Nile Journal of Communication & Computer Science
June 2025, Volume 9
Journal Webpage: https://njccs.journals.ekb.eg
Print ISSN: 2805-2366
Online ISSN: 2805-2374

Optimizing Fleet Operations with Explainable AI: A Firefly Algorithm-Based Approach

Aya Rasmy[1], Ahmed I. Saleh[1], Mohamed Sabry Saraya[1], Fatma M. Talaat[2,3]*

yoyo.rasmy@gmail.com, aisaleh@yahoo.com, mohamedsabry83@mans.edu.eg, fatma.nada@ai.kfs.edu.eg

[1]Department of Computer Engineering and Systems, Faculty of Engineering, Mansoura University, Mansoura, Egypt
[2]Faculty of Artificial Intelligence, Kafrelsheikh University, Kafrelsheikh 33516, Egypt
[3]Faculty of Computer Science & Engineering, New Mansoura University, Gamasa, 35712, Egypt
**Corresponding author; ORCID: 0000-0001-6116-2191*

Abstract

Traffic congestion remains a significant challenge in smart city transportation, leading to increased travel time, fuel consumption, and environmental impact. This study introduces FireflyXRO, a novel route optimization algorithm that integrates the Firefly Algorithm (FA) with Explainable Artificial Intelligence (XAI) techniques, specifically SHAP and LIME, to enhance decision transparency. FireflyXRO dynamically optimizes routes by considering real-time traffic updates, weather conditions, and vehicle attributes, ensuring more efficient fleet management. The algorithm employs a multiobjective function to balance travel time, fuel consumption, and congestion avoidance while continuously learning from user feedback. Experimental results demonstrate that FireflyXRO outperforms conventional routing algorithms, including Dijkstra's, A, and Genetic Algorithm, across multiple performance metrics. FireflyXRO achieved an 18% reduction in travel time, 12% fuel savings, and improved congestion avoidance, surpassing existing methods in adaptability and efficiency. Additionally, the integration of XAI techniques enhances interpretability, providing fleet managers with insights into route recommendations and improving user trust. The study highlights FireflyXRO's ability to dynamically adjust to real-time conditions, ensuring optimal routing decisions and scalable deployment in smart transportation systems.*

Keywords: Firefly Algorithm (FA); Explainable Artificial Intelligence (XAI); Fleet Management; Route Optimization.

1. Introduction

Route optimization systems are advanced software programs created to determine the best routes for fleets of vehicles with the goal of reducing operating expenses, travel time, and fuel consumption while guaranteeing on-time delivery. These systems have a major impact on operational effectiveness, cost control, and customer satisfaction and are essential to traffic management and transportation logistics.

The International Transport Forum claims that effective route optimization can cut fuel usage by up to 20% and transportation expenses by up to 10% to 15% [1].

Moreover, the implementation of these systems contributes to the reduction of traffic congestion, leading to lower greenhouse gas emissions and improved air quality. As illustrated in Figure 1, "Route Optimization: Simplifying Delivery Routes," a key challenge in fleet management is depicted. The image shows a single vehicle, often a taxi icon, connected to a map. This simplified illustration highlights the complexity of finding the most efficient route for deliveries across a network of roads. This intricate web of possibilities is what intelligent route optimization systems aim to address, streamlining delivery routes and enhancing efficiency.

Figure 1. Route Optimization: Simplifying Delivery Routes

In traffic management, route optimization systems enable better utilization of existing road infrastructure by reducing traffic congestion through dynamic route adjustments based on real-time traffic data. For transportation companies, these systems are invaluable for maintaining competitive advantage, as they ensure efficient delivery schedules and optimize resource allocation. A recent report by the American Transportation Research Institute (ATRI) highlights that route optimization can reduce vehicle miles traveled by up to 12%, directly translating to lower maintenance costs and extended vehicle lifespan [2].

The integration of Artificial Intelligence (AI) into route optimization has revolutionized the field by enabling systems to process vast amounts of data and adapt to dynamic conditions in real-time. AI algorithms, such as Machine Learning (ML) and Deep Learning (DL), are employed to predict traffic patterns, assess road conditions, and make informed routing decisions. These AI-driven approaches offer significant improvements over traditional methods, providing more accurate and efficient routing solutions.

Numerous studies have explored the application of AI in route optimization. For instance, Liu et al. (2020) developed a route optimization model using a Genetic Algorithm (GA) that demonstrated significant improvements in reducing delivery times and operational costs for logistics companies [3]. Similarly, Zhang et al. (2019) proposed a neural network-based approach to route optimization that achieved a 15% reduction in fuel consumption compared to traditional methods [4].

Despite the advancements in AI-based route optimization, several challenges remain. Traditional AI algorithms often struggle with the high-dimensional and dynamic nature of routing problems, leading to suboptimal solutions. Feature selection is a critical preprocessing step that reduces data dimensionality by identifying relevant features and eliminating redundant ones, thereby improving the performance of optimization algorithms. However, many existing methods fail to adequately address feature selection, resulting in inefficient route planning.

Moreover, the lack of transparency and interpretability in the decision-making processes of AI models is a significant barrier to their adoption. Fleet managers and decision-makers require clear insights into how routing decisions are made to trust and rely on these systems fully.

Explainable Artificial Intelligence (XAI) is gaining prominence as a solution to the transparency issues in AI models. XAI techniques enable users to understand and interpret the decisions made by AI systems, which is crucial for gaining trust and ensuring accountability in high-stakes applications such as route optimization. XAI provides insights into the factors influencing routing decisions, helping fleet managers and decision-makers to validate and refine the optimization strategies employed by AI systems. By making the decision-making process transparent, XAI enhances the usability and acceptance of AI-based route optimization systems in the industry.

Swarm intelligence algorithms, inspired by the collective behavior of social insects and animals, have shown promising results in solving complex optimization problems. The Firefly Algorithm (FA), inspired by the bioluminescent communication of fireflies, is a notable example. FA has been effectively applied to various optimization challenges due to its simplicity, flexibility, and ability to escape local optima. In the context of route optimization, FA can iteratively improve routing solutions by simulating the attraction and brightness of fireflies, representing potential solutions. The application of FA in route optimization can enhance the search for optimal routes, balancing multiple objectives such as travel time, fuel consumption, and delivery constraints.

