RESPOND: A Modular Platform for Urban Emergency Response Research and Decision Support

Ammar Bin Zulqarnain*, Jose Paolo Talusan*, Kelly Napier†, Corey Gens‡, Jennifer Higgs‡, Colleen Herndon‡, Ayan Mukhopadhyay*, and Abhishek Dubey*,
*Vanderbilt University, Nashville, TN, USA
†Nashville Fire Department, Nashville, TN, USA
‡Metro Government of Nashville and Davidson County, Nashville, TN, USA

*{ammar.bin.zulqarnain,jose.paolo.talusan,ayan.mukhopadhyay,abhishek.dubey}@vanderbilt.edu
†kelly.napier@nashville.gov
‡{corey.gens,jennifer.higgs,colleen.herndon}@nashville.gov

***Abstract*—Growing urban populations strain fire/Emergency Medical Services (EMS) systems, creating societal-scale concerns where decisions about station siting (strategy) and dispatch policies (operations) unfold in a tightly coupled cyber-physical loop. The core challenge lies in validating different approaches since direct experimentation on real populations is infeasible. Prior efforts address isolated components, treating strategic siting heatmaps and operational dispatch heuristics as separate problems. They lack a unified, incident-level simulator to expose the critical cross-policy trade-offs between siting and dispatch. We present RESPOND (REsponse Simulation Platform for Operations, Navigation, and Dispatch), a modular, incident-level, Operational Decision Support System. RESPOND holistically integrates these previously siloed functions, including: (i) optimal station placement, (ii) apparatus allocation, (iii) dispatch policies, (iv) travel time and service time models, and (v) survival modeling for incident prediction. The platform's engine replays historical incidents at unit resolution and stress-tests counterfactual futures (e.g., station moves, demand surges). A planner-facing interface surfaces key metrics (SLA compliance, 90th Percentile (P90) response time) for deliberation. Evaluations demonstrate reproduction of observed response patterns and reveal policy trade-offs. The result is a unifying platform that transforms fragmented analysis into an operational decision environment, enabling safe and rigorous evaluation of coupled station placement and dispatch policies through simulation.**

***Index Terms*—Emergency Response, Decision Support Systems, Simulation, Resource Allocation, Facility Location**

I. Introduction

Urban population growth has led to significant increases in emergency call volumes, placing immense pressure on fire departments and emergency medical services (EMS) [1]. Agencies are expected to meet strict response-time standards, such as those specified in NFPA 1710, even as resources remain constrained by fiscal and operational limits [2]. This high-stakes domain represents a canonical, societal-scale cyber-physical system (CPS) at the nexus of civil infrastructure and social need, one that tightly couples computational logic with physical infrastructure and critical, human-in-the-loop decision-making by agency planners and dispatchers. The effective management of this CPS is a rapidly growing national priority; the U.S. disaster management market is projected to reach $87.01 billion by 2034, driven by an increasing frequency of natural and man-made disasters [3]. In this high-pressure environment, fire chiefs must coordinate interdependent choices, balancing long-term infrastructure strategy (station siting & apparatus mix) with real-time operational tactics (dispatch protocols & deployment), all under a tight budget and service-level constraints [4]. These decision classes are coupled: the effectiveness of a real-time dispatch policy is bounded by the static station layout, and the value of adding a station depends on the policies used to deploy its units [5], [4].

With real-world experiments constrained by risks and cost, simulation remains the primary tool for evaluating policy and design alternatives [6]. Yet prior work remains fragmented [4]. Strategic planning largely relies on covering and p-median formulations for station siting [7], but these static models overlook dynamic realities such as unit busy fractions and concurrent demand, biasing realized coverage and response times [8], [5]. In parallel, dispatch research optimizes operational tactics via decision-theoretic/SMDP formulations [9] while treating station layout as fixed, and digital-twin efforts focus on hazards (e.g., smoke dispersion) rather than apparatus siting or dispatch [10], [11]. Consequently, a unified, high-fidelity testbed to evaluate coupled trade-offs between strategic siting and dynamic dispatch in a single closed-loop environment remains missing. We summarize our contributions as follows:

- **Unified simulation platform**: We introduce RESPOND, a modular CPS framework that integrates strategic planning and operational dispatch evaluation within a single environment.
- **Composable architecture**: We design a plug-and-play system across data, policy, routing, service-time, and simulation layers, enabling systematic comparison of methods.
- **Scenario-driven decision support**: We enable rigorous

evaluation of counterfactual scenarios with direct comparison to baseline operations, supporting data-driven planning.
- **Practitioner-oriented system**: We couple an incident-level simulator with an interactive interface to support real-world deployment and decision-making.

