NEXUS

D6.1 AI DEMONSTRATORS

PROJECT ACRONYM	NEXUS
PROJECT START DATE	01/10/2024
PROJECT DURATION (MONTHS)	24
GRANT	10177985
CALL	HORIZON-JU-ER-2023-01
TOPIC	HORIZON-ER-JU-2023-EXPLR-02
CONSORTIUM COORDINATOR	STAM
TITLE OF THE DELIVERABLE	D6.1 – AI demonstrators
WORK PACKAGE	WP6 — AI and Data Science Implementation in Metro Operation
TYPE OF DELIVERABLE	R — Document, report
DISSEMINATION LEVEL	PU - Public
STATUS – VERSION, DATE	Final, 09/12/2025
SUBMISSION DATE	10/12/2025

Co-funded by the European Union

Co-funded by the European Union. Views and opinion expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Europe's Rail Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them. The project 'NEXUS' is supported by the Europe's Rail Joint Undertaking and its members. UK participants in this project are funded by the UKRI.

NEXUS

AUTHORS/CONTRIBUTORS

Name	Organisation	Contribution
Michael Schmeja, Alexander Stocker, Alexander Zincke	ViF	Chapter 1, 5, 6 and Outlook
Marin Marinov, Hing Yan Tong, Lydia Egbo, Patrick Bannon	AU	Chapter 3 and 4
Luca Oneto, Simone Minisi	UNIGE	Chapter 2

QUALITY CONTROL

	Name	Organisation	Date
Primary reviewer	Takeru Shibayama, Bernhard Rüger	TUW	24/09/2025
Secondary reviewer	Giuseppe Rizzi	UITP	01/10/2025
Quality check	Pietro De Vito	STAM	07/10/2025

VERSION HISTORY

Version	Date	Author	Summary of changes
01	05/09/25	Michael Schmeja	First version for internal Review
02	07/10/25	Michael Schmeja	Second version after internal Review
03	08/12/25	Michael Schmeja	Third version after external Review

APPROVED FOR SUBMISSION BY

Name	Organisation	Approval date
Pietro De Vito	STAM	09/12/2025

NEXUS

LEGAL DISCLAIMER

This project has been funded by the European Union's Horizon Europe research and innovation programme under grant agreement No. 101177985. UK participants in this project are funded by the UKRI. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them. The information in this document is provided "as is", and no guarantee or warranty is given that it is fit for any specific purpose. The Nexus project Consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.

NEXUS

Table of contents

List of figures

List of tables

Executive summary

This deliverable, D6.1 “AI Demonstrators”, builds on the earlier D6.3 “AI in Future Metro Operations”, which outlined potential AI and data science use cases for metro systems. D6.1 moves from concept to practicable application presenting four demonstrators: crowding prediction (UNIGE), demand forecasting, timetable creation (ASTON), and uncleanliness detection in vehicle interiors (VIF). Each demonstrator includes data collection methods, applied algorithms, key results, and lessons learned, along with an evaluation of its practical relevance. The deliverable also provides a brief overview of research into technology acceptance and trust, highlighting their importance for successful AI adoption in metro operations.

KEYWORDS

Machine Learning (ML), Artificial Intelligence (AI), Data Science, Metro Operations, Technology Acceptance

Social Media link:

@nexus-project

For further information please visit nexus-project.eu

LIST OF ABBREVIATIONS AND ACRONYMS

Acronym	Meaning
ASAP	As Soon As Possible
AD	Anomaly Detection
AI	Artificial Intelligence
ALTAI	Assessment List for Trustworthy Artificial Intelligence
AMT	Azienda Mobilità e Trasporti
API	Application Programming Interfaces
ARPAL	regional environmental protection agency
B2B	Business-to-business
B2C	Business-to-Consumer
CCTV	Closed-Curcuit Television
EC	European Commission
EDA	exploratory data analysis
FACTS4WORKERS	EU-project FACTories which are attractive to WORKERS
GA	Grant Agreement
HTTP	Hypertext Transfer Protocol
IS	Information System
JSON	JavaScript Object Notation
KoM	Kick-off Meeting
KPI	Key Performance Indicator
MAE	Mean Absolute Error

Acronym	Meaning
ML	Machine Learning
PEOU	Perceived Ease of Use
PU	Perceived Usefulness
ResNet	Residual Network
TACO	Trash Annotations in Context
TAM	Technology Acceptance Model
TRA	Theory of Reasoned Action
UTAUT	Unified Theory of Acceptance and Use
WP	Work Package
XAI	Explainable AI
YOLO	You only look once

1 INTRODUCTION AND OBJECTIVES

This deliverable, *D6.1 "AI Demonstrators*", builds upon the work presented in *D6.3, "AI in Future Metro Operations*", which offered a desktop research overview of potential applications of artificial intelligence and data science in future metro systems. However, that previous report has already identified promising AI/Data Science use cases and provided an initial outlook on how these technologies could support and enhance metro operations.

In the present deliverable *D6.1"AI in Future Metro Operations*", we move beyond the conceptual stage and explore a set of AI and data science demonstrators in greater depth. For each demonstrator, we describe the approach to data collection, detail the procedures, algorithms, and scripts applied, present the results obtained, and discuss the main challenges encountered along with the lessons learned. Finally, we provide an evaluation of the demonstrator's effectiveness and relevance within the context of future metro operations.

D6.1 features four AI and data science demonstrators addressing the following topics: prediction of crowding based on exogenous data sources (UNIGE), demand forecasting during network expansion (AU), timetable creation using GTFS feeds (AU), and detection of uncleanliness in vehicle interiors (VIF). These demonstrators apply a variety of AI and data science techniques, including image classification and other advanced analytical methods, to explore practical and impactful use cases for enhancing future metro operations.

In addition, the D6.1 further explores research related to technology acceptance and trust in the context of AI and data science adoption, providing a snapshot of current insights into how these factors influence the successful implementation of such technologies in metro operations.

As regards the practical application of our findings, we are engaged in intensive discussions with our project partners (AMT and Sofia Metro) and with the metro operators on the Advisory Board. Over the course of the second year, it will become clear whether and under what conditions a real-world deployment of the respective tools is technically feasible and economically viable.

2 PREDICTION OF CROWDING BASED ON EXOGENOUS DATA SOURCES

Genoa's metro system, operated by Azienda Mobilità e Trasporti (AMT), is a vital part of the city's public transportation network. Spanning a significant 7.1 kilometres along a striking northwest-to-southeast route, the metro is centred around a single, yet crucial, line. Since its debut in 1990, the system has expanded through multiple phases to better serve both urban and suburban areas in response to increasing transport needs. By connecting the Brin station in the Rivarolo district to Piazza Principe and Brignole, Genoa's two main railway hubs, the metro serves not only daily commuters but also enhances regional connectivity. In this section the developed ML models used to predict metro station crowding will be described.

2.1 DATA COLLECTION

The predictive models for this demonstrator were developed using a comprehensive and heterogeneous dataset covering the entirety of 2024. The data was collected to build a holistic view of urban mobility patterns in Genoa, Italy, with the primary goal of predicting passenger traffic at key metro stations.

The data sources are categorized as follows:

- **Metro Passenger Data**: This is the core dataset and the primary prediction target. It consists of historical passenger counts, aggregated into 15-minute intervals, for both incoming (IN) and outgoing (OUT) traffic. Data was collected for all 8 metro stations: Brignole, Brin, Darsena, De Ferrari, Dinegro, Principe, Sarzano, and S. Giorgio.
- **Bus System Data**: Historical data on bus stop crowdedness and bus line traffic was integrated. This data serves as a crucial proxy for overall city mobility and a leading indicator for passenger load at metro stations, which often serve as interchanges.
- **Weather Data**: Historical weather information, including temperature, precipitation, and general conditions, was sourced from the regional environmental protection agency (ARPAL). Weather patterns, particularly rain, have a significant and demonstrable impact on public transport usage.
- **City Events Data**: A log of significant public events (e.g., concerts, sporting events, festivals) was compiled. This data is essential for explaining and predicting anomalous spikes in passenger traffic that are not tied to regular daily or weekly patterns.

2.1.1 EXPLORATORY DATA ANALYSIS

Before model development, a thorough exploratory data analysis (EDA) was conducted to understand the underlying patterns and relationships in the data. This was a critical step to guide feature engineering (Zheng, 2018) and modelling strategies. Key findings included:

- **Temporal Patterns**: Time-series analysis of passenger data revealed strong daily, weekly, and seasonal patterns as shown in Figure 1. Weekday traffic showed distinct morning and evening

peaks corresponding to commuter travel, while weekend traffic was generally lower and more spread throughout the day as shown in Figure 2.

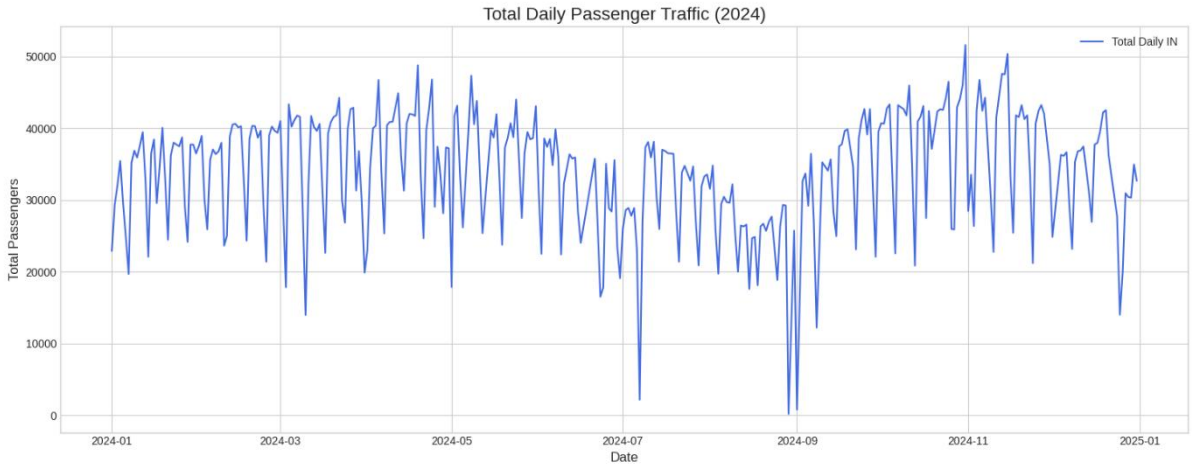

Figure 1 Daily passenger traffic in 2024

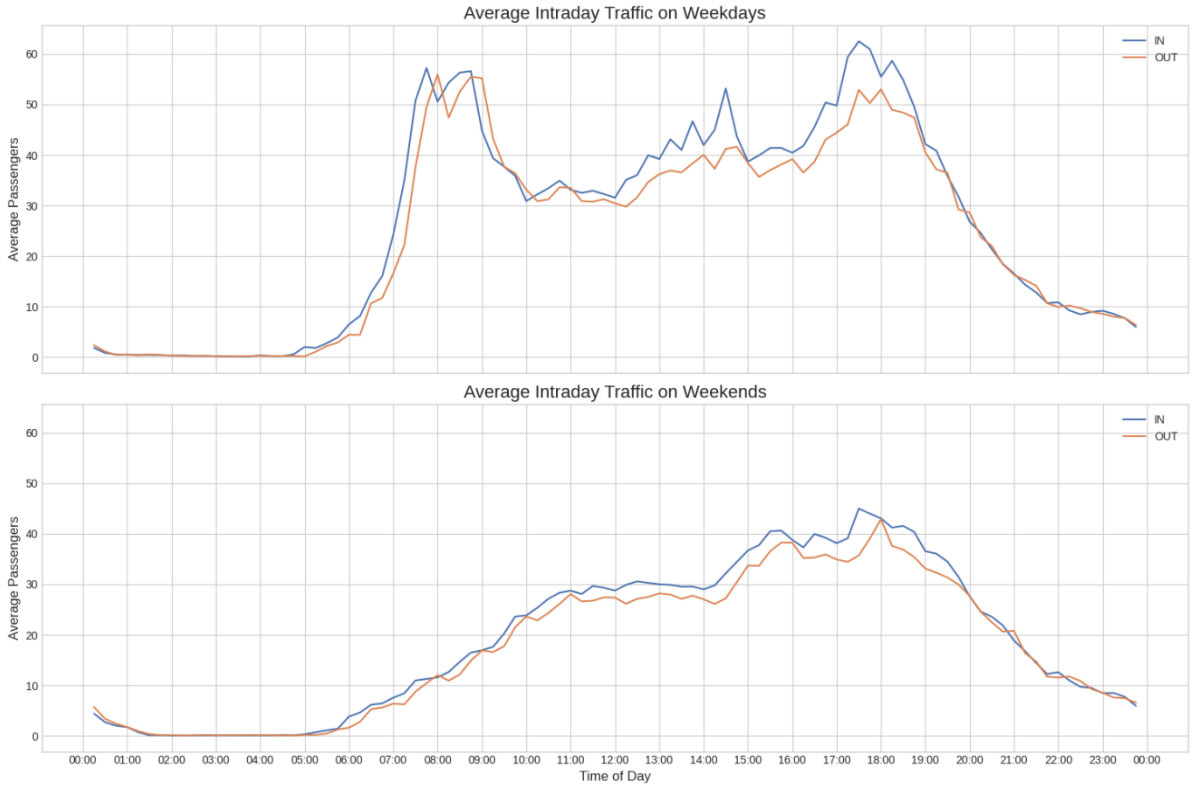

Figure 2 Intraday traffic patterns weekdays vs weekends

- **Weather Impact**: Correlation analysis confirmed that weather patterns have a significant and demonstrable impact on public transport usage. For instance, a clear correlation was found between rainfall and increased passenger flow, as people opt for public transport over walking or cycling as displayed in Figure 3.

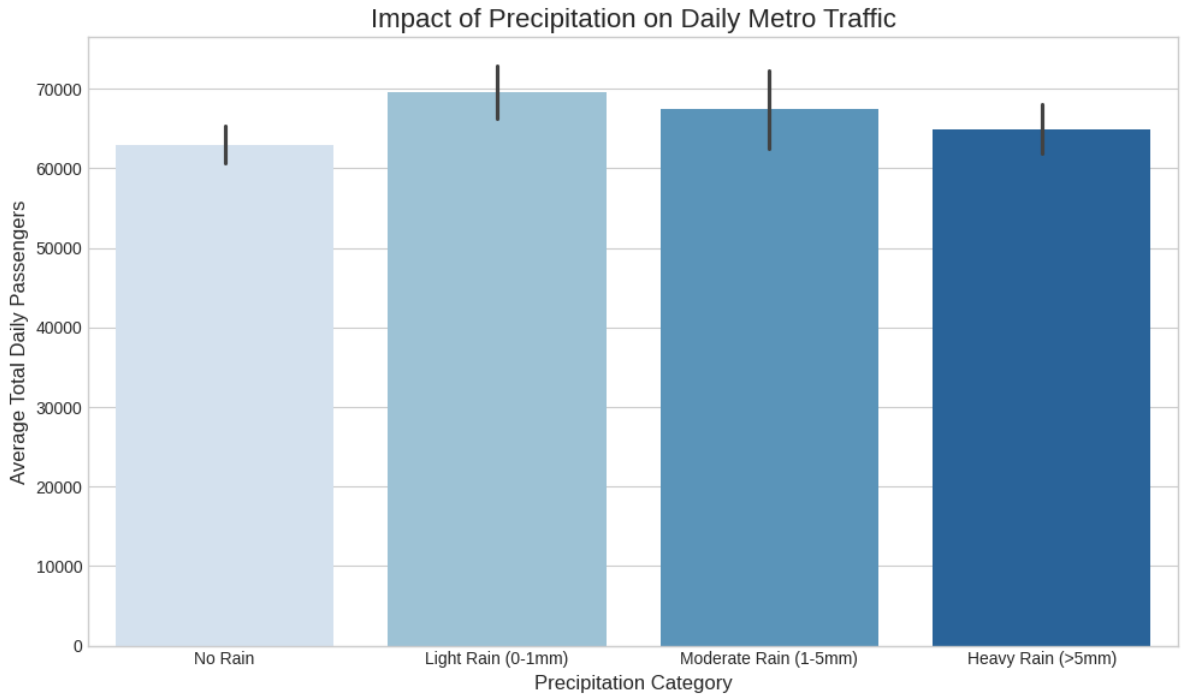

Figure 3 Impact of precipitation on daily passenger traffic

- **Inter-Station Dynamics**: Comparing passenger volumes across stations highlighted their different roles. Brignole and Brin are the stations with higher passenger volumes, attracting passengers from the two main valleys in Genoa: Val Polcevera and Val Bisagno (Figure 4).

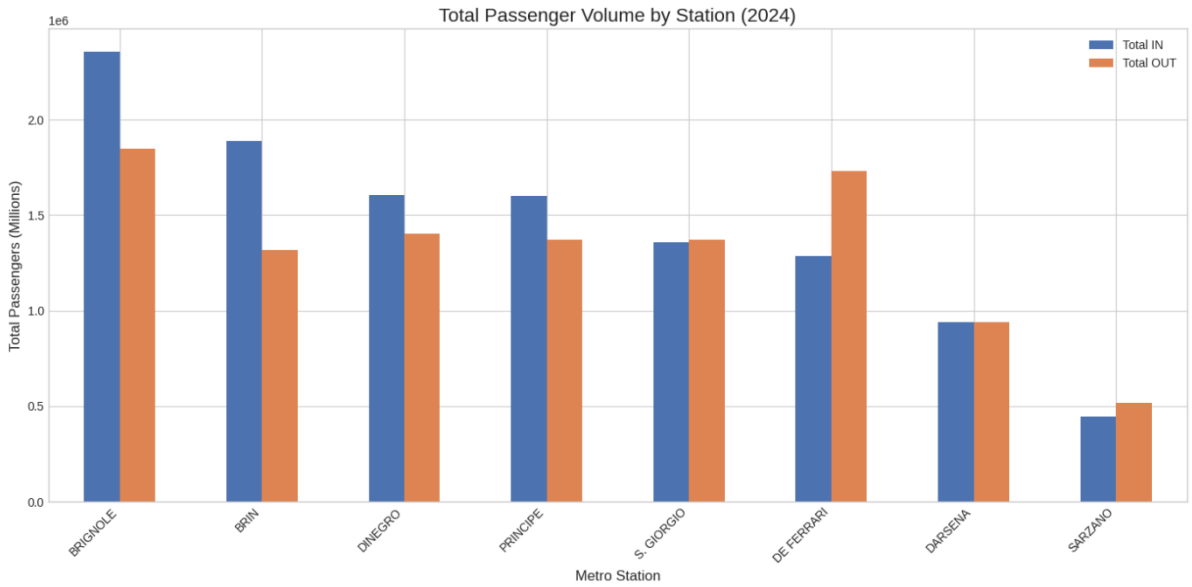

Figure 4 Passenger volumes by station

2.1.2 DATA PARTITIONING

For model training and evaluation, the dataset was partitioned chronologically to respect the time-series nature of the data:

- **Training Set**: Data from January 1, 2024, to September 30, 2024.
- **Test Set**: Data from October 1, 2024, to December 31, 2024.

2.1.3 DATA-GOVERNANCE CHECKLIST

This demonstrator adheres to strict data governance protocols to ensure the integrity, security, and privacy of the mobility data used.

- **Data Origin & Ownership:**
  - *Passenger Counts:* Proprietary data provided by the local public transport operator.
  - *Weather Data:* Sourced from public meteorological API services.
  - *Calendar/Events:* Publicly available holiday and event schedules.
- **Privacy & Anonymity:**
  - Data is strictly **aggregated**.
  - No PII (Personally Identifiable Information) is collected, processed, or stored.
  - The system complies with GDPR by design, as individual passenger trajectories are not tracked.
- **Licensing:**

- The underlying passenger count datasets are restricted and require specific data-sharing agreements for external use.

- **Security Controls:**
  - The model runs within an isolated Docker container, limiting access to the host file system.

2.2 DESCRIPTION OF PROCEDURE, ALGORITHMS, AND SCRIPTS

The demonstrator is designed as a Python-based prediction engine, exposed via a RESTful API implemented using the Flask framework (https://flask.palletsprojects.com/en/stable/). The system is designed for containerized deployment using Docker (www.docker.com).

2.2.1 SYSTEM ARCHITECTURE & MODELLING STRATEGY

The core of the system is the NexusModel class, which encapsulates all logic for data handling, feature retrieval, and prediction. The prediction strategy is based on a time-shift approach, where different models are trained for different prediction horizons. This allows the system to leverage the most relevant features for each timeframe.

The models are divided into two categories:

- **Short-Term Models (Shifts 0-4)**: These models provide high-fidelity predictions for near-future intervals: 0, 15, 30, 45, and 60 minutes from the current time. They are trained on a rich feature set including recent bus traffic and fine-grained temporal data.
- **Long-Term Model (Shift >4)**: A single, more generalized model provides predictions for any interval greater than 60 minutes. This model relies on more general calendar and seasonal features, as immediate mobility data becomes less relevant.

A separate model is trained for each combination of station, direction (IN/OUT), and short-term shift. This granular approach results in a total of 80 short-term models (8 stations * 2 directions * 5 shifts) plus 16 long-term models (8 stations * 2 directions), ensuring that each model is highly specialized for its specific context.

2.2.2 ALGORITHM

The development process involved a rigorous evaluation of several machine learning algorithms. We tested a range of models, including linear regressions (Freedman, 2005), (Shalev-Shwartz & Ben-David, 2014), Gradient Boosting Machines (Hastie, Tibshirani & Friedman, 2009), and ensemble methods (Opitz & Maclin, 1999). The Random Forest Regressor[1] (Ho, 1995) was ultimately selected

[1] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

as it consistently performed the best in capturing the complex, non-linear relationships within the data (e.g., the combined effect of a holiday and bad weather).

This ensemble algorithm was chosen for several key reasons:

- **Robustness**: It is less prone to overfitting compared to single decision trees.
- **Non-linearity**: It can effectively capture complex, non-linear relationships between features (e.g., the combined effect of a holiday and bad weather).
- **Confidence Intervals**: The ensemble nature of Random Forests allows for a robust estimation of prediction confidence. By analysing the variance in predictions across the individual trees in the forest, we can construct a percentile-based confidence interval. The _predict_with_confidence method implements this by calculating the 2.5th and 97.5th percentiles of the predictions from all trees to form a 95% confidence interval. This is achieved by leveraging the ensemble nature of the Random Forest algorithm, following a method consistent with the principles of Quantile Regression Forests (Meinshausen, 2006). Instead of only using the final averaged prediction, we capture the individual prediction from every decision tree within the forest for a given data point. This collection of predictions forms a non-parametric empirical distribution of the forecast. From this distribution, the method calculates the 2.5th and 97.5th percentiles to construct a robust 95% prediction interval. This approach is powerful because it makes no assumptions about the underlying error distribution (e.g., normality) and directly uses the variance in predictions across the trees as a measure of model certainty.

To ensure optimal performance, **grid search cross-validation**[2] (Oneto & Anguita, 2019) was used for systematic hyperparameter optimization for each model. The primary performance metric guiding this tuning process was the Mean Absolute Error (MAE) (Hodson, 2022), as it provides a clear, interpretable measure of the average error in terms of passenger counts.

2.2.3 IMPLEMENTATION AND SCRIPTS

nexus_model.py - The Core Prediction Engine

NexusModel Class: This class orchestrates the entire prediction process.

- **Initialization (__init__)**: Upon instantiation, it loads all 96 pre-trained models (.joblib files) and the pre-computed test feature sets (.csv files) into memory. This ensures low-latency predictions at runtime.
- **Prediction Flow (predict method)**:
  - The method accepts a current datetime and a future interval in minutes.
  - It rounds the current time to the nearest 15-minute block (_round_to_15_minutes).
  - It calculates the shift (e.g., a 40-minute interval corresponds to shift 2, as 40 // 15 = 2).

[2] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

- **Feature Retrieval**:
  - For **short-term** predictions (shifts 0-4), the system performs a lookup in the pre-loaded test dataframes (self.datasets_in, self.datasets_out). It finds the feature vector corresponding to the calculated target time, station, and platform. This is a demonstrator-specific implementation to simulate a real-time feature engineering pipeline.
  - For **long-term** predictions (shift >4) or if a short-term lookup fails, the system falls back to the _build_features_manual method. This function generates features based on calendar data (minute-of-day interval, day of week, hour, season) and one-hot encodes the station and platform.
- **Prediction & Confidence**: The appropriate model is selected, and the _predict_with_confidence method is called to generate both a point prediction and a 95% confidence interval.
- **Output Formatting**: The results for all stations and platforms are compiled into a structured JSON object.

app.py - The Flask API

- **Deployment Wrapper:** This script wraps the NexusModel in a Flask web service, making it accessible over HTTP.
- **Model Loading:** The entire NexusModel instance, including all loaded models and datasets, is loaded from a single compressed pickle[3] file (nexus_model.pbz2). This simplifies deployment by bundling all necessary assets into one file.
- **Endpoints:**
  - /predict (GET): The main prediction endpoint. It accepts datetime and interval as query parameters.
  - /status (GET): A simple health-check endpoint that returns an "OK" status.
- **Input Validation (@validate_input decorator):** A robust validation layer ensures that all incoming requests have the required parameters, that the datetime format is correct (ISO 8601), and that the values fall within a valid operational range. This improves API stability and provides clear error messages to clients.

The training phase of the project is not delivered as part of the final demonstrator. This is a deliberate choice, as the training process is computationally intensive and requires human oversight for validation and potential retraining. Instead, the focus was to deliver a self-contained forecasting tool. The entire application has been delivered as a **Docker container**, which encapsulates the model, its dependencies, and the API. This ensures that the system is platform-independent and can be run by users who are not experts in machine learning, making it highly portable and easy to deploy.