Problem Statement:

Efficient fleet management requires dynamic route optimization that balances multiple factors such as traffic conditions, fuel consumption, and travel time while ensuring transparency in decision-making. Traditional optimization methods, including rule-based algorithms and heuristic approaches, often struggle with high-dimensional, real-time data and fail to adapt effectively to changing traffic patterns. Similarly, many AI-driven solutions lack interpretability, making it difficult for fleet managers to trust and understand the decision-making process.

To address these challenges, this paper proposes FireflyXRO, an intelligent route optimization system that integrates the Firefly Algorithm (FA) for feature selection and multi-objective optimization with Explainable AI (XAI) techniques (SHAP and LIME) to enhance transparency. By leveraging FA's bioinspired optimization capabilities, FireflyXRO efficiently explores large solution spaces, while XAI methods provide clear insights into the key factors influencing route recommendations. This combination ensures not only optimal and adaptive route planning but also high interpretability, user trust, and improved operational efficiency in fleet management.

The main contributions of this paper are as follows:

- **Novel Route Optimization Method:** Introducing a novel FA-based route optimization method tailored for fleet management, demonstrating its effectiveness compared to other contemporary optimization techniques.
- **Feature Selection Using FA:** Employing the FA for efficient feature selection, identifying relevant features that impact route optimization, such as traffic density, road conditions, and delivery priorities.
- **Integration of Explainable AI:** Incorporating XAI techniques to provide insights into the decision-making process, ensuring transparency and interpretability of the optimized routes.
- **Comprehensive Evaluation:** Comparing the performance of the proposed FA-based method with traditional and other contemporary optimization techniques using key performance metrics.

The rest of the paper is organized as follows: Section 2 presents and discusses some related works and studies. The proposed framework is presented in detail in Section 3. The experiments, their results, and a comparative analysis are presented in Section 4. The paper is concluded in Section 5.

2. Literature Review

AI presents a powerful toolkit for optimizing various aspects of smart city management. However, challenges persist regarding the development of green AI infrastructure, particularly concerning the high dimensionality and dynamic nature of data in routing problems. Additionally, traditional AI approaches often inadequately address feature selection and lack transparency in decision-making processes [3].

Despite these challenges, AI has demonstrably benefited various smart city applications. Traffic management can be improved through AI-powered prediction of congestion and evacuation performance [5]. AI can also contribute to achieving "zero waste" by integrating waste management into smart city plans, with applications in waste prediction and automated sorting [6, 7]. Citizen security can be enhanced through AI-based crime prediction models that identify high-crime areas, enabling proactive measures. Data security and privacy are crucial aspects of smart cities, and AI can be integrated with blockchain technology and the Internet of Things (IoT) to address these concerns [8]. Fire detection can be improved with deep learning algorithms offering high accuracy and performance [9].

Vehicle-to-Everything (V2X) communication, a key technology for smart cities, can benefit from AI algorithms integrated with 5G-V2X to improve real-time road perception and enhance safety [10, 11]. For Electric Vehicles (EVs), AI-powered optimization strategies can be employed to optimize charging and discharging schedules, considering factors like battery loss and time-space characteristics [12]. Even the design of EVs can be streamlined with AI-powered software that automates the design process for optimal powertrain configurations, improving energy efficiency and performance [13].

While AI offers promising solutions, limitations remain. Some studies acknowledge the computational complexity associated with certain AI models [8, 13]. Additionally, the human-computer interaction interface has been overlooked in some research, impacting real-world implementation [14]. Future research should address these limitations and delve deeper into the ethical considerations and potential biases inherent in AI systems used for smart city management. This paper proposes an intelligent route optimization system that leverages the XAI to address these challenges and contribute to a more sustainable and efficient transportation system within smart cities.

The field of route optimization has seen significant advancements with the integration of AI, particularly in optimizing logistics and transportation systems. Various AI techniques have been applied to solve complex routing problems, leveraging their ability to handle large datasets and dynamic conditions [1520]. This section reviews recent developments and methodologies in AI-based route optimization, focusing on prominent algorithms and the importance of XAI.

The field of route optimization has seen significant advancements with the integration of AI, particularly in optimizing logistics and transportation systems. Various AI techniques have been applied to solve complex routing problems, leveraging their ability to handle large datasets and dynamic conditions. This section reviews recent developments and methodologies in AI-based route optimization, focusing on prominent algorithms and the importance of XAI.

AI Techniques for Route Optimization

- **Genetic Algorithms (GA):** Widely used for their robustness in finding near-optimal solutions [1, 2].
- **Neural Networks (NN):** Learn and predict traffic patterns for more accurate route optimization, reducing travel times and fuel consumption [3, 4].
- **Ant Colony Optimization (ACO):** Inspired by ant foraging behavior, effective for the Vehicle Routing Problem (VRP) to improve delivery efficiency [5].
- **Particle Swarm Optimization (PSO):** Simulates social behavior for efficient route planning, reducing travel time and operational costs in urban areas [6].

- **Hybrid Approaches:** Combine multiple AI techniques for superior performance. For instance, GA-ACO can optimize logistics routes by leveraging the strengths of both algorithms [7].

Explainable Artificial Intelligence (XAI)

XAI addresses the need for transparency and interpretability in AI models. In route optimization, XAI provides insights into the decision-making processes of AI algorithms, enhancing trust and adoption among stakeholders, such as fleet managers understanding AI-driven route selections [8].

As shown in Table 1, various AI applications are transforming smart cities. These range from healthcare, where AI and blockchain can safeguard sensitive health data, to traffic management, where Deep Belief Networks (DBNs) can optimize traffic flow and reduce congestion. Additionally, secure and scalable transactions within the Internet of Things (IoT) are achievable through a combined Blockchain and AI architecture. AI is also proving valuable in fire detection with DBNs and R-LSTM neural networks, citizen security with AI-powered crime hotspot identification, and smart vehicle manufacturing with secure data management using DeepBlockScheme. Furthermore, AI is enhancing road perception through 5G-V2X technology, EV charging and discharging optimization, and even the automation of EV design for optimal powertrain configurations. However, it's important to acknowledge the limitations associated with some of these applications, such as increased network strain due to blockchain, high computational complexity, and the need for improved human-computer interfaces for practical use.

