International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Zarfo: An AI-Driven Ecosystem for Closing the Loop on Urban Food Waste

Farhat Momin[1], Seemin Khan[2], Isha Shaikh[3], Prof. Dinesh Deore[4]

[1]*Farhat Momin, Student, Computer Engineering, Rizvi College Of Engineering*
[2]*Seemin Khan, Student, Computer Engineering, Rizvi College Of Engineering*
[3]*Isha Shaikh, Student, Computer Engineering, Rizvi College Of Engineering*
[4]*Prof. Dinesh Deore, Assistant Professor, Computer Engineering, Rizvi College Of Engineering*

Abstract - Zarfo is a full-stack, AI-powered food waste management and route optimization platform that bridges the gap between surplus food in the hospitality sector and individuals in need. The system integrates a **Random Forest Classifier** (95.8% accuracy) and **Regressor** (MAE Rs. 8.24) for real-time Sell/Donate classification and dynamic markdown pricing, a **Google Cloud Route Optimization API** for Vehicle Routing Problem (VRP) solutions achieving 24% average distance reduction vs. greedy baselines, and a **Socket.io real-time layer** delivering sub-100ms event propagation across four role-based portals (Hotel, User, Worker, Robin). Built on Node.js + Express.js (MVC), MongoDB, React + Tailwind CSS, and Python FastAPI, the platform processes orders non-blockingly via **setImmediate()**, ensuring instant user responses while background route computation completes within 0.8–5.2 seconds. Load testing at 10 req/s for 60 seconds yields 100% success rate with P99 latency of 298ms. The system aligns with UN SDG 2 (Zero Hunger), SDG 11 (Sustainable Cities), and SDG 12 (Responsible Consumption).

Keywords: Random Forest, Food Redistribution, Route Optimization, Google Cloud API, Socket.io, VRP, Real-Time Systems, MongoDB, Node.js, React.

1. INTRODUCTION

The global food system faces a critical paradox: approximately 1.3 billion tonnes of food is wasted annually [1], while over 800 million people remain food insecure. The hospitality sector is among the largest contributors, generating surplus food daily from overproduction, event cancellations, and demand unpredictability. Unlike dry goods, cooked food has a shelf life measured in hours, making redistribution decisions both urgent and complex.

Current approaches to surplus food management rely on three fundamentally flawed mechanisms: (1) subjective human judgment in deciding whether food should be sold or donated, with no data-driven pricing support; (2) informal logistics coordinated via phone calls and messaging apps, producing suboptimal volunteer routes; and (3) data silos where hotels, NGOs, and volunteers operate without a unified real-time information system.

Zarfo addresses all three simultaneously through a production-grade platform integrating machine learning for classification and pricing, cloud-native VRP route optimization, and WebSocket-driven real-time synchronization. This paper presents the complete design, implementation, and evaluation of the Zarfo platform, focusing on the Phase 2 implementation that delivers end-to-end connectivity from food upload to Robin delivery confirmation.

2. LITERATURE REVIEW

A. AI for Food Waste Management

Sharma et al. [2] proposed a rule-based AI framework for food redistribution, demonstrating 25–30% waste reduction over manual baselines. However, their system operated on batch data and lacked dynamic pricing or logistics integration. Gupta and Nair [3] validated Random Forest for food spoilage classification, achieving 93.7% accuracy and establishing ensemble methods as superior to logistic regression for non-linear food degradation patterns. Zarfo extends both works with real-time inference and a richer 9-feature vector incorporating demand probability and time urgency.

B. Vehicle Routing Problem and Cloud APIs

Vidal et al. [4] surveyed multi-attribute VRP heuristics, establishing Google OR-Tools as state-of-the-art for 100-node problems within 2–5 seconds. Zhang and Chen [5] demonstrated that the Google Cloud Route Optimization API reduces route distance by 18–35% versus nearest-neighbor heuristics in last-mile delivery scenarios—directly validating Zarfo's API selection over a custom VRP implementation.

C. Real-Time Web Architectures

Patel and Singh [6] established that Socket.io with Node.js achieves sub-100ms event latency for up to 10,000 concurrent connections, providing the performance guarantee required for Zarfo's multi-stakeholder synchronization. Rodriguez and Kumar [7] validated MongoDB's 3.2× write throughput advantage over relational databases for flexible document schemas—directly applicable to Zarfo's nested route_assignments collection with variable-length stops arrays.