[3] https://docs.python.org/3/library/pickle.html

2.2.4 PREDICTION PROCESS

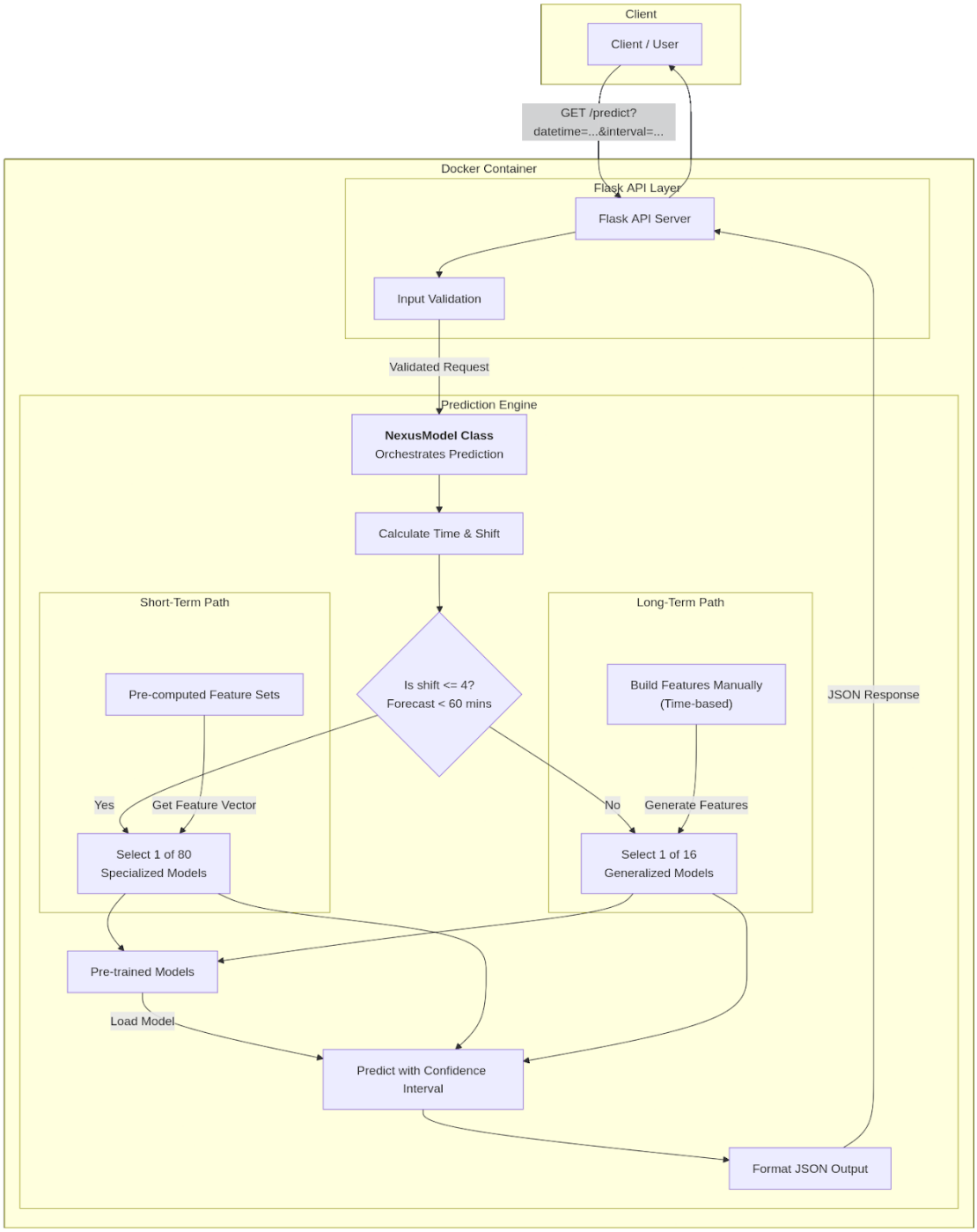

Figure 5 The prediction process

The diagram in Figure 5 illustrates the architecture of the predictive demonstrator, which is designed as a containerized, API-driven service. The flow begins with a client request and ends with a structured JSON response.

1. **Client Request:** An external user or service initiates a prediction by sending an HTTP GET request to the /predict endpoint. The request must include the datetime and interval parameters.
2. **Docker Container & API Layer:**
    a. The entire system is packaged within a **Docker Container**, ensuring it is portable and can run in any environment without dependency issues.
    b. The entry point is a **Flask API Server** (app.py). Its primary job is to handle incoming HTTP requests.
    c. Before any processing occurs, a **Validation Decorator** intercepts the request to ensure all parameters are present, correctly formatted (ISO 8601 for datetime), and within valid operational ranges. This makes the API robust and provides clear error feedback.
3. **Prediction Engine (NexusModel Class):**
    a. Once validated, the request is passed to the core **NexusModel** class (nexus_model.py), which orchestrates the entire prediction process.
    b. **Step 1: Time Calculation:** The engine first calculates the "shift" – the number of 15-minute intervals into the future the prediction is for.
    c. **Step 2: Critical Decision:** This is the core of the system's intelligence. The engine checks if the forecast is for the near future (shift <= 4, i.e., within 60 minutes) or the long term prediction (shift > 4).
4. **The Dual Prediction Paths:**
    a. Short-Term Path (Forecast < 60 mins):
        i. The system selects one of **80 highly specialized models**, each trained for a specific station, direction (IN/OUT), and 15-minute shifts.
        ii. To make a prediction, it performs a fast lookup in the **Pre-computed Feature Sets** (.csv files) that were loaded into memory at startup. This simulates a real-time feature pipeline and provides the rich, dynamic data needed for high-accuracy short-term forecasts.
    b. Long-Term Path (Forecast > 60 mins):
        i. The system selects one of **16 generalized models;** each trained for a specific station and direction.
        ii. Instead of looking up dynamic data (which is no longer predictive), it **builds features manually** using stable, cyclical information such as the time of day, day of the week, and season.
5. **Prediction and Formatting:**
    a. **Step 3: Prediction with Confidence:** Regardless of the path taken, the selected model is retrieved from the **Model Store** (the collection of pre-trained .joblib files). The engine then generates not only a point prediction but also a 95% confidence interval by analyzing the variance across the trees in the Random Forest model.
    b. **Step 4: JSON Formatting:** The results for all stations and platforms are compiled into a single, well-structured JSON object.

6. **Response:** The JSON object is returned to the Flask layer, which sends it back to the client as the HTTP response.

Sample API Response Snippet:


```
{
  "datetime": "2024-11-05T18:00:00",
  "interval": 30,
  "prediction": {
    "de_ferrari": {
      "platform_1": {
        "in": 152.75,
        "in_confidence": {
          "lower": 135.0,
          "upper": 170.5
        },
        "out": 120.1,
        "out_confidence": {
          "lower": 105.0,
          "upper": 135.0
        }
      },
      "platform_2": { ... }
    },
    "brignole": { ... }
  }
}
```


This structured output is machine-readable, providing both a point estimate and a measure of its uncertainty, which is critical for operational decision-making.

2.3 RESULTS

The models were rigorously tested against the hold-out test set (data from October 1, 2024, to December 31, 2024). The experiments were designed not only to measure performance but also to validate the system's core architectural choice: using specialized models for short-term predictions and a general model for long-term forecasts.

2.3.1 METHODOLOGY AND BASELINE MODEL

The system employs a two-tiered modelling strategy:

1. **Short-Term Models (Shifts 0-4):** High-fidelity models trained on a rich feature set, including dynamic data such as recent bus traffic, weather, and city events. These are used for predictions up to 60 minutes.
2. **Long-Term Model (Shift >4):** A generalized model trained only on stable, cyclical features (e.g., time of day, day of week, season).

For these experiments, the **Long-Term Model serves as the official baseline**. It is the model used by the system for all predictions beyond 60 minutes. The performance of the short-term models is measured by their improvement over this baseline.

The baseline performance, as measured by Mean Absolute Error (MAE), is:

- **Baseline MAE (IN): 7.70**
- **Baseline MAE (OUT): 7.63**

2.3.2 QUANTITATIVE RESULTS

Table 1 compares the MAE of the specialized short-term models against the long-term baseline. The results for Shifts 5 and 6 are included to demonstrate why the system switches to the baseline model after 60 minutes.

Table 1 Prediction horizons

PREDICTION HORIZON (MINUTES)	MODEL TYPE	MAE (IN)	% IMPROVEMENT VS. BASELINE (IN)
0	Short-Term	7.18	6.68
15	Short-Term	7.16	6.91
30	Short-Term	7.28	5.38
45	Short-Term	7.49	2.72
60	Short-Term	7.58	1.45
> 60	Long-Term	7.70	0.00
75 (For analysis only)	Short-Term	7.84	-1.84
90 (For analysis only)	Short-Term	7.99	-3.78

PREDICTION HORIZON (MINUTES)	MODEL TYPE	MAE (OUT)	% IMPROVEMENT VS. BASELINE (OUT)
0	Short-Term	7.23	5.18
15	Short-Term	7.20	5.64
30	Short-Term	7.33	3.85
45	Short-Term	7.42	2.67
60	Short-Term	7.58	0.66
> 60	Long-Term	7.63	0.00
75 (For analysis only)	Short-Term	7.73	-1.27
90 (For analysis only)	Short-Term	7.86	-3.02

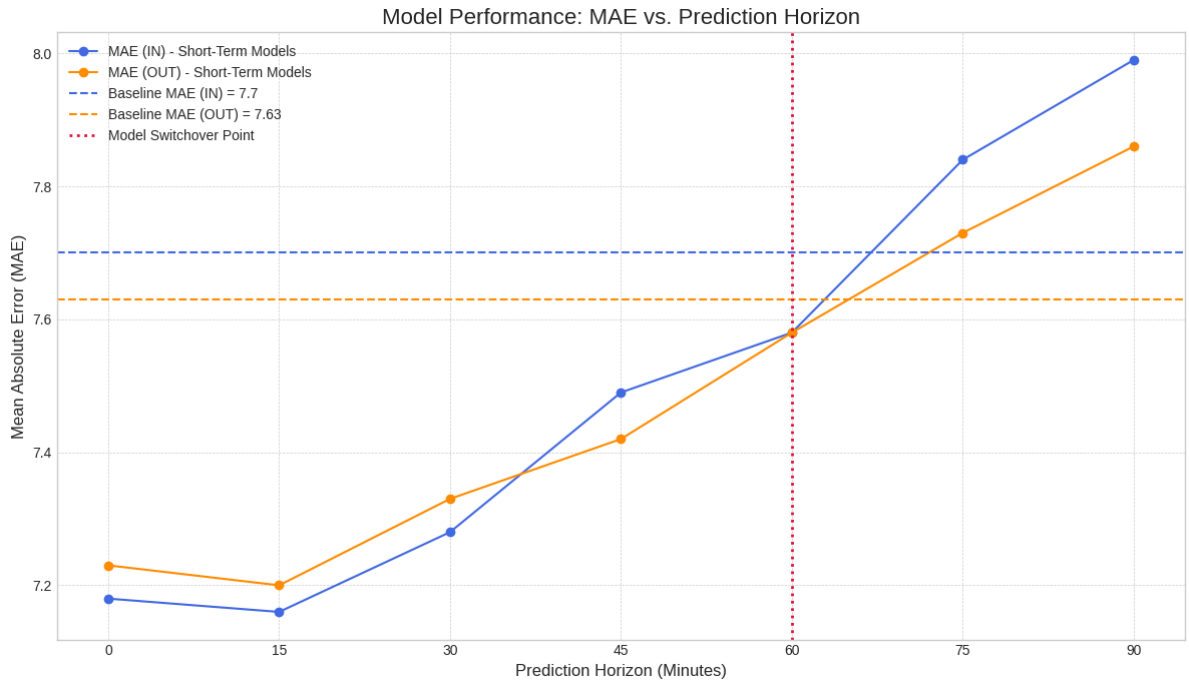

Figure 6 MAE influences the prediction horizon

Figure 7 Performance improvements of short-term models compared with the baseline

Note: Negative improvement indicates the model performed worse than the baseline. The system correctly uses the Long-Term Model for these horizons.

2.4 CHALLENGES & LESSONS LEARNED

Several challenges were encountered and addressed during the development of this demonstrator, leading to valuable insights.

- **Challenge: Simulating Real-Time Feature Engineering:** A full-scale production system would require a complex, low-latency data pipeline to compute features in real-time. Building this was outside the scope of the demonstrator.
    - **Lesson Learned:** The approach of pre-calculating feature vectors for a specific test period and loading them into memory (_get_features_from_dataset) proved to be an effective shortcut. It allows the demonstrator to showcase the potential of the high-fidelity short-term models without the overhead of a real-time pipeline. This highlights a key area for future development towards a production system.
- **Challenge: Differentiating Short-Term vs. Long-Term Dynamics:**
  Early tests showed that a single model struggled to perform well across all time horizons.
    - **Lesson Learned:** The dual-model strategy (specialized short-term vs. generalized long-term) was a breakthrough. It acknowledges that the predictive signals for "in 15

minutes" are fundamentally different from those for "in 3 hours." This separation of concerns significantly improved overall model performance.

Lesson Learned: The Value of Confidence Intervals: Providing a single number as a prediction can be misleading. By calculating and exposing confidence intervals, the demonstrator provides a more precise assessment of the future. This allows end-users to understand the potential range of outcomes and make more informed, risk-aware decisions.

2.5 EVALUATION

The following evaluation was conducted to rigorously assess the performance and effectiveness of the developed demonstrator. The primary objectives were to quantify the accuracy of the predictive models against a hold-out test set, validate the architectural decision of using a dual-model strategy, and assess the overall operational utility of the system. The evaluation is structured into a quantitative analysis of the results and a qualitative assessment of the demonstrator's design and features.

2.5.1 ANALYSIS OF RESULTS

- **Short-Term Models Provide Significant Uplift (0-60 mins):** The results prove the value of the specialized models. For near-future predictions, leveraging dynamic data sources provides a substantial accuracy boost, peaking at a **6.68% error reduction** for 15-minute forecasts. This confirms that real-time factors are powerful predictors of immediate passenger behaviour.
- **Validation of the 60-Minute Switchover Point:** The most critical insight comes from the performance degradation after 60 minutes as shown in Figure 6 and Figure 7. The results for Shifts 5 and 6 show that continuing to use the short-term modelling strategy becomes detrimental, with performance dropping **up to 3.78% below the baseline**. This happens because the short-term models attempt to find patterns in dynamic data that is no longer relevant, effectively overfitting to noise. In contrast, the simpler long-term model correctly relies on more stable, cyclical signals. This data provides a clear, empirical justification for the architectural decision to switch to the long-term model for any forecast beyond 60 minutes.
- **Robust and Intelligent System Design:** The evaluation validates that the two-tiered system is not just a design choice but an optimal strategy. The demonstrator intelligently selects the best model for the given time horizon: the high-precision model when its features are relevant, and the stable baseline model when they are not.

2.5.2 QUALITATIVE EVALUATION

Beyond quantitative metrics, the demonstrator was evaluated on its operational utility and design.

1. **Intelligent Architecture:** The data-driven validation of the 60-minute switchover point highlights the system's sophisticated and effective design.
2. **Actionable Insights:** The system provides stable, low-latency predictions via a robust API. The inclusion of confidence intervals further enhances the output, allowing end-users to understand the model's certainty and make risk-informed decisions.
3. **Deployment-Ready:** The entire system is containerized, and its assets are bundled, making it easy to deploy and maintain.

In conclusion, the evaluation confirms the demonstrator's success. It not only achieves high accuracy in its primary operational window (0-60 minutes) but also demonstrates a robust, data-backed architecture for handling predictions across all time horizons.

2.6 REPRODUCIBILITY & ARTEFACTS

To ensure that the results presented in this demonstrator can be independently verified and extended by external teams, we have consolidated the necessary code, models, and environment specifications. The workflow relies heavily on containerization to minimize environment-specific errors.

Artefacts Available:

- **Source Code:** The full Python codebase, including nexus_model.py (logic), app.py (API interface), and the Docker configuration files, is available in a git repository shared with project partners.
- **Pre-trained Models:** The compressed artifact nexus_model.pbz2 contains the 96 pre-trained Random Forest models (.joblib) and the pre-computed feature sets required for the short-term prediction path.
- **Docker Container:** A Dockerfile is provided to build the exact runtime environment, including dependencies listed in requirements.txt (e.g., Flask, scikit-learn, pandas).

Steps to Reproduce:

1. **Environment Setup:** Ensure Docker is installed. Clone the repository and navigate to the root directory.
2. **Build:** Execute the build command (e.g., docker build -t nexus-predictor) to create the image. This process automatically installs dependencies and copies the nexus_model.pbz2 file into the container.
3. **Run:** Launch the container exposing port 5000.
4. **Verification:** Use the /status endpoint to verify the service is running, then issue a POST request to /predict using the test dataset dates (October 1, 2024 – December 31, 2024) to reproduce the MAE metrics cited in Section 2.3.

2.7 CROSS-DEMO KPI TABLE AND BASELINE COMPARISONS

To facilitate a consistent evaluation of the project's AI assets, we have summarized the demonstrator’s performance using standardized Key Performance Indicators (KPIs). These metrics measure the added value of the specialized AI components against a "Baseline"—a simplified or naïve reference model representing a standard or generalized approach.

For the Nexus Passenger Prediction Demonstrator, the Baseline is defined as the "Long-Term Model." This model uses only static, cyclical features (season, day of week, time of day) without access to real-

time feature vectors. It represents a standard view of passenger traffic, against which the "Short-Term" AI demonstrator is measured.

Table 2 CROSS-DEMO KPI table

KPI CATEGORY	METRIC	BASELINE METHOD (GENERALIZED MODEL)	DEMONSTRATOR METHOD (SPECIALIZED AI)	PERFORMANCE DELTA
Accuracy	Mean Absolute Error (MAE) - IN Direction	7.70 (Avg. error per 15-min interval)	7.18 (at 0-min horizon)	+6.68% Accuracy Improvement
Accuracy	Mean Absolute Error (MAE) - OUT Direction	7.63 (Avg. error per 15-min interval)	7.23 (at 0-min horizon)	+5.18% Accuracy Improvement
Reliability	Confidence Interval	N/A (Point prediction only in standard forecasting)	95% CI (Quantile Regression Forest)	Adds quantification of uncertainty
Responsiveness	Adaptation to Real-Time Dynamics	Static (Ignores weather/traffic shifts)	Dynamic (Updates every 15 mins)	High fidelity to immediate conditions

Interpretation of Results:

The comparison highlights that while the Baseline model provides a stable, "average" expectation of traffic suitable for long-term planning, the Demonstrator's specialized short-term models significantly reduce error by capturing non-linear dependencies between immediate bus traffic, weather, and passenger flow. The 6.68% reduction in error validates the computational cost of maintaining the specialized short-term feature pipeline.

3 TIMETABLE CREATION USING GTFS FEEDS

Artificial Intelligence (AI) is increasingly being applied in public transportation to enhance operational efficiency, automate data processing, and improve passenger service delivery (Saki & Soori, 2025). One area where AI shows promise is in timetable creation, a task traditionally performed using static rules and manual data manipulation (S. R. Khokale et al., 2025). The rise of standardized open data formats, such as the General Transit Feed Specification (GTFS), enables AI to be integrated more easily into the public transport planning pipeline.

GTFS is a widely adopted standard that provides a structured, machine-readable representation of transit systems. It includes detailed information on stop locations (stops.txt), route configurations (routes.txt), scheduled trips (trips.txt), and stop-level arrival and departure times (stop_times.txt). This dataset forms a comprehensive view of transit operations and provides the foundation for applying AI to timetable creation.

Timetable generation remains a critical component of public transport operations, yet the process in Figure 8 continues to rely heavily on manual design rules and static service planning tools. While research into automated timetable creation has made progress in recent years, much of the existing work focuses on large-scale networks in Asia and North America, often with proprietary datasets or simulation environments that limit transferability. Medium-sized European tram networks, such as the West Midlands Metro, present a different context where operators increasingly release open schedule data through the General Transit Feed Specification (GTFS), but systematic AI-based frameworks for timetable creation remain underexplored. This study contributes to closing this gap by developing a conceptual pipeline that demonstrates how clustering, supervised learning, and constraint-based optimization can be integrated to generate draft timetables directly from GTFS feeds. The framework is tailored to the structure of West Midlands Metro data, yet generalisable to other regional systems that publish open feeds. Beyond its methodological design, the use case highlights a pathway for transport agencies to leverage publicly available data and artificial intelligence in creating adaptive, efficient, and passenger-oriented timetables.

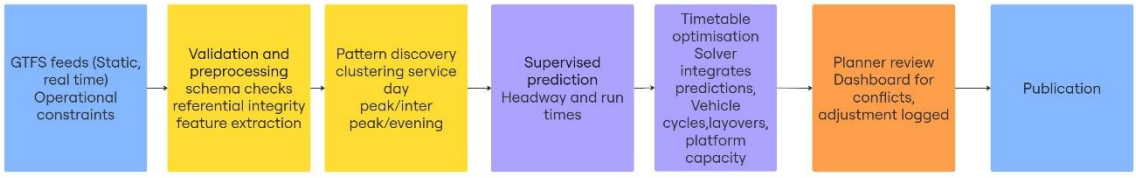

Figure 8 Flowchart of process used in research

3.1 DATA COLLECTION

3.1.1 STUDY AREA AND DATA

West Midlands Metro serves the West Midlands region with tram services that traverse a mix of urban corridors and city centre streets. The analysis uses the standard GTFS tables agency, routes, trips, stop_times, stops, calendar, calendar_dates, frequencies, shapes, and feed_info as defined in the GTFS Schedule. [1]. The regional dataset is distributed through Transport for West Midlands programmatic interfaces and is also indexed in third party archives for reproducibility checks [2].

GTFS schedule datasets encode routes, trips, stops, calendars, and stop times. These tables (Table 3) provide the canonical description of when vehicles should arrive and depart, on which route, in which direction, and on which service days. The GTFS reference enumerates required files such as routes, trips, stops, stop_times, calendar, and calendar_dates as seen in table 1 below.

Table 3 GTFS elements and role in the workflow

GTFS file	Key fields used	Role in workflow
agency.txt	agency_id, agency_name, agency_timezone	Operator metadata and feed context
routes.txt	route_id, route_short_name, route_long_name, route_type	Line identification and reporting labels
trips.txt	route_id, service_id, trip_id, direction_id, shape_id	Defines each scheduled trip and its direction, links routes to stop sequences
stop_times.txt	trip_id, arrival_time, departure_time, stop_id, stop_sequence, timepoint	Ordered times at stops for each trip, basis for headway and reference segment time calculations
stops.txt	stop_id, stop_name, stop_lat, stop_lon, parent_station	Stop locations and names, hub identification
calendar.txt	service_id, monday to sunday flags, start_date, end_date	Regular service patterns and validity windows
calendar_dates.txt	service_id, date, exception_type	Exceptions for special days and events
frequencies.txt	trip_id, start_time, end_time, headway_secs	Headway based service when used, informs period targets
shapes.txt	shape_id, shape_pt_lat, shape_pt_lon, shape_dist_traveled	Geometry and cumulative distance for segment distance shares
feed_info.txt	feed_publisher_name, feed_start_date, feed_end_date, feed_version	Version control for provenance and governance

Transport for West Midlands maintains an official API that exposes timetable and real time predicted departures. Public archives such as Transitland also record dated snapshots of the Transport for West

Midlands schedule feed which currently aggregates multiple agencies including West Midlands Metro. These sources confirm the availability of both schedule and real time data for a reproducible analytical workflow [3].

The West Midlands Metro operates between Birmingham and Wolverhampton with a zonal structure. The network map and zonal map clarify stop ordering and interchange points, which are important for validating stop sequences and for constraint definitions in the timetabling stage.

3.1.2 DATA GOVERNMENT CHECKLIST

All datasets originate from open public sources under TfWM's data licence, which permits reuse with attribution. The following checklist summarises the project's data-governance approach:

Table 4 Data Governance checklist

Checklist	Governance Dimension	Practice Implemented
✓	Origin & Licensing	TfWM GTFS static and real-time feeds; Creative Commons Attribution licence.
✓	Data Minimisation	Only operational attributes (route, stop, time) used; no personal or passenger data processed.
✓	Validation & Integrity	Automated schema and referential integrity checks executed at ingestion.
✓	Security & Privacy	All processing conducted on local or secured institutional servers; no personal identifiers handled.
✓	Retention & Versioning	GTFS snapshots stored with version tags; logs maintained for six months for reproducibility.
✓	Compliance	Conforms with UK Open Government Licence and institutional data-management policy.

3.2 DESCRIPTION OF PROCEDURE, ALGORITHMS, AND SCRIPTS

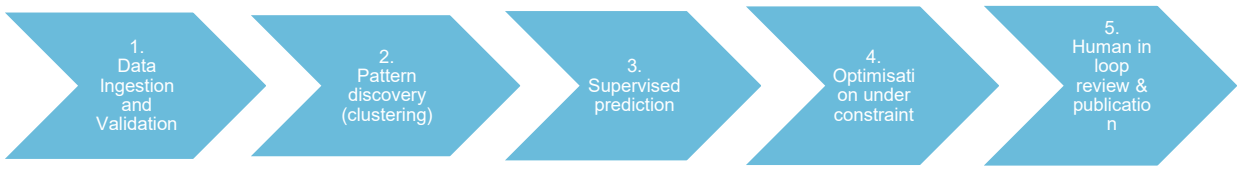

Figure 9 Conceptual AI Timetable Creation Pipeline

Developing an AI-supported timetable from GTFS data requires more than applying a single model in isolation. The process involves multiple stages of data preparation, analysis, prediction, and decision-making that must be logically connected in order to produce a timetable that is both operationally feasible and passenger oriented. In literature, scheduling frameworks are often presented as sequential

processes where raw data is refined step by step into usable outputs (Blanco et al., 2019; Ceder, 2016). Following this logic, the present study does not propose a specific algorithmic solution but instead outlines a conceptual pipeline, a methodological design that demonstrates how AI can be systematically integrated into the workflow of timetable creation.

The value of such a pipeline is twofold. First, it allows researchers and practitioners to visualise the relationship between different methodological components, from data ingestion to optimisation. Second, it provides a blueprint that can be tested in future studies or adapted by operators when resources permit. Importantly, the pipeline reflects the characteristics of West Midlands Metro and its open GTFS feed, which means it is not an abstract framework but one that aligns with real data availability and operational constraints in a medium-sized European tram system.

Figure 9 illustrates this conceptual pipeline, which consists of five main stages: (1) ingestion and validation, (2) pattern discovery, (3) predictive modelling, (4) optimisation, and (5) human-in-the-loop timetable release and monitoring.

3.2.1 INGESTION AND VALIDATION

The process begins with the collection of timetable data from the GTFS feed. These files contain information about routes, trips, stops, and operating calendars. Before any modelling can be attempted, the data must be checked carefully. Validation involves ensuring that all tables follow the correct schema, that links between files (such as trips and stop times) are consistent, and that the calendar correctly captures service days across the planning horizon. These checks are consistent with recommendations outlined in GTFS best practice guidelines [1]. Creating a data dictionary that explains how each GTFS field will be used in modelling is also an important step, since it allows researchers and practitioners to trace how raw schedule data are transformed into analytical features. As Ceder (2016) notes, the strength of any scheduling model depends heavily on the quality and structure of the underlying timetable data.

3.2.2 PATTERN DISCOVERY FOR TIME BANDING

Once the data are validated, the next task is to identify natural divisions in the service day. Public transport rarely operates at a constant frequency: early mornings are often quieter, peak periods around commuting hours are more intense, and evenings typically show a gradual reduction in service. To capture these differences, unsupervised learning methods such as k-means clustering or Gaussian mixture models can be applied to features derived from the stop-time records. These features might include the average interval between departures, observed trip durations, or typical dwell times at stops.

This form of segmentation has been successfully applied in the literature. For example, van der Knaap et al. (2024) used clustering techniques on large railway datasets to identify homogeneous demand periods, which were then used as inputs to be scheduling models. Applying a similar approach to West Midlands Metro data would allow the system to classify each trip into categories such as morning peak, inter-peak, or evening peak, forming the basis for subsequent prediction and optimisation.

3.2.3 SUPERVISED PREDICTION FOR HEADWAYS AND RUNNING TIMES

After the service day is segmented into time bands, supervised learning can be used to predict the key timetable variables. Two important quantities are the expected headway (the time interval between consecutive vehicles) and the running time between successive stops. Predictive models may draw on both static features (such as stop spacing or zone transitions from GTFS) and dynamic features (such as recent delay patterns from real-time feeds).

Earlier studies have shown the feasibility of this approach. Yu et al., (2011) demonstrated how regression models could predict bus arrival times using historical stop-time data. More recently, advanced deep learning methods such as temporal convolutional networks and transformer models have been applied to capture the non-linear dynamics in transit operations(Zhang et al., 2024). These models allow planners to move beyond fixed rules of thumb and instead generate data-driven predictions that reflect the actual operating environment.