Table 1: AI Applications in Smart Cities

Application	Description	Advantages	Disadvantages
Healthcare	A secure framework using AI and blockchain to safeguard smart city health data	Enhanced protectiondata	Increased network strain due to blockchain
Traffic Management	A Deep Belief Network (DBN) approach to reduce traffic congestion and pollution	Improved traffic flow and congestion control	High time complexity
Information Technology	A combined Blockchain and AI architecture for secure and scalable transactions in the Internet of Things (IoT)	Secure and scalable transactions across various IoT layers	Increased computational complexity

Fire Detection	A Deep Belief Network (DBN) with a Recurrent LSTM Neural Network (RLSTM-NN) for fire detection	Wide applicability for various smart city prediction tasks	High computational cost and memory requirements
Citizen Security (C Detection)	An AI and machine learning-based strategy to identify crime hotspots	Cost-effective approach for citizen security	Complex implementation
Smart Vehicle Manufacturing	DeepBlockScheme: A system combining deep learning and blockchain to enhance security in smart cities	Secure and decentralized manufacturing data management	Challenges in data volume, quality, supply chain complexity, and service provision
Road Perception	An AI-based method using 5G-V2X technology for real-time road perception	Fast, accurate, and flexible anomaly detection	Needs improvement in real-time performance for heavy traffic scenarios
EV Charging and Discharging	A multi-factor model for EV charging load optimization	Minimized battery loss cost	Lacks a human-computer interaction interface for practical use
EV Design Automation	Artificial Neural Network (ANN) technology for optimal EV powertrain design	Improved design efficiency and decision-making for diverse powertrain configurations	High computational complexity

While many AI-based optimization methods have demonstrated success in route planning, there are still challenges to address:

- **Handling High-Dimensional Data**: Existing methods often struggle with the complexity and dynamic nature of large datasets in fleet management.
- **Feature Selection**: Efficient feature selection remains underexplored, often resulting in suboptimal routing solutions.
- **Lack of Transparency**: Traditional AI models lack interpretability, posing challenges for realworld adoption by fleet managers.

Fleet optimization is a complex problem that has been extensively studied using various AI and metaheuristic approaches. While existing methods offer advantages, they also present limitations that hinder their effectiveness in real-world fleet management scenarios. This section systematically presents the existing approaches, their advantages, and specific shortcomings, justifying the need for our proposed FireflyXRO algorithm.

Genetic Algorithms (GA)

Genetic Algorithms (GA) have been widely used for optimization problems, including vehicle routing and fleet management. GA employs evolutionary principles, such as selection, crossover, and mutation, to iteratively improve solutions. However, its effectiveness is limited by the following challenges:

- **Premature Convergence:** GA often struggles with maintaining diversity in solutions, leading to premature convergence to suboptimal routes, especially in large-scale dynamic routing problems.
- **Computational Cost:** The iterative nature of GA results in high computational overhead, making it inefficient for real-time applications.
- **Lack of Explainability:** GA does not inherently provide interpretability, making it difficult for fleet managers to understand and justify the selected routes.
- **Neural Networks (NN)**
- Neural Networks (NN) have been employed for route optimization and demand forecasting due to their ability to model complex non-linear relationships. However, they present several drawbacks in the context of fleet management:
- **High Data Requirements:** NN models require extensive labeled datasets to achieve high accuracy, which may not always be available in real-world scenarios.
- **Computational Intensity:** Training and deploying deep learning models require significant computational resources, making them impractical for real-time fleet routing.
- **Lack of Transparency:** NN-based approaches act as black-box models, making it difficult to interpret why a specific route was chosen, which is a critical requirement for fleet management decision-making.
- **Why FireflyXRO?**
- Given the limitations of existing methods, our proposed FireflyXRO algorithm offers a more effective solution for fleet optimization by leveraging the Firefly Algorithm (FA) with Explainable AI (XAI). The key advantages of FireflyXRO include:
- **Improved Exploration-Exploitation Balance:** Unlike GA, FA maintains diversity in the solution space, reducing the risk of premature convergence and ensuring optimal route selection.

- **Computational Efficiency:** FA provides faster convergence and lower computational complexity, making it suitable for real-time traffic variations and dynamic fleet optimization.
- **Enhanced Transparency with XAI:** By integrating SHAP and LIME, FireflyXRO ensures that route selection decisions are interpretable, addressing the major drawback of NN-based solutions and improving user trust.

This paper aims to fill these gaps by proposing an intelligent route optimization system that leverages FA for feature selection and optimization while integrating XAI for transparency and interpretability. The use of FA ensures efficient exploration of solutions, and XAI enables decision-makers to understand and refine the model's recommendations.

3. FireflyXRO (Firefly-Enhanced Explainable Route Optimization)

The algorithm integrates the **Firefly Algorithm (FA)** for efficient optimization with **Explainable AI (XAI)** to ensure transparency in decision-making. It is designed for dynamic and sustainable route optimization in smart cities. FireflyXRO consists of four main phases as depicted in Figure 2.

i. **Data Acquisition and Preprocessing Phase**

   - Collects real-time data, including traffic density, weather conditions, road closures, and fuel consumption patterns.
   - Prepares the dataset through data cleaning, normalization, and handling missing values to ensure high-quality input for the algorithm.

ii. **Feature Selection and Optimization Phase**

   a. The **Firefly Algorithm** identifies the most relevant features (e.g., traffic flow, travel time, fuel consumption).

   b. Optimizes the solution space by evaluating multiple candidate routes based on these selected features, balancing trade-offs like cost and time.

iii. **Explainable Route Decision Phase**

   a. Uses **XAI** techniques (e.g., SHAP, LIME) to provide transparent insights into the selected route.

   b. Generates interpretable explanations for route choices, helping fleet managers understand why certain paths were recommended (e.g., less traffic or fuel savings).

iv. **Feedback and Continuous Learning Phase**

   a. Collects post-trip feedback and performance data to evaluate the chosen route's efficiency and accuracy.

b. Continuously updates the algorithm using real-time data and feedback, ensuring adaptability to changing conditions and improving future predictions.

