create repair jobs for each pothole. If the required repair material for a given defect exceeds the UGV's payload capacity, the job is split into smaller tasks, each limited to the payload capacity. This allows the required amount to be delivered to the location for any ratio of UGV's payload capacity to road defect size while utilising the ORS API as-is.

The nomenclature for job ID is configured to effectively analyse the time and effort required to accomplish larger potholes with complex jobs. For instance, if job_id "5" is broken into smaller jobs, the smaller jobs shall be given IDs 501, 502, 503, …and so forth.

4.2.7. Route planning and visualisation

Potholes can be visualised on a Folium map, with their respective details displayed on the mouse hovering over each pothole. Fig. 7 (a) illustrates how potholes are visualised. Details such as defect ID, defect type, area, volume, and the repair material required for each road defect are visible in the tooltip when the mouse pointer is hovered over them.

In the next step, optimised routes were planned for the UGVs using the ORS API, with the postcode's pin location acting as the base station (hence the starting and ending point) for all the UGVs.

If the count of road surface defects is very high or the total material required is high, the ORS API shall only assign jobs that can be accomplished within a single trip for each UGV in the fleet. To overcome this API limitation, the module is configured to iteratively call the API to plan successive trips for the UGVs until all pending repair jobs are assigned. For this route planning, each trip starts and ends at the base station, which is assumed to be an inexhaustible source of repair material.

The ORS API uses several parameters including but not limited to the vehicle profile (comprises of its dimensions, weight, max speed, minimum radius for curvature, etc.), traffic signals, roads and junctions involved, their load-bearing capacities, their curvature, permissible height, width, count of lanes on multi-lane highways, and valid directions (esp. for one-way roads) to compute the optimised route plan and estimated time of arrival. For simplicity, this module uses a standard vehicle profile of a car (often referred to as a "passenger car") benchmarking on the basis of efforts by Robotiz3d's SPV – ARRES Prevent [20]. A video of their working prototype is available at [49]. Effectively using the time-distance matrix and the metrics as mentioned above, the API responds with an optimised plan for all the UGVs in the fleet for the given set of jobs.

These planned routes are overlaid on the same Folium map for effective visualisation. Fig. 7 (b) shows the routes scheduled for a fleet of 3 UGVs (represented with 3 colours) assigned to repair the defects in this neighbourhood.

4.2.8. Results and discussion

The route information for each UGV and pothole can be utilised to analyse various KPIs (using relevant charts and visualisations, as shown in Figs. 8, 9, 10, and 11), such as distance vs. time per UGV, number of jobs per UGV, distance travelled by each UGV, repair material deployed per UGV, and payload carried over time by each UGV. Decision makers can use these for real time comparisons and analysis. (See Fig. 12.)

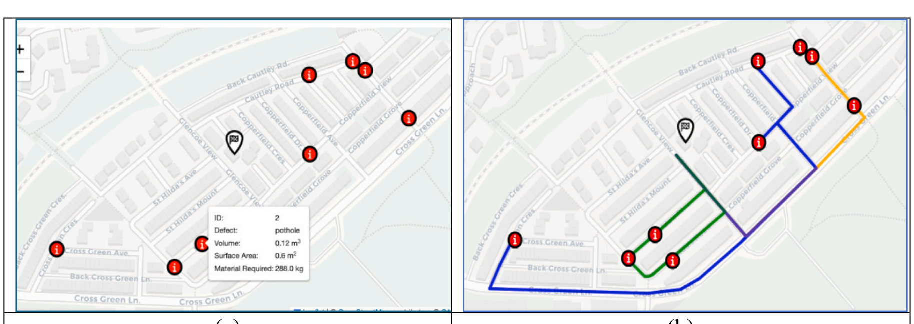


Fig. 7. Visualisations of (a) Potholes in a neighbourhood and (b) Route planning.

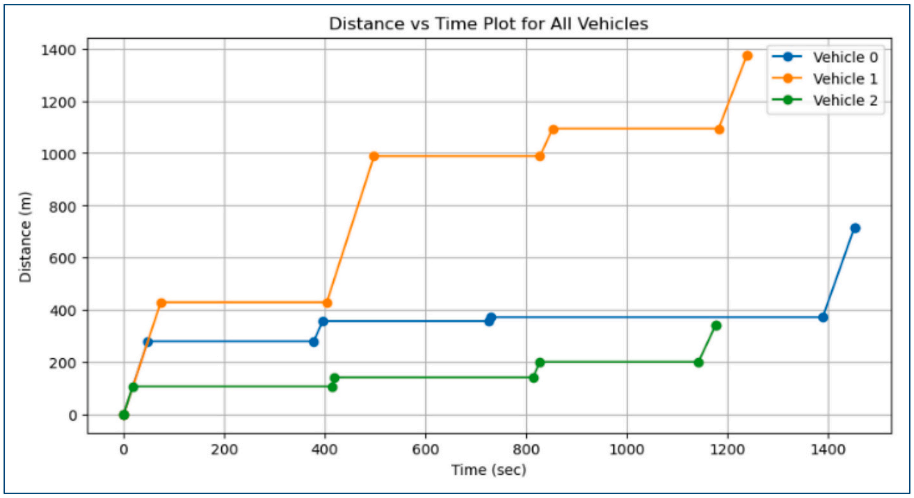


Fig. 8. Distance vs. Time per UGV.