3.2.4 TIMETABLE SYNTHESIS UNDER CONSTRAINTS

Predicted headways and running times cannot be adopted directly; they must be transformed into a timetable that respects operational rules. This stage uses optimisation techniques to generate a feasible schedule that accounts for cycle times of vehicles, minimum layover periods at termini, crew shift requirements, and platform availability at interchange points. Blanco, et al., (2019) developed a mixed-integer programming model for metro timetabling that incorporates such constraints, showing that optimisation can produce schedules that are both mathematically consistent and operationally realistic.

3.2.5 HUMAN-IN-THE -LOOP REVIEW AND PUBLISH

Even if algorithms propose a timetable, final responsibility rests with human planners. At this stage, planners review suggested changes through interactive tools that highlight where the new timetable deviates from the current one. Decision support dashboards can summarise key impacts, such as expected improvements in headway regularity, changes in vehicle minutes, or resolution of platform conflicts. This interactive oversight reflects the principles described by Ceder (2016), who emphasises that timetable planning must always combine automated methods with professional expertise.

Incremental control with real-time feedback

Once a timetable is published, service conditions will still vary. An online adjustment layer can make small changes in near real-time, for example by recommending a short hold at a terminus to re-space services more evenly. Reinforcement learning has been explored as a tool to make these micro-adjustments dynamically while respecting the published schedule (Ai et al., 2022). It has been demonstrated that such methods can improve headway regularity and passenger waiting times without requiring wholesale timetable redesign. In this context, real-time adjustments would act as a fine-tuning mechanism, ensuring stability for passengers while enhancing operational reliability.

3.3 RESULTS

Implementation Roadmap

The successful deployment of an AI-supported timetable system requires a phased approach that builds capacity while mitigating risks.

Phase one: **data readiness**

The foundation of the system lies in data quality. The first step is to establish automated pipelines that can access Transport for West Midlands (TfWM) GTFS feeds and real-time updates [2]. Validation routines must be developed to identify missing stop times, inconsistent identifiers, and irregular service calendars. A reliable dataset ensures that later stages of the pipeline are not compromised by upstream errors.

Phase two: **pattern discovery and prediction**.

Once data integrity is assured, offline experiments can be conducted using historical GTFS snapshots and archived real-time feeds from platforms such as [3]. Clustering methods can be applied to identify recurring service patterns, while predictive models can estimate headways and running times under different conditions. At this stage, the objective is proof-of-concept testing in a controlled environment rather than live deployment.

Phase three: **optimisation and integration**.

The next stage involves embedding predictive outputs into an optimisation solver that respects operational rules, including vehicle cycle times, platform capacity, and crew schedules. Previous studies have shown that mixed-integer and constraint programming approaches can generate feasible timetables that align with both passenger demand and operator constraints (Blanco et al., 2019). The outputs of this phase would be draft timetables suitable for expert review.

Phase four: **deployment and real-time adjustment**.

The final stage involves operational rollout with a monitoring layer capable of making minor real-time adjustments. Reinforcement learning approaches, as explored by Ai et al., (2021), can be used to smooth headway irregularities caused by delays while preserving the integrity of the published timetable. Transparency mechanisms such as logging and audit trails must be incorporated to ensure accountability for every adjustment.

3.4 CHALLENGES & LESSONS LEARNED

3.4.1 LIMITATIONS

Although the proposed framework offers a structured pathway for AI-supported timetable creation, several limitations must be recognised. The reliability of GTFS feeds remains a concern, as errors, missing values, or inconsistent identifiers can compromise the quality of downstream modelling. Inaccurate or incomplete real-time updates further reduce the robustness of learning-based approaches unless supplemented with strong validation and anomaly detection. Models trained primarily on routine historical patterns may also fail to capture the impact of rare but critical disruptions such as strikes, infrastructure failures, or large public events, highlighting the need for disruption-sensitive modelling. Similarly, optimisation outcomes are only as realistic as the operational parameters they incorporate; inaccurate estimates of cycle times, layover periods, or platform capacities could yield schedules that are mathematically feasible but operationally impractical. The real-time adjustment layer poses additional challenges, as frequent small changes to departures may undermine passenger trust if poorly

communicated or applied too close to departure. Finally, advanced methods such as deep neural networks raise issues of interpretability, as opaque model outputs may erode operator confidence unless accompanied by transparent governance, audit trails, and human oversight (Ceder, 2016; Urban Institute, 2025). These risks do not negate the potential of the framework but underscore the importance of incremental deployment, stakeholder involvement, and strong governance safeguards.

3.4.2 ETHICAL AND GOVERNANCE CONSIDERATIONS

While AI provides opportunities for efficiency, its integration into public transport timetabling raises important ethical and governance challenges. The proposed pipeline deliberately incorporates a human oversight layer to ensure that algorithmic outputs do not bypass professional judgment. This addresses the need for accountability and transparency in transport planning.

Governance frameworks, such as those outline by the Urban Institute [4], stress that the deployment of AI in public services must prioritise safety, accessibility, and fairness. In the transport context, this means ensuring that AI-generated timetables do not inadvertently disadvantage specific groups of passengers, such as late-shift workers, residents in peripheral zones, or people with reduced mobility. Workforce impacts must also be considered. Automating aspects of timetable creation may reduce manual workload but could also alter job roles for schedulers and planners. Proactive communication, training, and role adaptation are necessary to maintain trust within the organisation.

Ethical considerations further extend to the explainability of AI decisions. If a timetable recommendation cannot be understood by operations staff, its adoption risks undermining confidence in both the system and its governance. Therefore, interpretability features, audit trails of model outputs, and transparent reporting of optimisation trade-offs are integral to this use case.

3.5 EVALUATION

For an AI-supported timetable to be credible, it must be assessed through a transparent and comprehensive evaluation framework. Evaluation is not limited to technical model accuracy but must encompass service reliability, resource efficiency, and passenger experience. Figure 10 shows the implementation roadmap for timetable creation. This study proposes a multi-dimensional protocol that combines quantitative indicators with qualitative assessments. Each phase produces specific outputs that feed into the subsequent stage, creating a structured pathway from raw data to operational decision support.

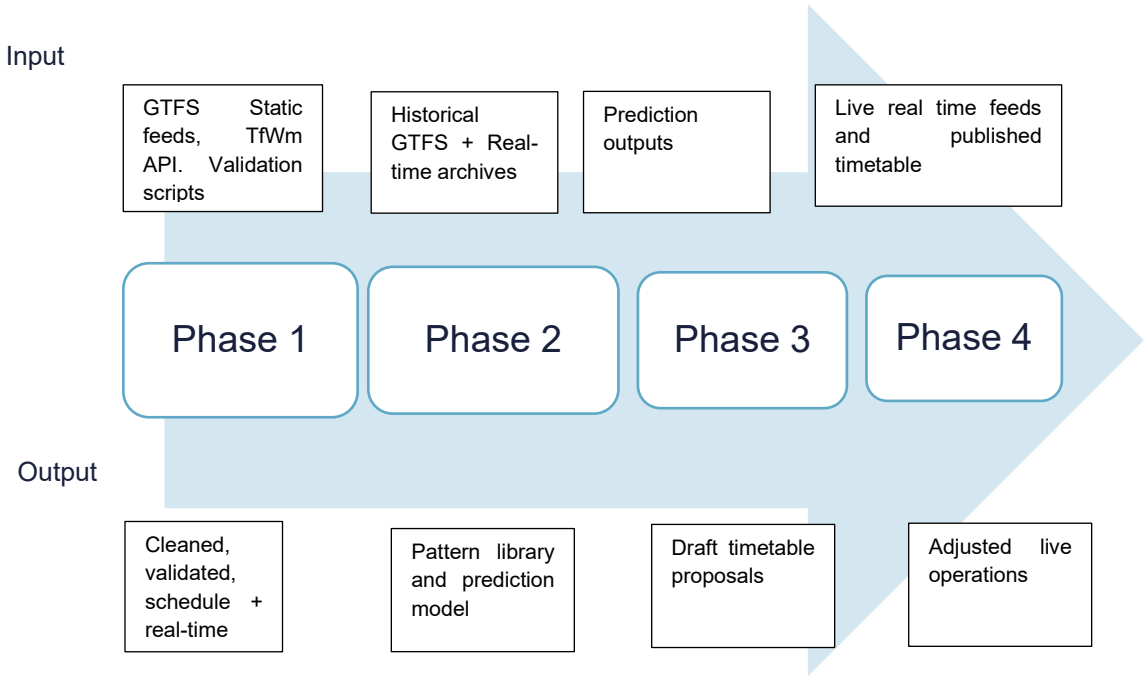

Figure 10 Implementation roadmap for timetable creation

The first dimension concerns the degree to which the timetable reflects intended service levels. Metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) provide a statistical measure of how closely predicted or generated departure times match actual operations. For high-frequency services such as West Midlands Metro, where passenger experience depends more on regular headways than on exact clock times, the Headway Regularity Index (Ceder, 2016) is particularly relevant. Complementing this, the percentage of departures that fall within an acceptable deviation of evenly spaced intervals offers a simple yet practical indicator of regularity.

A second dimension relates to how well the timetable utilises limited operational resources. Indicators such as vehicle cycle time utilisation and layover sufficiency capture whether vehicles and crews can realistically adhere to the schedule. At key interchange points like Wolverhampton Station or central Birmingham stops, platform conflict counts provide further evidence of whether the proposed timetable is operationally viable (Blanco et al., 2019).

Passenger-facing metrics. Evaluation also requires a passenger-centred lens. Transfer success rates measuring how many connections between tram services or with buses fall within acceptable transfer windows reflect the convenience of the timetable. Similarly, distribution of waiting times and the span of first-to-last service are essential for accessibility, particularly for commuters, students, and late-evening travellers.

A timetable cannot be evaluated only under typical weekday conditions. It must also be tested against weekend schedules, holiday periods, and special-event days that disrupt regular demand patterns. Out-of-sample validation using these varied calendar contexts helps ensure that AI-generated schedules are not overfitted to routine conditions. Recent work by Müller-Hannemann et al.,(2022) demonstrates how a machine-learning-based robustness oracle can efficiently approximate a system's performance

under diverse disruption scenarios, enabling rapid evaluation of timetable resilience across multiple operational contexts

Finally, no timetable should be adopted without input from practitioners. Structured evaluation workshops with schedulers, operations managers, and control room staff provide qualitative assessments of feasibility, interpretability, and compliance with organisational norms. This human-in-the-loop assessment complements quantitative metrics by addressing factors that cannot easily be measured.

3.5.1 EXPECTED CONTRIBUTION

Despite these limitations, the proposed use case offers several significant contributions to both First, it integrates methodological strands that have typically been addressed separately in the literature such as clustering for demand segmentation(Van Der Knaap et al., 2024), supervised prediction of arrival times (Yu et al., 2011), or optimisation of line planning (Blanco et al., 2019) into a unified conceptual pipeline.

Second, it grounds this integration in open GTFS data and Transport for West Midlands' real-time API, demonstrating that AI-enabled timetabling can be developed in contexts where transparency and replicability are possible, rather than relying solely on proprietary datasets.

Third, the framework is operationally relevant, as it explicitly accounts for practical constraints such as vehicle cycles, crew schedules, and platform capacity, distinguishing it from more theoretical treatments.

Fourth, it emphasises evaluation and governance by embedding technical performance metrics alongside passenger-centred outcomes and structured expert review, thereby promoting a holistic approach to assessing timetable quality.

Finally, the study contributes a forward-looking research pathway: it provides a blueprint that future studies can implement, refine, and extend, enabling comparative benchmarking of modelling approaches and empirical assessment of improvements in reliability, efficiency, and passenger experience.

4 PASSENGER DEMAND FORECASTING DURING NETWORK EXPANSION IN WEST MIDLANDS METRO, BIRMINGHAM: A LOG–LINEAR APPROACH

Accurate forecasting of metro passenger demand is central to effective transport planning and investment decision making. Demand forecasts guide infrastructure expansion, capacity allocation, fare setting, and long-term financial sustainability of transit systems [5]. Traditional statistical models such as linear or multiplicative forms have been widely applied in transport demand studies, yet they often face limitations when applied in dynamic urban environments where demand depends on economic, demographic, and policy factors (Godwin 1992; Button, 2010).

In this study, a log linear demand model is employed to estimate and forecast ridership for the West Midlands Metro. The model incorporates key independent variables like employment growth rate represented as GDP, population, network length (km), average fare while accounting for seasonal and trend effects as shown in the flowchart in Figure 11 below. Log linear models are particularly suitable because they enable direct elasticity interpretations, percentage changes in independent variables translate into proportional changes in demand.

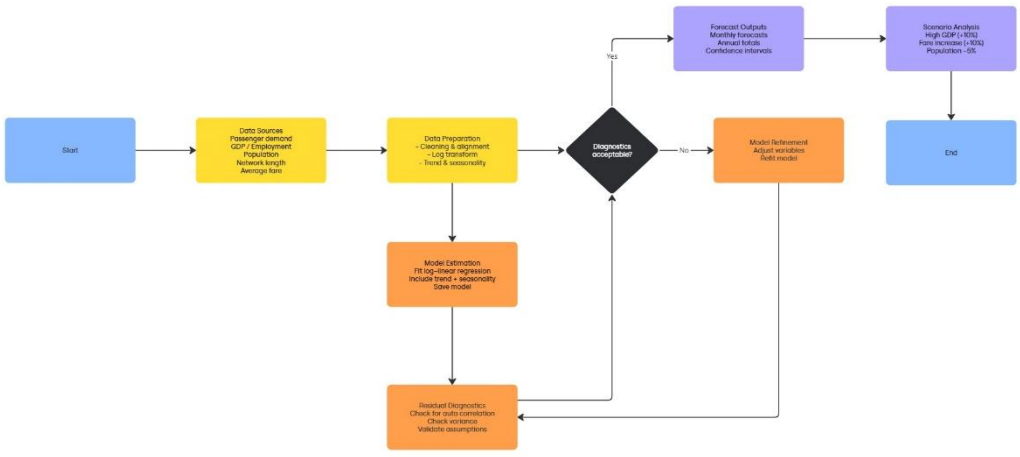

Figure 11 Flowchart of process involved

The analysis extends beyond baseline forecasts by conducting scenario analysis. Scenarios allow stress testing of demand forecasts under alternative assumptions, such as higher economic growth or

fare increases, thereby providing planners with insight of the robustness of capacity and policy decisions. This combination of baseline projections, elasticities and scenario testing contributes to knowledge and practical planning in metropolitan rail system.

4.1 DATA COLLECTION

4.1.1 DATA AND STUDY AREA

This research focuses on the West Midlands Metro network, a light rail system that connects Birmingham to Wolverhampton and serves as a central element of the region's public transport infrastructure. The network began operations in 1999 with the opening of Line One, a 20.4 km route between Birmingham and Wolverhampton. Since then, the network has undergone several expansion projects to meet the needs of a growing urban population and to support regional economic development [6];(Transport for West Midlands, 2023).

Currently, the network is undergoing one of its most ambitious expansion programmes. Five major extension projects are planned or recently completed, which together will extend the network length significantly and connect Birmingham, Wolverhampton, and Dudley more effectively. These projects include the Birmingham Eastside Extension, Birmingham Westside Extension, Wolverhampton City Centre Extension, East Birmingham–Solihull Extension, and the Wednesbury to Brierley Hill Extension. Collectively, these investments are designed not only to improve mobility but also to catalyse broader regeneration, including new housing, employment opportunities, and commercial development in the West Midlands [7].

The West Midlands is an ideal case study for demand forecasting due to the scale of its urban growth and its transport transformation. Birmingham's population has been growing steadily, with projections showing sustained increases in the coming years [8]. Wolverhampton and Dudley are also experiencing demographic and economic changes linked to new housing and employment investments. These trends, combined with large-scale infrastructure expansion, make accurate passenger demand forecasts critical for planning capacity, financial sustainability, and long-term policy development.

This chapter therefore sets the foundation for the forecasting analysis. Section 2.1 introduces the study area by detailing the West Midlands Metro and its extension projects. Section 2.2 presents the data sources, including ridership, demographic, economic, and network information. Section 2.3 explains the data preparation process, which involved harmonising these diverse datasets into a coherent monthly time series suitable for econometric modelling.

4.1.1.1 STUDY AREA

The West Midlands Metro is a light rail system that connects Birmingham to Wolverhampton, forming a critical component of the region's public transport network. Operations began in 1999 with Line One, a 20.4 km corridor between Birmingham and Wolverhampton. Since its launch, the metro has been expanded incrementally to serve a growing urban population and to underpin regional economic development [9].

At present, the network is undergoing one of its most ambitious expansion programmes since inception. Five major extension projects, either recently completed or underway, will significantly increase network

coverage and integrate additional urban centres such as Dudley into the system. These projects are designed not only to expand transport connectivity but also to stimulate regeneration through housing delivery, job creation, and increased accessibility to business, leisure, and education facilities [7].

The five key projects are outlined below:

1. Birmingham Eastside metro Extension.
   The Birmingham Eastside Metro Extension expands the network towards Digbeth, directly linking to the High Speed 2 (HS2) station at Curzon Street. The project comprises a 1.7 km twin-track route running from Bull Street to a new terminus at High Deritend. Construction began in 2021 and was completed in May 2024.
2. Birmingham Westside Metro Extension.
   The Birmingham Westside Metro Extension on the other hand planned to extend to the Centenary square (840m). This extension was in 2 phases with the first phase to Centenary Square completed in December 2019 and the second phase of the project extending Edgbaston, opened for passengers in 2022 [7].
3. East Birmingham and Solihull Mero.
   The East Birmingham to North Solihull Metro Extension is planned as part of the wider Sports Quarter Project, which includes the proposed 60,000-seat Birmingham City FC stadium at Bordesley. The extension is designed to provide a direct tram connection between the city centre and the new stadium, improving accessibility for residents and visitors. Although construction has not yet commenced, the project represents a key future expansion of the metro network into East Birmingham.
4. Wolverhampton City Centre Metro Extension.
   The Wolverhampton City Centre Metro Extension was completed in 2023, adding approximately 0.7 km of track within the city centre. The project introduced key new stops at both the bus station and the railway station, creating an integrated transport hub. By improving connectivity across modes, the extension encourages a modal shift from private cars to public transport, supporting sustainable mobility goals.

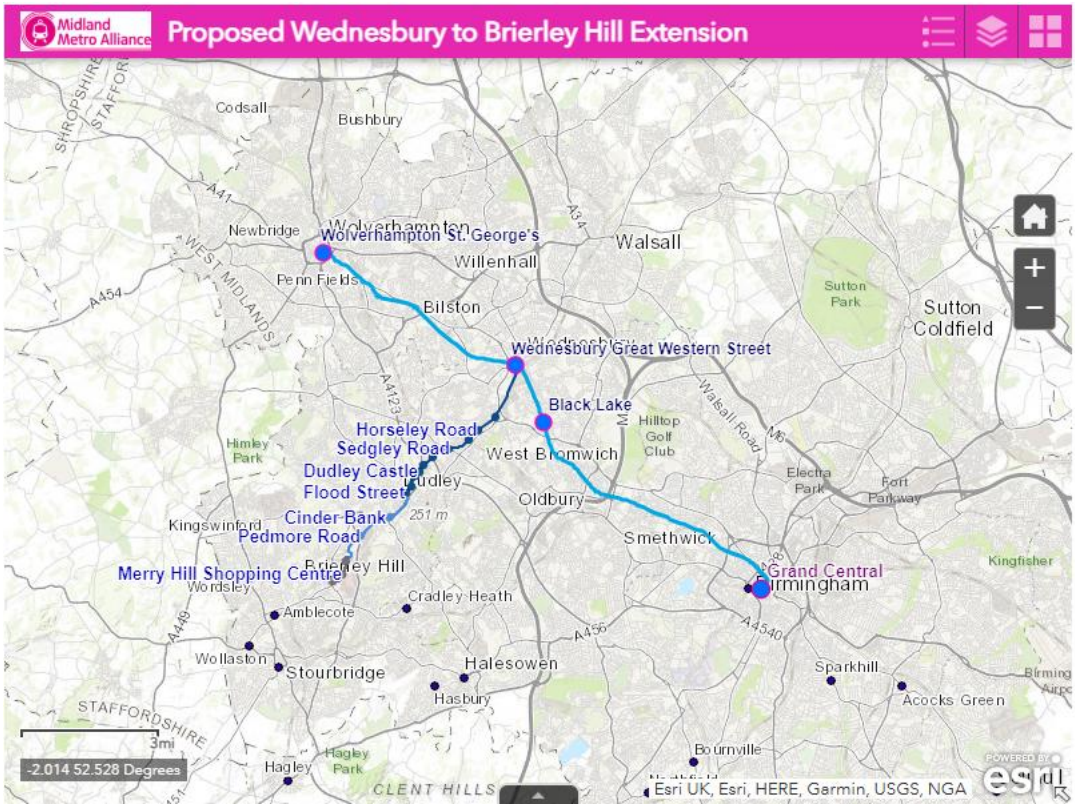

Figure 12 Proposed Wednesbury to Brierley Hill Extension

The Wednesbury to Brierley Hill Metro Extension in Figure 12, which forms the focal case study of this research, constitutes an 11 km expansion branching from the existing line at Wednesbury Great Western Street before passing through Tipton and Dudley enroute to Brierley Hill. This extension is particularly significant as it expands the network beyond Birmingham and Wolverhampton to incorporate Dudley, thereby strengthening regional connectivity. The project is being implemented in two phases: the first introduces nine new stops, while the second will add a further five. Both phases are scheduled for completion in autumn 2025. Given its scale and regional importance, this study examines the potential impact of the extension on passenger demand, with particular attention to anticipated ridership growth following the integration of Dudley into the metro system.

The Wednesbury to Brierley Hill extension is central to this study because it expands the metro beyond the Birmingham–Wolverhampton corridor into Dudley, introducing a new urban population into the network catchment area. The forecast of passenger demand after this extension forms the main applied component of this research.

4.1.1.2 DATA SOURCES

To develop a forecasting model for metro passenger demand, this study combines transport, demographic, and economic data from multiple sources. Table 5 summarises the key variables.

Table 5 Data Sources and variables

Variable	Unit	Source	Frequency	Description
Passenger Demand	Hundred thousand	Transport for West Midlands (TfWM)	Monthly	Total monthly ridership
Population	persons	Office of National Statistic (ONS), Birmingham City Council	Monthly	Estimated city population
Network_km	km	Midland Metro Alliance	Monthly	Metro network length
AvgFare	Currency (£)	Midland Metro Alliance, TfWM	Monthly	Average fare per trip
GDP	Percentage	Office of National Statistics (ONS)	Monthly	Employment growth rate
Expansion Indicators	Index	Derived (0,1)	Monthly	Expansion phase marker

Data collection combined secondary sources (ONS, TfWM, Midland Metro Alliance) with derived variables. Passenger demand data from TfWM was obtained for January 2013 to May 2025, forming the dependent variable in the forecasting model. Independent variables included population, economic growth, average fares, and network length.

4.1.2 DATA GOVERNANCE CHECKLIST

To ensure responsible handling of all data used in this study, a structured governance protocol was followed. Table 4 below summarises the main governance categories:

Table 6 Data governance checklist

Checklist	Governance Dimension	Description
✓	Data Origin	Passenger counts sourced from TfWM (confidential). Population and employment from ONS. GDP proxies from EY regional forecasts. Network km from Midland Metro Alliance.
✓	Licensing & Permission	TfWM data used strictly for internal research; redistribution prohibited. Public datasets used under standard open-data licences.
✓	Privacy & Security	No personal data used. All datasets contain aggregate flows; risk of re-identification is negligible. Secure storage implemented for TfWM data.
✓	Data Minimisation	Only variables required for modelling (demand, GDP, fares, population, network km) retained.
✓	Retention Policy	TfWM data to be deleted when project concludes or upon request. Derived variables (log transforms, annual aggregates) may be retained.
✓	Ethical Use	Analysis adheres to principles of transparency, replicability, and non-misuse. No operationally sensitive details disclosed publicly.

This framework ensures that the study aligns with the best practices for data governance and meets academic and organisational standard for responsible data management.

4.2 DESCRIPTION OF PROCEDURE, ALGORITHMS, AND SCRIPTS

4.2.1 DATA PREPARATION

Given the differences in data availability and frequency, careful preparation was required to construct a consistent monthly dataset.

- Population: Population data was obtained from the Office for National Statistics (ONS) annual mid-year estimates for 2013-2023 Birmingham and Wolverhampton. Since the model required monthly inputs, each annual value was treated as constant for all twelve months of the corresponding year. To extend the series for 2024 and 2025, projected growth rates were applied. Specifically, annual growth was calculated using the compound growth formula and applied to 2023 base to generate values for both cities.Dudley population was incorporated from 2025, coinciding with the opening of the Wednesbury–Brierley Hill extension, which brought the area into the metro's catchment.
- GDP / Employment Growth: Employment rates for Birmingham, Wolverhampton, and Dudley were sourced from ONS for 2013–2023. Values for 2024 and 2025 were calculated by applying the EY UK Regional Economic Forecasts projected growth rates (0.9% for Birmingham, 0.6% for the West Midlands). To the 2023 baseline, for example: $Employment_{2024} = Employment_{2023} * (1+0.009)$.
  Population-based growth rates were also derived using the compound annual growth formula: $(P_{2021}/P_{2011})^{1/10}-1$
  These projected values were then used to estimate employment for 2024–2025. A weighted average across Birmingham, Wolverhampton, and Dudley was computed, with weights determined by each city's share of the total population, to generate the final employment growth series for the model. [10]
- Average Fare: Fare levels were obtained from TfWM announcements (e.g., £6.90 peak and £5.40 off-peak day tickets). Since detailed fare series are not public, the most frequent fare categories were used as proxies, interpolated to monthly data where necessary [11].
- Network Length: Metro length was updated as each extension was completed (e.g., Westside, Wolverhampton, Eastside projects). Monthly records were created to align expansions with passenger demand data.
- Expansion Indicator: A binary (0/1) marker was constructed. Value = 0 before a new extension became operational; Value = 1 from the date of opening. This controls for structural changes introduced by new infrastructure.

4.2.2 MODEL SPECIFICATION

The forecasting framework adopts a log-linear econometric model. A wide range of demand forecasting approaches exist, from simple extrapolations of past growth to more detailed econometric formulations. The choice of technique is highly dependent on the availability and quality of data. According to the Airports Council International (ACI) guidelines, three broad techniques are typically employed in passenger and cargo demand forecasting: time series models, simple growth rate methods, and econometric models [12].

For the present study, the econometric approach was selected because it allows a multivariate specification in which passenger demand is explained by multiple socio-economic and system-level drivers. This approach captures the sensitivity of demand to changes in independent variables and produces elasticities that are directly interpretable for policy and planning.

Independent variables were chosen based on their expected influence on metro ridership in the West Midlands. These included:

- Population (proxy for demographic growth).
- Gross Domestic Product (GDP) (proxy for employment and income growth).
- Network length (km) (extent of metro infrastructure).
- Average fare (price of travel).
- Expansion indicators (binary variables capturing major system extensions).

Historical data for these variables were compiled from multiple official sources, ensuring consistency and coverage across the study period.

