Multimodal Learning for Operational Risk Detection in Delivery Route Planning

Prateek Gupta[1], Daniel Antunes Pedrozo[1], Jorge Augusto Meira[1], Antonio Ken Iannillo[1], and Danilo D'Aversa[2]

[1] SnT, University of Luxembourg, 29 Av. John F. Kennedy, 1855 Luxembourg City, Luxembourg
{prateek.gupta,daniel.antunes,jorge.meira,antonioken.iannillo}@uni.lu
[2] Gulliver Luxembourg S.à r.l. danilo.daversa@gullivernet.lu

Abstract. We propose a privacy-aware multimodal framework for detecting operational risks in last-mile delivery routes by integrating spatial, temporal, and semantic information. Our method fuses geospatial embeddings derived from latitude–longitude coordinates or representative zip-code centroids, ensuring GDPR compliance when precise locations are withheld, with temporal features from delivery attempt timestamps and semantic embeddings extracted by a locally hosted large language model (LLM). To capture localized delivery failure patterns, we incorporate sequence-to-pattern mining at the zip-code level, enabling the model to recognize high-risk operational contexts. This unified architecture is designed to maintain predictive robustness even when partial location data is missing, ensuring routes remain analyzable despite privacy constraints. Empirical evaluation on real-world logistics datasets comprising approximately 2,000 routes, each visiting between 2 and 26 customers, shows that multimodal embeddings significantly enhance precision in detecting early failures compared to random baselines during validation. However, test-set results reveal generalization challenges and overfitting risks, suggesting that further work is needed on data augmentation, embedding fine-tuning, and regularization strategies. By enabling early, data-driven risk detection, our approach has the potential to support more reliable and sustainable last-mile delivery operations.

Keywords: Multimodal Machine Learning · Privacy-Aware Modeling · Operational Risk Detection · Sustainable Logistics · Large Language Models (LLMs)

1 Introduction

Last-mile logistics is one of the most challenging supply chain stages, which involves transportation from local depots to customer doorsteps. This final leg often represents a significant portion of total delivery costs, driven by routing complexity, urban congestion, and unpredictable customer behavior, such as customers being unreachable despite prescheduled time slots.

Accurate prediction of operational risks, such as delivery failures, is crucial to cost efficiency and service quality. We specifically measure precision around the first failure event to assess how reliably the model signals early warnings. This metric is particularly important in contexts involving large or heavy products that must be unloaded in a strict sequence, and early detection of potential failures can help prevent cascading disruptions to subsequent deliveries and mitigate costly downstream effects. This remains difficult for two key reasons. First, logistics data often contain private information, including precise customer locations and personalized delivery instructions, which are tightly regulated under frameworks such as the General Data Protection Regulation (GDPR) in Europe. Second, in practice, some customers withhold precise geographic location data for privacy reasons, leaving gaps that can render route-level machine learning models unusable if they rely exclusively on accurate spatial signals.

In this paper, we tackle both challenges with a privacy-aware multimodal architecture that integrates spatial, temporal, and semantic information to predict route-level risks. Even when precise coordinates are unavailable, we substitute them with representative latitude-longitude values derived from zip-code centroids, ensuring GDPR compliance and preserving model functionality. Our experiments on real-world logistics data show that this multimodal approach can improve early risk detection in modern last-mile delivery networks.

1.1 Related Work

Classical optimization approaches to last-mile delivery are modeled as variants of the Vehicle Routing Problem (VRP), which optimize cost-minimizing delivery routes subject to constraints such as vehicle capacity, time windows, or maximum tour length. These models provide strong theoretical guarantees, are computationally expensive, and do not take advantage of historical data collected during logistics operations. For example, previous work by Dettenbach et al. [2], proposed maintaining backup plans to handle interruptions before delivery begins. However, such strategies are limited in capturing real-world uncertainties, particularly those arising from delivery failures due to customer unavailability or localized risk patterns.

To address these limitations, recent research uses machine learning (ML) and deep learning (DL) techniques to leverage data-driven insights for predictive accuracy and adaptability in last-mile logistics. Sharma et al. [11], for instance, employed random forests to predict service failures driven by customer behaviors such as absence or refusal; however, their work remained at the customer level and did not integrate route-level spatial dependencies. Florio et al. [3], used the "pool and select" algorithm to build candidate delivery sequences using heuristics based on entry and exit zones, then select the most promising sequence through a learned regression model. Other works have used gradient-boosted trees [7] and deep learning for delivery time predictions [1][4].