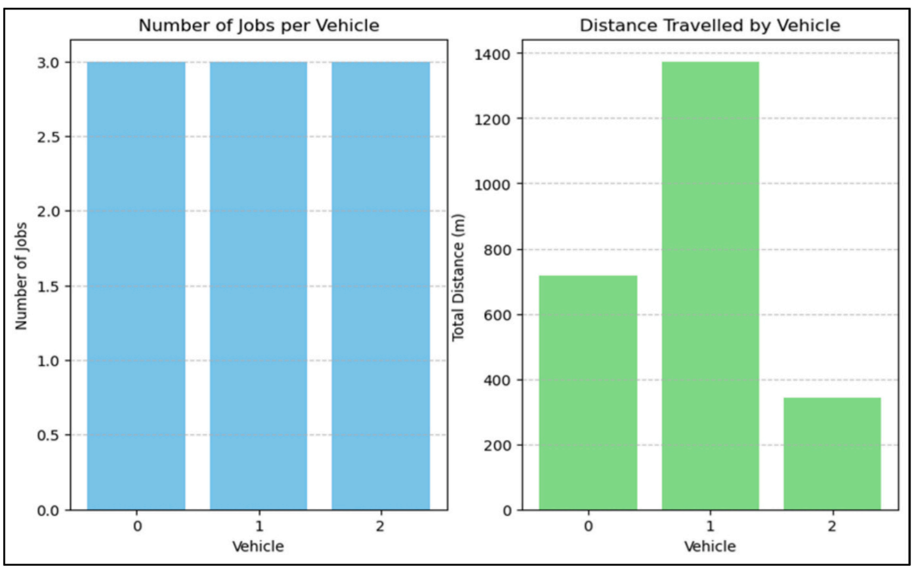


Fig. 9. Number of Jobs per UGV and Distance travelled by each UGV.

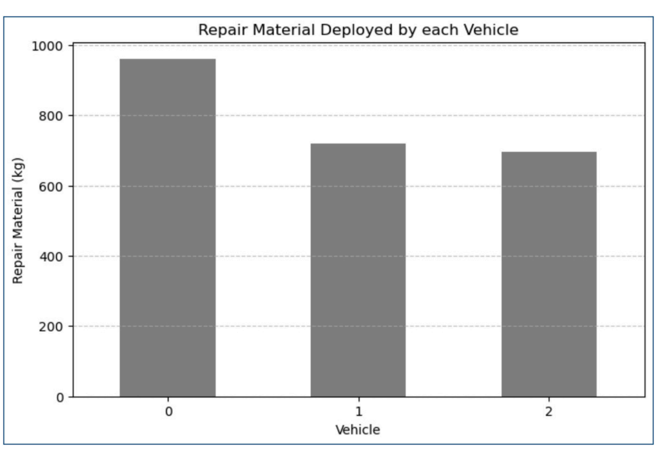


Fig. 10. Distance travelled by each UGV.

Fig. 8 shows the distance vs. time plot for each UGV. The slant line sections of the line plot denote that the bot is travelling, while the flat line sections denote that the bot is repairing a pothole since it is stagnant in a single place for that period. The chart shows that "bot 1" has travelled the longest, while the line plot for "bot 0" shows that its 3rd assignment must be a bigger pothole requiring more time to repair. "Bot 2" has travelled the least, implying that it has been assigned three potholes that lay very close to each other.

Fig. 9 shows two visualisations. The first is the count of pothole jobs assigned to each bot in the fleet, while the second shows the total distance travelled by each bot. These are different ways to show some of the information already available. Fig. 8 emphasises that various visualisation techniques can depict different or similar details to aid operational decision making. Similarly, Fig. 10 compares the amount of repair material delivered by each UGV. In Fig. 11, the line plot shows the amount of repair material carried in the UGV over time. After each repair, the UGV's payload weight is reduced by the material deployed at that pothole. Such information can support tactical supply chain decisions to ensure adequate material is available for maintaining operations during peak periods.

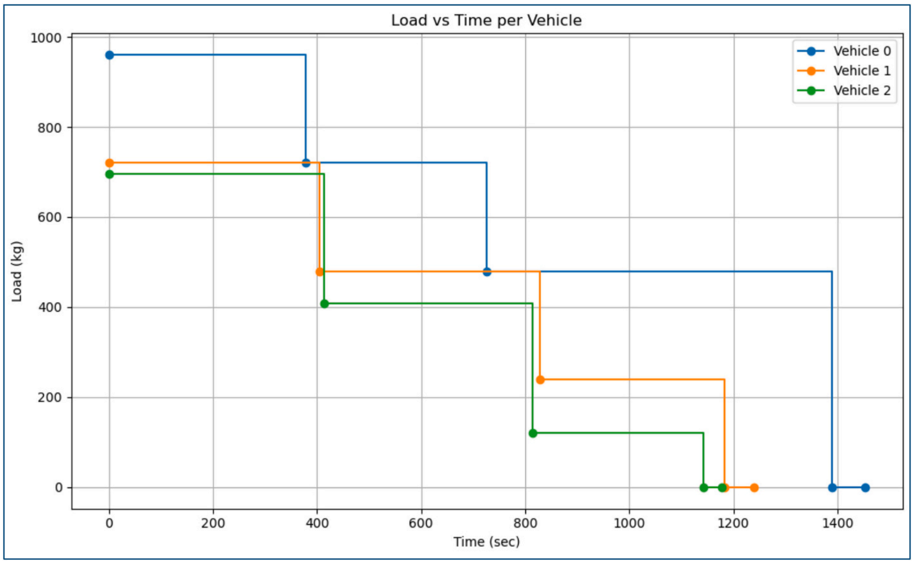


Fig. 11. Payload carried over time by each UGV.