Within the econometric framework, a log-linear specification was found to best represent the relationship between passenger demand and its determinants. Logarithmic transformation provides two advantages. First, it yields elasticity estimates in which coefficients can be interpreted as the percentage change in demand associated with a one percent change in the independent variable, holding other factors constant. Second, it helps stabilise the variance of time series that exhibit long-term growth trends such as GDP, population, and passenger demand.

The equation used is specified as:

$$\ln(D_t) = \beta_0 + \beta_1 \ln(GDP_t) + \beta_2 \ln(POP_t) + \beta_3 \ln(NETKM_t) + \beta_4 \ln(FARE_t) + \gamma t + \delta_m + \varepsilon_t$$

D_t = Passenger demand in month t

t= trend term

δ_m= seasonal effect for month, m

γt= deterministic time trend into capture long-term growth

ε_t = seasonal effect for month m, captured in dummy variables

GDP_t = Employment growth rate in month t

POP_t = Population in month t

$NETKM_t$ = size of metro network in kilometres in month t

$FARE_t$= Average fare in month t

4.2.3 ESTIMATION

The passenger demand model was estimated in R using the ordinary least squares (OLS) regression function lm(). The specification followed a log–linear structure, with the dependent variable defined as the natural logarithm of monthly passenger demand.

Independent variables included the logarithms of GDP, population, network size, and average fare. This log transformation allows direct interpretation of estimated coefficients as elasticities, a common approach in transport demand modelling.

To account for seasonality, a factor variable (month_f) representing the twelve calendar months was included in the regression. In addition, a deterministic time trend was added to capture systematic growth in demand beyond that explained by the regressors.

This modelling approach ensured that both long-term growth and seasonal variations were explicitly incorporated while retaining a straightforward elasticity-based interpretation of key economic and demographic drivers.

4.2.4 FORECASTING EXOGENOUS REGRESSORS

The demand model requires future values of the independent variables, specifically GDP, population, network length, and average fare. To provide these inputs, the historical monthly series for each variable were first converted into logarithmic form to match the log-linear specification of the demand model. Missing observations were checked, and a continuous monthly dataset was constructed to ensure regular frequency alignment across all regressors.

Each of the logged independent variables was then projected forward using autoregressive integrated moving average (ARIMA) models, estimated via the auto.arima() procedure in R. This approach automatically selects the optimal lag structure and differencing order based on information criteria, providing robust forecasts for non-stationary time series with trends. Forecasts were generated for a 90-month horizon, covering the period June 2025 to December 2032.

The resulting projections represent the expected trajectories of GDP, population, network size, and fare under the assumption that historical growth and trend patterns continue into the future. In addition to these statistical forecasts, deterministic features required by the demand model, such as the linear time trend and seasonal factors, were added to ensure consistency with the regression specification.

Generating Passenger Demand Forecasts

Passenger demand forecasts were generated by combining the estimated log–linear regression model with projected values of the exogenous regressors. Forecasts for GDP, population, network length, and fares were obtained through univariate ARIMA models, estimated using the auto.arima() function in R, as explained in the section above. These regressor forecasts were then incorporated into the demand equation to project monthly passenger volumes over the period from June 2025 to December 2032.

The log–linear structure of the model ensured that forecasted demand reflected proportional rather than absolute changes in the independent variables, a property consistent with established practices in transport demand modelling [5]. Seasonal patterns were explicitly captured using monthly indicator variables.

Forecasts were initially produced on the log scale, consistent with the model specification. These forecasts were then back transformed into passenger demand levels using the exponential function. The final outputs consisted of monthly forecasts of passenger demand that combined both macroeconomic determinants and systematic seasonal adjustments.

The complete estimation procedure, data transformations, and forecasting pipeline are provided in Appendix A, where the R code implementation is documented for reproducibility.

4.2.5 ALGORITHM DESCRIPTION

Table 7 Algorithm 1. Log- Linear Metro Demand Forecasting Framework with ARIMA projected regressors

Inputs	Monthly passenger demand (ridership count), GDP (UK employment growth rate), population (west midlands), network length (km), average fare (£)
Outputs	Monthly and annual passenger demand forecasts (baseline and scenario), 95% confidence intervals, elasticity estimates for GDP, population, network length, and fare.
Assumption	Log-linear functional form with constant elasticities; regressors are exogenous; monthly trend and seasonality capture unobserved dynamic

Step1. Initialisation
Load monthly demand and regressor data.
Create date index and month factor.
Set April–May 2022 demand to missing (construction closure) and linearly interpolate.
Transform demand, GDP, population, network length, and fare using natural logs.
Remove non-finite values and create a monthly trend variable.

Step2. Model Estimation
Fit the log linear model: $\ln(D_t) = \beta_0 + \beta_1 \ln(GDP_t) + \beta_2 \ln(POP_t) + \beta_3 \ln(NETKM_t) + \beta_4 \ln(FARE_t) + \gamma t + \delta m + \varepsilon_t$
Extract elasticities from coefficients.

Step3. Diagnostic Checks
Plot residual time series, ACF, histogram, and Q-Q plot.
Apply Ljung Box test for autocorrelation.

Step4. Forecasting Regressors
Convert each logged regressor to a monthly time series.
Fit ARIMA models using auto.arima().
Forecast each regressor to the chosen horizon.

Step5. Baseline Demand Forecast
Build a future dataset with projected regressors, extended trend, and month factors.
Predict future log demand and back transform with exp.
Compute 95 percent confidence intervals.
Aggregate monthly forecasts to annual totals.

Step6. Scenario Analysis
Create modified regressor paths:
High GDP: log_gdp + log(1.10) (10% increase)
High Fare: log_fare + log(1.10) (10% fare rise)
Slow Population: log_pop + log(0.95) (5% lower growth)
Recompute demand forecasts for each scenario.

Step7. Output
Export baseline and scenario forecasts in monthly and annual form.
Save model object, forecast tables, and figures.

4.3 RESULTS

4.3.1 MODEL COEFFICIENTS

The log-linear regression produced statistically significant coefficients for all independent variables. Table 8presents the estimates, standard errors and p-values. The co-efficient can be directly interpreted as elasticities, indicating the percentage change in passenger demand with a 1% change in each independent factor.

Table 8 Estimated Co-efficient of the Log-Linear Model

Variable	Estimate (β)	Std. Error	p-value	Elasticity interpretation
Intercept	-47.41	12.82	0.000	Baseline constant
Log_GDP	4.15	2.21	0.063	1% increase in GDP equals 4.15% demand (marginally significant)
Log_POP	1.23	0.74	0.099	1% increase in Population equals 1.23% demand (marginally significant)
Log_NETKM	9.26	1.97	0.000	1% increase in network length equals 9.26% demand (highly significant)
Log_Fare	-1.30	0.90	0.148	1% increase in fare equals 1.3% demand (not significant)

Table 4 shows the elasticities of passenger demand with respect to employment growth rate/ GDP, population, network length and fare. The result suggests that demand is highly sensitive to network expansion, with a 1% increase in network length leading to a 9.26% increase in demand, significant at the 1% level. GDP and population growth both positively influenced demand at elasticities of 4.15 and 1.23 respectively, although only marginally significant at the 10% level. The fare elasticity is negative, as expected, indicating that higher fares reduce demand, but the effect is not statistically significant.

4.3.2 RESIDUAL DIAGNOSTICS

Residual analysis was conducted to assess whether log-linear regression model met the standard of linear modelling. The time series plot of residuals Figure 13 shows that residuals fluctuate around zero but with some visible persistence, indicating possible autocorrelation.

The histogram and overlaid density plot of residuals suggest approximate normality, though some skewness is present. This is supported by the Normal Q–Q plot, where most points follow the theoretical 45-degree line, but deviations occur in the tails, showing that extreme values are not perfectly captured.

The residual-versus-fitted plot shows no strong pattern, suggesting homoscedasticity (constant variance), although there is moderate clustering around the fitted values.

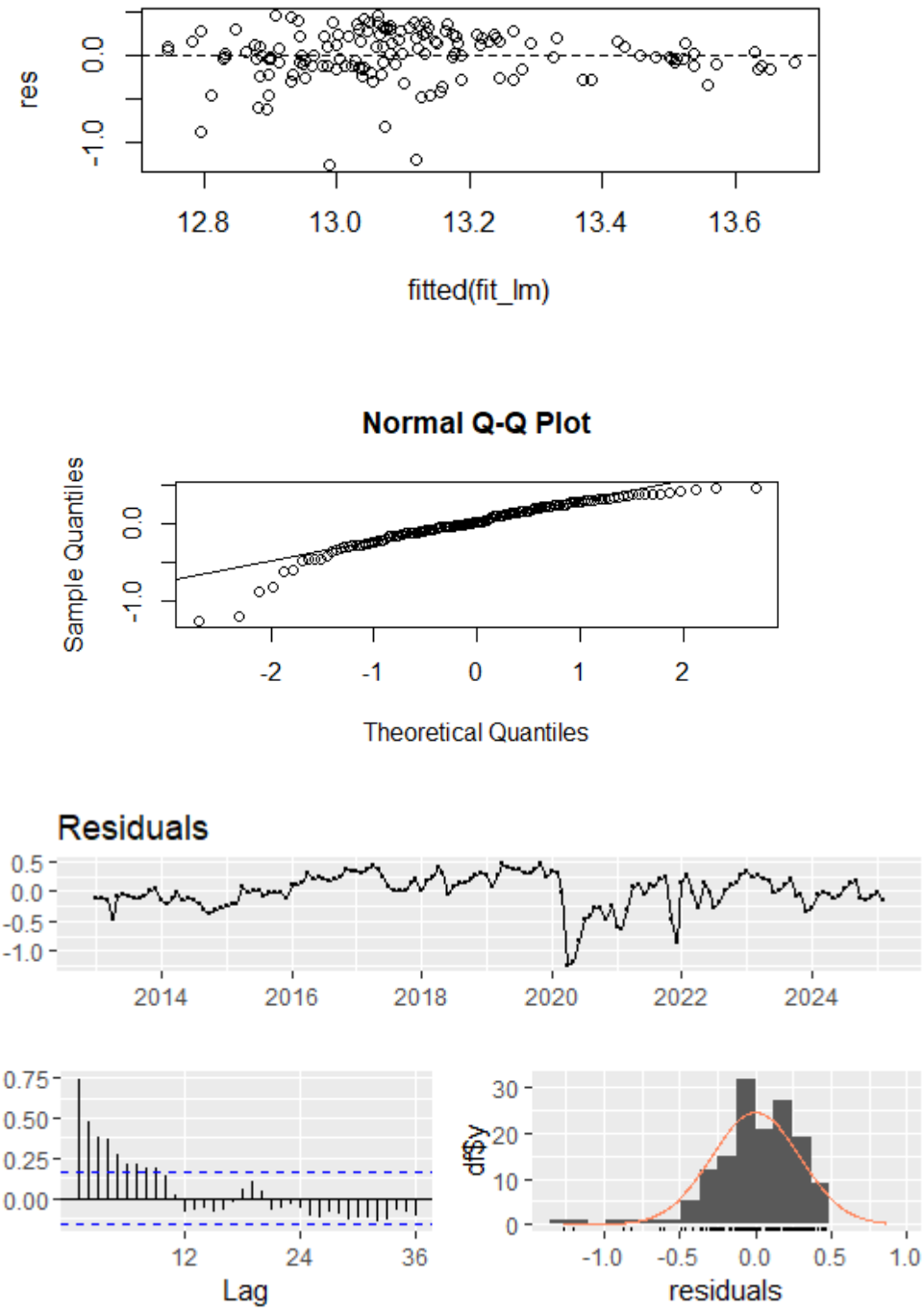

Figure 13 Residual diagnostics, Q-Q Plot and Residual vs Fitted plot

Ljung-Box test

data: Residuals

Q* = 210.13, df = 24, p-value < 2.2e-16

The Ljung–Box test was highly significant (Q* = 210.13, p < 0.001), which indicates that residual autocorrelation remains in the model. This suggests that while the log-linear specification captures the main systematic variation in demand, some time-dependent effects remain unexplained.

4.3.3 FORECAST PLOT

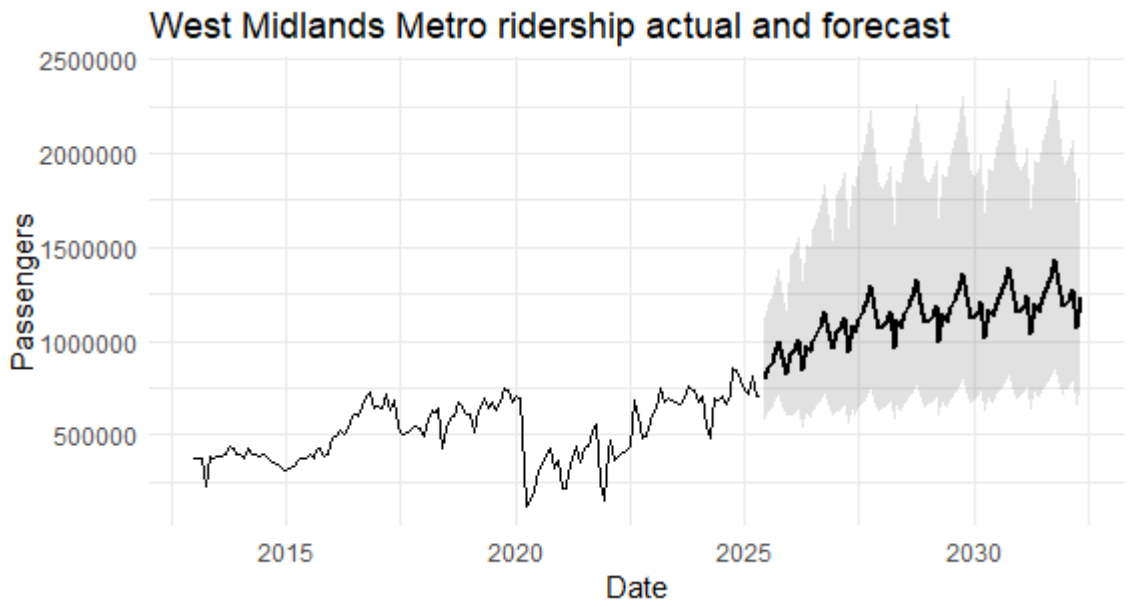

Figure 14 West Midlands Metro ridership actual and forecast

Figure 14 present the actual and forecasted West Midlands Metro between 2013 to 2031. The solid black line on the left represents observed ridership, while the line on the right shows predicted values generated from the log-linear regression model described in section 3. Seasonal and trend effects were explicitly modelled and exogenous regressors (GDP, Population, fare and network kilometres) were projected forward using time series methods.

The shaded region around the forecast denotes 95% confidence interval, reflecting the range of uncertainty associated with future predictions. As expected, the width of the interval expands as the forecast horizon increases, a common feature of time series models due to compounding uncertainty over longer horizons [13].

The forecast indicates a steady upward trend in metro ridership through 2032, driven primarily by projected economic and demographic growth. However, the forecast also retains visible seasonal fluctuations, with peaks in certain months of the year and troughs in others, consistent with historical

seasonal patterns. This suggests that the model successfully captured both long term structural drivers and short-term dynamics of passenger demand.

From a policy perspective, the results imply that service planners should prepare for sustained ridership growth under baseline economic and demographic conditions. The confidence interval highlights the potential variability of outcomes, underlining the importance of adaptive planning to account for economic shocks or unexpected events such as pandemics [14].

4.3.4 FORECAST TABLE (ANNUAL TOTALS)

Table 9 shows the annual totals of forecasted metro passenger demand for the period 2025 - 2032. The values are shown in millions of passengers, with 95% confidence intervals. To provide a policy-relevant perspective, the monthly forecasts were aggregated into annual totals. Annual demand forecast offers a clearer picture of Long-term ridership growth and are more directly usable for capacity planning, financial projections and infrastructure investment [15]

Because actual ridership data was available up to May 2025, the forecast horizon spans from June 2025 to December 2032. This produces partial-year totals for 2025 (June – December) to December 2032, (while 2026-2032) reflect complete forecasted years.

Table 9 presents the projected annual ridership alongside 95% confidence intervals

Year	Forecast Demand	Lower 95% CI	Upper 95% CI
2025	6.229.068	4.472.124	8.676.270
2026	11.935.722	7.557.706	18.852.065
2027	13.323.368	7.759.483	22.879.473
2028	13.676.891	8.043.962	23.257.071
2029	14.039.808	8.335.108	23.651.795
2030	14.405.691	8.624.946	24.064.002
2031	14.777.016	8.914.001	24.499.677
2032	13.939.481	8.473.420	22.934.867

The forecasts indicate a strong upward trajectory in passenger demand between 2026 and 2031, with annual ridership projected to increase by approximately 24% over this period (from 11.9 million to 14.8 million). The 95% confidence intervals widen slightly over time, reflecting greater uncertainty in long-horizon forecasts, yet the growth trends remain robust across all scenarios.

The 2032 total remains high relative to the start of the forecast window, although slightly below the 2031 peak, which is consistent with stabilising regressor projections at the end of the horizon. Confidence intervals widen through time, which reflects increasing uncertainty at longer horizons.

4.4 CHALLENGES & LESSONS LEARNED

4.4.1 DISCUSSION

The elasticities derived from the log-linear model provide important evidence on the drivers of metro ridership. Demand was found to be highly sensitive to GDP and population, confirming that macroeconomic and demographic growth are central determinants of urban transit demand [14] . The positive effect of network length validates the role of infrastructure expansion in stimulating passenger

uptake, consistent with findings in Wardman (2012) that service improvements are a key determinant of mode choice.

On the other hand, the negative fare elasticity underlines the importance of fare policy. Excessive fare increase could discourage ridership, reducing the ability of metro systems to achieve broader goals such as congestion mitigation and modal shift [16]. The scenario analysis further highlighted that while GDP shocks yield the largest variation in ridership outcomes, fare adjustments can also substantially influence demand, making pricing policy choices highly consequential.

4.4.2 LIMITATIONS

The model assumes that the regressors, that is, the GDP, Population, fare and network length are exogenous and not affected by metro demand. For example, increased ridership can influence fare-setting or prompt earlier network expansions.

Second, the model excludes other important factors such as service reliability, competition from ridesharing or buses, and changes in urban land use, which could influence demand.

Third, the forecasts of exogenous variables (GDP, population, fares, and network length) were generated using ARIMA models. While ARIMA is effective for capturing trends and short-term autocorrelation, it relies on historical statistical patterns. Alternative macroeconomic projections, policy-driven inputs, or structural assumptions could yield different demand trajectories.

Finally, structural breaks such as the COVID-19 pandemic illustrate the vulnerability of long-term forecasts to unforeseen disruptions. The results should therefore be interpreted as conditional on the continuation of historical dynamics rather than as precise predictions of future ridership.

4.4.3 CONCLUSION

This study applied a log-linear regression model to forecast West Midlands Metro ridership through 2032, using GDP, population, fare, and network length as independent variables. The results confirmed that demand is elastic with respect to economic and demographic growth, positively influenced by infrastructure expansion, and negatively affected by fare increases.

Baseline forecasts indicated continued growth in ridership, while scenario analysis demonstrated that outcomes are highly sensitive to GDP and fare assumptions. These findings underline the importance of integrating scenario-based planning into transport policy to address uncertainty.

Policy implications include the need to align metro investment with expected demand growth, safeguard affordability to support ridership retention, and build resilience against external shocks. Future research should incorporate additional determinants such as service quality and competing modes and explore non-linear modelling approaches including machine learning and hybrid models, to improve predictive accuracy in complex urban systems.

4.5 EVALUATION (SCENARIO ANALYSIS)

While the baseline forecasts in Table 10 provide a central estimate of future ridership, transport planning decisions often require considering alternative futures. To this end, Scenario analyses was taken to evaluate the robustness of demand forecast to changes in key drivers such as GDP growth and fare policy.

Scenario analysis differs from the statistical confidence intervals in section 4.4. Confidence intervals reflect uncertainty in the model's error term, whereas scenarios allow structural changes in independent variables. The log-linear specification of the model enables a straightforward elasticity-based interpretation: a one percent change in independent variable is associated to a proportional change in demand, holding other factors constant (Train, 2009). Scenarios were therefore constructed by modifying the future paths of selected regressors while holding others at baseline trajectories. More so like simulating uncertain incident to determine what the demand would look like. The result of the analysis can be seen in Table 10 and Figure 15 below.

Table 10 Summary of the annual ridership forecasts across scenarios

Year	Scenario	Forecast (Millions)	Lower 95% CI (millions)	Upper 95% CI (millions)
2025	Fare +10%	5.50	3.76	8.06
2026	Fare +10%	10.54	6.41	17.34
2027	Fare +10%	11.77	6.61	20.94
2028	Fare +10%	12.08	6.85	21.31
2029	Fare +10%	12.40	7.09	21.69
2030	Fare +10%	12.72	7.33	22.09
2031	Fare +10%	13.05	7.57	22.51
2032	Fare +10%	12.31	7.19	21.07
2025	High GDP +10%	9.25	5.48	15.62
2026	High GDP +10%	17.73	9.89	31.79
2027	High GDP +10%	19.79	10.42	37.61
2028	High GDP +10%	20.32	10.71	38.57
2029	High GDP +10%	20.86	11.00	39.57
2030	High GDP +10%	21.40	11.28	40.62
2031	High GDP +10%	21.95	11.55	41.73
2032	High GDP +10%	20.71	10.90	39.35
2025	Population –5%	5.85	4.43	7.72
2026	Population –5%	11.21	7.57	16.60
2027	Population –5%	12.51	7.79	20.10
2028	Population –5%	12.84	8.06	20.45
2029	Population –5%	13.18	8.35	20.82
2030	Population –5%	13.53	8.62	21.22
2031	Population –5%	13.87	8.90	21.63
2032	Population –5%	13.09	8.44	20.28

High GDP Growth Scenario (+10%)

To capture the effect of stronger than expected macroeconomic performance, GDP growth was raised by 10 percent relative to the baseline forecast. Results indicate substantial demand increases. For instance, ridership in 2030 reaches 21.4 million passengers, compared with 14.4 million under baseline conditions, representing a nearly 50 percent uplift. Even by 2032, when long-horizon uncertainty is greatest, ridership remains robust at 20.7 million passengers compared to 13.9 million baselines. This scenario highlights the pivotal role of economic growth in shaping metro demand.

Fare Increase Scenario (+10%)

The fare increase scenario simulated a 10 percent rise in real average fares. As expected, ridership decreases across the forecast horizon, reflecting the negative elasticity of demand with respect to price. By 2030, demand falls to 12.7 million passengers, around 12 percent lower than baseline. In 2032, ridership remains depressed at 12.3 million compared to 13.9 million in the baseline. These results underscore the sensitivity of demand to fare policy and the need for cautious balancing between financial sustainability and ridership growth (Small& Verhoef, 2007).

Slower Population Growth Scenario (-5%)

In this scenario, population growth was reduced by 5 percent relative to baseline projections. Demand declines modestly, with ridership reaching 13.5 million passengers in 2030, compared with 14.4 million baselines. By 2032, demand is 13.1 million, slightly below the 13.9 million baseline projection. This scenario suggests that while demographic trends matter, their impact is less immediate than macroeconomic shifts or fare adjustments.

Figure 15 Graph and bar chart showing annual ridership forecasts across scenarios

4.5.1 POLICY RECOMMENDATION

For the West Midlands, this evidence suggests that sustaining affordable fares while investing in network expansion is essential to support projected growth. Policymakers should embed demand scenarios into long-term planning frameworks, ensuring that service capacity and funding strategies are robust under both optimistic and adverse economic conditions. Integrating transport planning with wider economic development and land-use policies will further strengthen ridership growth and help deliver broader sustainability and congestion-reduction objectives.

Future Work:

Future work will consider applying XGBoost regression as a complementary approach to the baseline log linear model for monthly demand with exogenous regressors. This method can capture nonlinear effects and interactions among GDP, population, fares, and network kilometres, with model validation done using rolling origin cross validation. Results would be benchmarked against the current specification on the same folds, with SHAP analyses used for interpretability while elasticities continue to be reported from the log linear model.

Disclaimer:

Raw monthly passenger counts for West Midlands Metro are confidential at the request of Transport for West Midlands. This report shares derived outputs such as figures, tables, coefficients, and code. Access to the underlying ridership data can be provided to examiners or reviewers on request subject to approval from Transport for West Midlands.

4.6 DATA DESCRIPTION IN DETAIL

This appendix provides a detailed overview of the datasets used in the study, including definitions, units of measurement, and data sources. All variables were transformed to natural logarithms (log) to facilitate interpretation in elasticity terms.

Dependent Variable

- Passenger Demand (demand).
  - Definition: Monthly number of passenger journeys on the West Midlands Metro.
  - Unit: Number of passengers (counts).
  - Source: Transport for West Midlands (TfWM) operational data, internal ridership records.
  - Transformation: Logarithm applied after ensuring non-zero values.

Independent Variables

1. Gross Domestic Product (GDP).
   - Definition: UK real GDP at constant prices.
   - Unit: £ millions (chained volume measure, seasonally adjusted).
   - Source: UK Office for National Statistics (ONS).
   - Transformation: Natural log.
   - Justification: GDP is widely used as a proxy for economic activity and correlates strongly with transport demand (Goodwin, 1992; Button, 2010).
2. Population (population).
   - Definition: Estimated population of the West Midlands metropolitan area.
   - Unit: Number of residents.

- Source: ONS Mid-Year Population Estimates.
- Transformation: Natural log.
- Justification: Population growth increases the pool of potential transit users, influencing long-run ridership (Wardman, 2012).

3. Network Length (network_km).
   - Definition: Total length of metro track in operation.
   - Unit: Kilometers.
   - Source: TfWM expansion reports, project documentation.
   - Transformation: Natural log.
   - Justification: Infrastructure expansion increases service coverage and accessibility, which in turn generates demand (Williams & Rao, 2017).
4. Average Fare (avg_fare).
   - Definition: Real average fare per passenger journey.
   - Unit: British Pounds (£), adjusted for inflation using the UK Consumer Price Index (CPI).
   - Source: TfWM fare tables and ONS CPI data.
   - Transformation: Natural log.
   - Justification: Fare levels directly influence demand due to price sensitivity, as supported by transport elasticity studies (Small & Verhoef, 2007).

Additional Variables

- Trend (trend).
  - Definition: Sequential index of months to capture long-term underlying growth or decline in demand not explained by regressors.
  - Unit: Integer (1, 2, 3, …).
  - Source: Constructed variable.
- Seasonality (month_f).
  - Definition: Dummy variables for calendar months to capture seasonal patterns such as summer peaks or winter slowdowns.
  - Unit: Factor variable (January = 01, February = 02, … December = 12).
  - Source: Derived from observation date.
- Expansion Indicator (expansion_ind) [optional inclusion].
  - Definition: Binary variable equal to 1 in months when a major expansion project opened, 0 otherwise.
  - Unit: Dummy (0/1).
  - Source: TfWM project documentation.
  - Note: This was not used in the main model due to lack of future projections but could be incorporated in future work with known expansion timelines.

4.7 REPRODUCIBILITY AND ARTEFACTS

To ensure that the forecasting results presented in this study can be independently repeated, all computational steps have been fully documented and implemented in reproducible R code. The complete workflow, including data preparation, model estimation, scenario construction, and forecast generation, is provided in algorithm description. This includes:

The artefacts include:

- R scripts for data cleaning, log-transformation, regression estimation, ARIMA forecasting of regressors, demand prediction, scenario analysis, and visualisation.
- Model objects, including the saved fitted regression model (fit_lm.rds) and forecast outputs (forecast_monthly_baseline.csv, forecast_annual_baseline.csv).
- Software environment specifications, detailing package versions recorded in session_info.txt as recommended for scientific reproducibility.
- Data-access instructions, describing how West Midlands Metro administrative ridership data and ONS demographic indicators should be integrated.