II. Related Works

Prior efforts in emergency response are divided among three distinct streams, each evolving in parallel with minimal interaction.

The first focuses on strategic planning, primarily fire station siting. Extensive research has addressed optimal station locations using classical facility location models such as covering formulations and p-median formulations [12], [7]. These approaches identify station sites to maximize demand coverage or minimize response distances, providing important insights into long-term resource allocation. However, a critical limitation is their static nature: they typically assume each station's units are always available to respond [8]. In practice, emergency units are often occupied with ongoing incidents, so coverage is probabilistic rather than deterministic. Neglecting dynamic unit availability can lead to suboptimal layouts when evaluated under realistic call patterns [8]. In essence, a layout that is optimal on paper can underperform in practice.

The second stream concentrates on dispatch optimization. The dispatch problem was formulated as a semi-Markov decision process within a decision-theoretic framework, aiming to minimize response times under uncertainty. This approach leverages techniques such as policy iteration and Monte Carlo Tree Search to compute near-optimal dispatch decisions in real time [9]. This work was extended by introducing an online pipeline that continuously adapts the dispatch policy to changing emergency conditions [13]. These contributions represent the state of the art in dispatch optimization, demonstrating significant response time reductions through smarter allocation of calls to units. However, dispatch optimizations inherently assume fixed infrastructure. Station locations and fleet distributions are taken as given, so this line of work cannot answer strategic "what-if" questions about how performance might change if a station were opened or relocated.

The third stream comprises high-fidelity simulations and digital twins. FireCom [10] is a digital twin platform that models urban fire incidents by integrating 3D city data, real-time sensor feeds, and physical fire dynamics to simulate hazard evolution. It provides rich insight into fire progression and impact, supporting decisions on evacuation and resource allocation, but does not simulate the dispatch or placement of emergency response units. Similarly, simulation-based call-center twins evaluate how organizational structures (e.g., consolidating regional centers) affect call handling performance [14]. These tools stop at the call-center boundary, omitting field deployment and coupled dispatch-siting decisions.

III. System Overview

We organize the description in a blocks-first order as shown in Figure 1 and tie each block to its role in simulation and analysis. The structure follows: (§III-A) Architecture & Dataflow, (§III-B) Data & Ingestion, (§III-C) Policy Layer, (§III-D) Travel-Time Estimation, (§III-E) Service-Time Modeling,(§III-F) Simulation Kernel (with an embedded MDP interface) and (§III-G) Analytics & Reporting.

Notation. We use *incident* to mean a single Computer Aided Dispatch (CAD) event requiring a response; *apparatus* to mean a responding vehicle (e.g., Engine, Medic); and *beat/service zone* to denote polygons used by zone-based dispatch policies.

A. Architecture & Dataflow

The platform has four cooperating layers and a simulation kernel:

1) **Data & Ingestion** (§III-B): normalizes historical incidents, station inventories, apparatus logs, and geographic assets into a canonical, versioned form.
2) **Policy Layer** (§III-C): plugin dispatch strategies (e.g., zone-based or nearest-available policies) that map state observations to one or more dispatch actions.
3) **Travel-Time Layer** (§III-D): multi-modal ETA computation via network routing and a fast historical interpolation model, with hybrid fallbacks.
4) **Service-Time Layer** (§III-E): on-scene duration modeling via empirical sampling or Model-based forecasters.
5) **Simulation Kernel** (§III-F): a discrete-event engine that advances incidents and vehicles through the operational lifecycle and exposes a clean MDP interface for learning-based dispatch.