Fig. 12. Command to use the importer.py module.

4.3. Data visualisations

Extracting meaningful insights from the visualisations is the key aspect of data-driven projects. The present work provides a framework for enabling access to various real-life data, which can be used in different decision-making scenarios. This data-driven framework allows for building visualisations using the available parameters like GPX data (location), time, distance travelled, repair material quantity and deployment per repair. It also creates possibilities for employing digital twins for life-like simulations, which can be another data source for analysis and insightful decision-making.

4.4. Digital twin integration

This project utilised Webots to create a digital twin of the road network of the neighbourhood from the corresponding OSM map. The importer.py python module of Webots is used for this. The figure below shows the command used to generate the webots world from the OSM map file:

The command takes two arguments: –input to pass the OSM map file, and –output to rename the output digital twin (webots world) file.

Upon executing the command, it runs the Python module and prints various logs on the console. The first two lines are error logs, indicating some errors while generating the start/end of a specific road. However, that is expected since we are working on a rectangular cropped portion of the real-world map, and roads can lie on or beyond the boundaries of the map in the input file, causing the importer module to face conflicting road/lane properties.

The third line is a warning log indicating that the module could not determine the country. It is believed that this could be due to the input map being very small in this use case and not including any major landmarks. However, this shortfall is surprising since it is not entirely impossible to determine the country given the geo-coordinates.

The following lines capture the success logs, verbosely indicating the count of various OSM objects encountered while parsing the input OSM file. Table 3 enlists them. Lastly, the logs also mention crucial cartographic information such as the reference coordinates, offsets, the GCS and projections used, their formats, and the units of measure. Finally, the conclusion of the program is acknowledged with “Done.”. Figs. 13 and 14 show the digital twin generated for the area under consideration:

4.5. Fault tolerance

Another crucial objective is to build a robust fault-tolerant system. This feature shall make the system resilient to bot failures amidst task execution. Although bot-failure detection can be achieved through sensor technology and validated through simulations and live tests, this feature currently focuses notionally on handling the failure. Here, a “failure” was defined simply as a bot not working. Thus, in case of a bot

Table 3
List of OSM objects parsed from input file and generated in output file successfully.

OSM Objects	Count
Roads	105
Crossroads (junctions / round-abouts)	59
Buildings	369
Trees	15
Barriers	2
Rivers	0
Areas (forest, water bodies, etc.)	0
Parking lines	0

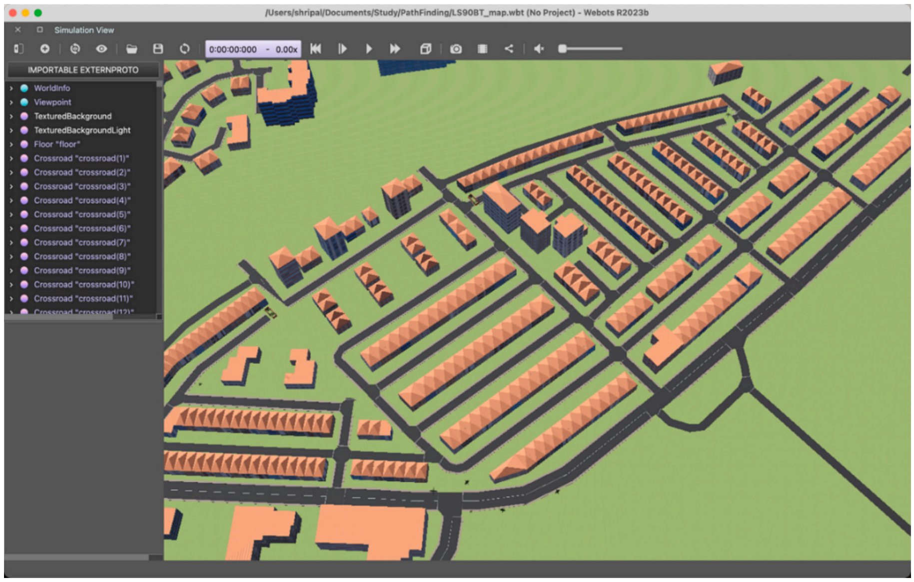


Fig. 13. Digital Twin of the neighbourhood (~100,000 m^2 residential area) of LS90BT (Top View).

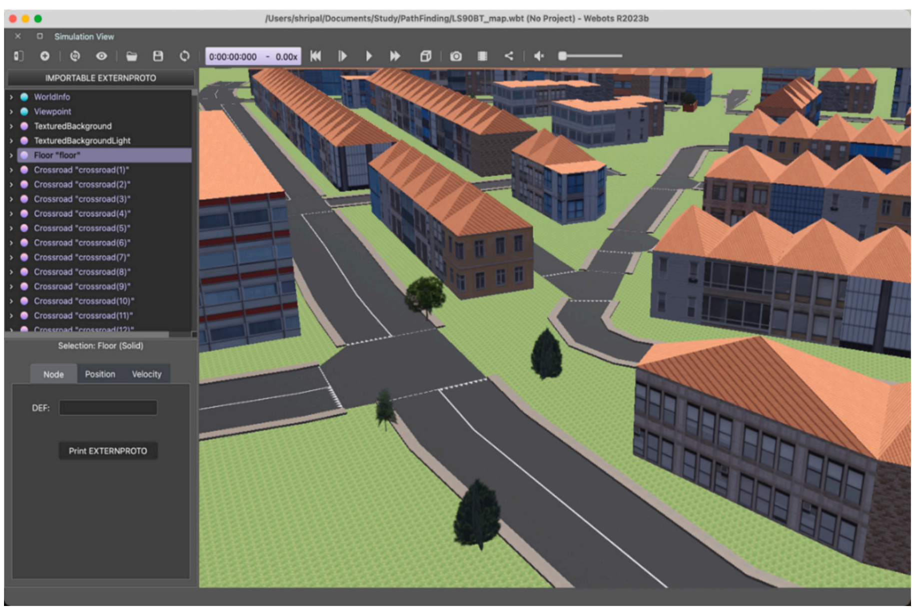


Fig. 14. Digital Twin of the neighbourhood (~100,000 m^2 residential area) of LS90BT (close-up view).

failure, the system records the "bot_id" and the "time_of_failure" (in seconds). Then, reassigns the pending jobs among the remaining bots in the fleet and plan the routes again.