Because the raw ridership dataset is confidential, replication requires requesting the data directly from Transport for West Midlands (TfWM). Once obtained, the workflow will yield comparable results, apart from minor deviations resulting from differences in ARIMA model re-estimation.

4.8 CROSS MODEL KPI TABLE AND BASELINE COMPARISONS

To benchmark the predictive performance of the proposed log linear model, three standard baseline models were estimated on the same monthly demand series:

(1) a seasonal naive model that repeats the previous year's monthly values,

(2) a historical mean model that forecasts a constant long run average, and

(3) a linear trend with monthly seasonal indicators fitted directly to observed demand.

Performance was assessed using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) to enable consistent comparison across specifications.

Table 7: Model Performance Comparison Using Shared KPIs

Model	**RMSE**	**MAE**	**MAPE**
Seasonal naive	190908	144296	35.6 percent
Historical mean	161030	139436	34.1 percent
Trend plus season	139798	110562	28.3 percent
Log linear with exogenous regressors	**126100**	**102638**	**23.9 percent**

The baseline results demonstrate the expected pattern. Both the seasonal naive and historical mean models perform poorly, with MAPE values exceeding 34 percent, confirming that models relying solely on past patterns or average behaviour cannot adequately capture structural changes in demand. Incorporating a deterministic trend and monthly seasonal dummies produces a notable improvement, reducing RMSE to 139798 and MAPE to 28.3 percent, which shows that regular seasonal fluctuations are an important component of metro demand.

The proposed log linear model achieves the strongest performance across all three KPIs. RMSE is reduced to 126100, MAE to 102638, and MAPE to 23.9 percent, marking a substantial accuracy gain relative to the best baseline. This improvement reflects the model's ability to incorporate exogenous drivers such as GDP, population, fare levels, and network kilometres, which influence long term ridership but are invisible to pure time series baselines. The comparison confirms that metro ridership is shaped by broader macroeconomic and infrastructural dynamics that cannot be captured by historical patterns alone. As a result, the log linear specification is the most suitable approach for strategic forecasting during periods of network expansion and economic uncertainty.

4.9 OPERATIONAL NOTES FOR DEPLOYABLE COMPONENTS

While the model implemented here is primarily research-oriented, its structure is suitable for integration into a practical forecasting pipeline used by metro operators. To support deployment, the following operational guidelines are proposed:

Configuration Requirements

- Requires R (≥ 4.0) and specific libraries: forecast, dplyr, ggplot2, lubridate, writexl.
- Input datasets must follow consistent column naming, monthly date indexing, and complete historical coverage.

Routine Monitoring

- Periodically evaluate model drift by comparing monthly forecasts with observed demand.
- Re-estimate the model annually or when substantial structural changes occur (e.g., network extensions).

Logging

- Archive model outputs, forecasts, and diagnostic tests.
- Save residual plots and KPI evaluations for continuous monitoring.

Fail-Safe Behaviours

If exogenous data cannot be updated (e.g., missing GDP forecasts), system should revert to:

- A trend plus seasonality fallback model, or
- A last-observed-value extrapolation.

Validation Procedure

Before deployment or model updates:

- Conduct a rolling origin back test (e.g., 12-month forecasting window).
- Compare RMSE, MAPE, and elasticity stability with previous model versions.
- Ensure no violation of core assumptions (no severe autocorrelation, no explosive residual variance).

5 ANOMALY DETECTION APPLIED ON METRO OPERATION

CCTV (video) surveillance is abundant in metro operations to ensure the safe and secure travel of the metro users. In addition to these important tasks, CCTV surveillance can be utilized to provide additional benefits like managing crowd control during peak hours. For this demonstrator data from CCTV surveillance is used for another task, namely for uncleanliness and litter detection.

Littering is the phenomenon that trash and grime are left behind in public places instead of being disposed properly in the provided trash bins. Literature discussing this issue, its social and cultural aspects as well as how to remedy it goes back at least 50 years (see e.g. Robinson et al, 1976).

Notably, what studies have found is that people are more likely to litter where litter is already present (Tehan et al, 2017 and references therein). Besides the effect that present litter promotes additional littering, litter also affects the perceived safety of people in this environment. Present litter and graffiti tend to cause concerns about perceived safety which negatively impacts the acceptance and usage of public transport (Debosc et al, 2012). There exist studies that further investigate gender-specific differences in the relationship of fear of crime and perceived safety in transit environments (Loukaitou-Sideris, 2014). It is, therefore, the aim of metro operators to keep their infrastructure clean also from this perceived safety perspective.

By identifying litter, debris and general pollution in the metro and its infrastructure as quickly as possible, the cleaning process can be adapted dynamically with the goal to respond such events fast and efficiently.

5.1 DATA COLLECTION

5.1.1 ETHICAL AND PRIVACY ISSUES

When it comes to CCTV surveillance, privacy and ethical issues are key concerns when working with such kind of data. As it is generally forbidden to discard litter in the metro infrastructure a comprehensible concern arises that the detection of the uncleanliness event in the CCTV footage might be utilized to implement a fining/suing process. Data might be used to alert police and/or safety personnel as the littering person might be identified and even be traced across several cameras. That is a legitimate concern and indeed, literature discusses such public littering and monitoring strategies (Alharbi et al, 2025). In addition, uncleanliness detection might be used to assess the performance of cleaning personnel, and this has been already discussed in literature (Jayasinghe et al, 2019).

The aim of this use case is totally unrelated to these far-reaching consequences and focuses solely on identifying littering and grime in the metro environment to ensure that a fast and efficient cleaning process can take place. The aim is to keep the metro environment clean and keep it attractive for metro users.

To counteract the previously mentioned concern related to privacy and ethics, two strategies are employed in the selection of CCTV data. First of all, it is technically beneficial to use images that do not

contain any people. The simple reason for this is that people in the image hinder the visibility and do not allow for scanning the entire space for littering. Consequently, images for training and testing are ideally free of people. Of course, choosing such images is only possible for this use case scenario but not in real world operation. If this method is utilized in real time evaluation of CCTV footage, anonymization software needs to be employed to ensure the privacy of the metro users.

5.1.2 DATA SETS

For the training and testing of clean/unclean environments dedicated data sets are needed. Unfortunately, the large standard image training data sets do not contain such classes. Therefore, dedicated data sets had to be obtained, which are quite small.

Clean/Littered Road Classification (Kaggle Data set #1): this data set comprises 237 images of clean and littered road environments (Figure 16) and an example of each class is shown in the following.

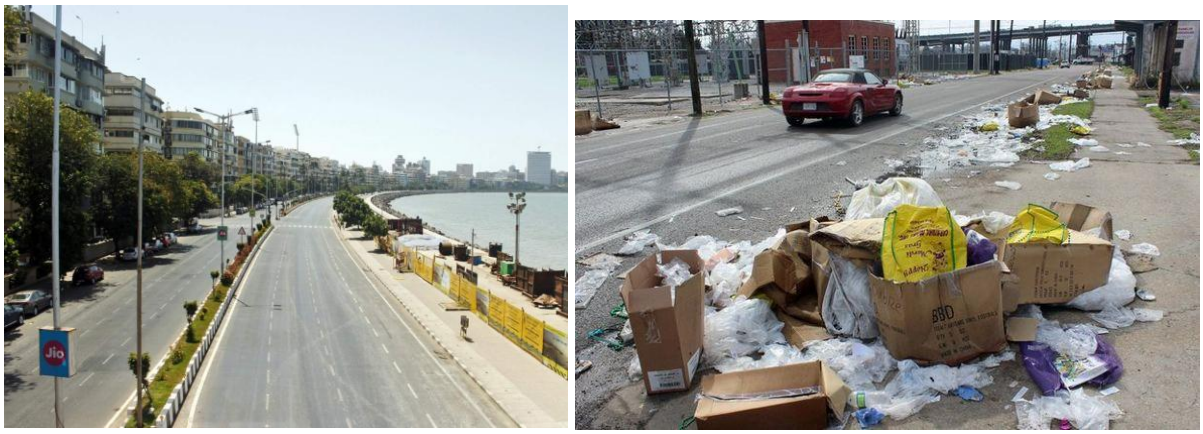

Figure 16 Examples of clean/unclean road environments

Visual Pollution Dhaka Streets (Kaggle Data set #2): this data set contains 1400 images of visual pollutants like actual trash, but also only visually annoying elements like wiring, advertisements etc. Only a subset of this data set contains actual littering.

In total, about 540 images were obtained that served as a training dataset to describe clean/unclean road environments and were used to train the classification-based methods.

Data Set #3: plitter: A dataset that heavily focuses on litter and various other classes, acquired from roboflow. It contains 12,752 images with localized labels, split into train, validation and test sets.

Data Set #4: TACO: The **TACO (Trash Annotations in Context)** dataset is a publicly available image dataset designed for litter detection and segmentation in real-world settings. It contains over 6,000 images with more than 10,000 manually segmented annotations covering various types of trash, such as plastic, metal, paper, and glass, captured in diverse urban and natural environments. The dataset follows the COCO format, enabling easy integration with standard computer vision frameworks. TACO aims to support research in environmental monitoring, waste management, and the development of autonomous litter detection systems. This dataset could not yet be used in our experiments. Should we have further issues with sourcing relevant data we will experiment with TACO and with merging some

labels of the above datasets with TACO to generate a larger dataset, to boost generalisation of the transfer-learned models. Metro-related data sets are even more difficult to obtain, and no suitable data set could be found on the common platforms. Therefore, ca 50 images from Instagram channel “trash train” were used (Instagram Trash Train data set) as test data set for uncleanliness detection (Figure 17).

5.1.3 DATA-GOVERNANCE CHECKLIST

The datasets used in this study are publicly available and licensed under CC0, MIT, or CC4 terms. In addition, we evaluate model performance on a small set of 50 images that were publicly uploaded to Instagram as part of a contest. These images are used exclusively for evaluation and are not included in any training data. Their limited, non-commercial use for research purposes is consistent with typical fair-use considerations.

- Data Origin & Ownership
  - The datasets are sourced from kaggle, roboflow and TACO, as described above.
  - The trash-train dataset is comprised of images on Instagram. The uploaders of the individual images are the continued copyright owners of the images.
- Privacy & Anonymity
  - We took special care to use datasets without any humans depicted and only using images recorded in public spaces. Any humans preset in the datasets are anonymized.
  - Our use-case strictly focuses on uncleanliness of equipment and environment.
- Security Controls
  - Because all datasets used in this study consist of publicly available information, no special confidentiality measures are required. Security controls are limited to standard best practices, such as controlled access to project files, protection against unauthorized modifications, and secure storage, to ensure data integrity and proper research reproducibility.

Figure 17 Exemplary images of unclean metro environments (from Instagram Trash Train data set)

5.2 DESCRIPTION OF PROCEDURE, ALGORITHMS, AND SCRIPTS

Training machine learning algorithms for image detection and classification requires generally very large image data sets and a lot of associated computing resources. For the use case demonstrator a different approach was chosen, namely transfer learning. Transfer learning means, that a model is being used that has been pretrained for a usually different (image classification) task. To make this pretrained model better suited for the use case of interest, generally a short training phase with a small use-case dedicated training data set is conducted.

However, for the present use case only a very small number of images could be collected that show unclean metro environments. The obtained data set is far too small to be usable for actual training. Therefore, a zero-shot approach was utilized.

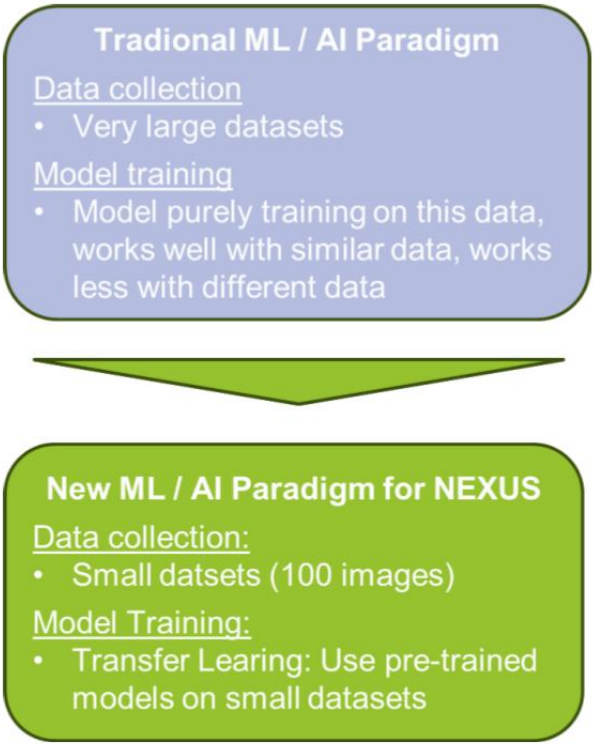

Figure 18 Traditional vs. Transfer learning approach

In more detail, the procedure in the current use case was as follows: the residual network (ResNet) model was chosen as a powerful pretrained image classification model (He et al, 2016). Despite that it having been published in 2015/2016 it still remains a powerful image classification method (Figure 18). ResNet models are deep convolutional neural networks known for their ability to train very deep networks by using residual connections to prevent vanishing gradients.

Originally, the ResNet model was pretrained on the ImageNet 2012 classification dataset (Russakovsky et al, 2015) that consists of 1000 classes and includes ca. 1.28 Mio training images. However, these 1000 classes only contain generic classifying elements like aeroplane, bicycle, dog, sofa etc but not more abstract classes like clean, unclean. Differently sized versions of the ResNet models exist that vary by the number of layers.

Consequently, the ResNet50 model (has 50 layers) was trained on a small data set containing littered places to learn the new clean/unclean environment classes (Figure 19).

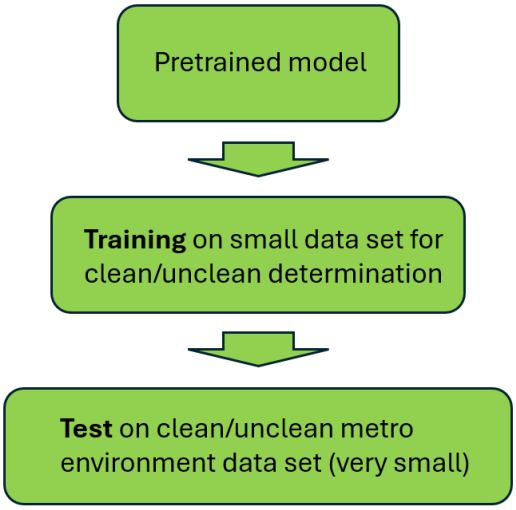

Figure 19 Uncleanliness training pipeline

Only after this step, ResNet50 becomes usable to determine clean and unclean environments. In the final step, this modified ResNet50 model is then tested on a small number of available unclean metro images to determine the prediction performance.

In technical terms, the machine learning part was implemented in Python using the PyTorch library.

After being trained on the clean/unclean classes, it reaches more than 98% of training accuracy and is then tested on the clean/unclean metro environment data set (Figure 20). The modified ResNet50 model achieves 86% accuracy in the uncleanliness classification task on the test set.

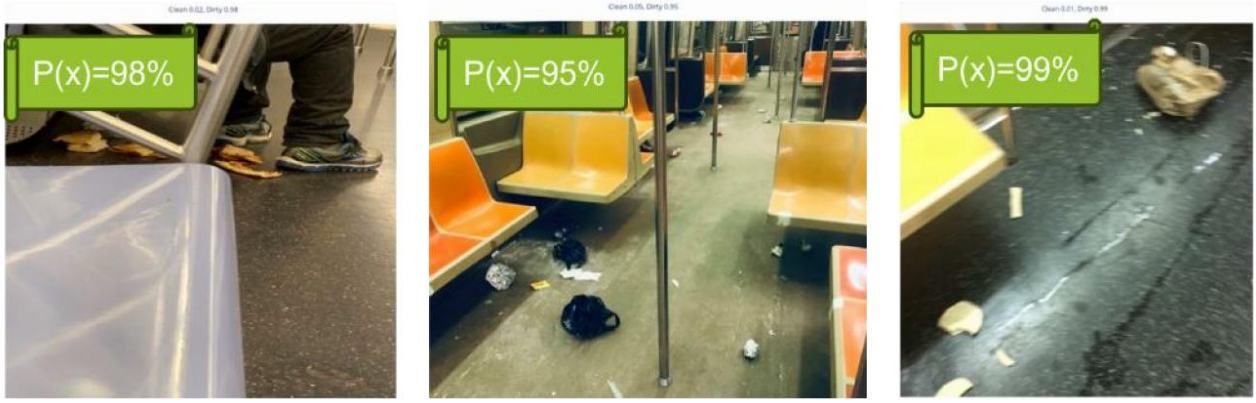

Figure 20 Uncleanliness determination confidence of the modified ResNet50 model

5.3 RESULTS

The goal of this work was to explore automated methods for detecting uncleanliness in public transport environments. We investigated two approaches: (i) image classification using a ResNet architecture, and (ii) object detection using a YOLO framework. Both models were trained on publicly available litter datasets. The classification models were trained on a dataset produced by merging dataset #1 and #2 and the Object Detection model was trained on plitter, which contains annotated examples of discarded packaging, bottles, and related waste. To assess real-world performance, we evaluated the models on a small, custom dataset of 50 images collected in the New York City metro system, which presents much harsher, cluttered, and less structured conditions than curated benchmarks.

In parallel, we considered the suitability of anomaly detection (AD) methods for this task. While we could not implement AD due to the absence of an appropriate dataset, we outline in this section why AD offers a promising future direction for “uncleanliness detection,” especially for amorphous forms of dirt such as stains, dust, or liquids. We could not yet source a dataset of normal CCTV images of clean metro trains and stations. In the second part of NEXUS we hope that by working together with the metro operators in the project we can attain a sizeable dataset.

5.3.1 UNCLEANLINESS CLASSIFICATION WITH RESNET

We first implemented a ResNet-based classifier (He et al., 2016) fine-tuned on the plitter dataset. The model was trained to predict whether an image contained visible litter or appeared clean.

The best model was achieved after just 2 epochs of retraining with ResNet50 pretrained on Imagenet1k_V2. We replaced the last layer with 2048 fully connected neurons and 2 output neurons and softmax as output activation function. We treat our “trash train” dataset as test set, but do not use it to choose any hyperparameters besides the number of epochs. We are forced to skip a validation step because of the limited data available for this experiment. We achieved the following results.

5.3.1.1 TRAINING PERFORMANCE ON MERGED URBAN LITTER DATASETS

ResNet50 attains a very good fit to the binary classification task. After only 2 epochs its training scores are as demonstrated in Figure 21, Figure 22, Figure 23 below.

Training Set			
TARGET / OUTPUT	Clean	Unclean	SUM
Clean	112 20.86%	7 1.30%	119 94.12% 5.88%
Unclean	1 0.19%	417 77.65%	418 99.76% 0.24%
SUM	113 99.12% 0.88%	424 98.35% 1.65%	529 / 537 98.51% 1.49%

Figure 21 ResNet50 Classification Training Confusion Matrix

Class Name	Precision	1-Precision	Recall	False Negative Rate	F1 score	Specificity (TNR)	False Positive Rate (FPR)
Clean	0.9912	0.0088	0.9412	0.0588	0.9655	0.9976	0.0024
Unclean	0.9835	0.0165	0.9976	0.0024	0.9905	0.9412	0.0588

Figure 22 ResNet50 Classification Training Performance Metrics

Accuracy	0.9851
Misclassification Rate	0.0149
Macro-F1	0.9780
Weighted-F1	0.9850

Figure 23 ResNet50 Classification Training Accuracy Metrics

5.3.1.2 TEST PERFORMANCE ON "TRASH TRAIN" DATASET

Test Set			
TARGET / OUTPUT	Clean	Unclean	SUM
Clean	0 0.00%	7 14.00%	7 0.00% 100.00%
Unclean	0 0.00%	43 86.00%	43 100.00% 0.00%
SUM	0 0.00% 0.00%	50 86.00% 14.00%	43 / 50 86.00% 14.00%

Figure 24 ResNet50 Classification Test Confusion Matrix

Class Name	Precision	1-Precision	Recall	False Negative Rate	F1 score	Specificity (TNR)	False Positive Rate (FPR)
Clean	0.0000	0.0000	0.0000	1.0000	0.0000	1.0000	0.0000
Unclean	0.8600	0.1400	1.0000	0.0000	0.9247	0.0000	1.0000

Figure 25 ResNet50 Classification Test Performance Metrics

Accuracy	0.8600
Misclassification Rate	0.1400
Macro-F1	0.4624
Weighted-F1	0.7953

Figure 26 ResNet50 Classification Test Accuracy Metrics

We observe that the misclassified images are 1 with worse than average image quality, 1 with faecal matter (which does not appear in the training set (at least not prominently)) and 5 with spills of different fluids, which are also not a distinguishing factor in the training dataset. Following this experiment, we hoped that with the more modern YOLO architecture and by using object-detection or anomaly-detection methods we could increase the performance even further or provide additional value by highlighting the areas in the image where a problem was detected.

5.3.2 UNCLEANLINESS CLASSIFICATION WITH YOLO-CLS

For the object detection tasks, we settled on the Ultralytics Python API which provides access to modern YOLO models. Since this platform also provides a classifier based on YOLO be chose to run the same experiment we did with ResNet50 with YOLO11x-cls. We froze all but the last layer and set it to the two classes we have in our dataset and trained it on our data with default arguments.

5.3.2.1 TRAINING PERFORMANCE ON MERGED URBAN LITTER DATASETS

The reduced dataset size is due to the automatic dataset splitting of the pipeline, and some images being filtered out of the dataset due to a mismatch in formats or parsing errors. The remaining training dataset is trained to 100% accuracy within a couple of epochs and the best model also reaches 100% accuracy on the validation set Ultralytics reserves automatically. In binary classification, Top-5 accuracy is always 100%.

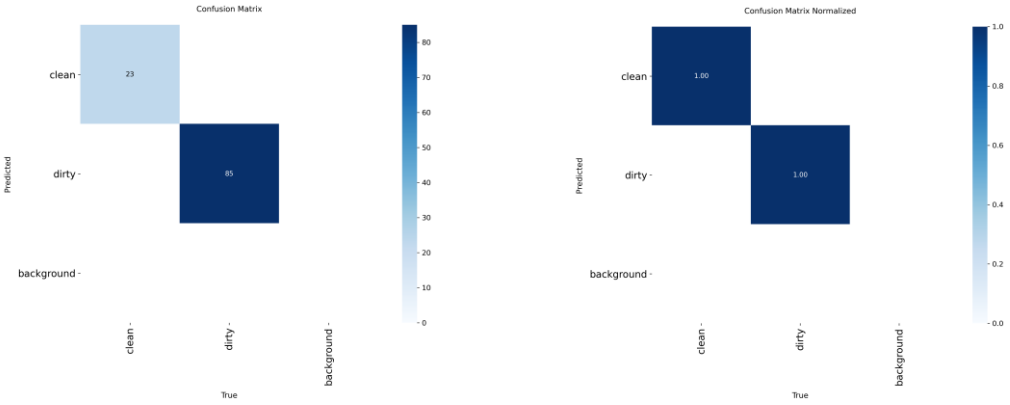

Figure 27 YOLO11x-cls validation confusion matrices

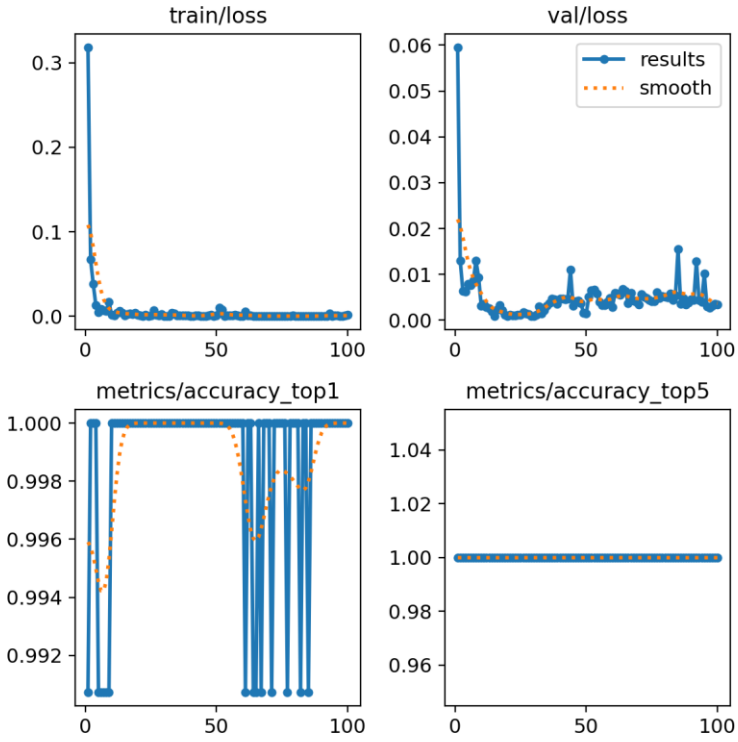

Figure 28 YOLO11x-cls Training and validation performance over epochs

5.3.2.2 TEST PERFORMANCE ON "TRASH TRAIN" DATASET

When evaluated on the "trash train" test set, the model scores 86% accuracy, identical to the ResNet model. Since this outcome produces identical performance metrics and confusion matrix, we kindly refer the reader to the figures in 5.3.1.2 (Figure 24, Figure 25, Figure 26).

5.3.3 UNCLEANLINESS DETECTION WITH YOLO

We next implemented an object detection pipeline using YOLO (You Only Look Once; Redmon et al., 2016). YOLO has become one of the most widely used detectors due to its balance of speed and accuracy. We chose an object detection approach to find out whether the added background invariance often exhibited by models that need to localize and classify often translates to our problem. For our proof of concept, we trained YOLO on the plitter dataset, containing multiple classes of uncleanliness, such as litter, cups, etc., and tested its ability to localize and classify waste in metro images. For comparing the models, we scored the Object Detection model by accepting any image in which the model detected at least one object as unclean.

5.3.3.1 TRAINING ON PLITTER DATASET

We used the Ultralytics Python-API to retrain the pretrained detection model YOLO11x on the plitter dataset. We froze the first 22 layers of the pretrained model and left all other settings at their default values.

