

وليد عبدالجواد

Metro_AI_Resource_Article.docx

 Midocean University

Document Details

Submission ID

trn:oid::2945:400539721

Submission Date

Jul 21, 2026, 2:50 PM GMT+3

Download Date

Jul 27, 2026, 10:33 PM GMT+3

File Name

Metro_AI_Resource_Article.docx

File Size

14.3 KB

5 Pages

1,472 Words

8,628 Characters

0% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

-  **0 Not Cited or Quoted 0%**
Matches with neither in-text citation nor quotation marks
-  **0 Missing Quotations 0%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 0%  Internet sources
- 0%  Publications
- 0%  Submitted works (Student Papers)

Integrity Flags

1 Integrity Flag for Review

-  **Hidden Text**
12 suspect characters on 1 page
Text is altered to blend into the white background of the document.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- 
0 Not Cited or Quoted 0%
 Matches with neither in-text citation nor quotation marks
- 
0 Missing Quotations 0%
 Matches that are still very similar to source material
- 
0 Missing Citation 0%
 Matches that have quotation marks, but no in-text citation
- 
0 Cited and Quoted 0%
 Matches with in-text citation present, but no quotation marks

Top Sources

- 0%  Internet sources
- 0%  Publications
- 0%  Submitted works (Student Papers)

From Reactive to Autonomous:

How AI Can Transform Metro Crowd Management

A technical overview of an AI-based dispatch and timetabling system for Riyadh Metro, built on Automatic Fare Collection (AFC) data, CCTV computer vision, and machine learning.

The Problem: Metro Crowd Management Is Still a Manual, Reactive Process

Modern metro systems are seeing rapid ridership growth, and Riyadh Metro is no exception. Yet at most stations, crowd management still works the same way it did a decade ago: an Operations Control Centre (OCC) controller watches CCTV feeds, notices a platform getting crowded, and manually decides to add an extra train — often after the crowding has already become a problem.

This reactive approach has three structural weaknesses:

- **It depends on human attention.** A controller monitoring dozens of camera feeds at once cannot catch every early warning sign, especially during multi-station surges.
- **Timetables are static.** They're written in advance based on historical averages and don't adapt to real-time or even weekly shifts in demand.
- **Data sources are siloed.** Automatic Fare Collection (AFC) gate data, CCTV footage, and the timetable/incident system are never combined into a single, unified signal — so no part of the system has the full picture.

This project set out to replace that reactive workflow with an **autonomous, predictive system**: one that fuses multiple real-time data sources, forecasts crowding before it happens, and automatically generates dispatch decisions and updated timetables — with no manual data entry or controller intervention required for routine operations.

System Architecture: Three Data Sources, One Signal

The system is built around a six-stage pipeline that fuses three complementary data streams into a single crowd-intelligence signal, which then feeds two cooperating AI models.

1. Automatic Fare Collection (AFC) gates supply exact passenger entry/exit counts per station, per hour. From these, four derived features are computed: entries, exits, net flow, and a running cumulative occupancy rate. AFC is precise for counting gate transactions, but it can't measure how many passengers are physically on the platform at a given moment.

2. CCTV + YOLOv8 computer vision fills that gap. Cameras covering three platform zones (front, middle, rear) capture a frame every 30 seconds, and a YOLOv8 person-detection model converts each frame into a headcount and per-zone density score. Unlike AFC, CCTV measures real-time platform occupancy directly — and it keeps working even if a fare gate malfunctions.

3. The timetable and incident system supplies operational context: scheduled trains, delay minutes, and incident severity — which lets the models tell the difference between ordinary demand fluctuations and crowding caused by a service disruption.

These three streams are combined using a weighted **data fusion formula**:

$$\text{fused_occupancy} = 0.60 \times \text{AFC_occupancy} + 0.40 \times \text{CCTV_headcount}$$

AFC is weighted more heavily because it's an exact transaction count, while CCTV acts as a real-time corrective signal for what AFC alone can't see. This fused occupancy value becomes the primary input feature for both AI models described below.

The Dataset

The models were trained and validated on a dataset covering **12 stations across 16 weeks** of Line 3 operations, sampled hourly across 19 operating hours a day. That works out to **25,536 records**, each describing one station at one hour on one day, across 34 combined AFC, CCTV, timetable, and derived features.

The dataset reflects a real operational challenge: **class imbalance**. Most station-hours require no additional trains, but the highest-severity “Critical” class (requiring three extra trains) is still well represented — over 7,000 records. Handling this imbalance correctly, without letting the model under-predict the rare-but-critical cases, was a key design constraint for the modeling approach below.

Two Cooperating AI Models

Rather than one monolithic model, the system uses two purpose-built models that share the same fused input signal:

Model 1 — Dispatch Classifier (Random Forest)

A Random Forest Classifier (400 trees, balanced_subsample class weighting) predicts, for the current hour, how many extra trains are needed: 0, +1, +2, or +3. The class weighting directly addresses the imbalance problem — it's specifically tuned so the model doesn't miss rare Critical-demand events, which are the most operationally costly to get wrong.