4.6. Optimisation through comparison

The project enables optimisation through planning and performance comparisons based on varying fleet sizes. For example, for the same set of potholes, when routes are planned for fleets of three and two UGVs while keeping all other parameters and constraints constant, analysing the total time and distance travelled by the UGVs allows for effective comparison and decision-making. Practically, this decision-making process depends on several factors, including the severity and urgency of the potholes and the cost, fuel consumption, and maintenance of the UGVs.

The total distance travelled, the total time spent in the repair process, and the number of jobs fulfilled by each UGV using a fleet of 2 UGVs are shown in the charts in Fig. 15 and Fig. 16:

These charts can be compared with the earlier charts in Fig. 8 and Fig. 9, which denote the plans and estimates for a fleet of 3 UGVs) for data-driven decision-making.

This project can optimise resource utilisation through effective planning. Table 4 shows a comparison between the two scenarios. The total distance travelled by 3 UGVs in the first experiment was 2450 m. While the total distance travelled by 2 UGVs to fulfil the same repair jobs was 2800 m. Similarly, the total time spent travelling (not considering the time spent in repairing at each site), also referred to as "inter-travel time", in the first experiment using 3 UGVs, was 410 s. While the inter-travel time for 2 UGVs was 680 s. Upon comparison, it is evident that the fleet of 3 UGVs can accomplish the same jobs by travelling 12.5 % less distance and spending 39.7 % less time compared to the fleet of 2 UGVs. In fact, the total time spent using 3 UGVs is 1440 s. This is approximately 40 % less than the total time spent using 2 UGVs. Fig. 17 visually illustrates the comparison between cases 1 and 2 from Table 4.

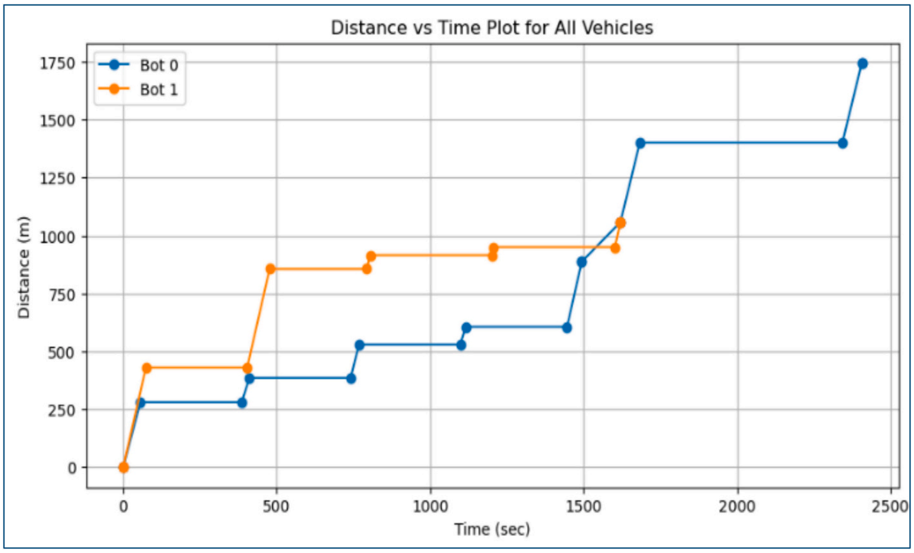


Fig. 15. Distance vs Time chart for Fleet of 2 UGVs.

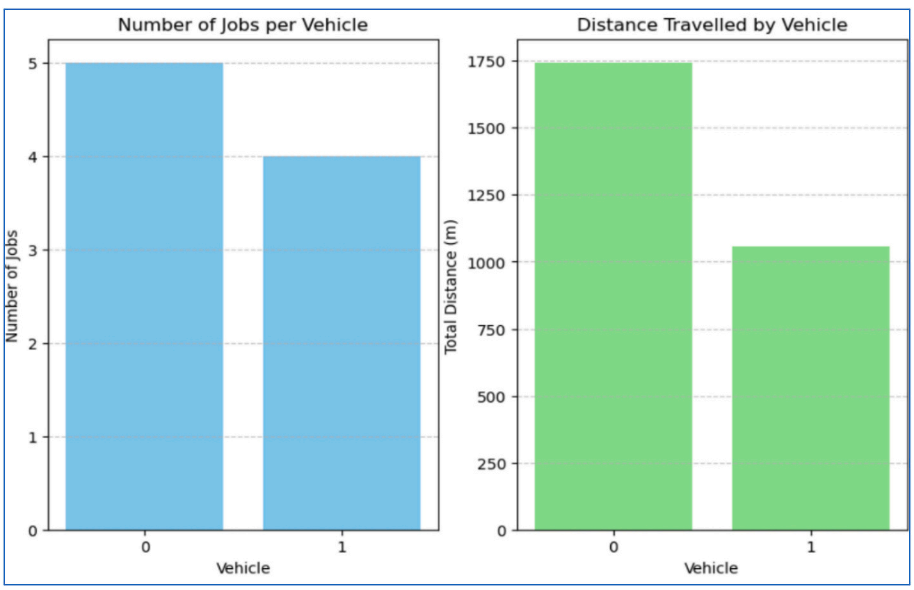


Fig. 16. Number of Repair Jobs and Total distance travelled by each UGV in the fleet.