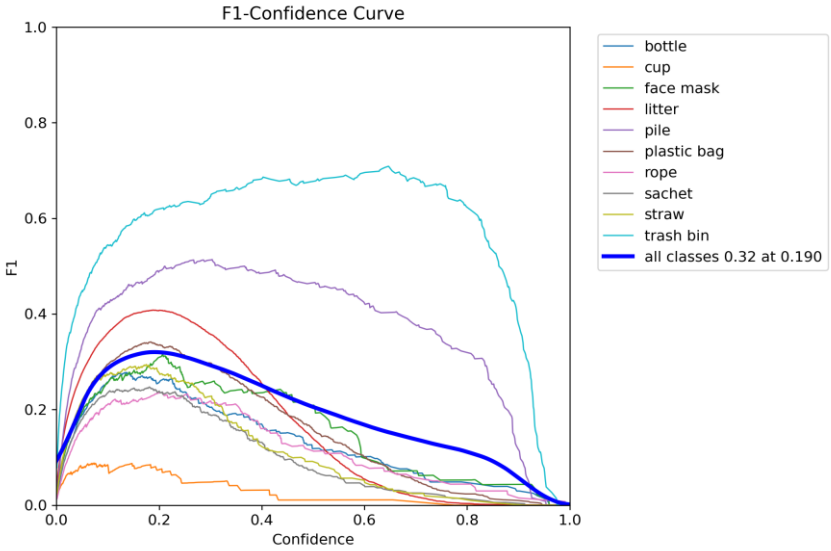

Figure 29 YOLO11x F1-Confidence curve

The **F1–Confidence plot** exhibited undesirable behaviour during training, with the curve failing to reach a stable peak across the confidence threshold range. Ideally, this plot should demonstrate a well-defined peak, indicating the model's ability to balance precision and recall at an optimal confidence level, but this can only be seen for the "trash bin" class. Instead, the curve appeared flat and inconsistent, suggesting that the model struggled to achieve a meaningful trade-off, which undermines its reliability in practical deployment.

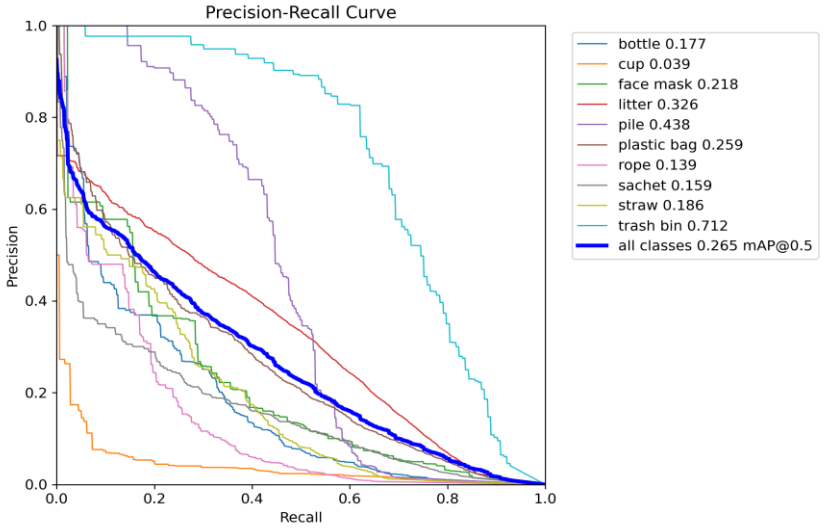

Figure 30 YOLO11x Precision-Recall curve

The **Precision–Recall curve** revealed further shortcomings, as the relationship between the two metrics was neither smooth nor indicative of strong discriminative capability. A desirable curve would trend toward the upper-right corner, like the curve for the “trash bin” class, reflecting both high recall and high precision across thresholds. However, the irregular and low-lying curve implied that the detector frequently misclassified objects, either missing true positives or generating excessive false positives, thus demonstrating poor generalization.

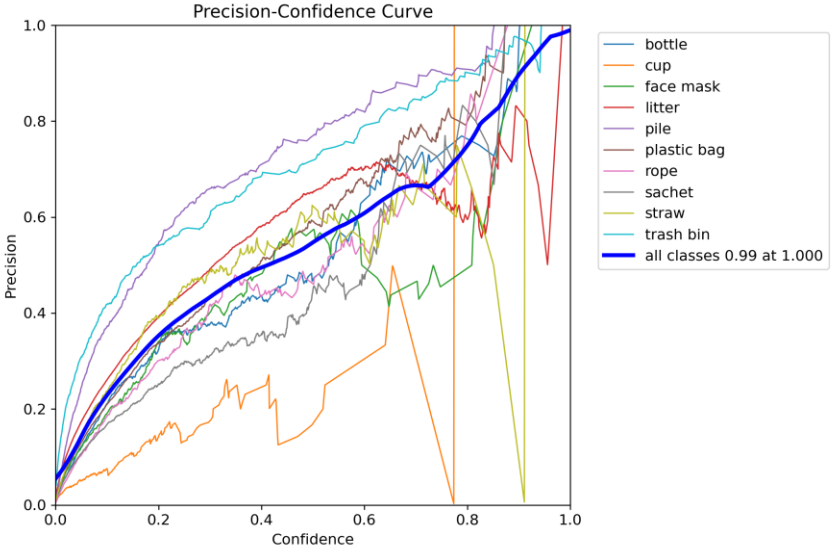

Figure 31 YOLO11x Precision-Confidence curve

The **Precision–Confidence plot** should be (near-)monotonic increasing: as the confidence threshold rises, low-confidence detections are filtered out, false positives usually fall, and precision typically

improves, ideally plateauing at high precision as the threshold becomes slightly stricter. If a plateau is reached, the lower it extends, the better. In our run, precision remained flat and even dipped at around a 0.7 confidence threshold—an undesirable signature consistent with many high-confidence false positives. The confidence threshold controls the FP/FN trade-off; higher thresholds should not generally hurt precision, so a non-increasing curve points to labelling noise, overconfident misclassifications, or class imbalance that the scorer failed to capture.

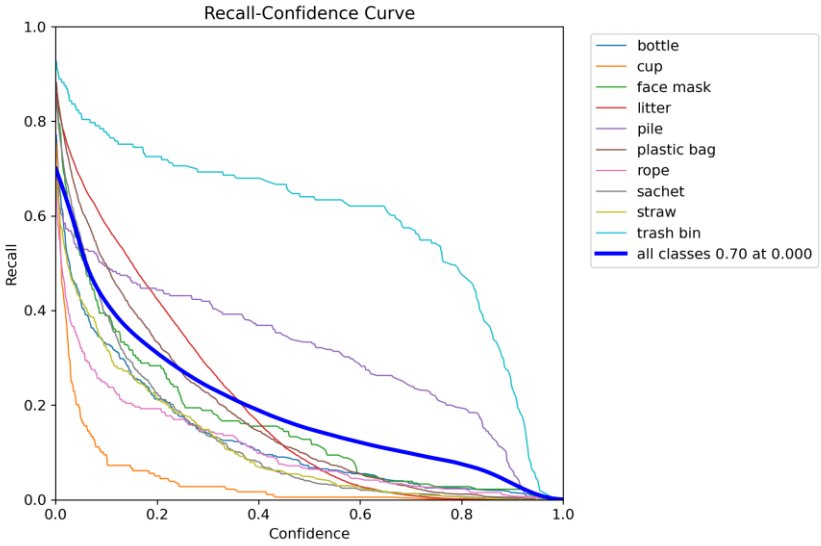

Figure 32 YOLO11x Recall-Confidence curve

The **Recall–Confidence plot** is expected to decrease as the threshold increases: lowering the threshold yields more predicted positives and thus higher recall; while raising it suppresses detections and increases false negatives. Our curve showed recall collapsing sharply. Such behaviour indicates the model is missing many true objects (low sensitivity). Lower confidence thresholds should raise recall, so failure to do so suggests data/anchor mismatch, or inadequate feature learning.

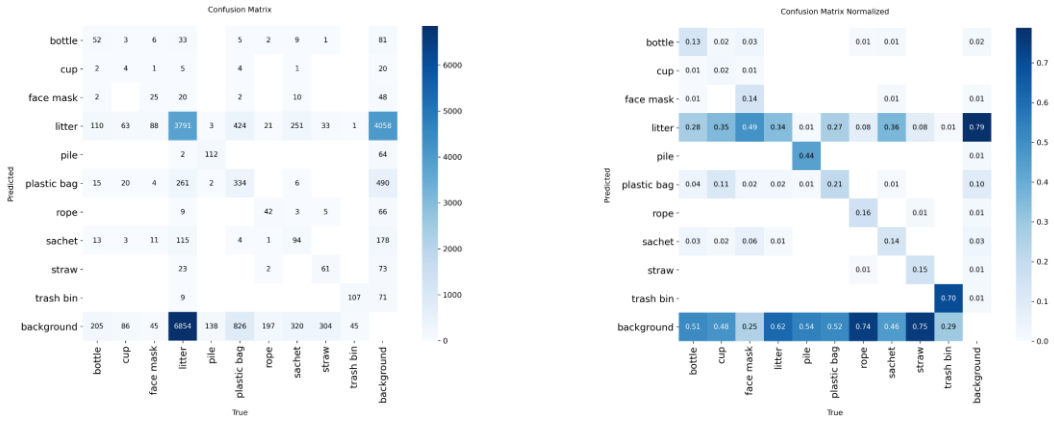

Figure 33 YOLO11x validation confusion matrices. Absolute counts (left) and normalized (right).

In the normalized confusion matrix, we can see that the model misclassifies a lot of classes as litter, e.g. 49% of face masks or 35% of cups, yet 62% of litter instances are misclassified as background. This suggests a strong class imbalance towards litter samples and inconsistent labelling between litter and non-litter classes in the training dataset.

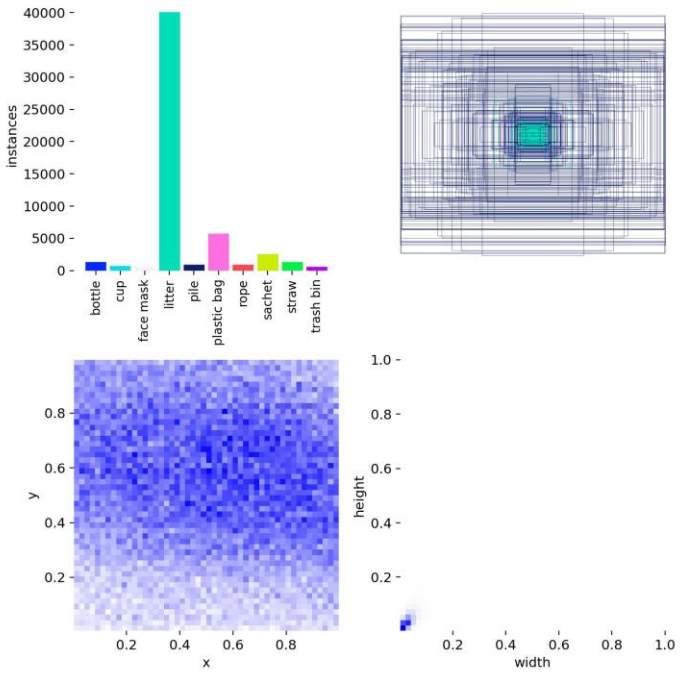

Figure 34 plitter dataset analysis

In the dataset analysis plot, we can see a few difficulties, which explain the undesirable training performance. The dataset is highly imbalanced towards litter instances as can be seen on the top left, explaining poor recall and precision scores. On the top right we can see that while most bounding boxes are small and similar in size, the variance within the dataset is extremely high, with some bounding boxes being as large as the image itself, whereas some bounding boxes are only a few pixels in size, making detecting and classifying them unreliable. The bottom left is the heatmap of bounding box centres, which shows a concentration around a band between 40% and 60% height, but that bias doesn't seem so extreme that it would be a large issue. The bottom right plot visualizes distribution of aspect ratios and sizes of bounding boxes. While they seem appropriate for a litter dataset, it also suggests that litter that has been photographed up-close might be misclassified or missed by the model entirely, due to the model expecting objects to only occupy small parts of the image.

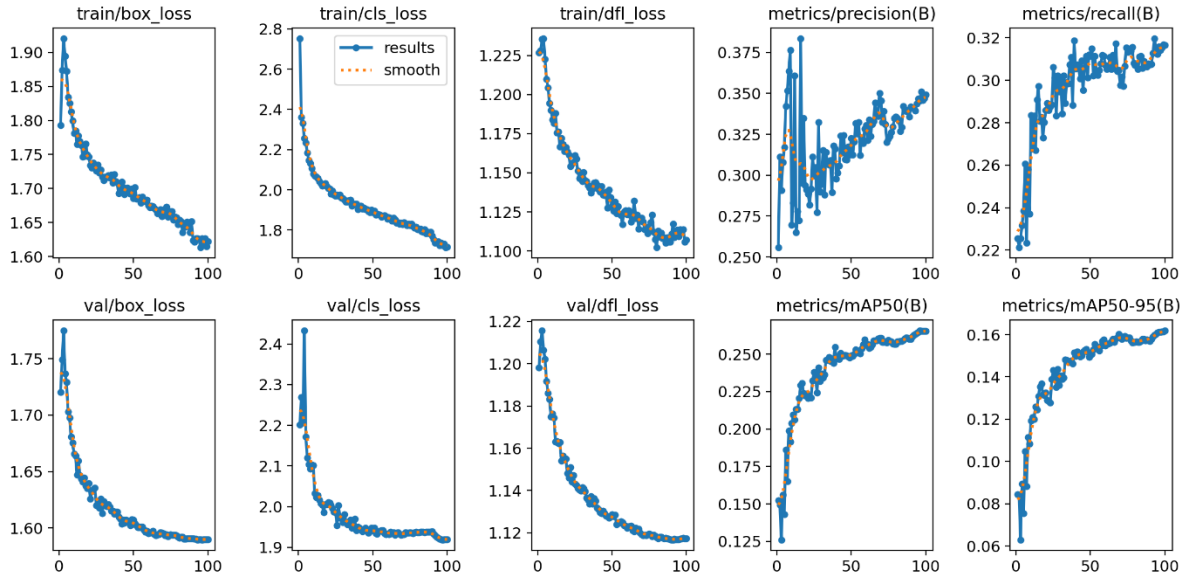

Figure 35 YOLO11x Training and validation performance over epochs

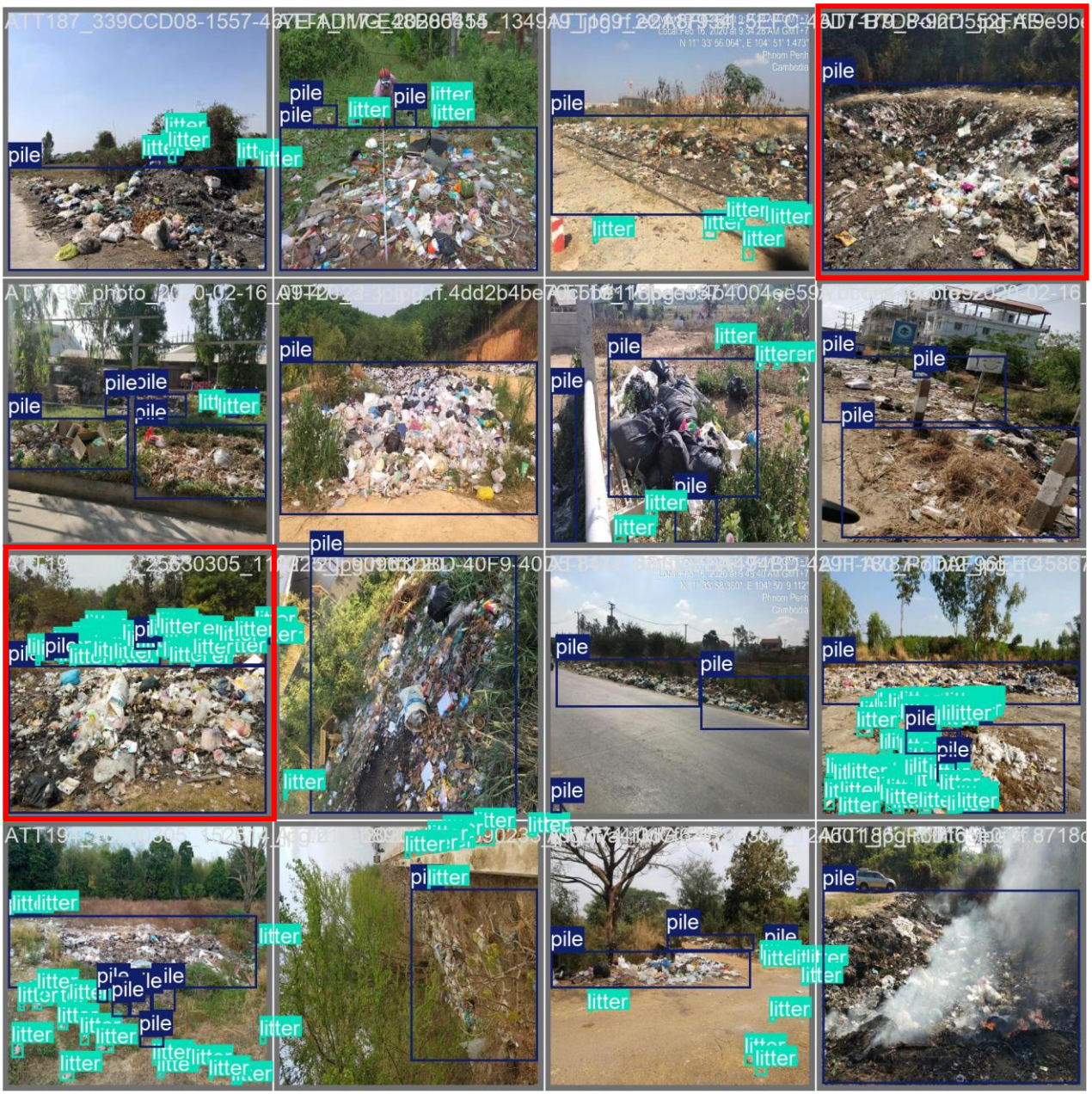

Figure 36 Example Validation Batch labels

Figure 37 Example Validation Batch predictions

By looking at the predicted and true labels of a validation batch in Figure 36 and Figure 37, we can see in images 4 and 9, some issues with the dataset. Some images with piles of garbage are labelled as a single pile, while others have labels of the contents of the pile. The model is punished during training and scoring for being unable to perfectly reproduce the seemingly arbitrary mix of piles of litter and individual litter classifications and instances. We observe that the annotations, especially for the labels pile and litter are very inconsistent, so the scores attained by the model are difficult to interpret. All things considered, the model does not seem to learn the placement and sizes of bounding boxes perfectly, although many of the predictions in the validation set examples above produce rather sensible outputs, e.g. examples 9. Furthermore, the baseline YOLO11x model scores only marginally worse than the Roboflow 3.0 OD (Fast), which scores 35.8% mAP@50, 43.3% Precision and 39.7% Recall on the plitter dataset. This is still not a very high score, which hardens our suspicions about the quality

– or high difficulty - of the training dataset, especially given that the base YOLO11x we use scores a mAP50-95 on the validation set of the famous COCO benchmark.

5.3.3.2 TEST PERFORMANCE ON "TRASH TRAIN" DATASET

When we evaluate the trained YOLO11x model on our "trash train" dataset. If any object is detected in the image, we treat it as an "unclean" classification. The model detects instances of uncleanliness in 22 images, which translates to the following metrics.

Test Set			
TARGET / OUTPUT	Clean	Unclean	SUM
Clean	0 0.00%	28 56.00%	28 0.00% 100.00%
Unclean	0 0.00%	22 44.00%	22 100.00% 0.00%
SUM	0 0.00% 0.00%	50 44.00% 56.00%	22 / 50 44.00% 56.00%

Figure 38 YOLO11x Classification Test Confusion Matrix

Class Name	Precision	1-Precision	Recall	False Negative Rate	F1 score	Specificity (TNR)	False Positive Rate (FPR)
Clean	0.0000	0.0000	0.0000	1.0000	0.0000	1.0000	0.0000
Unclean	0.4400	0.5600	1.0000	0.0000	0.6111	0.0000	1.0000

Figure 39 YOLO11x Classification Test Performance Metrics

Accuracy	0.4400
Misclassification Rate	0.5600
Macro-F1	0.3056
Weighted-F1	0.2689

Figure 40 YOLO11x Classification Test Accuracy Metrics

We believe that the suboptimal zero-shot performance observed when applying YOLO trained on litter to detect uncleanliness in the form of e.g. spills can be attributed to the inherent characteristics of the object detection algorithm. YOLO operates by classifying regions of the image into predefined classes based on learned feature representations. When the algorithm encounters an object or region that does not closely match the features of the classes it has been trained on, such as e.g. spills, it may fail to assign a meaningful class label. In such instances, YOLO typically defaults to classifying the region as background, which our evaluation method deems a misclassification of the image.

This underscores the need for comprehensive training data to enable better generalization to new object categories in object detection tasks.

5.3.3.3 SINGLE CLASS MODE TRAINING

Since our method used to test and evaluate the model does not give any weight to the predicted classes of the detected objects, we repeat our experiment by merging all class labels into a single class. This is akin to setting the object misclassification weight to 0% and the instance detection weight to 100% and is still suitable for our evaluation method on the test set, as any kind of instance detected lets us classify the entire image as unclean.

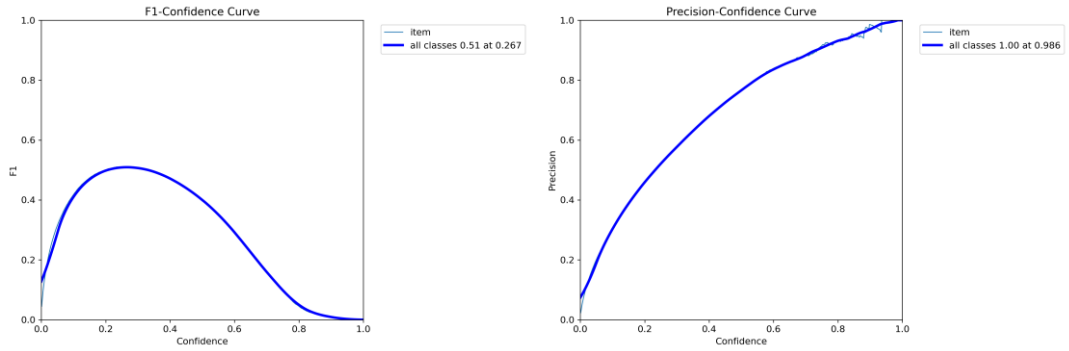

Figure 41 YOLO11x merged-classes F1-confidence (left) and Precision-Confidence (right) curves

Immediately we can see that evaluation metrics like the F1-confidence and precision-confidence (Figure 33) curves have shown noticeable improvement. This suggests that the model is now more decisive in identifying objects and better calibrated in its confidence scores, at least in terms of reducing false positives. By simplifying the classification task, the model can focus more on the object localization aspect without the added complexity of distinguishing between multiple classes.

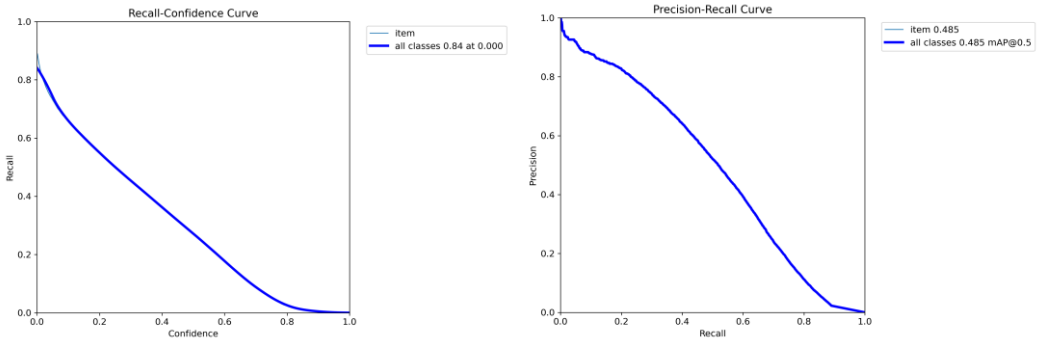

Figure 42 YOLO11x merged-classes Recall-confidence (left) and Precision-Recall (right) curves

However, despite the improved F1 and precision confidence metrics, the recall-confidence and recall-precision curves remain poor—largely due to the arbitrary nature of the bounding boxes in the dataset. The visible improvement in the Precision-Recall curve can be explained by the improvement of the precision in the single class setting, but since many of the ground truth annotations are seemingly inconsistent or imprecise, it is difficult for the model to learn reliable localization patterns. As a result, even correctly identified objects may not count as true positives under standard IoU thresholds, dragging down recall. This highlights a limitation in the dataset itself rather than the model, suggesting that further progress may require cleaner, more carefully curated dataset.