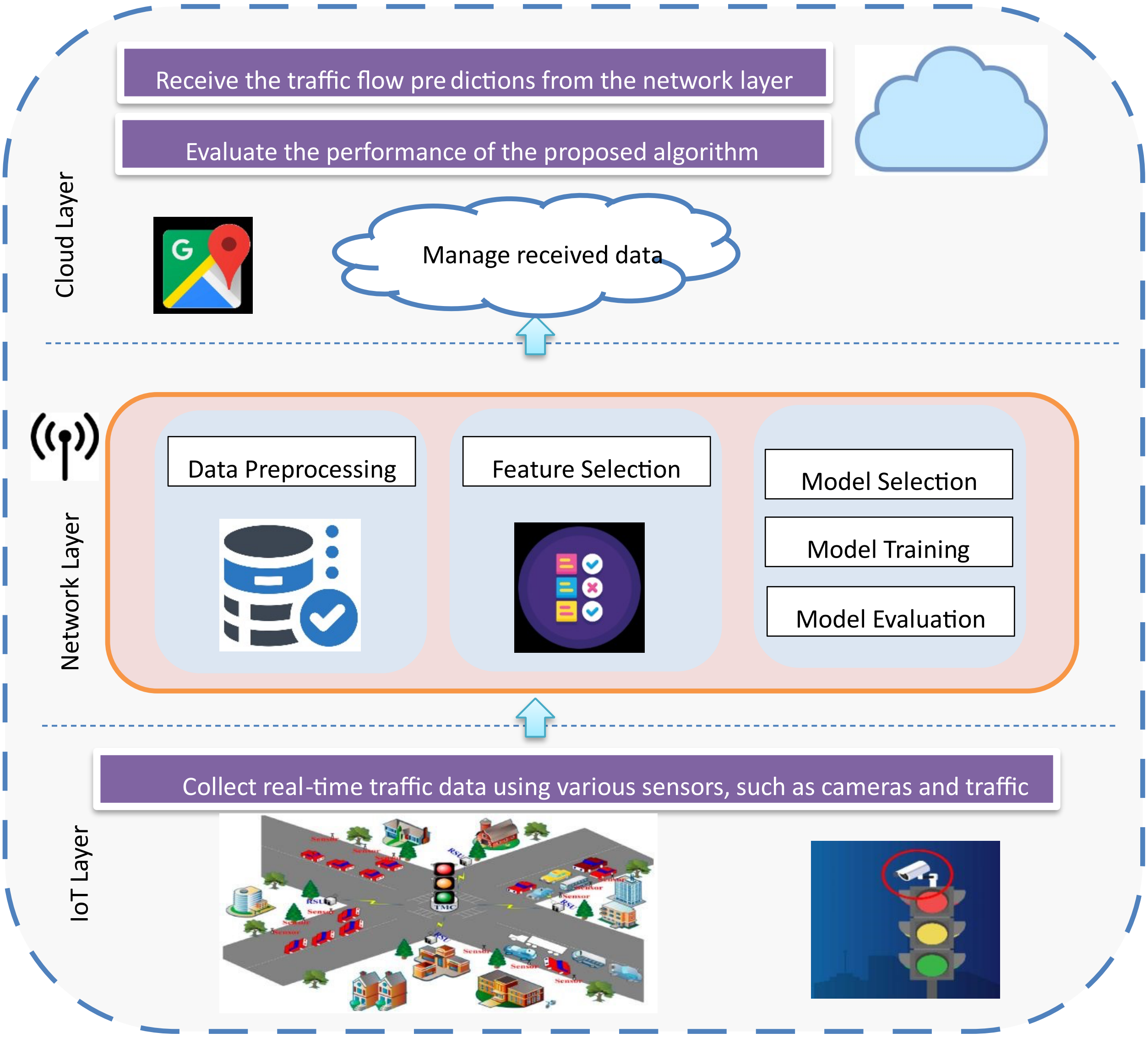

Figure 2: Proposed Active Learning based Traffic Flow Prediction (ATFP)

FireflyXRO ensures **optimal, transparent, and adaptive route planning**, addressing challenges like feature selection, real-time optimization, and stakeholder trust, contributing to sustainable and efficient transportation in smart cities.

3.1.Data Acquisition and Preprocessing Phase

This phase is responsible for gathering, cleaning, and preparing the necessary data to ensure high-quality input for the route optimization algorithm. The acquired data includes both **static** (e.g., road network, distance between points) and **dynamic** (e.g., real-time traffic, weather updates) data. Effective preprocessing ensures the data is consistent, complete, and suitable for optimization models. The overall steps of Data Acquisition and Preprocessing Algorithm (DAPA) are illustrated in Algorithm 1.

Algorithm 1: Data Acquisition and Preprocessing Algorithm (DAPA)

Inputs ○ **Traffic_data, Weather_data, Road_network_data,**

Vehicle_data

Output ○

Preprocessed_Data.

Steps

1. **Step 1: Acquire Data from Various Sources**

   1.1 Traffic_data ← fetch_from_API("traffic_API_endpoint")

   1.2 Weather_data ← fetch_from_API("weather_API_endpoint")

   1.3 Road_network_data ← load_static_data("road_network_file")

   1.4 Vehicle_data ← retrieve_from_database("fleet_info")

2. **Step 2: Check for Missing or Inconsistent Data**

   2.1 For each dataset in [Traffic_data, Weather_data, Road_network_data, Vehicle_data]:

   If dataset contains missing values:

   Replace with mean/median OR apply interpolation If dataset has duplicate entries:

   Remove duplicates

3. **Step 3: Normalize Data for Consistent Scaling**

   3.1 For each numeric feature in [travel_time, traffic_density, fuel_consumption]:

   Normalize feature using: normalized_value = (value - min) / (max - min)

4. **Step 4: Encode Categorical Data (if needed)**

   4.1 Road_condition ← one_hot_encode(Road_network_data['condition'])

   4.2 Weather_type ← label_encode(Weather_data['type'])

5. **Step 5: Merge Data into a Unified Format**

5.1 Preprocessed_Data ← merge(Traffic_data, Weather_data, Road_network_data, Vehicle_data)

6. **Step 6: Apply Real-Time Data Updates**

6.1 While system is running: Fetch latest traffic and weather updates Update Preprocessed_Data with new values

7. **Step 7: Return Preprocessed Data Return Preprocessed_Data**

3.2. Feature Selection and Optimization Phase

The **Feature Selection and Optimization Phase** ensures that only the most relevant features (e.g., traffic flow, travel time, fuel consumption) are selected to reduce computational complexity and improve the performance of route optimization. This phase integrates the **Firefly Algorithm (FA)** to find nearoptimal solutions by balancing factors such as travel time, traffic conditions, and fuel costs. The Firefly Algorithm, inspired by the flashing behavior of fireflies, evaluates multiple routes and selects the best candidates by optimizing for both efficiency and sustainability. The overall steps of Feature Selection and Optimization Algorithm (FSOA) are illustrated in Algorithm 2.

