Methodology

✓ Detailed research process
✓ Data collection & preparation
✓ Feature selection rationale
✓ Modeling choices & justification
✓ Evaluation methodology
✓ Optimization methodology
✓ Ethical/production considerations

1. Research Approach

Our methodology follows a **data-driven applied research process** designed to understand, model, and optimize delivery route performance.
We apply a **quantitative, experimental approach**, using historical operational data to identify patterns that lead to delays and inefficiencies.
The research process includes:

1. **Problem understanding** – define operational inefficiencies (delays, distance deviations).
2. **Data exploration** – examine distributions, correlations, and temporal patterns.
3. **Feature selection** – identify which variables influence delays most strongly.
4. **Model design and training** – build predictive models to quantify delays and inefficiencies.
5. **Optimization design** – use predictions to propose route and scheduling improvements.
6. **Evaluation** – measure performance against baseline operational metrics.

Each step builds toward a production-ready intelligent decision-support system.

---

2. Data Collection & Description

The dataset consists of historical last-mile delivery records. It includes:

- Route-level data (RouteID, StopID, DriverID)
- Planned delivery structure (IndexP, DistanceP, time windows)
- Actual delivery behavior (IndexA, DistanceA, ArrivedTime)
- Temporal context (DayOfWeek, WeekID)
- Geographic indicators (Country, Depot)

No external sources or sensors were used; all data is pre-recorded operational data.
This aligns with the project requirement of using historical data only and enables reproducible research.

---

3. Data Preprocessing

We performed the following preparation steps to ensure clean and analyzable data:

3.1 Cleaning

- Removed incomplete rows with missing arrival times.
- Normalized time formats into comparable timestamps.
- Standardized categorical fields (DriverID, RouteID).
- Detected and removed extreme outliers (e.g., negative distances).

3.2 Feature Engineering

We created domain-specific derived features that directly relate to the problem:

- **Delivery Delay:**
  `delay = ArrivedTime - LatestTime`
- **Distance Deviation:**
  `distance_diff = DistanceA - DistanceP`
- **Stop Sequence Deviation:**
  `sequence_diff = IndexA - IndexP`
- **Time window tightness:**
  `window_size = LatestTime - EarliestTime`

These engineered features significantly improve model interpretability and predictive performance.

---

4. Feature Selection & Rationale

We selected features that have **direct causal influence** on delivery timing and route efficiency.
Our selection was based on:

A. Domain knowledge

Logistics research shows delays are affected by:

- Stop order

- Time windows
- Driver behavior
- Route length
- Temporal patterns (day of week)

B. Exploratory Data Analysis

Correlation plots and statistical tests showed:

- Strong relationships between delay and arrival time behavior
- High importance of planned vs. actual distance
- Clear weekly and daily seasonality patterns
- DriverID explaining systematic differences in performance

C. Model-based relevance

Tree-based feature importance confirmed the importance of:

- LatestTime
- DistanceP and DistanceA
- IndexP and IndexA
- DriverID
- DayOfWeek

Thus, our chosen features are justified by domain theory, statistical evidence, and model-driven selection.

---

5. Modeling Methodology

We use a **two-level modeling strategy**:

5.1 Baseline Predictive Models

- **Random Forest** (for delay classification: on-time vs delayed)
- **XGBoost** (for delay regression: predict minutes late)

Why these models?

- Handle tabular operational data extremely well
- Robust to noise and missing features
- Provide explainability (feature importance)

- Fast to train and tune

5.2 Advanced Temporal Model (PyTorch LSTM)

- Routes are **sequences** of stops → early delays propagate
- LSTM is ideal for sequential patterns, time dependencies, and ordered data

Rationale:
The LSTM captures how delay accumulates along the route, which static models cannot.

Training & Validation

We use:

- 70/15/15 train-validation-test split
- k-fold validation for fairness
- Standard metrics: MAE, RMSE, accuracy, F1-score

This ensures reliability and reproducibility.

---

6. Optimization Methodology

Once delay predictions are available, we design an optimization layer:

6.1 Diagnose the cause

Using model outputs, we identify whether a delay is caused by:

- Tight time windows
- Inefficient stop order
- High distance deviation
- Driver inconsistency
- Day-of-week congestion

6.2 Generate improvements

We test interventions such as:

- **Reordering stops** using heuristic sequencing (e.g., nearest-neighbor based on distance order).
- **Driver reallocation** when a driver shows consistent lateness.
- **Time-window smoothing**, spreading deliveries with tight windows earlier.

- **Route load balancing** by shifting stops across days with lighter workloads.

6.3 Simulation

We simulate each intervention using predicted delay outcomes to estimate:

- Fewer delayed stops
- Reduced distance deviation
- Earlier arrival adjustments

This is the "production intelligence" layer of the system.

---

7. Ethical, Safety & Production Considerations

- No private customer data used; dataset contains only operational info.
- Models are explainable and auditable.
- All experiments are reproducible via scripts and documented environments.
- The optimization does not override human judgment; it provides decision support.