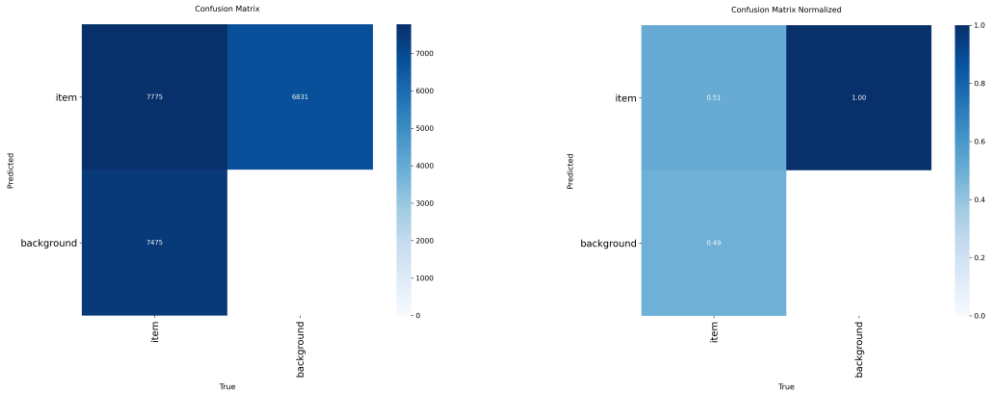

Figure 43 YOLO11x merged-class validation confusion matrices

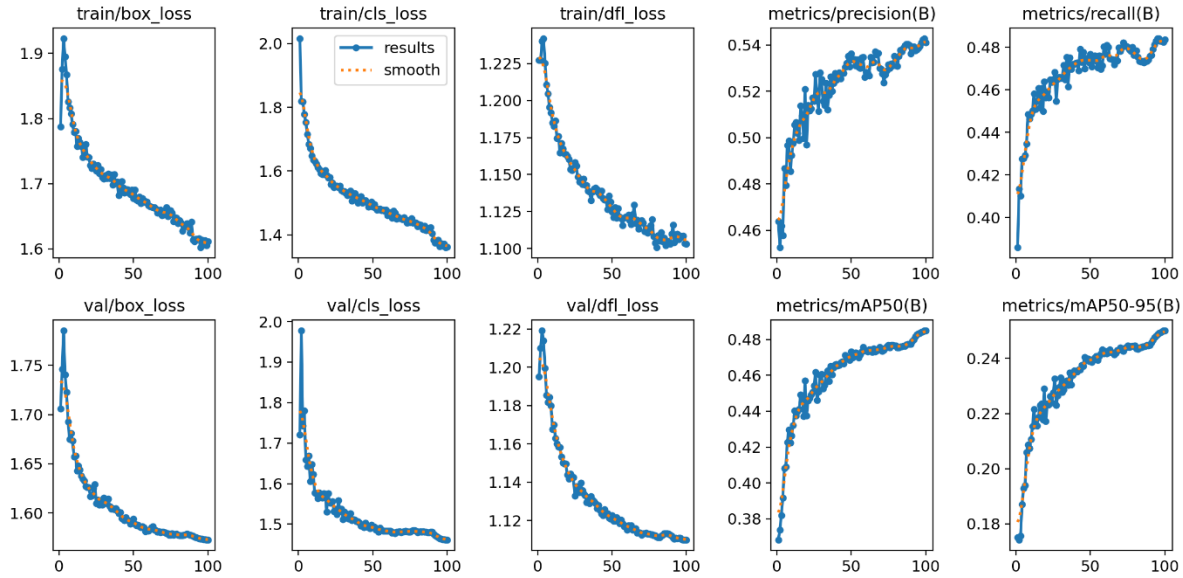

Figure 44 YOLO11x merged-class Training and validation performance over epochs

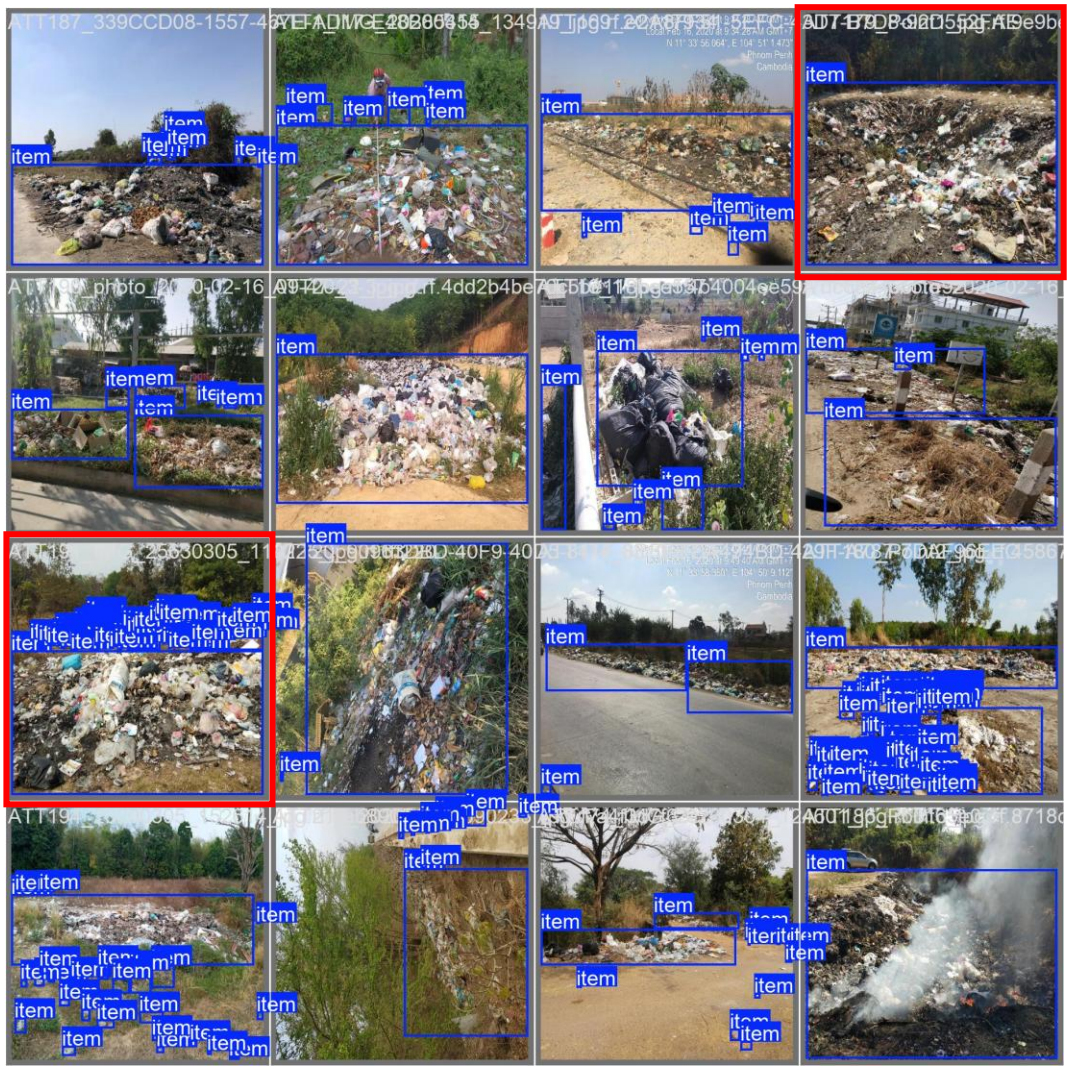

Figure 45 Example Validation Batch merged-class labels

Figure 46 Example Validation Batch merged-class predictions

Similar to the multiclass object detection model, the single class object detection predicts bounding boxes that would make more sense to a human observer but still differ significantly from the ground truth. The comparison between the aforementioned images 4 and 9 again highlights particularly bad offenders.

5.3.3.4 SINGLE CLASS MODE EVALUATION

We evaluated the trained object detection model in the same manner as described in 5.3.3.2, any detection of an object in a test set marks the image as classified unclean. The metrics below show the results of diluting the label, by merging the different uncleanliness classes into a single category. This broadens the range of objects the model considers, which leads to a slight increase in the test set score.

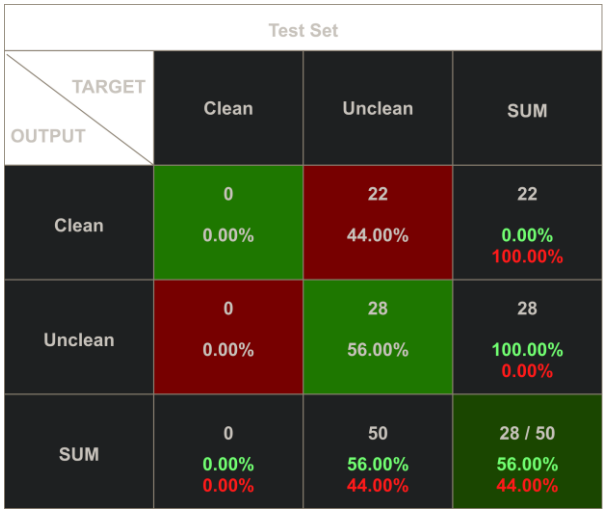

Figure 47 YOLO11x merged-class Classification Test Confusion Matrix

Class Name	Precision	1-Precision	Recall	False Negative Rate	F1 score	Specificity (TNR)	False Positive Rate (FPR)
Clean	0.0000	0.0000	0.0000	1.0000	0.0000	1.0000	0.0000
Unclean	0.5600	0.4400	1.0000	0.0000	0.7179	0.0000	1.0000

Figure 48 YOLO11x merged-class Classification Test Performance Metrics

Accuracy	0.5600
Misclassification Rate	0.4400
Macro-F1	0.3590
Weighted-F1	0.4021

Figure 49 YOLO11x merged-class Classification Test Accuracy Metrics

5.4 CHALLENGES & LESSONS LEARNED

5.4.1 WORKING WITH LIMITED DATA

A central challenge of this project was the **scarcity of domain-specific data**. At the outset, there was essentially no annotated dataset of "dirty" metro environments available for training. The only testing material we had was a small but topical set of **50 images from a particularly unclean New York City metro station**. Despite the limited scale and extremity of this dataset, it provided a valuable benchmark for evaluating the transferability of public models to real-world conditions.

5.4.2 TRANSFER LEARNING FROM PUBLIC LITTER DATASETS

Given the lack of in-domain training data, we turned to **public litter datasets such as plitter**. While these datasets mostly feature outdoor waste in daylight conditions, they provided a strong starting point for **transfer learning with pre-trained architectures**. With minimal effort in data augmentation, preprocessing, or image curation, both classification and object detection models achieved encouraging performance on the metro test set.

- **Classification (ResNet-50):** The classifier was able to correctly identify **43 out of 50 images** as clean or unclean, showing that a standard architecture fine-tuned on the merged datasets 1&2 can generalize surprisingly well, even under severe domain shift.
- **Object Detection (YOLO):** YOLO demonstrated strong competence on object categories represented in plitter, accurately localizing items such as bottles, cups, and wrappers in the metro scenes. Cases where the detector "ignored" anomalies such as spills or stains were not failures per se, but a reflection of the dataset's labelling scope — in fact, this behaviour highlights the model's strong alignment with its training objectives.

5.4.3 INSIGHTS ON DOMAIN ADAPTATION

The project underscored how **domain differences** (urban vs. metro settings) shape model performance. Importantly, even without tailored preprocessing or targeted image selection and dataset curation, transfer learning from public datasets produced **reasonable and interpretable results** in the zero-shot experiments with the "trash train" dataset. This demonstrates the maturity of existing architectures and methods and opens the door to further improvements as the dataset grows and improves. Furthermore, the strong out-of-the-box performance of the models enables efficient dataset creation methods like model assisted labelling or human-in-the-loop training.

5.4.4 LESSONS LEARNED

Several valuable lessons emerged from this exercise:

1. **Transfer learning works.** Even with a small, extreme test set and no specialized augmentation, pre-trained models delivered meaningful results in the zero-shot setting.
2. **Classification is a strong baseline.** A simple ResNet-50 classifier proved robust, offering a reliable first-pass assessment of cleanliness with minimal tuning.

3. **Detection models follow their data.** YOLO's behaviour on metro images highlighted the importance of dataset coverage, but also its strength in faithfully replicating the categories it was trained on.

5.5 EVALUATION

5.5.1 EVALUATION OF CLASSIFICATION MODELS

The ResNet-50 architecture, fine-tuned on a merged subset of public litter datasets, demonstrated strong baseline performance. When tested on the "Trash Train" dataset, consisting of 50 real-world metro images, the classifier achieved an accuracy of 86%. Misclassifications were concentrated in cases where uncleanliness was not represented by distinct litter objects (e.g., stains, spills), or where image quality was poor.

From an evaluation perspective, two key observations emerged:

1. **Robustness under domain shift** – Despite being trained primarily on outdoor litter in daylight conditions, the ResNet classifier generalized reasonably well to indoor metro environments, which are harsher and less structured. This indicates that the underlying feature representations learned by ResNet are transferable, even with minimal fine-tuning.
2. **Limitations of binary classification** – The model was effective in identifying prominent litter items but struggled with subtle or amorphous forms of dirt. A binary clean/unclean label may be too coarse for real-world application, where gradations of cleanliness and diverse sources of uncleanliness are important.

Taken together, the evaluation suggests that classification models are a strong first baseline but lack sufficient granularity to fully address the problem domain.

5.5.2 EVALUATION OF YOLO-BASED CLASSIFICATION

To compare architectures, a YOLO11x-based classifier was trained under similar conditions. On the public training data, the YOLO classifier achieved 100% accuracy within a few epochs, reflecting the simplicity of the binary classification task. When evaluated on the "Trash Train" dataset, YOLO achieved 86% accuracy—identical to ResNet.

This parity in performance indicates that the architecture itself is not the primary bottleneck. Rather, the scarcity and lack of diversity and specificity of training data remain the limiting factor. Both ResNet and YOLO converge towards similar levels of generalization when applied to out-of-domain test data, reinforcing the need for richer datasets.

5.5.3 EVALUATION OF YOLO OBJECT DETECTION

YOLO was also employed for object detection, offering the advantage of localizing specific instances of uncleanliness. Training on the *plitter* dataset yielded mixed results.

5.5.3.1 PERFORMANCE ON TRAINING DATA

The evaluation metrics during training (F1–Confidence, Precision–Recall, and related curves) highlighted several shortcomings:

- Precision–Recall curves were irregular and low, pointing to frequent misclassifications.
- Confidence–based metrics revealed non-monotonic behaviour, suggesting noisy labels and overconfident false positives.
- The confusion matrix showed significant misclassifications, with many classes mislabelled as "litter" and a high proportion of litter misclassified as background.

These results reflect dataset-level issues, including strong class imbalance, inconsistent annotation of bounding boxes, and variability in object size and scale.

5.5.3.2 PERFORMANCE ON METRO DATA

On the "Trash Train" dataset, YOLO detected uncleanliness in 22 of 50 images. However, detections were largely constrained to object categories represented in training data (e.g., bottles, wrappers), with amorphous anomalies like spills consistently missed. When trained in *single-class mode* (merging all categories into one), evaluation curves improved, particularly for precision. Nevertheless, recall remained low due to inconsistent annotations and noisy ground truth, though performance on the "Trash Train" dataset slightly improved to 28 out of 50 images.

Overall, while YOLO demonstrated the capability to highlight and localize relevant objects, its generalization to unrepresented categories of uncleanliness was poor. These findings suggest that object detection is promising but dataset quality is the key limiting factor.

5.5.4 CHALLENGES IN DATASET QUALITY

A recurring theme in the evaluation is the pivotal role of dataset quality. Several challenges were identified:

- **Limited availability of metro-specific data**: No large-scale annotated dataset of metro uncleanliness exists, restricting evaluation to a small 50-image set.
- **Class imbalance and annotation inconsistency**: Public datasets such as *plitter* exhibited skewed distributions (e.g., overrepresentation of litter vs. other waste types) and inconsistent annotation policies.
- **Mismatch of object categories**: Amorphous or contextual uncleanliness (stains, grime, liquids) is absent from training data, limiting detection capabilities.

These dataset challenges directly impacted evaluation outcomes, producing models that performed strongly on curated benchmarks but less reliably under real-world conditions.

5.5.5 REPRODUCIBILITY & ARTEFACTS

Our experiments are set up as a python-based monorepo. Dependencies and binary are managed byt he project manager uv, and the experiments were conducted inside docker containers on GPU enabled gitlab-runners. The recipes on how to run the experiments are present in the standard automation files. Our code does include some internal libraries for gathering the data for the experiment, however these dependencies can be simply removed and the data put in place by any other method.

Artefacts Available:

- **Source Code:** The Python codebase is available in a git repository shared with project partners.

Steps to Reproduce:

- If a (gpu-enabled) gitlab runner is available, as that's necessary is to push the code to a gitlab repo. Otherwise, the code can be run directly using uv, both inside and outside docker containers. The exact docker image used is documented in the gitlab-ci declaration file.
- The results can be verified using the produced evaluation metrics and plots.

5.5.6 CROSS-DEMO KPI TABLE AND BASELINE COMPARISONS

Due to the very constrained dataset, only simple baseline comparisons for the models can be provided.

Table 11 CROSS-DEMO KPI TABLE AND BASELINE COMPARISONS

Metric Type	Model Type	Baseline (uniform random)	Model performance
Accuracy	Transfer-learned ResNet	50%	86%
Accuracy	YOLO11x-od (merged class)	50%	56%
Accuracy	YOLO11x-cls	50%	86%

5.5.7 OVERALL EVALUATION

The evaluation demonstrates that automated uncleanliness detection is feasible but strongly data-dependent. Pretrained architectures, when fine-tuned on limited litter datasets, achieved respectable performance even under domain shift. However, generalization beyond object-based litter remains an open challenge. Improving dataset quality, expanding to metro-specific imagery, and exploring anomaly detection will be critical to advancing the robustness and applicability of the system in real-world deployments.

Despite current limitations, the models already demonstrate promising performance that could possibly enable **semi-automated, human-in-the-loop dataset creation**. This makes labelling for new, metro-specific datasets more efficient and feasible, and sets the stage for continuous model improvement as additional data becomes available.

6 ASPECTS OF TRUST AND ACCEPTANCE BY USING AI

6.1 MOTIVATION

The widespread integration of artificial intelligence (AI) into information systems (IS) represents a transformative shift in technology, bringing with it both opportunities and complex challenges for user adoption. While classical models of technology acceptance, such as the Technology Acceptance Model (TAM) and the Unified Theory of Acceptance and Use of Technology (UTAUT), provide a valuable theoretical foundation, their constructs are often insufficient for the unique characteristics of AI—namely, its opacity, unpredictability, and dynamic nature.

Trust in an AI algorithm is not only a consequence of its perceived usefulness or ease of use, but a critical prerequisite for adoption. The report addresses the psychological foundations of this trust and examines how factors such as outcome feedback and accountability and the explainability of a system influence user behaviour. It further examines the workplace impact of AI, distinguishing between systems that assist and those that paternalize. The latter, while often well-intentioned, proves ethically problematic and a reliable path to project failure. To guide developers, the report concludes by outlining the technical and ethical foundations of trustworthy AI, providing a practical framework for building systems that are not only powerful, but also fair, transparent, and centred on human users.

6.1.1 THE EVOLVING LANDSCAPE OF INFORMATION SYSTEMS

For decades, information systems were based on predictable, rule-based logic. A traditional IS was a traceable tool; its internal workings were clear, and its output was a direct result of its programming. The challenge for user adoption was its function and efficiency. However, AI, particularly with machine learning models, has introduced a new paradigm. These algorithms learn from data, creating complex internal structures that even their creators may not fully understand. This "black-box problem" means AI can make decisions without clear reasoning, bringing a new level of opacity and uncertainty. This shift requires a re-evaluation of how users interact with and accept technology. The unpredictability and lack of transparency in AI are the central issues that require a new approach to trust.

6.1.2 SCOPE AND DEFINITIONS

For clarity and consistency, this report will use the following formal definitions:

- **User Acceptance:** The behavioural intention and subsequent adoption of a new technology system by an individual or organization.
- **User Trust:** A user's belief in the reliability, competence, and integrity of a system or its output.
- **Information System (IS):** A structured system for the collection, storage, and processing of data, designed to support organizational or individual tasks.

- **AI Algorithm:** A component of an IS that uses machine learning, natural language processing, or other techniques to perform tasks that typically require human intelligence, often exhibiting learning and adaptation.

6.2 FOUNDATIONAL MODELS OF TECHNOLOGY ACCEPTANCE

6.2.1 THE TECHNOLOGY ACCEPTANCE MODEL (TAM): A BEHAVIORAL FOUNDATION

Developed by Fred Davis in the 1980s, the Technology Acceptance Model (TAM) emerged to address why new systems, despite their technical promise, frequently failed to gain user traction (Davis, 1989). TAM posits that a user's motivation to adopt a new technology is primarily influenced by two cognitive factors:

1. **Perceived Usefulness (PU):** The degree to which a person believes that using a particular system will enhance their job performance or help them accomplish desired tasks.
2. **Perceived Ease of Use (PEOU):** The degree to which a person believes that using a particular system will be free of effort.

According to TAM, these two beliefs shape a user's attitude toward using the technology, which, in turn, influences their behavioural intention and, ultimately, their actual use. The model (Figure 50) streamlined the Theory of Reasoned Action (TRA), a more general psychological theory, to create a parsimonious framework specifically for the context of technology adoption. Later extensions, such as TAM2, incorporated additional factors like "subjective norm," which refers to the influence of social pressure from peers or supervisors on technology acceptance (Venkatesh & Davis, 2000).

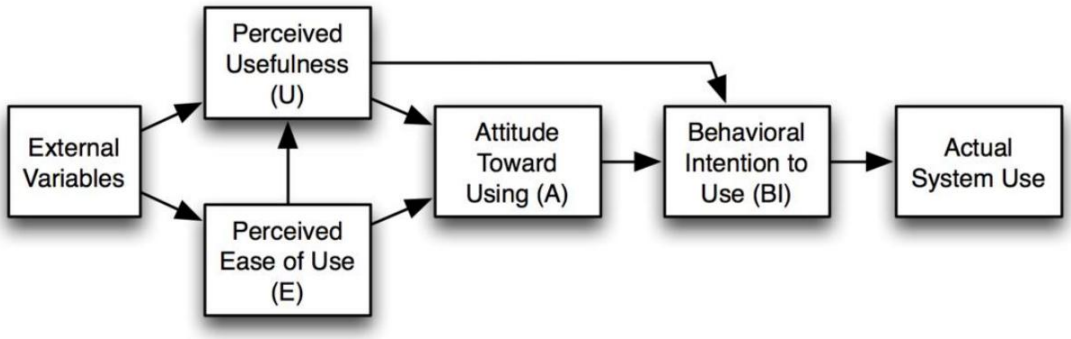

Figure 50 The basic Technology Acceptance Model (Davis 1989)

Despite its widespread use, TAM has notable limitations in the context of contemporary technology. The model's individual-centric perspective, limited focus on social and contextual factors, and static nature make it less relevant for technologies that learn and evolve. TAM assumes that all users will behave in a similar way and does not account for individual differences in personality or experience,

nor does it provide a framework for the complex social dynamics or cultural factors that can influence adoption.

6.2.2 THE UNIFIED THEORY OF ACCEPTANCE AND USE OF TECHNOLOGY (UTAUT): A UNIFIED VIEW

The Unified Theory of Acceptance and Use of Technology (UTAUT) was developed to synthesize the constructs of eight prior models, including TAM, into a single, comprehensive framework (Venkatesh, Morris, Davis, & Davis, 2003). This model aimed to provide a more holistic understanding of user intentions and subsequent usage behaviour in an organizational context. The theory is built on four key determinants of behavioural intention and use:

1. **Performance Expectancy:** The extent to which an individual believes that using the system will help them achieve gains in job performance. This is the strongest predictor of use intention.
2. **Effort Expectancy:** The degree of ease associated with the use of the system.
3. **Social Influence:** The degree to which an individual perceives that important others believe they should use the new system.[8] This factor is particularly significant in mandatory work environments.
4. **Facilitating Conditions:** The belief that the necessary organizational and technical infrastructure exists to support the use of the system.

UTAUT further refines its predictive power by incorporating moderating factors such as age, gender, experience, and the voluntariness of use. A longitudinal study found that UTAUT (Venkatesh, V., & Davis, F. D., 2000). could account for a significant portion of the variance in a user's intention to use a system (70%) and actual usage behaviour (50%). However, the model has been critiqued for its complexity, with one analysis noting that it presents a large number of independent variables to predict user intention and behaviour.

6.2.3 THE DELONE AND MCLEAN IS SUCCESS MODEL: A MULTIDIMENSIONAL FRAMEWORK

The DeLone and McLean IS Success Model offers a multidimensional framework for evaluating the overall success of an information system (DeLone & McLean, 1992). The model, first proposed in 1992 and refined a decade later, is considered one of the most influential theories in IS research and is widely used to evaluate systems across various industries. It is composed of six interrelated dimensions:

1. **Information Quality:** The quality of the system's output, including its accuracy, reliability, and trustworthiness.
2. **System Quality:** The overall quality of the system itself, including its ease of use, functionality, and flexibility.
3. **Service Quality:** The quality of the support provided by the IS developer or support department, such as technical support and user training.
4. **Use/Usage Intentions:** The manner in which the system is used or the user's intention to use it.
5. **User Satisfaction:** The extent to which a user is pleased or content with the system.

6. **Net Benefits:** The extent to which the IS contributes to the success of its users or the underlying organization, whether in a positive or negative way.

While the model provides a comprehensive, holistic view of success, it has several critical limitations, particularly in the context of AI. The framework relies heavily on self-reported data, which can be subject to bias, and does not account for the dynamic nature of systems that change over time. Most importantly, it lacks a specific framework for analysing the impact of technology, such as AI, that can fundamentally alter the context in which a system operates and the psychological and social dynamics involved.

6.3 RECONCEPTUALIZING ACCEPTANCE AND TRUST IN THE AGE OF AI

6.3.1 THE AI-SPCIFIC CHALLENGE: THE "BLACK BOX" AND NEW DIMENSIONS OF UNCERTAINTY

The foundational models of technology acceptance were developed for a world of predictable, rule-based software. The black-box problem inherent in many AI systems, however, fundamentally changes the dynamic between user and machine (Zen, 2024; Blattner, 2025). Unlike traditional systems, which have a traceable, linear logic, deep learning models operate in a way that is too complex for human comprehension (Zen, 2024). This opacity directly challenges classical notions of "System Quality" and "Information Quality" where reliability and trustworthiness are typically assumed to be verifiable.

This challenge has led to the emergence of trustworthiness as a core construct. In traditional models like TAM and UTAUT, trust is not a central, independent variable; rather, it is an implicit byproduct of a system's perceived ease of use and usefulness. A user first evaluates a system's quality and utility, and if these are high, they may implicitly trust it. However, with AI's inherent opacity, a user must have a pre-existing level of trust in the algorithm's capabilities and its underlying logic, even if unseen, before they can even begin to evaluate its perceived usefulness or ease of use. This makes trust a prerequisite for, rather than a consequence of, the initial acceptance process. This new emphasis is supported by a range of research that identifies trustworthiness, along with transparency and explainability, as an essential building block for AI (AIHLEG, 2019; Li, Wu, Huang, & Luan, 2024; Tranberg, n.d.).

6.3.2 ADAPTING CLASSICAL CONSTRUCTS FOR AI SYSTEMS

Classical constructs can be reinterpreted to accommodate the unique characteristics of AI. The concept of **Perceived Usefulness**, for instance, now encompasses a user's belief in the AI's ability to learn and adapt, not just perform a fixed function. New concepts have been introduced, such as the belief that AI expands an individual's capabilities (growth) and does not impede them (non-deskilling) (Davis, 1989; Venkatesh & Davis, 2000; Kelly et al., 2023).

The multidimensional framework of the DeLone and McLean IS Success Model can also be adapted to account for AI-specific challenges. A comparative analysis demonstrates how the core dimensions of IS success are challenged and redefined by the introduction of AI.

Table 12 Mapping Classical IS Success Constructs to AI System Dimensions

Classical IS Success Construct	AI Reinterpretation/Challenge	Relevant Source Information
Information Quality	Shifts from the accuracy, timeliness, and reliability of static data to the accuracy and trustworthiness of a dynamic, learning model's output. The presence of inaccurate or limited knowledge in the AI's responses can directly erode this dimension.	(DeLone & McLean, 2003; McLean, 2003)
System Quality	Moves beyond basic functionality and features to encompass technical robustness, security, and resilience. The system must not only work but must do so in a way that is dependable and safe from malicious or unforeseen inputs.	(DeLone & McLean, 2003; AIHLEG, 2019)
Use/Usage Intentions	While still a measure of a user's adoption, this construct is now heavily influenced by the psychological factors of trust, transparency, and the perceived risk of using the system.	(Hosanagar, Ahn, & Almaatouq, n.d.; Li et al., 2024)
User Satisfaction	Is now tied to new cognitive constructs like "non-deskilling" and the feeling that the AI augments rather than replaces one's capabilities, leading to higher cognitive job satisfaction.	(DeLone & McLean, 2003; EU CORDIS, 2014)

This table (Table 12) illustrates that the core principles remain relevant, but their measurement and meaning must be expanded. A system that produces information that is simply accurate is no longer sufficient; its information must also be trustworthy, especially given the black-box nature of the underlying algorithms. Similarly, user satisfaction with an AI system depends not only on its ease of use but also on the user's perception that it is empowering them rather than making their skills obsolete.

6.4 THE PSYCHOLOGY OF TRUST AND BIAS IN AI ADOPTION

6.4.1 THE FOUNDATIONS OF HUMAN-AI TRUST

Research has shown that a user's trust in an AI system is more driven by **outcome feedback** than by **interpretability**. A study found that participants built trust over time based on whether following the AI's advice helped or hurt their performance on recent predictions. This challenges the idea that transparency is the main driver of trust. It suggests that while explainable AI (XAI) is essential for technical purposes, it may not be the primary psychological lever for user acceptance and hence adoption (Hosanagar et al., n.d.; Juniper.net, n.d.; Li et al., 2024). The user's focus is on the practical result ("Is this useful?") rather than the technical process ("Can I explain this?"). This dynamic can differ for experts and non-experts. Experts, with their deep domain knowledge and scepticism, may require more detailed explanations to satisfy their need to dig deeper into the AI's logic. In contrast, non-experts

may find complex explanations confusing and instead prioritize an AI that provides clear, simple feedback that demonstrates its value (Hosanagar et al., n.d.).