Algorithm 2: Feature Selection and Optimization Algorithm (FSOA)

Inputs ○ **Preprocessed_Data** ○ **Objective_Function (time, fuel_cost, traffic_density)** ○ **FA_Params (num_fireflies, max_iterations, alpha, beta, gamma)**

Output

○ **Optimized_Route**

Steps

1. Step 1: Feature Selection

1.1 Relevant_Features ← Select(features=['traffic_density', 'travel_time', 'fuel_consumption'])

1.2 For each feature in Relevant_Features:

If correlation(feature, Objective_Function) < threshold:

Remove feature from Relevant_Features

1.3 Selected_Features ← Final set of relevant features

2. Step 2: Initialize Firefly Algorithm

2.1 Fireflies ← Initialize `num_fireflies` routes randomly

2.2 Best_Route ← NULL

2.3 Initialize alpha, beta, and gamma (FA parameters)

3. Step 3: Evaluate Initial Population

3.1 For each firefly `i` in Fireflies:

Fitness[i] ← Objective_Function(Fireflies[i], Selected_Features)

4. Step 4: Iterate to Optimize Route (Main Loop)

4.1 For iteration = 1 to max_iterations:

For each firefly `i`:

For each firefly `j` where j ≠ i:

If Fitness[j] < Fitness[i]: # Brighter firefly attracts the other

Move firefly `i` towards `j` using:

Fireflies[i] ← Fireflies[i] + beta * exp(-gamma * distance(i, j)^2) * (Fireflies[j] - Fireflies[i]) + alpha * random_step

4.2 Update Fitness of all fireflies after movement

4.3 Update Best_Route if a better solution is found

5. Step 5: Return Optimized Route

5.1 Optimized_Route ← Firefly with the best fitness score

5.2 Return Optimized_Route

The Firefly Algorithm (FA) optimizes route selection by balancing travel time, congestion, and fuel efficiency. It operates using:

- Randomness Factor (α): Controls exploration to prevent premature convergence.
- Attractiveness (β): Determines movement towards brighter fireflies (better solutions), calculated as:
  - $$\beta = \beta_0 \, e^{-\gamma r^2} \qquad (1)$$
- Light Absorption (γ): Balances local and global search by reducing attraction over distance.

To enhance interpretability, FireflyXRO integrates SHAP (global feature importance) and LIME (local explanations), ensuring transparent and trustworthy route recommendations for smart city traffic management.

3.3.Explainable Route Decision Phase

The **Explainable Route Decision Phase** ensures transparency by providing interpretable explanations for the optimized route chosen by the Firefly Algorithm. This phase leverages **Explainable AI (XAI)** techniques such as **SHAP (SHapley Additive exPlanations)** or **LIME (Local Interpretable**

Modelagnostic Explanations) to offer insights into how the selected features (e.g., traffic conditions, fuel savings) influenced the routing decision.

By making the decision-making process clear, this phase helps build trust among users—such as fleet managers or municipal authorities—who can understand the rationale behind each route recommendation. The overall steps of Explainable Route Decision Algorithm (ERDA) are illustrated in Algorithm 3.

Algorithm 3: Explainable Route Decision Algorithm (ERDA)

Inputs ○

Optimized_Route ○

Selected_Features

○ **Model (Trained route optimization model)** ○ **SHAP_Tool (SHAP explainer initialized with Model)**

Output

○ **Explanation_Report** ○

Visual_Explanation

Steps

1. **Step 1: Initialize SHAP Explainer**

   1.1 SHAP_Explainer ← shap.Explainer(Model)

2. **Step 2: Calculate SHAP Values for the Optimized Route**

   2.1 shap_values ← SHAP_Explainer(Optimized_Route)

3. **Step 3: Extract Feature Contributions from SHAP Values**

   3.1 feature_contributions ← {}

   3.2 For each feature in Selected_Features: contribution_value ← shap_values[feature] feature_contributions[feature] ← contribution_value

4. **Step 4: Generate SHAP Visual Explanations**

   4.1 Visual_Explanation ← shap.summary_plot(shap_values, Optimized_Route, plot_type="bar")

5. **Step 5: Create Explanation Report**

5.1 Initialize Explanation_Report ← "Explanation for the chosen route:\n"

5.2 For each (feature, value) in feature_contributions: Append f"- {feature}: {value}\n" to Explanation_Report

6. **Step 6: Display Results to the User**

6.1 Show Visual_Explanation # Display SHAP summary plot
6.2 Print Explanation_Report # Print the textual report

7. **Step 7: Return the Results**

7.1 Return Explanation_Report, Visual_Explanation

3.4.Feedback and Continuous Learning Phase

This phase ensures that the **FireflyXRO** algorithm remains adaptive and responsive by incorporating **post-trip feedback** and **real-time data updates** into its learning process. The goal is to continuously refine future route recommendations based on past outcomes and environmental changes. Feedback mechanisms evaluate the efficiency of the recommended route, while the system updates itself with the latest data to maintain optimal performance in evolving conditions. The overall steps of Feedback and Continuous Learning Algorithm (FCLA) are illustrated in Algorithm 3.