Most of the previous work in ML for last-mile logistics focuses on demand forecasting, predicting estimated time of delivery, and planning routes [10][9] [6][8][12][5]. Research aimed at predicting delivery failures, such as a customer

not being at home, is relatively limited. This gap arises for several reasons. Failed deliveries are rare in many operational datasets, making supervised learning difficult. Another reason is that success and failure are often tied to privacy-sensitive signals such as at-home presence, responsiveness, or special delivery instructions, which fall under GDPR protections.

In our work, we do not use time windows as a predictive feature, as each company defines and enforces time windows differently. Instead, we rely on the actual time of delivery attempt recorded in historical data, which provides more precise and interpretable temporal signals. Our approach focuses on predicting service-level delivery failures by modeling interactions with preceding stops along the route. Using a proprietary logistics dataset in which delivery outcomes are explicitly labeled, we demonstrate that it is feasible to learn meaningful risk patterns in real-world, privacy-compliant settings.

Unlike prior studies that assume the full availability of precise latitude and longitude data, we explicitly address scenarios where customers withhold geographic location information due to privacy concerns. To our knowledge, no existing multimodal architecture integrates a privacy-aware fallback mechanism that substitutes missing coordinates with representative zip-code centroids while maintaining predictive performance. Our work fills this gap by explicitly designing for robustness under partial spatial data availability.

The rest of the paper is organized as follows. Section 2 presents the training pipeline and model architecture, including input representations, graph attention layers, and transformer encoding. In Section 3, we describe the experimental setup covering hyperparameters, embedding dimensions, sequence mining settings, baseline comparisons, and performance analysis. Section 4 concludes with a summary of contributions and directions for future work.

2 Model

2.1 Overview

We propose a multimodal neural architecture designed for early detection of operational risks in last-mile delivery. The model integrates geospatial, semantic, temporal, and pattern-based signals into a unified framework and explicitly models spatial relationships via graph attention mechanisms. Our architecture emphasizes early detection by combining position-aware loss functions and evaluation metrics tailored for sequential prediction.

2.2 Input Representations

Each time step t in a delivery sequence is represented by the following modalities:

- **Geospatial features**: GPS coordinates are encoded into a high-dimensional latent space
$$\mathbf{x}_t^{\text{geo}} \in \mathbb{R}^{d_{\text{geo}}}$$
using a pretrained *LocationEncoder*, inspired by vision-language models, to capture spatial context beyond the raw coordinates.

– **Semantic embeddings**: Contextual information from textual delivery instructions is extracted using a frozen LLaMA 3.2 1B model, yielding embeddings

$$\mathbf{x}_t^{\text{llama}} \in \mathbb{R}^{d_{\text{llama}}}.$$

Since decoder-only models like LLaMA do not produce a single pooled embedding by default, we extract the hidden state of the first token from the final layer to serve as a sentence-level representation. This vector is normalized before being concatenated with other features

– **Temporal features**:
  - **Time-of-day cyclic features:** hour and minute encoded via sine and cosine transformations to model periodicity,

$$\sin\left(2\pi\frac{\text{hour}}{24}\right), \quad \cos\left(2\pi\frac{\text{hour}}{24}\right), \quad \sin\left(2\pi\frac{\text{minute}}{60}\right), \quad \cos\left(2\pi\frac{\text{minute}}{60}\right).$$

  - **Day-of-week cyclic features:** day-of-week encoded similarly with sine and cosine transforms,

$$\sin\left(2\pi\frac{\text{dow}}{6}\right), \quad \cos\left(2\pi\frac{\text{dow}}{6}\right),$$

where $\text{dow} \in \{0, \ldots, 6\}$.
  - **Weekend indicator:** a binary feature that indicates whether the day is Saturday or Sunday.
  - **Time segment one-hot encoding:** hour-of-day is discretized into five segments $\{[0, 7), [7, 12), [12, 17), [17, 21), [21, 24)\}$, encoded as a one-hot vector.

These features are concatenated to form the temporal feature vector

$$\mathbf{x}_t^{\text{time}} \in \mathbb{R}^{d_{\text{time}}}.$$

– **Pattern embeddings**: Summaries of frequent delivery success and failure patterns are encoded as

$$\mathbf{x}^{\text{pattern}} \in \mathbb{R}^{d_{\text{pattern}}}.$$

All features are concatenated to form the time-step input:

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{x}_t^{\text{geo}} \\ \mathbf{x}_t^{\text{llama}} \\ \mathbf{x}_t^{\text{time}} \end{bmatrix} \in \mathbb{R}^{d_{\text{in}}},$$

where

$$d_{\text{in}} = d_{\text{geo}} + d_{\text{llama}} + d_{\text{time}}.$$

2.3 Architecture

The sequence $\{\mathbf{x}_t\}_{t=1}^T$ is first linearly projected into a shared latent space of dimension d_{model}:

$$\mathbf{h}_t^{(0)} = W_{\text{in}}\,\mathbf{x}_t + \mathbf{b}_{\text{in}}.$$

When pattern embeddings are available, we inject them as additive biases:

$$\mathbf{h}_t^{(0)} \leftarrow \mathbf{h}_t^{(0)} + W_{\text{pattern}}\,\mathbf{x}^{\text{pattern}}.$$