Model 2 — Timetable Forecaster (Gradient Boosting)

A Gradient Boosting Regressor (300 trees, max depth 4) forecasts occupancy rate up to **four weeks ahead**, retrained weekly to stay current. Its output feeds a fixed set of autonomous decision rules that convert a predicted occupancy percentage directly into a dispatch action and headway adjustment — for example, above 92% predicted occupancy triggers “+3 trains, -6 min headway,” while below 30% triggers a train reduction to save operating cost.

Experimental Results

The results validate both the modeling approach and the value of fusing multiple data sources:

Metric	Result
Dispatch classifier accuracy	99.7%
Macro F1-score (all 4 classes)	0.995
Misclassifications (test set of 5,244)	14 total, all adjacent-class (e.g., +2 predicted as +1/+3)
Timetable forecaster error (MAE)	2.87% occupancy
Autonomous 4-week timetable	6,384 time slots computed → 3,501 insertions, 1,445 reductions, 1,822 Critical alerts
CCTV camera network uptime	87.9% fully operational, 8.1% partial, 4.1% offline (auto-fallback to AFC)
YOLOv8 average detection confidence	93.5%

Two results stand out. First, the classifier **never once missed a Critical-demand event** in testing — every error was a near-miss between adjacent classes, never a dangerous “no action” prediction when action was actually needed. Second, comparing AFC-only estimation against the combined AFC+CCTV configuration confirms that CCTV isn't redundant: it adds genuinely new information (direct occupancy measurement, per-zone density, and resilience to gate failure) that AFC alone structurally cannot provide.

Explainability: Why the Model Made Each Decision

An autonomous dispatch system is only useful if operators can trust it — which means every decision needs to be explainable, not just accurate. The project incorporates **SHAP (SHapley Additive exPlanations)** analysis to quantify exactly which input features drove each individual prediction.

Across all dispatch classes, SHAP analysis consistently identifies **fused occupancy rate** and **delay minutes** as the two strongest predictors — which lines up with operational intuition (crowding and service disruption are the two most obvious real-world drivers of dispatch need) and gives human reviewers a clear, auditable reason behind every AI-generated decision.

From Model to Tool: The Interactive Dashboard

The trained models don't stop at a notebook — they're packaged into a browser-based interactive dashboard with three main views:

- **AI Timetable view** — all 6,384 computed time slots, filterable by station, day, hour, or alert severity, with color-coded status.
- **Station Deep-Dive view** — the full four-week plan, occupancy trend, and peak-hour profile for any single station.

- **Analytics & Trends view** — action distribution, Critical-alert frequency, and weekday-vs-weekend demand comparison across the full 16-week history.

A later addition to the dashboard mirrors a familiar consumer experience: a **“Check Status by Date”** search, styled after flight and train booking sites. Users pick any date within the model's four-week forecast horizon (bounded automatically so out-of-range dates are rejected gracefully), and the tool instantly surfaces that date's dispatch status — normal, high, or critical — along with how many trains were added or removed. On page load, it automatically shows today's status first, so the most relevant information is visible immediately with zero clicks.

Limitations

The results are strong, but four limitations should be acknowledged honestly before this moves toward real-world deployment:

1. **Synthetic dataset.** The training data mirrors realistic operational patterns but isn't drawn from genuine OCC historical exports.
2. **CCTV pipeline not yet live.** The vision pipeline is architecturally complete but has been validated against simulated camera data, not a live IP camera feed.
3. **No ground-truth validation.** AFC-derived crowd estimates are a proxy; they haven't yet been checked against independent manual headcounts.
4. **Four-week forecasting ceiling.** Accuracy is expected to degrade past the four-week horizon the model was designed and tested for.

Future Work

Each limitation maps directly to a proposed next step:

- **Real data integration** — retrain and validate on genuine OCC operational exports.
- **Live API connection** — connect the CCTV pipeline to a real IP camera feed for true real-time inference.
- **LSTM-based forecasting** — extend the timetable forecaster beyond four weeks using a sequence model better suited to long-range seasonal patterns.
- **OCC mobile app** — push Critical alerts directly to controllers' phones with a one-tap approval step, keeping a human checkpoint during the transition to full autonomy.
- **Multi-line scaling** — extend the approach from a single line to network-wide, cross-line capacity balancing.

Conclusion

This project demonstrates that fusing AFC data, CCTV computer vision, and timetable context — processed through two purpose-built, cooperating machine learning models — can turn metro crowd management from a manual, reactive process into a proactive, explainable, and largely autonomous one. With 99.7% dispatch accuracy, a fully autonomous four-week timetable, and SHAP-backed transparency behind every decision, the results provide a credible technical foundation for the real-data validation and live deployment work outlined above.

This article summarizes the methodology, results, and conclusions of an AI/ML graduate research project on autonomous crowd management for Riyadh Metro, covering system architecture, dataset design, model training, experimental evaluation, and an interactive decision-support dashboard.