The simulation kernel is implemented in C++, and the front end in JavaScript; the platform exposes a stable API layer for data ingestion, policy modules, and analytics. Analysts define a *scenario specification* (time window, station and apparatus configurations, dispatch policy, routing, and service-time models), which acts as a configuration layer governing all modules. Each module exposes a standardized interface via an abstract base class: demand-generating modules output a canonical *incident object*—encoding timestamp, location, category, and resource requirements—consumed agnostically by downstream modules (Dispatch, Routing, Service-Time) regardless of whether the source is historical or synthetic. Upon submission, the Data Layer re-indexes geographic assets, the Travel-Time Layer recalculates ETA matrices, and the Kernel reinitializes station inventories—ensuring full consistency before replay begins. The identical machinery supports *(a)* historical replay and *(b)* counterfactual runs.

B. Data & Ingestion

We curate and standardize operational data (incidents, stations, and geography) and, when needed, synthesize near-future demand using a survival-regression generator to support reproducible simulation and analysis.

- **Incident logs**: timestamped calls with locations (points/zones), categories (e.g., non-structured/structured fire), and requested resources; and strictly time-ordered.
- **Stations/apparatus**: station geocodes, beat/service-zone polygons, per-station apparatus inventories (types, counts), and optional staffing windows.
- **Geographic assets**: OpenStreetMap-derived road graph (for network routing), GeoJSON for beats/zones.

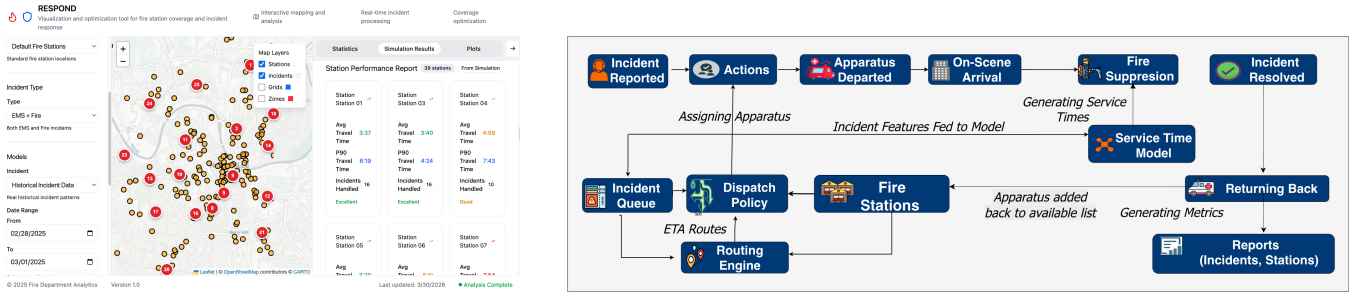

Fig. 1: RESPOND platform overview: (left) user interface with simulation controls, a geospatial map of incidents and stations, and simulation results; (right) simulation kernel showing how each reported incident triggers policy and routing queries, simulates dispatch through on-scene arrival, samples service time, returns apparatus to the pool, and emits KPIs/reports.

- **Travel-time matrices** : per (station-zone, incident-zone) mean and variance estimated from apparatus GPS traces to support historical interpolation.

Station-network editing. Analysts may (i) retain the baseline map; (ii) drag-and-drop relocations; (iii) add candidate stations (pre-populated apparatus sets); (iv) close/consolidate stations; (v) re-roster apparatus across stations; and (vi) add an *optimized* station, where layouts are precomputed offline for each grid resolution (0.5 and 1.0 mile) and budget $K \in \{1, \ldots, 5\}$, and the user selects a $(K, \text{resolution})$ option at simulation time. When beats exist, FIREBEATS uses predefined beats; otherwise, NEAREST is applied by default.