Table 4
Comparison between the repair plans for the neighbourhood of LS90BT using 2 UGVs and 3 UGVs.

Metric	Case 1	Case 2	% change (from Case 1 to Case 2)
Fleet Size (number of UGVs in the fleet)	2	3	50 % more
Total Material Delivered	2470 kg	2470 kg	0 %
Total Time spent	2440 s	1440 s	40.98 % less
Total Inter-Travel Time	680 s	410 s	39.7 % less
Inter-Travel Time per UGV	340 s	136.67 s	59.8 % less
Total Distance Travelled	2800 m	2450 m	12.5% less
Distance Travelled per UGV	1400 m	816.67 m	41.67 % less
Total Trips Planned (planning each trip involves computational resources too)	2	1	50 % less
Number of Trips per UGV	1.5	1	33.33 % less

The repair aligns with the UK default standards – CD 226 [50] and CD 236 [51] from Design Manual for Roads and Bridges by Highways England. Furthermore, the bituminous mixtures used for maintenance operation also align with BS EN 12697 [52] and CM 231 [53] of the Highways England where specific mention is given to surface repairs, including potholes.

4.7. Benefits and impact

The project demonstrates optimisation of time and cost. It encourages deploying more UGVs to accomplish the repair jobs in less time [28]. Although this logic is self-evident, in practice, it is equally possible that prevailing operational conditions might warrant a 2 UGV deployment as outlined in the comparison above. Moreover, it promises to provide increased safety by refraining from putting humans (labourers) in accident-prone zones on the roads, as highlighted in the current state-of-the-art per [26].

As demonstrated by the implementation above, this data-driven

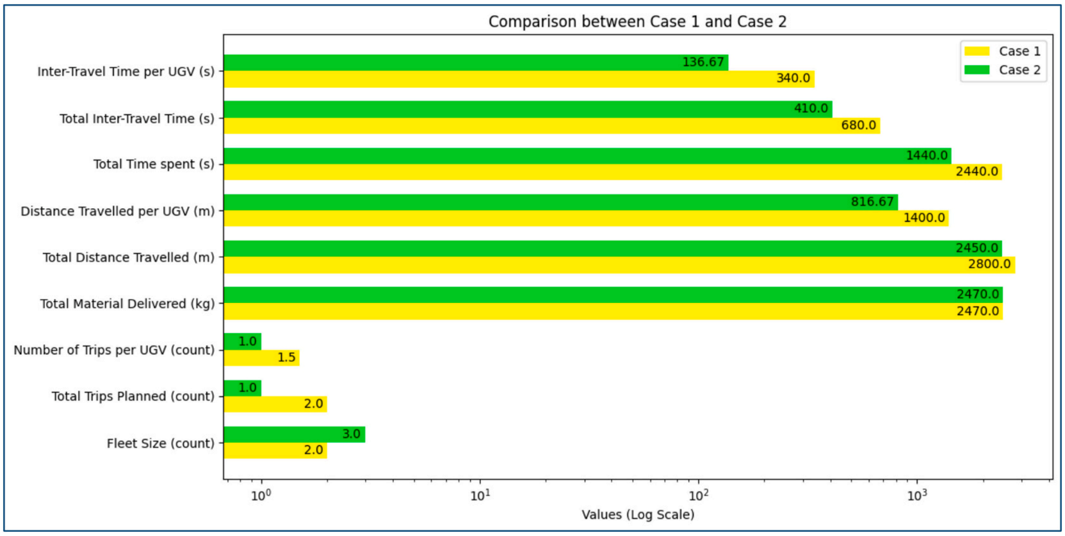


Fig. 17. Comparison between case 1 and case 2.

framework enables data-backed decision-making based on different priorities, such as low cost, urgency, etc.

This project not only automates and optimises the repair process but also enables ways to track and monitor workload execution. Furthermore, it also allows micro-monitoring on each of the UGVs. For instance, monitoring the total distance travelled and total load carried by each UGV over weeks or months can help study their health and manage maintenance costs.

This project can be extended to custom vehicle profiles and incorporate heterogeneous fleets in which different UGVs serve their specific purpose, or each UGV may have its own unique capacities and strengths.

Road maintenance works cause unexpected diversions and detours, resulting in chaos and frustration among the drivers. It also causes excessive fuel combustion, contributing to a carbon footprint [54]. The turn-around time of road maintenance can be reduced drastically if it is accomplished during night-time and the roads are available by the day [55]. Especially when the right kind of cold-asphalt mixture [27] is used, it is even possible to reopen the roads immediately [25,26]. Thus, this data-driven framework not only allows for accurate planning through the deployment of UGV fleets that are precise in their work but also promises a better and more sustainable solution towards greener smart cities of the future.