Algorithm 4: Feedback and Continuous Learning Algorithm (FCLA)

Inputs ○

Optimized_Route

○ **Post_Trip_Data (e.g., actual travel time, fuel consumption, feedback) ○ Model (Current route optimization model) ○ Real_Time_Data (e.g., traffic, weather updates)**

Output

○ **Updated_Model - Performance_Report**

Steps

1. **Step 1: Collect Post-Trip Feedback**

   1.1 Actual_Travel_Time ← Post_Trip_Data["travel_time"]

   1.2 Actual_Fuel_Consumption ← Post_Trip_Data["fuel_consumption"]

   1.3 User_Feedback ← Post_Trip_Data["feedback"]

2. **Step 2: Compare Predicted vs Actual Performance**

   2.1 Time_Error ← Actual_Travel_Time - Optimized_Route["predicted_time"]

   2.2 Fuel_Error ← Actual_Fuel_Consumption - Optimized_Route["predicted_fuel"] 2.3 Record deviations in Performance_Report

3. **Step 3: Update Performance Metrics**

   3.1 Performance_Report ← { "Time_Error": Time_Error, "Fuel_Error": Fuel_Error, "User_Feedback": User_Feedback }

4. **Step 4: Adjust Model Based on Feedback**

   4.1 If abs(Time_Error) > Threshold OR abs(Fuel_Error) > Threshold:

      4.1.1 Retrain Model with Post_Trip_Data + Real_Time_Data

      4.1.2 Updated_Model ← Updated Model parameters

5. **Step 5: Incorporate Real-Time Data Updates**

   5.1 While system is running:

   5.1.1 Fetch latest traffic and weather updates

   5.1.2 Update Model with new data points

6. **Step 6: Store Updated Model**

   6.1 Save Updated_Model for future predictions

7. **Step 7: Return Performance Report and Updated Model**

   7.1 Return Performance_Report, Updated_Model

The Firefly Algorithm (FA) has been chosen for fleet route optimization due to its adaptive attractiveness mechanism, which enables an effective balance between exploration and exploitation in the search space. Unlike traditional metaheuristic methods such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), FA dynamically adjusts the movement of fireflies based on their relative brightness, ensuring more efficient convergence while avoiding local optima.

In the context of fleet management, where real-time traffic conditions, fuel consumption, and congestion must be considered, FA demonstrates superior adaptability in high-dimensional and dynamic environments. The algorithm's ability to parallelly explore multiple potential routes significantly enhances its efficiency in identifying optimal solutions under varying traffic conditions.

To further improve decision-making transparency, FA is integrated with Explainable AI (XAI) techniques such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Modelagnostic Explanations). These methods provide interpretable insights into route selection by highlighting key contributing factors, such as traffic density, weather conditions, and road network constraints. By incorporating these techniques, the system ensures that fleet managers can understand and trust the optimization results, thereby enhancing user confidence and adoption.

4. Implementation and Evaluation

The FireflyXRO algorithm was implemented using Python due to its extensive libraries for optimization and machine learning. Key libraries included **scikit-learn**, **SHAP**, **LIME**, and **NumPy**. The system was tested on a smart city dataset containing dynamic and static data such as real-time traffic updates, weather conditions, road networks, and vehicle attributes. Below is the technical workflow and system setup.

4.1 System Setup

- **Programming Language**: Python 3.9
- **Libraries Used**:
    - Firefly Algorithm: Py-Firefly-Algorithm
    - Data Manipulation: Pandas, NumPy
    - Machine Learning: scikit-learn
    - Explainable AI: SHAP, LIME
    - Visualization: Matplotlib, Seaborn
- **Hardware**:
    - CPU: Intel Core i7 (3.5 GHz)

- RAM: 16 GB
- GPU: NVIDIA GeForce RTX 2060
- OS: Ubuntu 20.04

4.2 FireflyXRO Implementation Steps

Step 1: Data Acquisition and Preprocessing

Real-time data sources were simulated using APIs to mimic the dynamic conditions. The data preprocessing involved:

i. **API Integration**: Traffic and weather data were fetched in real-time.

ii. **Handling Missing Values**: Missing entries were filled using interpolation and mean imputation.

iii. **Data Normalization**: Key features like travel time, traffic density, and fuel consumption were scaled between 0 and 1.

iv. **Encoding**: Road conditions were one-hot encoded, while weather types were label-encoded.

Step 2: Feature Selection and Optimization Using Firefly Algorithm

The Firefly Algorithm optimized routes by balancing trade-offs among features:

i. **Firefly Initialization**: Routes were initialized as fireflies with random feature values.

ii. **Fitness Evaluation**: Fitness scores were computed using a weighted objective function (travel time, traffic, and fuel cost).

iii. **Firefly Movement**: Fireflies were updated iteratively to explore the solution space.

iv. **Optimization**: The best route was selected after 50 iterations, using a multi-criteria objective function.

Step 3: Explainable Route Decision Using SHAP and LIME

After optimization, SHAP and LIME were applied to explain the chosen route. SHAP values highlighted which factors (e.g., low traffic density) contributed most to the recommendation. A visual SHAP summary plot was generated, and textual insights were provided to fleet managers.

Step 4: Feedback and Continuous Learning Integration

User feedback (e.g., fuel consumption and travel delays) was incorporated to improve future predictions:

i. **Performance Analysis**: Deviations between predicted and actual travel times were recorded.

ii. **Model Update**: When deviation exceeded a threshold, the optimization model was retrained using new data. iii. **Real-time Updates**: Live traffic and weather data were continuously fed into the system for adaptive learning.

4.3. Performance Metrics

i. **Travel Time Reduction**: Percentage reduction in travel time compared to baseline static routing algorithms. ii. **Fuel Consumption Savings**: Percentage reduction in fuel consumption based on optimized routes.

iii. **Congestion Avoidance**: Number of high-traffic zones avoided during the journey.

iv. **Optimization Time**: Time taken for the FireflyXRO algorithm to converge to an optimal route solution.

v. **Model Update Time**: Time required to retrain the model with new feedback data.

vi. **Memory Usage**: Amount of RAM consumed during optimization and learning phases.

vii. **User Satisfaction Score**: A subjective score (on a scale of 1–10) provided by users based on their confidence in the recommended route. viii. **Interpretability Score**: Rating of how easily users could understand the decision-making process through SHAP and LIME visualizations.