Synthetic Incident Generation. The platform's modular architecture extends to incident generation, allowing planners to "plug and play" custom demand models beyond default historical replay. As one such implementation, we include a synthetic incident generator employing a feature-conditioned exponential survival model to produce realistic arrivals with complex spatial and temporal structures. In this module, the service area is partitioned into uniform grid cells (e.g., 1-mile resolution), each maintaining a set of covariates $x_g(t)$ (e.g., hour-of-day, day-of-week, recent demand), which are then clustered based on historical incident density. For each cluster c, the model learns coefficients w_c that define the expected inter-arrival time $\mathbb{E}[\Delta T_g \mid x_g, c] = \exp(w_c^\top x_g)$ and the corresponding hazard rate $\lambda_g(x_g) = \exp(-w_c^\top x_g)$. A competing time-to-event sampling process generates the global incident timeline: for each cell g, a candidate inter-arrival time $\Delta t_g \sim \text{Exponential}(\lambda_c(x_g(t)))$ is drawn. The next incident occurs at $\min_g \Delta t_g$ in the corresponding cell g^*. The clock then advances, all covariates $x_g(\cdot)$ are updated, and the process repeats. Finally, an incident category is sampled from the empirical conditional distribution $p(\text{category} \mid c)$ to preserve realistic mixtures, allowing this survival-regression approach to reproduce diurnal/weekly seasonality and spatial heterogeneity. We point out that our tool is completely configurable to different modeling paradigms, e.g., in theory, one could model category-specific arrival processes and even encode conditional dependencies (e.g., a fire triggering a secondary crash). However, sparse classes made per-category survival models high-variance in our data. To balance heterogeneity with statistical stability, we fit a single inter-arrival process across categories and then sample incident types from cluster-specific empirical frequencies, preserving spatial–temporal mix while avoiding overfitting.

C. Policy Layer (Dispatch & Constraints)

Policies implement `getAction(State)` $\rightarrow$ `vector<Action>`, returning one or more dispatch assignments that satisfy incident category requirements.

Rule-based Dispatch Policies: We have implemented two dispatch policies: (1) *NEAREST* which greedily minimizes ETA under apparatus-type constraints, with optional tie-breakers on current workload or home-station proximity. (2) *FIREBEATS* which adheres to beat-based ordering: primary station $\rightarrow$ backups to preserve territorial integrity.

Extensibility: While the current release includes only *NEAREST* and *FIREBEATS*, the simulator is designed for drop-in policy extensions. The dispatch interface (`getAction(State)`) and a central feasibility validator allow adding learned or hybrid policies without modifying the simulation kernel. Hard constraints (apparatus typing, minimum coverage, mutual-aid triggers) remain enforced centrally to ensure safety.

D. Travel-Time Estimation (Routing Modules)

Accurate and configurable travel-time estimation is essential because *travel* dominates total response time and is the first lever impacted by dispatch policy or station layout changes. The simulator therefore exposes interchangeable routing modules:

OSRM network routing: The system provides path-accurate routing on an OSM-derived road graph with turn restrictions and one-ways; outputs include route polylines, distances, and ETAs. This serves as the transparent baseline and supports incident-specific route playback.

Interpolated historical model: A fast emulator stores mean and variance for (station-zone, incident-zone) travel times estimated from apparatus GPS traces. At runtime, it retrieves the reference pair, draws a travel time from a Gaussian, and scales by the ratio of actual to reference distance for minor OD mismatches. Nearest-k caching accelerates unseen pairs and captures travel time patterns.

E. Service-Time Modeling (Empirical & ML)

Empirical sampling. On-scene durations are sampled from stratified historical distributions by incident category, capturing heavy-tailed and multimodal distributions.

ML predictors. Context-aware regressors (e.g., random forest or gradient boosting) use features such as time-of-day/week, category, geography. Pipelines include encoders/scalers packaged with the model for portable inference.

F. Simulation Kernel (Event Model & State)

The kernel advances incidents and vehicles via a discrete-event engine and exposes an MDP-compatible interface so rule-based or RL dispatch policies can act on a fully observable state and receive reproducible environment transitions.