Another key psychological factor is **anthropomorphism**, the tendency to ascribe human-like qualities to non-human entities. Research indicates that systems perceived as more human-like, particularly those exhibiting positive emotions, are more likely to be trusted by individuals. However, this trust can be fragile. A user's trust is also contingent on a system's **accountability**. People need assurance that a clear process exists to handle AI-related issues and that specific parties, such as developers, can be held responsible for errors or harm. This is a crucial point, as people tend to perceive that robots have poorer controllability over tasks and may show less moral outrage toward algorithmic discrimination compared to human discrimination, though the company itself may be evaluated poorly (Li et al., 2024; Qlik.com, n.d.).

6.4.2 NAVIGATING USER BIASES: AUTOMATION AVERSION AND ALGORITHM BIAS

The relationship between humans and AI is often influenced by two opposing behavioural tendencies: **automation bias** and **algorithm aversion** (Tranberg, n.d.; Ajibade, 2018). Automation bias refers to an over-reliance on AI, a "blind preference for automated responses over manual sources of information". A classic example is blindly following a GPS unit despite manual signs that indicate a wrong turn. Conversely, algorithm aversion is a preference for manual responses, a distrust of automated systems even when it is clear that they are more accurate and reliable than human judgment. This bias often becomes more pronounced as the stakes of a decision rise.

The challenge for developers and designers is that the remedies for one bias can often exacerbate the other. For example, a recommended strategy to reduce automation bias is to highlight errors and emphasize the system's supportive rather than directive nature (Ajibade, 2018). However, this same practice can increase algorithm aversion, as users are quick to lose confidence in a machine when it visibly errs. A system that appears fallible may cause a user to over-correct and distrust the system entirely. This situation creates a delicate and difficult balancing act. A nuanced approach is required, where the system is transparent about its limits without being so overtly fallible that it erodes all user confidence. This "bias mitigation paradox" suggests that a single design choice can have opposing effects on different user biases, requiring a thoughtful approach to feedback and transparency.

6.5 THE IMPACT OF AI ON THE WORK ENVIRONMENT AND EMPLOYEE WELL-BEING

6.5.1 FROM TASK AUTOMATION TO JOB RE-SKILLING

AI is fundamentally changing the nature of work (McKinsey, 2025). The primary impact is not just the automation of low-value, repetitive tasks like data entry and invoicing but a shift toward freeing employees to engage in more creative and strategic activities (McKinsey, 2025). By automating cognitive functions like summarizing, coding, and reasoning, AI can lower skill barriers and enable a state of "superagency," where people and machines work together to increase personal productivity and creativity. For instance, AI-powered tools can handle routine processes in HR, finance, and

customer service, allowing human employees to focus on more complex, valuable work. This promises a future where human expertise is augmented, not replaced.

6.5.2 THE PATERNALISM-ASSISTANCE CONTINUUM

The way in which AI is implemented in the workplace determines its impact on employee well-being. AI systems can be designed along a continuum, with one end representing an **assistance model** and the other representing **AI paternalism**. AI paternalism is defined as the use of an AI system to make decisions or take actions on behalf of individuals without their input or consent (Khalpey-ai.com, 2024). This occurs when an AI is designed to prioritize certain values that may not align with the user's preferences, leading to a feeling of lost control and a lack of agency. An assistance model, by contrast, empowers and augments the user's capabilities, leaving the final decision and control in their hands (Khalpey-ai.com, 2024).

The railway industry provides a clear example of this continuum through the Grades of Automation (GoA) for trains (Ditzel, 2025). In low automation levels (GoA 0-1), a human operator is in full control, with the AI providing minimal or no support. At GoA 2, the AI takes on more of an **assistance model**, handling tasks like accelerating and stopping while the driver remains present to take over, monitor for obstructions, and respond to emergencies (Ditzel, 2025). This model augments the driver's capabilities without diminishing their authority. However, as automation increases to GoA 3 (driverless operation with an attendant) and GoA 4 (unattended operation), the system's role shifts from assistant to controller, eroding the human's "domain-specific autonomy" in making informed judgments (Ditzel, 2025; Min et al., 2025). A paternalistic model in this context, where AI assumes full control of high-stakes tasks like traffic management or safety without meaningful human oversight, can lead to distrust among experts and the public (Ditzel, 2025; Global Railway Review, 2025). This demonstrates that a paternalistic design model is not only ethically questionable but is also a direct path to user rejection and project failure.

6.5.3 PRESERVING JOB DIMENSIONS

To avoid the pitfalls of paternalism and foster a positive, human-centric work environment, it is essential to design AI systems that preserve or enhance key job dimensions (Table 13). The European-funded **FACTS4WORKERS** project provides a tangible example of a successful human-centric approach. The project's objective was to create "Factories for Workers" by empowering workers on the shop floor with smart factory infrastructure (EU CORDIS, 2014; EU EFFRA, 2014). The solutions were designed to support the inclusion of knowledge work, increase problem-solving and innovation skills, and lead to increased cognitive job satisfaction and productivity. By focusing on worker-centric solutions, the project directly contrasts the paternalistic model and provides a clear, actionable path for implementation. For example, the project's measurable indicators included increasing problem-solving and innovation skills of workers, as well as increasing cognitive job satisfaction (EU EFFRA, 2014).

Table 13 Impact of AI on work environment related to trust and acceptance

Dimension of Work Environment	Potential Positive Impacts of AI	Potential Negative Impacts of AI	Factors Influencing Impact
User Autonomy and Control	Automation of routine tasks, freeing up time for complex work; provision of self-service platforms; decision-making assistance	Algorithmic management reducing control over tasks and pace; potential for deskilling due to over-reliance; feeling of reduced agency	Design of AI systems (human-in-the-loop vs. fully autonomous); organizational culture and management approaches; user training and support
Task Variety	Automation of mundane tasks, allowing focus on more engaging work; potential for new roles and responsibilities	Feeling of reduced meaning and purpose if creative aspects are automated; potential for job displacement leading to lack of variety	Nature of the job and tasks; extent of AI integration; opportunities for reskilling and upskilling
Perceived Competence	Enhanced productivity and efficiency; access to vast amounts of information and insights; potential for skill development through AI tools	Over-reliance on AI potentially eroding critical thinking skills; feelings of inadequacy compared to AI capabilities	User confidence in own skills vs. confidence in AI; complexity and transparency of AI systems; opportunities for learning and growth

6.6 TECHNICAL AND ETHICAL PREREQUISITES FOR TRUSTWORTHY SYSTEMS

6.6.1 THE PILLARS OF TRUSTWORTHY AI

The responsibility for building trustworthy AI falls to the developers and organizations that create them. It is critical to distinguish between **trustworthiness** and **trust**. Trustworthiness is a technical and ethical concept that serves as a prerequisite for trust; it is the developer's responsibility to build a system that is, by design, lawful, ethical, and robust. This is achieved by adhering to a set of core technical and ethical pillars that ensure the system's internal integrity, such as fairness and transparency (AIHLEG, 2019). Trust, however, is a much broader psychological, social, and contextual concept. It is the user's belief and confidence in the system, which is influenced by the system's trustworthiness but is also shaped by factors beyond the developer's direct control. These factors include the changes in work processes, the social dynamics of the work environment, and other contextual circumstances. A system can be technically trustworthy but fail to gain user trust if it is deployed in a manner that erodes user autonomy or clashes with established social norms. Therefore, achieving a trustworthy AI system is the foundation, while building user trust requires a broader, human-centric approach that addresses the full work environment.

Achieving this requires a commitment to a set of core technical and ethical pillars, which have been formalized by organizations such as the European Commission and the National Institute of Standards and Technology (NIST) (AIHLEG, 2019). The essential building blocks of AI trustworthiness include:

- **Fairness:** Ensuring the system is unbiased and does not perpetuate harmful stereotypes or discrimination based on biases in training data.
- **Accountability:** Establishing clear responsibility for the system's actions and outcomes, particularly when it makes mistakes or causes harm.
- **Transparency & Explainability:** Designing systems to provide understandable reasoning for their decisions.
- **Technical Robustness:** Ensuring the system is safe, secure, and reliable, and can operate dependably even under unexpected conditions.
- **Privacy:** Protecting user data and ensuring proper data governance throughout the system's lifecycle.

6.6.2 BEST PRACTICES FOR DEVELOPERS

To translate these abstract ethical requirements into actionable steps, developers can follow a series of best practices (Table 14) throughout the AI development lifecycle. A useful guide for this process is the European Commission's **Assessment List for Trustworthy Artificial Intelligence (ALTAI)**, which serves as a self-assessment checklist for developers and organizations to ensure their AI systems align with the seven key requirements of trustworthy AI (ALTAI.insight-centre.org, n.d.).

Table 14 Technical Pillars of Trustworthy AI with Developer Best Practices

Trustworthy AI Pillar	Developer Best Practice
Fairness	**Start with the Right Data:** Use diverse and representative data sets and implement data augmentation techniques to reduce bias at the source. **Test for Fairness:** Continuously test the system for bias using fairness metrics and other evaluation methods.
Accountability	**Design for Traceability:** Ensure that a system's decisions can be traced back to its inputs and logic. **Implement Human Oversight:** Build in continuous human review and approval loops, especially for high-stakes decisions. The human developer remains responsible for the output.
Transparency & Explainability	**Prioritize Interpretability by Design:** Shift from post-hoc, "tacked on" explanations to building interpretability into the model's architecture from the outset. Use techniques like explainable AI (XAI) to help users understand system outputs.
Technical Robustness	**Conduct Rigorous Testing:** Validate AI-generated code and models for accuracy, security, and reliability. **Use Iterative Development:** Follow a continuous cycle of generation, review, testing, and refinement until the system meets all requirements.

These practices move beyond a simple checklist to advocate for a cultural shift in the development process. For instance, the focus on "interpretability by design" challenges the idea that transparency can be added later as an afterthought. True transparency requires developers to select and design models with interpretability in mind from the very beginning. Furthermore, the practice of continuous human oversight reinforces that AI is an assistant, not a replacement for human expertise.

6.7 SUMMARY AND ACTIONABLE RECOMMENDATIONS

Based on the ALTAI checklist and the current status of the model prototypes, the next steps for developing trustful AI expert systems for the railway industry must be a proactive, step-by-step process of risk identification and mitigation. Trustworthiness is a technical and ethical concept that is the developer's responsibility to build a system that is, by design, lawful, ethical, and robust through adherence to core technical pillars such as fairness, transparency, and accountability (AIHLEG, 2019). Building user trust, however, is a much broader psychological, social, and contextual concept that requires addressing the full work environment and is influenced by factors beyond a developer's direct control (AIHLEG, 2019). A cross-functional team, including developers and domain experts, should use the ALTAI checklist as a guide (ALTAI.insight-centre.org, n.d.) to identify and address risks which might be easily overlooked by each use case.

For each of the use-cases and demonstrators, it is **recommended for the next development phase to identify the required stakeholders**, **build cross-functional teams and work through the State-**

of-the-Art guidelines. This is a low hanging fruit as the amount of work involved is low, although creating a 360° perspective on possible pitfalls on building users' trust and acceptance for a later in time productive AI decision support system, allowing to take necessary actions before the models go live.

7 CONCLUSION AND OUTLOOK

The goal of the NEXUS project is to set an innovative benchmark, tackle key challenges, and lead European metros into a transformative future. Through optimization, analysis, energy, and service efficiency, NEXUS strives to develop innovative solutions for urban transport and metro transport of the future in two European cities (Genoa, Italy, and Sofia, Bulgaria).

The widespread adoption of digital technologies across all industries—particularly in local transport—has led to the generation of huge amounts of data. When analysed effectively, this data becomes a valuable asset that enables improved decision-making and the optimization of operational efficiency and overall system performance.

The main objective of this document is to provide readers with key insights into future metro operations, focusing on how AI and data science can be used to improve system performance. It examines relevant use cases and offers practical implementation guidelines to support the integration of these cutting-edge technologies into metro networks. It also introduces and contextualizes research related to technology acceptance and trust in the context of AI and data science adoption and provides an overview of current findings on how these factors influence the successful implementation of such technologies in metro operations.

This Deliverable, D6.1 “AI Demonstrators,” builds on the work in D6.3 “AI in Future Metro Operations,” which provided an overview of potential applications of artificial intelligence and data science in future metro systems based on desktop research. However, this earlier report already identified promising use cases for AI/data science and provided an initial outlook on how these technologies could support and improve metro operations.

In this report D6.1, “AI in demonstrators,” we go beyond the concept phase and examine a number of AI and data science demonstrators in greater detail. For each demonstrator, we describe the approach to data collection, explain the methods, algorithms, and scripts used, present the results achieved, and discuss the main challenges and lessons learned. Finally, we evaluate the effectiveness and relevance of the demonstrator in the context of future metro operations.

D6.1 comprises four demonstrators from the fields of AI and data science that address the following topics: predicting crowding based on exogenous data sources (UNIGE), demand forecasting during network expansion (AU), timetable creation using GTFS feeds (AU), and detection of dirt in the vehicle interior (VIF). These demonstrators apply a variety of AI and data science techniques, including image classification and other advanced analytics, to explore practical and impactful use cases for improving future metro operations. They illustrate the potential of AI technologies to improve operational efficiency, passenger experience, and service reliability.

To move from concept to practice, the use cases mentioned are further developed on the basis of detailed implementation concepts, taking into account architectures, data requirements, technical challenges, and possible integration paths. The document describes the data science and AI demonstrators that have been developed and deployed during the course of the project, thereby laying the groundwork for concrete results that can be applied in real metro systems.

The integration of AI, IoT, and big data will revolutionize metro systems and make urban public transportation a smarter, more efficient, and more sustainable means of mobility. These technologies will drive automation, optimize operational processes, and improve real-time decision-making, resulting in safer, more reliable, and seamlessly connected metro networks. By leveraging the power of AI-enabled analytics, IoT-enabled connectivity, and big data insights, metro systems can not only improve their operational efficiency, but also provide greater safety, sustainability, and a better experience for passengers.

The widespread integration of artificial intelligence into information systems represents a transformative change in technology that brings both opportunities and complex challenges for user acceptance. While classic models of technology acceptance, such as the Technology Acceptance Model and the Unified Theory of Acceptance and Use of Technology, provide a valuable theoretical foundation, their constructs are often insufficient to address the unique characteristics of AI—namely, its opacity, unpredictability, and dynamism.

Trust in an AI algorithm is not only a consequence of its perceived usefulness or user-friendliness, but also a crucial prerequisite for its acceptance. The report addresses the psychological foundations of this trust and examines how factors such as outcome feedback, accountability, and the explainability of a system influence user behaviour. It also examines the impact of AI in the workplace, distinguishing between systems that support and those that patronize. While the latter are often well-intentioned, they prove to be ethically problematic and reliably lead to project failure. As a guide for developers, the report concludes by outlining the technical and ethical foundations of trustworthy AI and provides a practical framework for developing systems that are not only powerful but also fair, transparent, and human-cantered.

For each use case and demonstrator, it is recommended that the necessary stakeholders be identified, cross-functional teams be formed, and state-of-the-art guidelines be reviewed in the second project year. This is considered an easily achievable goal, as the associated workload is relatively low, although it provides a 360° perspective on potential pitfalls in building user trust and acceptance for a later productive AI decision support system, allowing necessary measures to be taken before the models are put into operation.

All activities were closely coordinated and communicated with the sister work packages, in particular WP4 (models to support the analysis of the adaptability of underground railways) and WP5 (feasibility study on future train control systems). This ensured close integration of the research work and avoided unnecessary duplication. The use cases mentioned are still in the early stages of development and will be further developed and continued in the second year of the project.

As the transportation sector faces increasing demands for smarter, more adaptable infrastructure, understanding how AI and data science can drive metro systems forward is crucial. The deliverable aims to provide readers with the foundational knowledge necessary to understand the transformative potential of these technologies and their real-world applications.

8 REFERENCES

ACI. (2019). Airport Council International: Demand Forecasting Techniques. Retrieved from https://airportscouncil.org/wp-content/uploads/2020/03/CHAPTER-3-DEMAND-FORECASTING-TECHNIQUES.pdf

Ai, G., Zuo, X., Chen, G., & Wu, B. (2022). Deep Reinforcement Learning based dynamic optimization of bus timetable. Applied Soft Computing, 131, 109752. https://doi.org/10.1016/j.asoc.2022.109752

AIHLEG. (2019). Ethics Guidelines for Trustworthy AI. Retrieved from https://ai.bsa.org/wp-content/uploads/2019/09/AIHLEG_EthicsGuidelinesforTrustworthyAI-ENpdf.pdf

Ajibade, P. (2018). Technology Acceptance Model Limitations and Criticisms: Exploring the Practical Applications and Use in Technology-related Studies. Library Philosophy and Practice (e-journal), 1941. Retrieved from https://digitalcommons.unl.edu/libphilprac/1941/

ALTAI.insight-centre.org. (n.d.). Welcome to the ALTAI portal!. Retrieved from https://altai.insight-centre.org/

Birmingham City Council. (2018). Population and Census. Retrieved from https://www.birmingham.gov.uk/info/20057/about_birmingham/1294/population_and_census/3#:~:text=The%20population%20of%20Birmingham%20is%20projected%20to%20grow,to%20442%2C500%20in%202028%2C%20an%20increase%20of%204.7%25.

Blanco, V., Conde, E., Hinojosa, Y., & Puerto, J. (2019). An optimization model for line planning and timetabling in automated urban metro subway networks. https://doi.org/10.48550/ARXIV.1903.08617

Blattner, L. (2025). The AI Black Box: What We're Still Getting Wrong about Trusting Machine Learning Models. Hyperight. Retrieved from https://hyperight.com/ai-black-box-what-were-still-getting-wrong-about-trusting-machine-learning-models/

Button, K. (2010). Transport Economics. Edward Elgar.

Ceder, A. (2016). Public Transit Planning and Operation: Modeling, Practice and Behavior, Second Edition (0 edn). CRC Press. https://doi.org/10.1201/b18689

Chatsfield, C. (2000). Time-Series Forecasting. Taylor and Francis Group.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly, 13(3), 319-340.

DeLone, W. H., & McLean, E. R. (1992). Information systems success: The quest for the dependent variable. Information Systems Research, 3(1), 60–95. https://doi.org/10.1287/isre.3.1.60

DeLone, W. H., & McLean, E. R. (2003). The DeLone and McLean model of information systems success: A ten-year update. Journal of Management Information Systems, 19(4), 9-30. https://doi.org/10.1080/07421222.2003.11045748

DFT. (2018, November 29). Department of Transport: TAG Data Book. Retrieved from https://www.gov.uk/government/publications/tag-data-book

Ditzel, G. (2025). Autonomous Trains. Paper presented at the ODVA Conference, March 19, 2025. Retrieved from https://www.odva.org/wp-content/uploads/2025/03/2025-ODVA_Conference_Ditzel_Autonomous_Trains_FINAL.pdf

EU CORDIS. (2014). FACTS4WORKERS. Retrieved from https://cordis.europa.eu/project/id/636778

EU EFFRA. (2014). FACTS4WORKERS. Retrieved from https://portal.effra.eu/project/1426

EY. (2025, March 1). UK Regional Economic Forecast 2025. Retrieved from https://www.ey.com/content/dam/ey-unified-site/ey-com/en-uk/newsroom/2025/03/ey-uk-regional-economic-forecast-03-2025.pdf

Freedman, D. (2005). Statistical Models: Theory and Practice. Cambridge University Press.

Global Railway Review. (2025). How AI is putting the railway industry back on track. Retrieved from https://www.globalrailwayreview.com/article/197596/how-ai-is-putting-the-railway-industry-back-on-track/

Goodwin, P. B. (1992). A Review of New Demand Elasticities with Special Reference to Short and Long Run Effects of Price Changes. Journal of Transport Economics and Policy, 26(2), 155-169.

Gössling, S., Scott, D., & Hall, M. (2020). Pandemics, tourism and global change: a rapid assessment of COVID-19. Journal of Sustainable Tourism, 29(1), 1-20.

GTFS. (2023). General Transit Feed Specification. Retrieved from General Transit Feed Specification Reference: https://gtfs.org/

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning. Springer.

Ho, T. K. (1995). Random decision forests. In Proceedings of 3rd International Conference on Document Analysis and Recognition.

Hodson, T. O. (2022). Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not. Geoscientific Model Development, 15(14), 5481–5487.

Hoffman, R. R., Mueller, S. T., Klein, G., & Litman, J. (2023). Measures for explainable AI: Explanation goodness, user satisfaction, mental models, curiosity, trust, and human-AI performance. Frontiers in Computer Science, 5.

Hosanagar, K., Ahn, D., & Almaatouq, A. (n.d.). Why is it so hard for AI to win user trust?. Knowledge@Wharton. Retrieved from https://knowledge.wharton.upenn.edu/article/why-is-it-so-hard-for-ai-to-win-user-trust/

Juniper.net. (n.d.). What is explainable AI (XAI). Retrieved from https://www.juniper.net/us/en/research-topics/what-is-explainable-ai-xai.html

Khalpey-ai.com. (2024). The risks of AI paternalism on patient autonomy: A deeper exploration. Retrieved from https://khalpey-ai.com/the-risks-of-ai-paternalism-on-patient-autonomy-a-deeper-exploration/

Khokale, S. R., Jadhav, A., Chavan, R., Wani, S., & Iwanate, P. (2025). A Survey Paper on Timetable Generator Using AI Methods. International Research Journal on Advanced Engineering Hub (IRJAEH), 3(03), 860–864. https://doi.org/10.47392/IRJAEH.2025.0122

Leanware.co. (n.d.). Best Practices for Using AI in Code Generation. Retrieved from https://www.leanware.co/insights/best-practices-ai-software-development

Li, Y., Wu, B., Huang, Y., & Luan, S. (2024). Developing trustworthy artificial intelligence: insights from research on interpersonal, human-automation, and human-AI trust. Frontiers in Psychology, 15. Retrieved from https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2024.1382693/full

Mahoney, T., Varshney, K. R., & Hind, M. (2020). AI Fairness through Robustness. IBM Corporation. Retrieved from https://research.ibm.com/publications/ai-fairness-through-robustness

McKinsey & Company. (2025). Superagency in the Workplace: Empowering People to Unlock AI's Full Potential at Work. Retrieved from https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/superagency-in-the-workplace-empowering-people-to-unlock-ais-full-potential-at-work

Meinshausen, N. (2006). Quantile Regression Forests. J. Mach. Learn. Res., 7, 983–999.

Midland Metro Alliance. (2024). Wednesbury to Brierley Hill Metro Extension. Retrieved from https://metroalliance.co.uk/projects/wednesbury-to-brierley-hill-extension/

Mim, M. S., Gatsi, F. A., Ying, E., & Kumar, S. (2025). Ensuring user autonomy in AI systems. Science Policy Review. Retrieved from https://sciencepolicyreview.pubpub.org/pub/flqvfypt

Müller-Hannemann, M., Rückert, R., Schiewe, A., & Schöbel, A. (2022). Estimating the robustness of public transport schedules using machine learning. Transportation Research Part C: Emerging Technologies, 137, 103566. https://doi.org/10.1016/j.trc.2022.103566

NIST. (n.d.). Trustworthy and Responsible AI. Retrieved from https://www.nist.gov/trustworthy-and-responsible-ai

O'Neill, O. (2025). Paternalism in the NHS: The ghost in the machine. The Lancet Digital Health. Retrieved from https://pmc.ncbi.nlm.nih.gov/articles/PMC11452825/

Oneto, L., & Anguita, D. (2019). Model Selection and Error Estimation in a Nutshell. Springer Publishing Company, Incorporated.

Opitz, D., & Maclin, R. (1999). Popular ensemble methods: an empirical study. J. Artif. Int. Res., 11, 169–198.

Ortúzar, J., & Willumsen, L. G. (2024). Modelling Transport. John Wiley & Sons.

Qlik.com. (n.d.). What is Explainable AI?. Retrieved from https://www.qlik.com/us/augmented-analytics/explainable-ai

Refraction.dev. (n.d.). Ethics and AI: Software Development Considerations. Retrieved from https://refraction.dev/blog/ethics-ai-software-development-considerations

Saki, S., & Soori, M. (2025). Artificial Intelligence, Machine Learning and Deep Learning in Advanced Transportation Systems, A Review. Multimodal Transportation, 100242. https://doi.org/10.1016/j.multra.2025.100242

Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press.

Small, K. A., & Verhoef, E. T. (2007). The Economics of Urban Transportation. Routledge.

TfWM. (2023, August). Transport for West Midlands. API documentation and schedule data endpoints for GTFS. Retrieved from https://api-portal.tfwm.org.uk/docs#/

Tranberg, P. (n.d.). Trust in AI can be too little and too much. DataEthics.eu. Retrieved from https://dataethics.eu/trust-in-ai-can-be-tool-little-and-too-much/

Transitland. (2023, August). Transitland API. Retrieved from Transitland: https://www.transit.land/feeds/f-transport~for~west~midlands

Urban Institute. (2025). Building Foundations for AI Exploration in Local Government. Retrieved from Urban Institute: https://www.urban.org/sites/default/files/2025-05/Building-foundations-for-AI-exploration-in-local-government.pdf

Urban Institute. (2025, January 8). Practical AI Insights for Local Leaders. Retrieved from Urban Institute: https://www.urban.org/research/publication/practical-ai-insights-local-leaders

Van Der Knaap, R. J. H., De Bruyn, M., Van Oort, N., Huisman, D., & Goverde, R. M. P. (2024). Clustering railway passenger demand patterns from large-scale origin–destination data. Journal of Rail Transport Planning & Management, 31, 100452. https://doi.org/10.1016/j.jrtpm.2024.100452

Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. Management Science, 46(2), 186-204.

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. MIS Quarterly, 27(3), 425-478.

Wardman, M. (2001). A review of British evidence on time and service quality valuations. Transportation Research Part E: Logistics and Transportation Review, 37(2), 107-128.

West Midland Metro. (2023). Metro Expansion Welcome to the future of West Midlands Metro. Retrieved from https://www.westmidlandsmetro.com/about/expansion/

West Midlands Metro. (2025, March 20). Fare review announced as Metro continues to invest in the network. Retrieved from https://www.westmidlandsmetro.com/fare-review-announced-as-metro-continues-to-invest-in-the-network/

WMCA. (2023, January 26). Work started on £43m West Midlands Metro depot expansion as more new trams arrive. Retrieved from https://www.wmca.org.uk/news/work-started-on-43m-west-midlands-metro-depot-expansion-as-more-new-trams-arrive/#:~:text=New%20extensions%20to%20the%20line%20in%20Birmingham%20and,more%20capacity%20needed%20to%20store%20and%20maintain%20them.

Yu, B., Lam, W. H. K., & Tam, M. L. (2011). Bus arrival time prediction at bus stop with multiple routes. Transportation Research Part C: Emerging Technologies, 19(6), 1157–1170. https://doi.org/10.1016/j.trc.2011.01.003

Zen the innovator. (2024). Lack of Explainability and Transparency in AI: The Black-Box Dilemma. Medium. Retrieved from https://medium.com/@ThisIsMeIn360VR/lack-of-explainability-and-transparency-in-ai-the-black-box-dilemma-5bb58776cd93

Zhang, K., Shi, Y., & Clarke, S. (2024). Sustainable Multi-Modal Transit Timetabling with Deep Learning. SSRN. https://doi.org/10.2139/ssrn.4884880

Zheng, A. C. A. (2018). Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. O'Reilly Media, Inc.