5. Conclusions

This paper successfully developed a data-driven framework for automating the pothole repair process using a fleet of unmanned ground vehicles (UGVs). The framework included data collection, fleet management, route planning, workload scheduling, fault tolerance mechanisms, and a vision to integrate with a digital twin environment for effective monitoring and analysis. The project demonstrated the potential for enhanced efficiency, cost-effectiveness, and safety in road maintenance operations, aligning with the vision of sustainable smart cities. The following conclusions can be summarised:

- By integrating the OpenRouteService (ORS) API into the route planning system, the efficiency and reliability of the UGV fleet's navigation could be significantly enhanced. This would leverage customisable routing options and up-to-date OpenStreetMap data to optimise performance in dynamic environments.
- The framework included a fault tolerance mechanism that detected bot failures and dynamically reassigned pending tasks to ensure continued operation and task completion.
- By integrating a digital twin environment and employing a modular design in Docker-based applications, scenario simulations, performance optimisation, and risk mitigation could be enabled—enhancing the system's reliability and robustness—while allowing for easy scalability and adaptation to different geographical areas and fleet sizes.
- Detailed visualisations and analytics provided insights into resource utilisation metrics through various KPIs such as distance travelled, repair material deployed, and workload distribution, enabling data-driven decision-making. Moreover, by automating the pothole repair process, the project reduced the need for human labour in hazardous areas, increasing safety and reducing the risk of accidents.
- The modelling analysis in Table 4 showed that the fleet of 3 UGVs per defined area (Fig. 13) can accomplish the same jobs by travelling approximately 10 % less distance and spending about 40 % less time than the fleet of 2 UGVs. The presented framework provides a robust and consistent verification that the project can optimise resource utilisation through effective planning.

Despite the significant achievements, the project faced several challenges and limitations. The poor documentation for the ORS Python wrapper required frequent referencing of the official API documentation, which slowed development. The job setup time defaulted to 0 due to missing parameters in the ORS Python wrapper. The ORS API's limitations on vehicle and job counts constrained the solution's scalability [56]. Furthermore, the vehicle ID reset issue in the ORS API response complicated scenarios involving specific vehicle attributes.

The key areas for future work recommendations are: 1. Heterogeneous Fleet Management: Extend the framework to manage heterogeneous fleets with bots with varied skills and capabilities. 2. User Interface Development: Develop a user-friendly interface for real-time monitoring and control of the UGV fleet and repair operations. 3. Utilise sensor data from simulations: The sensor data generated in digital twin environments can be used to optimise performance, enhance predictive maintenance, and improve real-world decision-making.

Funding body

There is no Funding support for this research.

CRediT authorship contribution statement

Shripal Mehta: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Abiodun B. Yusuf:** Writing – review & editing, Validation, Supervision, Project administration, Conceptualization. **Sepehr Ghafari:** Writing – review & editing, Validation, Supervision, Project administration.

Ethical statement

The paper has been submitted with full responsibility, following due ethical procedure, and there is no duplicate publication, fraud, plagiarism. None of the authors of this paper has a financial or personal relationship with other people or organizations that could inappropriately influence or bias the content of the paper.

Declaration of competing interest

No Competing interests are at stake and there is no conflict of interest with other people or organizations that could inappropriately influence or bias the content of the paper.

Data availability

Data will be made available on request.