Event-driven core: The kernel is a discrete-event simulator with a global clock and a priority queue of events, processed chronologically. Concurrency is native; multiple incidents and vehicles can be in-flight simultaneously. The incident lifecycle emits: *IncidentReport* $\rightarrow$ *DispatchDecision* $\rightarrow$ *VehicleDeparture* $\rightarrow$ *ArrivalOnScene* $\rightarrow$ *IncidentResolve* $\rightarrow$ *ReturnToStation*. Vehicle state transitions are explicitly modeled as: *Available* $\rightarrow$ *Dispatched* $\rightarrow$ *EnRoute* $\rightarrow$ *OnScene* $\rightarrow$ *Returning* $\rightarrow$ *Available*.

MDP interface for dispatch research: To support dispatch research, the kernel exposes a Gym-compatible MDP interface.

State representation. The `State` object provides full observability, including temporal context, station inventories, per-vehicle fleet status, and the active incident queue.

Action space. The discrete action space consists of all valid (station, vehicle, incident) dispatch assignments.

Transition dynamics. A call to `step(action)` advances the discrete-event simulation, updating all vehicle and incident states based on stochastic travel and service time models, and returns the next state.

Reward model. No default reward is shipped in the current release. Policy quality is therefore assessed via KPIs in §III-G (e.g., P90 travel/response time, SLA compliance, utilization, unserved calls). When a user-supplied reward module is provided, it is invoked without changes to the simulation kernel.

Episodes and horizons. `reset()` initializes stations/fleets; episodes run for configurable simulated durations (e.g., 1 day/week) and terminate on horizon.

G. Analytics & Reporting

Core KPIs: We report the following indicators for both historical replay and counterfactual evaluation: (1) **Travel Time** (dispatch $\rightarrow$ arrival): mean and P90, stratified by *station* and *incident category*; (2) **Service Time** (on-scene duration): mean by *incident category* and *unit type* with sensitivity to *multi-unit deployments*; (3) **Incident Loads per Station**: counts handled per station and apparatus type; and (4) **SLA Compliance**: % of incidents with ETA $\leq$ a configurable target (e.g., 5 minutes for first engine);

Online Dashboard: A series of views are included in the web dashboard for operators. The intent is to deploy the tool in a format that is easily accessible and readily available to decision makers. These include: (1) **Geospatial canvas**: stations, beats, incidents, and routes; optional overlays for SLA hits/misses and cross-beat responses; (2) **Station dashboards**:Incident load distributions, and travel/service time summaries per station and apparatus type; and (3) **KPI panels**: travel and total service time (mean/P90), SLA compliance gauges, and incident counts per station tables for benchmarking runs. The primary goal of this dashboard is to facilitate evidence-based decision-making by allowing planners to set a baseline run (e.g., historical data) and then visually and statistically compare it against one or more scenarios.

IV. Experiments, Results and Discussions

To demonstrate RESPOND's capabilities as both a modular research testbed and an operational decision support system, our evaluation is twofold. First, we conduct a fidelity assessment that leverages the platform's core modularity. We test and compare various configurations of its plug-in components (dispatch, travel time, and service time) against historical ground truth data. Second, we use this selected configuration to conduct exploratory "what-if" analyses, demonstrating the platform's utility for strategic and operational planning.

A. Experimental Setup

We conducted the study in a mid-sized U.S. metropolitan area, using the existing fire station network. The evaluation period spans from January 1, 2025, to July 20, 2025. Historical data was drawn from the city's logs, providing a real-world baseline for incident timestamps, locations, and unit response patterns. We test the fidelity of various module combinations within RESPOND. These configurations vary across three primary axes: **(1) Dispatch Policy:** NEAREST vs. FIREBEATS, **(2) Travel-Time Module:** OSRM vs.INTERPOLATED and **(3) Service-Time Module:** HISTORICAL vs. ML.

To evaluate both the simulator's fidelity and the impact of counterfactual scenarios, we define two categories of metrics:

1) **Fidelity Assessment Metrics:** Used to compare simulation outputs against the historical Ground Truth dataset. These include *Aggregate Outcomes* which include **Coverage % Difference** (number of incidents responded to within 320 seconds) and **MAE of Incident Counts** per station to validate spatial and load distribution and *Temporal Indicators:* **Total Service Time** and **Travel Time Mean Absolute Error (MAE)**

2) **Exploratory Analysis KPIs:** Core operational metrics reported by the RESPOND platform for "what-if" scenarios. These include Resource Specific Metrics: Mean and 90th-percentile (P90) Travel Time and Station incident counts.