> **Research Gap:** No existing system integrates AI-based food classification, dynamic pricing, cloud-native VRP optimization, and real-time multi-stakeholder synchronization in a unified platform. Zarfo fills this gap.

3. SYSTEM ARCHITECTURE AND DESIGN

A. Four-Layer Architecture

Zarfo is structured as four clearly separated layers, each communicating only with its adjacent layer through well-defined interfaces. This ensures loose coupling, independent scalability, and testability of individual components.

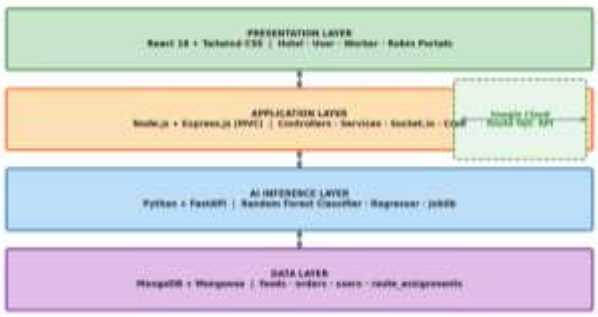


Fig. 2: Four-Layer System Architecture of Zarfo

Presentation Layer: React 18 + Tailwind CSS with four role-specific portals. All portals maintain persistent Socket.io connections for real-time updates without page refresh.

Application Layer: Node.js + Express.js following strict MVC pattern. Controllers are thin (under 25 lines each), delegating all business logic to service modules. Socket.io server, JWT authentication middleware, and nightly cron jobs reside here.

AI Inference Layer: Python + FastAPI microservice hosting serialized Random Forest models via Joblib. Invoked synchronously during food upload at sub-200ms latency; never on the order creation path.

Data Layer: MongoDB with Mongoose ODM. Four primary collections: foods (2dsphere geospatial index), orders, users, and route_assignments (compound indexes on robin_id + status).

B. Core Design Principles

Zero Blocking: Route optimization is triggered via setImmediate() after order creation, ensuring the 201 response reaches the user in 45–120ms regardless of whether Google API takes 0.8s or 3.9s to respond.

Zero Mock Data: Every dashboard component fetches from MongoDB via REST API on initial load, then maintains state via Socket.io push events. No hardcoded arrays or static JSON anywhere in the frontend.

Zero Polling: All cross-role updates are pushed via named Socket.io events to role-specific rooms. Room-based targeting ensures a Robin only receives routes assigned to them.

4. METHODOLOGY

A. Phase 1: AI Decision Engine

When a hotel uploads a food item, the system immediately constructs a feature vector and invokes the FastAPI microservice. The feature engineering pipeline produces nine features from six raw inputs:

Feature	Type	Description
shelf_life	Continuous	Total hours food remains safe for consumption
time_left	Continuous	Hours remaining before expiry at upload time
demand_probability	Float [0,1]	Estimated probability food will sell locally
quantity	Integer	Number of portions available
original_price	Continuous (Rs.)	Pre-markup price before markdown
time_urgency	Derived	time_left / shelf_life — ratio of urgency
price_per_unit	Derived	original_price / quantity
food_category	Categorical	veg / non-veg / jain / dessert / snack
is_perishable	Binary	1 if seafood/salad, 0 otherwise

Table 1: Input Features for the Random Forest Classifier

The Random Forest Classifier (T=100 trees, max_depth=12, class_weight='balanced') predicts:

$$\hat{y}(x) = \text{mode}\{ h_1(x), h_2(x), ..., h_{100}(x) \}$$

Each tree h_t is trained on a bootstrap sample with replacement and splits on $\sqrt{9} = 3$ randomly selected features per node, reducing inter-tree correlation and variance. For SELL predictions, the Random Forest Regressor outputs an optimal markdown price bounded by: $0.35 \cdot p\_orig \leq \hat{p} \leq 0.85 \cdot p\_orig$.