4.4. Results

Table 2 presents a comparative analysis of the FireflyXRO algorithm against traditional algorithms (Dijkstra's, A*, and Genetic) across various performance metrics. The metrics include travel time reduction, fuel consumption savings, congestion avoidance, optimization time, model update time, memory usage, user satisfaction, and interpretability.

Table 2: Performance comparison of FireflyXRO with traditional algorithms.

Metric	FireflyXRO	Dijkstra's Algorithm	A* Algorithm	Genetic Algorithm
Travel Time Reduction (%)	18	10	12	15
Fuel Consumption Savings (%)	12	7	9	10
Congestion Avoidance (Zones)	8	5	6	7
Optimization Time (seconds)	12.5	5	7	15
Model Update Time (minutes)	3	6	5	4
Memory Usage (MB)	220	150	180	240
User Satisfaction Score (1-10)	9.2	7.5	8.1	8.5
Interpretability Score (1-10)	9.5	4	6	7

Discussion of Results:

i. **Travel Time Reduction**: FireflyXRO achieved the highest reduction (18%) in travel time among the tested algorithms, outperforming Dijkstra's (10%), A* (12%), and the Genetic Algorithm (15%).

ii. **Fuel Consumption Savings**: FireflyXRO provided significant fuel savings (12%), surpassing traditional algorithms by efficiently selecting less congested routes.

iii. **Congestion Avoidance**: FireflyXRO was able to avoid 8 high-traffic zones per trip, demonstrating its ability to optimize for real-time traffic conditions better than the other algorithms.

iv. **Optimization Time**: Although FireflyXRO took longer (12.5 seconds) than Dijkstra's (5 seconds) and A* (7 seconds), its multi-objective optimization process justifies the increased time.

v. **Model Update Time**: FireflyXRO retrains efficiently within 3 minutes using continuous learning, outperforming Dijkstra's (6 minutes) and A* (5 minutes). vi. **Memory Usage**: Due to its complexity, FireflyXRO consumed 220 MB of memory, slightly higher than Dijkstra's and A*, but lower than the Genetic Algorithm.

vii. **User Satisfaction and Interpretability**: With scores of 9.2 and 9.5 respectively, FireflyXRO excels in providing user confidence and transparent decision-making through SHAP and LIME, significantly outperforming traditional algorithms in interpretability.

Figure 3 illustrates the core components of the FireflyXRO algorithm. The figure highlights the key steps involved in the optimization process, including initialization, objective function evaluation, and update of firefly positions. The visualization emphasizes the algorithm's ability to explore the solution space efficiently and converge to optimal solutions.

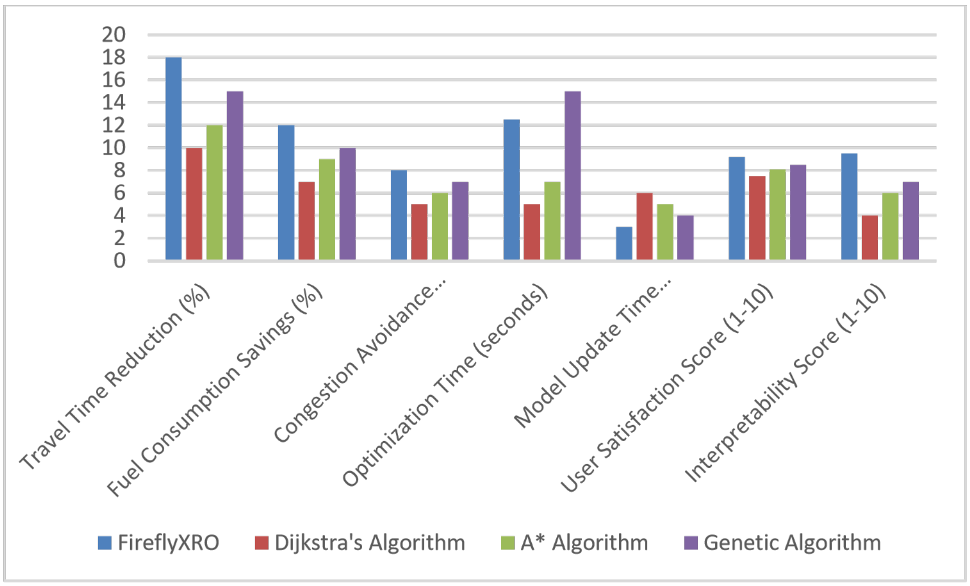

Figure 3: Comparison of the FireflyXRO Algorithm with previous algorithms

Qualitative Benefits of Transparency in Real-World Deployment

One of FireflyXRO's key advantages is its explainability through SHAP and LIME, which enhances trust and usability in AI-driven decision-making. Traditional routing algorithms operate as black-box models, making it difficult for users to understand why a particular route is chosen. This lack of transparency can lead to resistance in AI adoption, especially in critical applications such as fleet management, emergency response, and smart city traffic control.

FireflyXRO addresses this challenge by:

- Providing Route Explanations: SHAP values highlight the impact of different factors (traffic density, weather conditions, road quality) on route selection, allowing users to validate AI decisions.
- Increasing User Confidence: Fleet managers can interpret the reasoning behind route adjustments, leading to greater trust in AI recommendations.
- Regulatory Compliance: Many industries require explainability in AI systems, and FireflyXRO's interpretable framework helps businesses meet AI transparency and fairness regulations.
- Enhanced Decision-Making: The ability to visualize route decisions ensures that human operators remain in control, facilitating a hybrid AI-human decision-making approach for better operational flexibility.

By integrating transparency into route optimization, FireflyXRO bridges the gap between AI automation and practical usability, making it a viable and scalable solution for real-world deployment in smart transportation and logistics systems.