B. Fidelity Assessment

The objective of the fidelity assessment is to identify the simulation configuration that most closely reproduces the aggregate operational dynamics of the real system. We use MAE for key performance indicators, as summarized in Table I.

The `FIREBEATS` configurations demonstrate high fidelity in replicating the system's operational dynamics, particularly

TABLE I: Simulation Fidelity Comparison. Bold indicates best per column; underlined indicates second-best.

Configuration (Dispatch_Service_Routing)	MAE (counts) Incident Counts	Coverage Diff. (%)	MAE (s) Travel Time	MAE (s) Service Time	MAE (s) P90 per Station
FIREBEATS_ML_OSRM	6.0	15.4%	**121.9**	**794.8**	**170.8**
NEAREST_ML_INTERPOLATED	27.4	29.2%	283.8	905.6	557.7
FIREBEATS_ML_INTERPOLATED	7.6	3.7%	338.6	918.7	593.7
NEAREST_HISTORICAL_INTERPOLATED	30.3	28.9%	283.6	1201.9	602.3
NEAREST_HISTORICAL_OSRM	8.8	18.1%	122.7	1119.9	190.7
FIREBEATS_HISTORICAL_OSRM	**5.9**	15.7%	121.9	1122.4	171.6
NEAREST_ML_OSRM	8.8	17.9%	122.7	796.1	191.6
FIREBEATS_HISTORICAL_INTERPOLATED	7.5	**2.0%**	341.1	1287.6	598.8

in stations' assignment. The low MAE for Incident Counts Per Station (6.0) confirms that the zonal-based policy accurately mirrors the real-world policy used by Fire Departments, ensuring that simulated incident distributions closely align with ground truth.

While this FIREBEATS model sets a high benchmark, a broader look at the results reveals a clear performance hierarchy. A major performance differentiator is the routing method. The four INTERPOLATED models all perform poorly on **MAE Travel Time** (with errors ranging from 283.6s to 341.1s) and show high errors in **MAE Incident Counts Per Station**. This suggests an inability to capture travel dynamics and spatial incident load accurately.

This leaves the four OSRM configurations as the more viable candidates. Within this group, the choice of service time model is critical. The HISTORICAL models (NEAREST_HISTORICAL_OSRM and FIREBEATS_HISTORICAL_OSRM) show relatively better fidelity on **MAE Travel Time** (approx. 122s) but perform poorly on **MAE Service Time** (1119.9s and 1122.4s, respectively).

Conversely, the ML-based models are the only configurations to achieve better fidelity across all key metrics, demonstrating low MAE in both **MAE Travel Time** (approx. 122s) and **MAE Service Time** (approx. 795s). This establishes them as the best configurations.

Focusing on FIREBEATS_ML_OSRM achieves a slightly more robust fidelity across all metrics. It performs marginally better on both MAE Service Time (794.8s vs. 796.1s) and MAE Travel Time (121.9s vs. 122.7s), and more significantly on MAE Incident Counts (6.0 vs. 8.8) and MAE P90 Per Station (170.8 vs. 191.6).

However, FIREBEATS, has a critical operational drawback of being by design, a static, zonal-based policy fixed to the existing station layout. It cannot accommodate the "what-if" scenarios involving new or removed stations, as its dispatch logic relies on pre-defined "beats." In contrast, the NEAREST policy naturally handles any network modifications.

Given this constraint, we must select our baseline not only on raw fidelity but also on its suitability for exploratory analysis. As NEAREST_ML_OSRM is the other top-tier, high-fidelity configuration and is the *only* one that is functionally generalizable to the required station siting scenarios, we select it as the validated baseline for the analysis that follows for one of the scenarios.