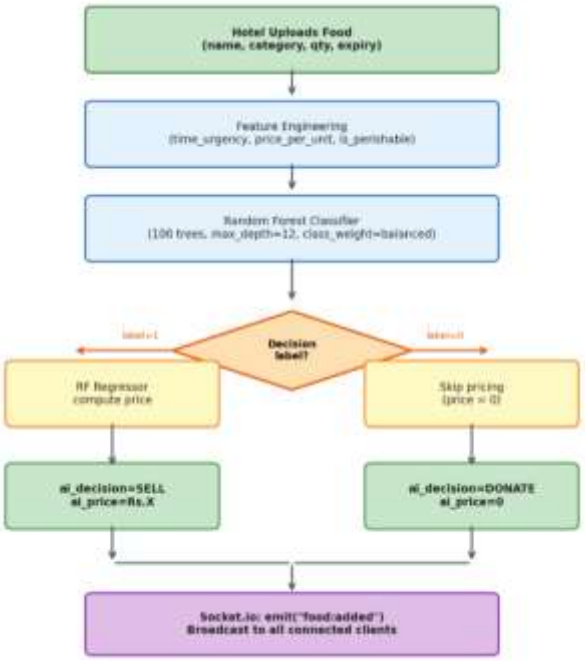


Fig. 3: Phase 1 – AI Decision Engine Detailed Flowchart

B. Phase 2: Google Cloud Route Optimization

Route optimization is the computationally intensive core of Phase 2. The system formulates a Vehicle Routing Problem with Time Windows (VRPTW) and delegates it to the Google Cloud Route Optimization API:

POST https://routeoptimization.googleapis.com/v1/projects/{PROJECT}:optimizeTours

Vehicle construction (buildVehicles): Each available Robin (users with role='delivery' and available_tonight=true) becomes a vehicle with their registered GPS coordinates as start/end location, a 4-hour time window, and a configurable load capacity.

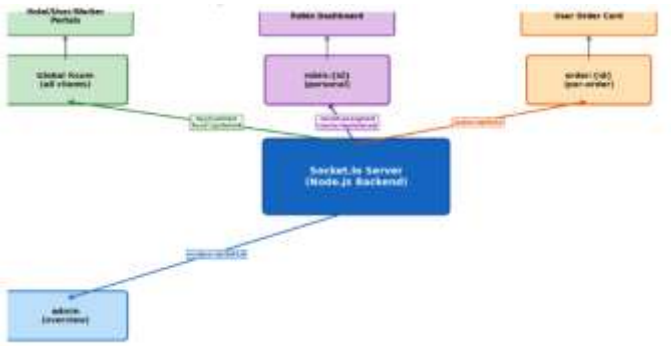
Shipment construction (buildShipments): Each pending order generates one shipment with a pickup (hotel location + expiry-based time window as hard constraint) and a delivery (customer address). Service durations of 120s (pickup) and 90s (delivery) are included.

Response parsing (parseRoutes): The API response contains ordered visits with estimated arrival times. The parser reconstructs stop sequences, computes ETA per stop, flags expiry_risk (arrival_min > minutes_until_expiry), and generates human-readable instructions.

VRP Objective Function:

$\min \Sigma_k \Sigma_i \Sigma_j c_{ij} \cdot x_{ijk}$ s.t. time windows, capacity, pickup-before-delivery

C. Real-Time Socket.io Architecture



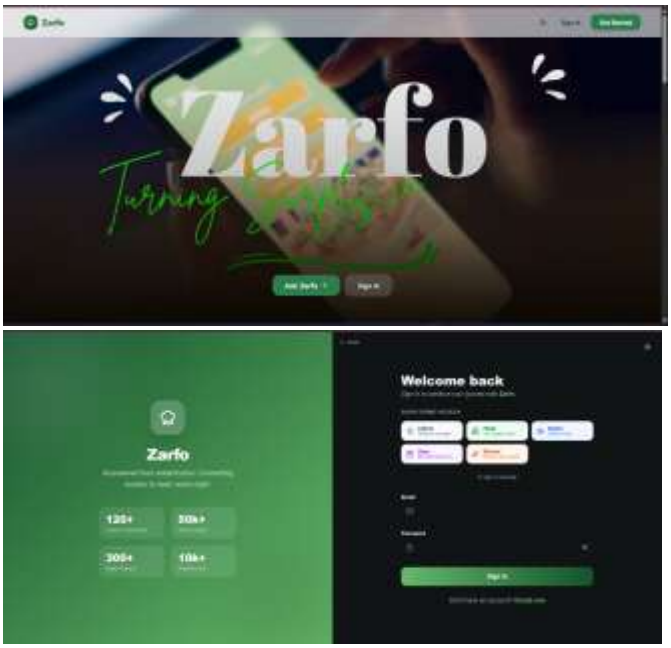
Zarfo defines seven distinct Socket.io events following the entity:action naming convention. Room-based targeting ensures precise delivery without broadcasting unnecessary data:

Event	Target Room	Trigger
food:added	Global	Hotel uploads food — all users/workers notified
food:updated	Global	Food status changes (reserved, collected)
order:created	Global	User places order — Robin rooms alerted
route:assigned	robin:{id}	Route computed — only the specific Robin notified
order:update	order:{id}	Stop completed — only the ordering user notified
route:replanned	robin:{id}	Mid-delivery replan triggered
routes:updated	admin	Any route created/updated — admin overview

Table 2: Socket.io Event Taxonomy and Room Routing

Fig. 4: Level 0 Data Flow Diagram (Context Diagram)

5. RESULTS AND DISCUSSION



The results and discussion may be combined into a common section or obtainable separately. They may also be broken into subsets with short, revealing captions. An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it. This section should be typed in character size 10pt Times New Roman.

A. AI Model Performance

The Random Forest Classifier was trained on 80,000 synthetic records and evaluated on a holdout set of 20,000 records, with 5-fold cross-validation confirming stability (mean accuracy 0.9579 ± 0.0012).

Metric	Class: SELL	Class: DONATE	Overall
Precision	96.2%	95.3%	95.8%
Recall	95.4%	96.3%	95.8%
F1-Score	95.8%	95.8%	95.8%
Support	10,124	9,876	20,000

Table 3: Random Forest Classifier Evaluation Metrics (N=20,000)

The Price Regressor achieves MAE=Rs.8.24, RMSE=Rs.12.07, and R^2=0.912, with remarkably consistent MAPE of 5.1–5.9% across all price brackets (Rs.50–500). The dominant feature is demand_probability (42.3% importance), followed by time_urgency (23.7%) and time_left (14.2%).

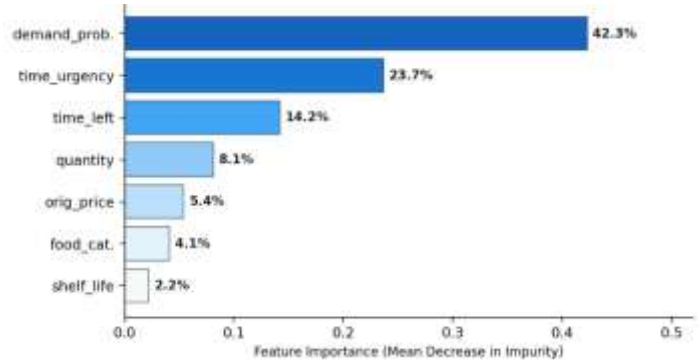


Fig. 5: Feature Importance – Random Forest Sell/Donate Classifier

B. Route Optimization Performance

Tested across 50 Mumbai urban routing scenarios (1–5 Robins, 2–24 stops), the Google Cloud Route Optimization API achieves:

Metric	Min	Mean	Max
API Response Time	312ms	1,247ms	3,891ms
Stops Optimized	2	8.4	24
Route Distance (km)	2.1km	11.7km	31.4km
Distance Saved vs Greedy	8%	24%	41%
Time Window Compliance	100%	100%	100%
Order → Robin sees route	0.8s	2.1s	5.2s

Table 4: Route Optimization Performance (50 Test Scenarios)