References

[1] N. Laver, Pothole Trip & Fall Compensation Claims | claims.co.uk ™, Accessed: Jan. 01, 2024. [Online].Available: https://www.claims.co.uk/slips-trips-falls/pothole-compensation-claims, 2025.
[2] RAC Motoring Services, RAC Pothole Index – Statistics and Data for UK Roads | RAC Drive [Online]. Available, https://www.rac.co.uk/drive/advice/driving-advice/rac-pothole-index-statistics-data-and-projections/, 2025.
[3] RAC Media Centre, RAC pothole-related breakdowns for 2023 jump by a third compared to 2022, Mynewsdesk, 2025 [Online]. Available, https://media.rac.co.uk/pressreleases/rac-pothole-related-breakdowns-for-2023-jump-by-a-third-compared-to-2022-3297039.
[4] LCRIG, A new industry report addresses local authority pothole struggles - LCRIG, Jan. 01, 2024 [Online]. Available, https://lcrig.org.uk/2023/09/19/a-new-industry-report-addresses-local-authority-pothole-struggles.
[5] J. Bergal, To Combat Potholes, Cities Turn to Technology [Online]. Available: https://stateline.org/2018/06/18/to-combat-potholes-cities-turn-to-technology/, Jan. 01, 2024.
[6] Kwik Fit, Pothole Impact on Drivers' Pockets as Cost from Damage Rises by 35% [Online]. Available: https://www.kwik-fit.com/press/pothole-impact-on-drivers-pockets, 2025.
[7] A.H. Geelani, H. Bashir, A. Aier, M. Alam, S.G. Dr Kumar, Potholes patching - a paper on materials, testing, and the optimum binder content, Int. Res. J. Mod. Eng. Technol. Sci. 4 (6) (2022) 2582–5208 [Online]. Available: https://www.irjmets.com/uploadedfiles/paper//issue_6_june_2022/25670/final/fin_irjmets1654776480.pdf.
[8] D. Mitra, AI-Based Pothole Detection for Enhanced Road Safety [Online]. Available: https://www.cyberswift.com/blog/ai-based-pothole-detection-for-enhanced-road-safety, Jan. 01, 2024.
[9] A. Niaz, S. Khan, F. Niaz, M.U. Shoukat, I. Niaz, J. Yanbing, Smart City IoT application for road infrastructure safety and monitoring by using digital twin, in: Presented at the 2022 International Conference on IT and Industrial Technologies (ICIT), IEEE Xplore, 2022, https://doi.org/10.1109/icit56493.2022.9989141.
[10] E. Wallin, Smart City Solutions for a More Sustainable Future [Online]. Available: https://proptechos.com/smart-buildings/smart-city-solutions/, 2025.
[11] A. Simondetti, N. Bachand, A. Delahunty, J. Griffith, J. Sustarevas, Autonomous robotics in the AEC practice, Technol. Des. 5 (2) (2021) 250–252, https://doi.org/10.1080/24751448.2021.1967072.
[12] B. Ali, D. Thahir, K. Al-Karmi, S. Esmaeili, A. Alateyah, Pothole patrol: An autonomous vehicle for locating and repairing potholes, in: 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), IEEE, Ankara, Turkey, Jun. 2022, pp. 1–6, https://doi.org/10.1109/HORA55278.2022.9799954.
[13] R.A.M. Homji, Intelligent Pothole Repair Vehicle [Online]. Available: https://oaktrust.library.tamu.edu/bitstream/handle/1969.1/4328/etd-tamu-2005B-MEEN-Minocher.pdf, Jan. 01, 2024.
[14] UK Research and Innovation, World first AI robot for tackling the pothole problem, 2025 [Online]. Available, https://www.ukri.org/news/world-first-ai-robot-for-tackling-the-pothole-problem/.
[15] R. Hafezzadeh, F. Autelitano, F. Giuliani, Asphalt-based cold patches for repairing road potholes – an overview, Constr. Build. Mater. 306 (2021) 124870, https://doi.org/10.1016/j.conbuildmat.2021.124870.
[16] Z. Patton, To Combat Potholes, Cities Turn to Technology [Online]. Available: https://www.governing.com/archive/sl-cities-potholes-fix.html, Jan. 01, 2024.
[17] Grams Infratech India Private Limited, About Us [Online]. Available: https://www.potholerepairindia.com/about-us.php, Jan. 01, 2024.
[18] Robotiz3d, Robotiz3d [Online]. Available, https://www.robotiz3d.com/, 2025.
[19] JCB, Pothole Pro [Online]. Available, https://www.jcb.com/en-gb/products/wheeled-excavators/hydradig/pothelepro, 2025.
[20] T. Stone, Robot to Detect and Assess Potholes being Developed by University of Liverpool, UK [Online]. Available: https://www.traffictechnologytoday.com/news/asset-management/robot-to-detect-and-assess-potholes-being-developed-by-university-of-liverpool-uk.html, Jan. 01, 2024.
[21] G.B. Road Safety, Robot uses state of the art technology to prevent potholes forming, Jan. 01, 2024 [Online]. Available: https://roadsafetygb.org.uk/news/robot-uses-state-of-the-art-technology-to-prevent-potholes-forming/.
[22] G. Roberts, AI pothole robot tested on Herts road - how did it do?, Jan. 01, 2024 [Online]. Available: https://www.watfordobserver.co.uk/news/24169496.ai-pothole-robot-tested-hertfordshire-road-first-time/.
[23] J. Yu, J. Jiang, S. Fichera, P. Paoletti, L. Layzell, D. Mehta, S. Luo, Road Surface Defect Detection – From Image-based to Non-image-based: A Survey [Online]. Available: https://arxiv.org/abs/2402.04297, Jan. 01, 2024.
[24] P.P. Chavan, P. Dhapare, P.S. Joshi, P.S. Karande, M.S. Kolwankar, Automated pothole filling machine, Int. J. Res. Appl. Sci. Eng. Technol. 11 (4) (2023) 4116–4118, https://doi.org/10.22214/ijraset.2023.51213.
[25] S. Bruno, G. Cantisani, A. D'Andrea, G. Del Serrone, P. Di Mascio, K. Knudsen, G. Loprencipe, L. Moretti, C. Polidori, S.T. Ingwersen, L. Venturini, M. Zani, A small robot to repair asphalt road potholes, Infrastructures 9 (2024) 210, https://doi.org/10.3390/infrastructures9110210.
[26] S. Bruno, G. Loprencipe, P. Di Mascio, G. Cantisani, N. Fiore, C. Polidori, A. D'Andrea, L. Moretti, A robotized raspberry-based system for pothole 3D reconstruction and mapping, Sensors 23 (2023) 5860, https://doi.org/10.3390/s23135860.
[27] G. Cantisani, A. D'Andrea, P. Di Mascio, L. Moretti, N. Fiore, M. Petrelli, C. Polidori, L. Venturini, Materials study to implement a 3D printer system to repair road pavement potholes, Transp. Res. Procedia 69 (2023) 91–98, https://doi.org/10.1016/j.trpro.2023.02.149.
[28] M. Coindreau, O. Gallay, N. Zufferey, Parcel delivery cost minimization with time window constraints using trucks and drones, Networks 78 (4) (2021), https://doi.org/10.1002/net.22019.
[29] Michelin Connected Fleet, Fleet Management | What Is Fleet Management | Complete Guide by MICHELIN Connected Fleet [Online]. Available, https://connectedfleet.michelin.com/blog/complete-guide-to-fleet-management/, Jan. 01, 2024.
[30] IBM, What Is a Digital Twin [Online]. Available, https://www.ibm.com/topics/what-is-a-digital-twin, 2025.
[31] K. Achuthan, N. Hay, M. Aliyari, Y.Z. Ayele, A digital information model framework for UAS-enabled bridge inspection, Energies 14 (19) (Sep. 2021) 6017, https://doi.org/10.3390/en14196017.
[32] H. Xia, Z. Liu, E. Maria, X. Liu, C. Lin, Study on City digital twin Technologies for Sustainable Smart City Design: a review and bibliometric analysis of geographic information system and building information modeling integration, Sustain. Cities Soc. 84 (2022) 104009, https://doi.org/10.1016/j.scs.2022.104009.
[33] U. Fayyad, D. Haussler, P. Stolorz, KDD for Science Data Analysis: Issues and Examples [Online]. Available: https://cdn.aaai.org/KDD/1996/KDD96-009.pdf, 1996.
[34] C. Schröer, F. Kruse, J.M. Gómez, A systematic literature review on applying CRISP-DM process model, Procedia Comput. Sci. 181 (1) (2021) 526–534, https://doi.org/10.1016/j.procs.2021.01.199.
[35] A. Azevedo, M. Santos, KDD, SEMMA and CRISP-DM: A Parallel Overview [Online]. Available: https://recipp.ipp.pt/bitstream/10400.22/136/3/KDD-CRISP-SEMMA.pdf, 2008.
[36] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, H. Liu, Feature selection: a data perspective, ACM Comput. Surv. 50 (6) (2018) 1–45, https://doi.org/10.1145/3136625.
[37] V. Kumar, S. Minz, Feature selection: a literature review, Smart Comp. Rev. 4 (3) (2014) [Online]. Available, https://faculty.cc.gatech.edu/~hic/CS7616/Papers/Kumar-Minz-2014.pdf.
[38] S. Roberts, What is Data Cleaning? Process and Tools [Online]. Available, https://www.theknowledgeacademy.com/blog/data-cleaning/.
[39] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, O. Tabona, A survey on missing data in machine learning, J. Big Data 8 (1) (2021), https://doi.org/10.1186/s40537-021-00516-9.
[40] D. Varma, A. Nehansh, P. Swathy, Data preprocessing toolkit: an approach to Automate data preprocessing, Int. J. Sci. Res. Eng. Manag. 07 (03) (2023), https://doi.org/10.55041/ijsrem18270.
[41] R.Y. Wang, D.M. Strong, Beyond accuracy: what data quality means to data consumers, J. Manag. Inf. Syst. 12 (4) (1996) 5–33 [Online]. Available: https://www.jstor.org/stable/40398176.
[42] V. Da Poian, B. Theiling, L. Clough, B.A. McKinney, J.D. Major, J. Chen, S.M. Horst, Exploratory data analysis (EDA) machine learning approaches for ocean world analog mass spectrometry, Front. Astron. Space Sci. 10 (2023), https://doi.org/10.3389/fspas.2023.1134141.
[43] P. Agarwal, A. Singhal, A. Garg, SDLC model selection tool and risk incorporation, Int. J. Comput. Appl. 172 (10) (Aug. 2017) 6–10, https://doi.org/10.5120/ijca2017915143.
[44] GeeksForGeeks, Software Engineering | Rapid application development model (RAD) - GeeksforGeeks [Online]. Available, https://www.geeksforgeeks.org/software-engineering-rapid-application-development-model-rad/, 2025.