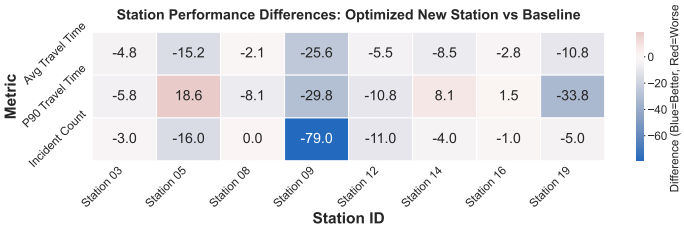

Fig. 2: Station performance differences after adding an optimally placed station. Cells report per-station deltas (*Avg/P90 Travel Time* in seconds; *Incident Count* in incidents).

C. Counterfactual Scenario-based Analysis

Leveraging the validated FIREBEATS_ML_OSRM and NEAREST_ML_OSRM configuration as our baselines, we demonstrate RESPOND's utility for strategic planning by simulating four "what-if" scenarios. We evaluate the impact of each scenario by analyzing the core KPIs, the planners required.

Scenario 1 & 2: Station Network Modification: To evaluate strategic siting decisions, we simulated two scenarios: (A) adding a new station near downtown and (B) removing a firestation near suburban area where the frequency of fire is low.

Adding an optimized station near the downtown (Scenario A) core led to improvements across multiple response metrics. As shown in the comparative plot in Figure 2, most stations experienced reductions in both average and 90th-percentile (P90) travel times, indicating faster service delivery for high-demand areas. The largest gains were concentrated around stations adjacent to the new site, where average travel times improved by 10–25 seconds and incident loads were redistributed away from previously overburdened stations. This suggests that the optimized addition effectively alleviated spatial imbalance in coverage and reduced peak congestion. In contrast, eliminating a suburban (Scenario B) station—produced localized degradation in both average and P90 travel times, primarily for nearby zones, with only minor ripple effects on the broader network.

Scenario 3: Apparatus Reallocation To model fleet adjustments, we simulated re-rostering apparatus at a single high-volume station near the downtown area, specifically by adding one Engine and one Medic unit to its existing fleet .

Adding apparatus capacity at a station near downtown led to measurable improvements concentrated around its neighboring

response zones. The average travel time decreased by up to 6–7 seconds for adjacent stations, reflecting faster unit availability and reduced dispatch overlap. Similarly, the P90 travel times show notable improvement for one of the neighboring stations (24 seconds), suggesting that the redistribution of apparatus helped alleviate prior bottlenecks in that corridor. The incident count distribution indicates a modest shift of calls nearby stations toward the allocated station, confirming a mild rebalancing of workload. Overall, the reallocation enhances coverage efficiency, improving response reliability for high-volume areas.

Scenario 4: Synthetic Demand Generation To demonstrate RESPOND's capability to analyze future or hypothetical demand, we test its integration with the synthetic incident generator (§III-B). We use the module to generate a new incident log for one month (August 2025) and run this synthetic data through our validated simulation configuration. The test serves to show the platform's ability to ingest and execute non-historical data streams.

For this synthetic month, the system achieved an average Mean Travel Time of 240.24 s and a P90 of 390.10 s. SLA compliance was 80.72%, with 180/223 incidents within SLA Compliance. These results indicate acceptable central tendency with a long-tail risk driven by peak-load conditions; the gap between the mean and P90 highlights that extreme cases dominate perceived service quality. The observed compliance shortfall flags potential bottlenecks under this hypothetical demand pattern.

V. Conclusion

This paper introduced **RESPOND**, an operational decision support system that bridges the critical gap between long-term strategic planning and real-time operational dispatch for urban emergency services. By unifying these once-siloed components, RESPOND provides a high-fidelity simulation environment, enabling planners to rigorously evaluate complex "what-if" scenarios such as station closures or fleet reallocations without risking public safety.