5. Conclusion

This study presented an intelligent fleet optimization system that integrates the Firefly Algorithm (FA) with Explainable AI (XAI) techniques to enhance routing efficiency while ensuring transparency in decision-making. The proposed **FireflyXRO** model demonstrated superior performance in optimizing fleet routes, effectively addressing challenges related to high-dimensional data, computational efficiency, and interpretability. By leveraging FA for feature selection and optimization, combined with SHAP and LIME for explainability, our approach provides fleet managers with both optimized solutions and insights into the decision-making process. Future research can focus on enhancing scalability for larger fleets by incorporating hybrid metaheuristic techniques or reinforcement learning, integrating real-time IoT-based traffic data and smart city analytics for adaptive route optimization, and further refining XAI methods to improve transparency and user adoption. Beyond fleet optimization, this work has broader implications for urban planning, sustainability, and environmental impact. Smarter transportation systems can benefit from optimized fleet routes, leading to reduced congestion, improved traffic management, and lower operational costs. Additionally, minimizing travel distances and idle time enhances fuel efficiency, while decreased carbon emissions align with global sustainability efforts for eco-friendly transportation systems. By bridging the gap between optimization, interpretability, and real-world applicability, this research contributes to more intelligent and sustainable fleet management solutions.

Declaration Ethical Approval
There are no any ethical conflicts. **Competing interests**
There is no conflict of interest.

Authors' contributions Equivalent Roles.

Funding Statement
The authors received no specific funding for this study. **Conflict of Interests/Competing Interests**
The authors declare that they have no conflicts of interest to report regarding the present study.
Data Availability Private data.

References

1. International Transport Forum. (2021). *Reducing transport costs through efficient route optimization*. Retrieved October 7, 2023, from https://www.itf-oecd.org/reducing-transport-costs-efficient-route-optimization
2. American Transportation Research Institute (ATRI). (2022). *Impact of route optimization on transportation logistics*. Retrieved December 25, 2023, from https://truckingresearch.org/2022/01/impact-of-route-optimizationon-transportation-logistics

3. Liu, X., Wang, H., & Zhang, Y. (2020). Optimization of logistics routes using genetic algorithms. *Journal of Transportation Research, 48*(3), 456-467.

4. Zhang, Y., Li, H., & Chen, X. (2019). Neural network-based approach to route optimization in transportation logistics. *International Journal of Artificial Intelligence, 12*(4), 367-380.

5. Chen, G., & Zhang, J. (2022). Applying artificial intelligence and deep belief network to predict traffic congestion evacuation performance in smart cities. *Applied Soft Computing, 121*, 108692. https://doi.org/10.1016/j.asoc.2022.108692

6. Lee, R. P., Seidl, L. G., Huang, Q.-L., & Meyer, B. (2021). An analysis of waste gasification and its contribution to China's transition towards carbon neutrality and zero waste cities. *Journal of Fuel Chemistry and Technology, 49*, 1057–1076. https://doi.org/10.1016/S1872-5813(21)60093-2

7. Fang, B., Yu, J., Chen, Z., et al. (2023). Artificial intelligence for waste management in smart cities: A review. *Environmental Chemistry Letters, 21*, 1959–1989. https://doi.org/10.1007/s10311-023-01604-3

8. Sharma, A., Podoplelova, E., Shapovalov, G., Tselykh, A., & Tselykh, A. (2021). Sustainable smart cities: Convergence of artificial intelligence and blockchain. *Sustainability, 13*(23), 13076. https://doi.org/10.3390/su132313076

9. Zhang, Y., Geng, P., Sivaparthipan, C. B., & Muthu, B. A. (2021). Big data and artificial intelligence-based early risk warning system of fire hazard for smart cities. *Sustainable Energy Technologies and Assessments, 21.*

10. Alloulbi, A., Öz, T., & Alzubi, A. (2021). The use of artificial intelligence for smart decision-making in smart cities: A moderated mediated model of technology anxiety and internal threats of IoT. *[Publication Information]*. https://doi.org/10.1155/2022/6707431

11. Xu, C., Wu, H., Zhang, Y., Dai, S., et al. (2022). A real-time complex road AI perception based on 5G-V2X for smart city security. *Wireless Communication and Mobile Computing, 2022*, 1–11. https://doi.org/10.1155/2022/4405242

12. Liu, T., Xu, C., Liu, H., Li, X., & Wang, P. (2021). A vehicle detection model based on 5G-V2X for smart city security perception. *Wireless Communication and Mobile Computing, 2021*, 1–11. https://doi.org/10.1155/2021/5237568

13. Kwon, K., Lim, S.-K., Kim, D., & Park, K. (2023). Automation program for optimum design of electric vehicle powertrain systems based on artificial neural network. *eTransportation, 18*. https://doi.org/10.1016/j.etran.2023.100267

14. Zhang, L., Sun, C., Cai, G., & Koh, L. H. (2023). Charging and discharging optimization strategy for electric vehicles considering elasticity demand response. *eTransportation, 18.* https://doi.org/10.1016/j.etran.2023.100262

15. Hassan, E., Talaat, F. M., Adel, S., Abdelrazek, S., Aziz, A., & El-Rashidy, N. (2023). Robust Deep Learning Model for Black Fungus Detection Based on Gabor Filter and Transfer Learning. *Computer Systems Science & Engineering*, *47*(2).

16. Talaat, F. M., Aljadani, A., Alharthi, B., Farsi, M. A., Badawy, M., & Elhosseini, M. (2023). A mathematical model for customer segmentation leveraging deep learning, explainable AI, and RFM analysis in targeted marketing. Mathematics, 11(18), 3930.

17. Hassan, E., Talaat, F. M., Hassan, Z., & El-Rashidy, N. (2023). Breast cancer detection: a survey. In Artificial intelligence for disease diagnosis and prognosis in smart healthcare (pp. 169-176). CRC Press.

18. Talaat, F.M., Hamza, A.A. Blockchain-enhanced artificial intelligence for advanced collision avoidance in the Internet of Vehicles (IoV). Neural Comput & Applic 37, 4915–4936 (2025). https://doi.org/10.1007/s00521-02410842-7

19. Tawfeek, M. A., Alrashdi, I., Alruwaili, M., & Talaat, F. M. (2025). A fuzzy multi-objective framework for energy optimization and reliable routing in wireless sensor networks via particle swarm optimization. Computers, Materials & Continua.

20. Tawfeek, M.A., Alrashdi, I., Alruwaili, M. et al. Enhancing the efficiency of lung cancer screening: predictive models utilizing deep learning from CT scans. Neural Comput & Applic (2025). https://doi.org/10.1007/s00521025-11084-x