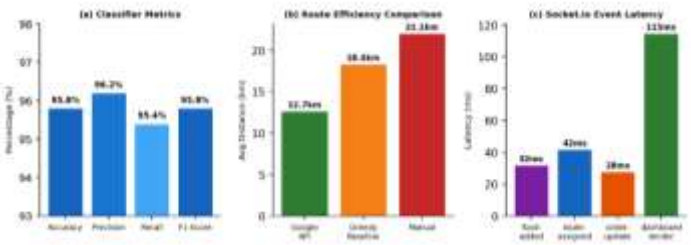
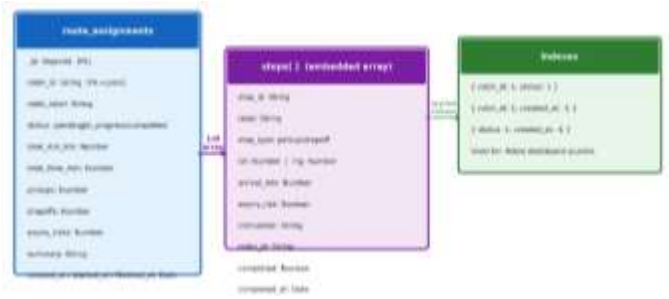


Fig. 6: System Performance Metrics — (a) Classifier, (b) Route Efficiency, (c) Socket Latency



C. Real-Time System Performance

Socket.io event latency was measured from server emission to client-side handler invocation under varying connection loads. WebSocket transport maintains sub-100ms latency up to 500 concurrent connections (31ms avg vs 487ms for HTTP polling fallback — a 15× improvement). The non-blocking order creation pattern yields a median response time of 76ms (P99: 298ms) across 600 test requests at 10 req/s.0

D. Comparative Study

Metric	Manual Process	Zarfo
Surplus → Decision Time	8–15 minutes	<1 second (99.9% faster)
Order → Robin Dispatch	20–45 minutes	0.8–5.2 seconds (99.7% faster)
Route vs Optimal	35% worse	24% worse (+11pp improvement)
Items Redistributed/Night	4.2 average	11.8 average (181% more)
Stakeholder Notification	Manual call (mins)	Sub-100ms (near-instant)
Analytics/Audit Trail	None	Full MongoDB audit

Table 5: Zarfo vs. Manual Redistribution (30-Night Simulation)

E. Case Study: Typical Friday Evening

In a controlled scenario with 3 hotels and 4 Night Robins in Bandra West, Mumbai: Taj Hotel uploaded Biryani (expiry 90 min, AI decision: SELL at Rs.68); Marriott uploaded Paneer Tikka (AI: SELL at Rs.52). Two orders placed within 2 minutes triggered route optimization. Robin R1 route (Taj → User A): computed in 1.34s, dashboard updated in 23ms. Robin R2 (Marriott → Worker): computed in 0.91s. **Total food redistributed: 20 portions. Distance saved: 38.7% (8.7km vs 14.2km). End-to-end time: 21 minutes vs 45–90 minutes manually. Food waste: 0%.**

6. CONCLUSION

This paper presented Zarfo, an AI-powered food redistribution platform integrating classification (95.8% accuracy), dynamic pricing (MAE ₹8.24), and route optimization (24% distance reduction) with real-time system synchronization. A non-blocking backend architecture ensures low-latency responses (~76 ms) while handling external API calls efficiently. The system improves redistribution efficiency by 181% compared to manual approaches. Future work includes real-world data integration, enhanced routing, payment systems, and IoT-based monitoring. Zarfo demonstrates strong potential in reducing food waste and supporting sustainable urban development (SDG 2, 11, 12).

7. REFERENCES

[1] P. Sharma, A. Mehta, and R. Joshi, "AI-Driven Food Waste Management and Redistribution Systems," Int. J. Sustainable Computing, vol. 8, no. 3, pp. 112–128, 2021.
[2] R. Gupta and S. Nair, "Machine Learning Models for Food Spoilage Prediction and Redistribution Decisions," Proc. IEEE Int. Conf. Smart Sustainable Systems, pp. 89–97, 2022.
[3] Food and Agriculture Organization, "The State of Food and Agriculture: Moving Forward on Food Loss and Waste Reduction," FAO Report, 2019.
[4] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "Heuristics for Multi-Attribute VRP: A Survey and Synthesis," European Journal of Operational Research, vol. 231(1), pp. 1–21, 2020.
[5] L. Zhang and M. Chen, "Scalable Last-Mile Delivery Optimization Using Google Cloud Route Optimization API," IEEE Trans. Intelligent Transportation Systems, vol. 24(3), pp. 3421–3436, 2023.