Our fidelity assessment validates this approach, demonstrating that a properly calibrated digital twin can reproduce historical operations with a quantifiable, known degree of error. RESPOND's primary contribution lies not in a single model, but in its modular and extensible architecture designed to evaluate different allocations of resources and scenarios in comparison with the baseline. As demonstrated, researchers can incorporate novel models such as context-aware service time predictors or AI-driven travel time engines directly into the platform's layers. This allows the community to isolate and improve individual components, using RESPOND as a shared testbed to validate their impact on system-wide performance. It thus provides a crucial, data-driven tool for iteratively enhancing the cyber-physical systems that underpin life-critical emergency response. To support reproducibility and follow-on research, we will release curated simulated datasets generated by RESPOND (including synthetic incident logs and scenario definitions), together with module configurations and evaluation scripts sufficient to reproduce the principal results reported here.

Future Work focuses on improving fidelity and leveraging it for advanced decision-support research. By developing more robust data cleaning pipelines, we aim to further reduce the reality gap. We plan to leverage this validated MDP interface to develop and evaluate novel, learning-based dispatch policies that can simultaneously optimize response time, resource utilization, and geographic equity under dynamic constraints.

Acknowledgment

This material is based on work sponsored by the Nashville Innovation Alliance. We thank the Nashville Fire Department and Nashville Metropolitan Information Technology Services for their invaluable feedback and domain expertise, which helped us better understand the subtleties and complexities of emergency response. This work was supported in part by the National Science Foundation (NSF) under Grants CNS-2238815 and CNS-1952011. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF. Results reported in this paper were obtained using the Chameleon testbed, which is supported by the NSF.

References

[1] City of Coeur d'Alene, "Keeping up with growth: Emergency calls have increased by 35%," City News Release, Coeur d'Alene Fire Department, 2025.

[2] Johnson City Fire Department, "NFPA 1710 Fact Sheet," Informational Brochure, Johnson City, TN Fire Dept., 2012, summary of NFPA 1710 response time and staffing standards.

[3] Statifacts, "US Disaster Management Market Size to Attain USD 87.01 Billion by 2034," July 2025, uRL: https://www.statifacts.com/outlook/us-disaster-management-market.

[4] D. Neira-Rodado, J. W. Escobar-Velásquez, and S. McClean, "Ambulances deployment problems: Categorization, evolution and dynamic problems review," *ISPRS International Journal of Geo-Information*, 2022.

[5] R. C. Larson, "A hypercube queueing model for facility location and redistricting in urban emergency services," *Computers & Operations Research*, pp. 67–95, 1974.

[6] G. M. Carter and E. Ignall, "A simulation model of fire department operations: Design and preliminary results," The New York City Rand Institute, New York, NY, Tech. Rep. R-632-NYC, 1970.

[7] R. L. Church and C. ReVelle, "The maximal covering location problem," *Papers of the Regional Science Association*, pp. 101–118, 1974.

[8] M. S. Daskin, "A maximum expected covering location model: Formulation, properties and heuristic solution," *Transportation Science*, pp. 48–70, 1983.

[9] A. Mukhopadhyay, Z. Wang, and Y. Vorobeychik, "A decision theoretic framework for emergency responder dispatch," pp. 588–596, 2018.

[10] R. H. Lewis, J. Jiao, K. Seong, A. Farahi, P. Navrátil, N. Casebeer, and D. Niyogi, "Fire and smoke digital twin – a computational framework for modeling fire incident outcomes," *Computers, Environment and Urban Systems*, 2024.

[11] W. S. Lim and et al., "Location optimization of urban fire stations: Access and service coverage," *Socio-Economic Planning Sciences*, 2024.

[12] S. L. Hakimi, "Optimum locations of switching centers and the absolute centers and medians of a graph," *Operations Research*, vol. 12, no. 3, pp. 450–459, 1964.

[13] A. Mukhopadhyay, G. Pettet, C. Samal, A. Dubey, and Y. Vorobeychik, "An online decision-theoretic pipeline for responder dispatch," *International Conference on Cyber-Physical Systems*, 2019.

[14] Y. Penverne, C. Martinez, N. Cellier, C. Pehlivan, J. Jenvrin, D. Savary, V. Debierre, F. Deciron, A. Bichri, Q. Lebastard, E. Montassier, B. Leclere, and F. Fontanili, "A simulation-based digital twin approach to assessing the organization of response to emergency calls," *npj Digital Medicine*, p. 385, 2024.