[45] ESRI, Spatial References | Documentation | ArcGIS Developers [Online]. Available, https://developers.arcgis.com/documentation/spatial-references/, Jan. 01, 2024.

[46] P.C. Gosling, E. Symeonakis, Automated map projection selection for GIS, Cartogr. Geogr. Inf. Sci. 47 (3) (2020) 261–276, https://doi.org/10.1080/15230406.2020.1717379.

[47] U.S. Geological Survey, Map Projections, Accessed: Jan. 01, 2024. [Online]. Available: https://pubs.usgs.gov/gip/70047422/report.pdf, 1993.

[48] GIScience, GIScience/openrouteservice [Online]. Available: https://github.com/GIScience/openrouteservice?tab=readme-ov-file, Jan. 01, 2024.

[49] Hertfordshire County Council [Online]. Available, https://twitter.com/i/status/1745017650424504748, 2024.

[50] Highways England, Design Manual for Roads and Bridges Pavement Design CD 226 Design for new pavement construction [Online]. Available: https://www.standardsforhighways.co.uk/tses/attachments/9654b4de-efa7-4843-8598-295019387077?inline=true, Nov. 2021.

[51] Highways England, Design Manual for Roads and Bridges Pavement Design CD 236 Surface course materials for construction [Online]. Available: https://www.standardsforhighways.co.uk/tses/attachments/815f3a3c-efe5-4555-9003-9ee38c2d080b?inline=true, Dec. 2022.

[52] BSI Group, Bituminous Mixtures. Test Methods, Jan. 2025, https://doi.org/10.3403/bsen12697.

[53] Highways England, Design Manual for Roads and Bridges Pavement Design CM 231 Pavement surface repairs [Online]. Available: https://www.standardsforhighways.co.uk/tses/attachments/df52250c-a3c3-4d8a-b55a-c418e82ad243?inline=true, Mar. 2020.

[54] J. Ma, L. Cheng, D. Li, Road maintenance optimization model based on dynamic programming in urban traffic network, J. Adv. Transp. 2018 (2018) 1–11, https://doi.org/10.1155/2018/4539324.

[55] C. Mudiono, N. Widyaningsih, Influence factors of Road maintenance for the Safety of Road users on the Jakarta - Tangerang toll segment, Int. J. Res. Rev. 9 (6) (2022) 135–143, https://doi.org/10.52403/ijrr.20220616.

[56] HeiGIT gGmbH, API Restrictions | Openrouteservice [Online]. Available, https://openrouteservice.org/restrictions/, Jan. 01, 2024.