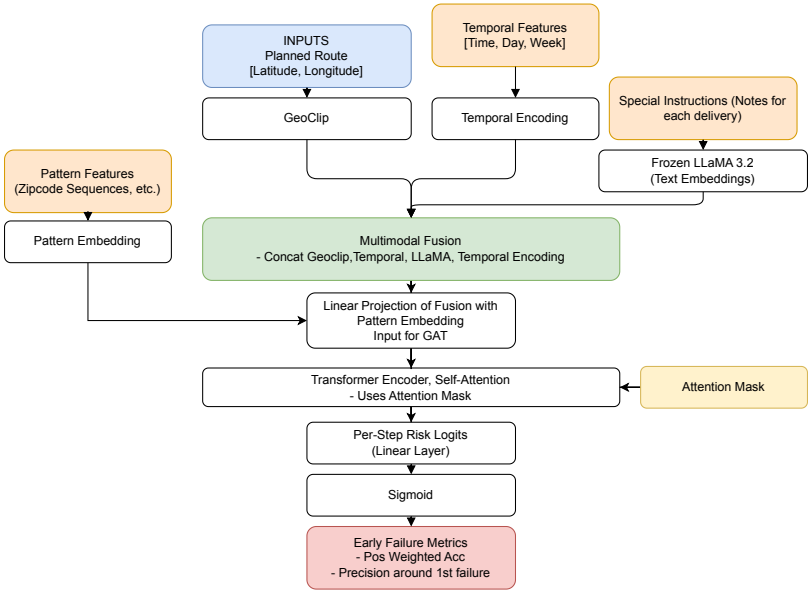

Fig. 1: Model Architecture

Graph Attention Layers. To capture spatial and topological dependencies among stops, we process the sequence with a two-layer Graph Attention Network (GAT). Importantly, edges operate *over the flattened latent representations* of stops across the batch:

$$\mathbf{h}^{\text{gat}} = \text{GAT}\big(\text{flatten}(\mathbf{h}^{(0)}),\ E\big),$$

where E is an edge index defining stop-to-stop connectivity. The GAT aggregates neighborhood information in latent space and reshapes the output back to sequence form. Residual connections yield:

$$\mathbf{h}_t^{(1)} = \mathbf{h}_t^{(0)} + \mathbf{h}_t^{\text{gat}}.$$

Transformer Encoding. The enhanced embeddings $\{\mathbf{h}_t^{(1)}\}$ are fed into a multilayer Transformer encoder, capturing temporal and global sequence dependencies:

$$\{\mathbf{h}_t^{\text{enc}}\}_{t=1}^{T} = \text{TransformerEncoder}\left(\{\mathbf{h}_t^{(1)}\}_{t=1}^{T}\right).$$

Finally, a linear layer projects to scalar logits:

$$\hat{y}_t = W_{\text{out}}\,\mathbf{h}_t^{\text{enc}} + \mathbf{b}_{\text{out}}.$$

2.4 Training Objectives

Our primary loss is a binary cross-entropy applied at each time step. However, to emphasize early detection and reduce delayed corrections, we introduce the **Sequential Penalty-Aware Contextual Loss (SPaCeLoss)**.

SPaCeLoss. Let the predicted logits be $\hat{\mathbf{y}} \in \mathbb{R}^{B \times T}$. Compute probabilities:

$$\mathbf{p} = \sigma(\hat{\mathbf{y}}).$$

Define binary cross-entropy:

$$\text{BCE}(\mathbf{p}, \mathbf{y}) = -\big[\mathbf{y} \cdot \log(\mathbf{p}) + (1 - \mathbf{y}) \cdot \log(1 - \mathbf{p})\big].$$

The focal modulation term is:

$$\mathbf{pt} = \mathbf{p} \cdot \mathbf{y} + (1 - \mathbf{p}) \cdot (1 - \mathbf{y}),$$

yielding:

$$\text{FocalWeight} = \alpha\,(1 - \mathbf{pt})^{\gamma}.$$

Hence the focal loss:

$$\text{FocalLoss} = \text{FocalWeight} \odot \text{BCE}(\mathbf{p}, \mathbf{y}).$$

To penalize late mistakes, SPaCeLoss introduces positional penalties:

$$\mathbf{w}_{\text{pos}} = \text{linspace}(1.0,\ 0.1,\ T).$$

Scaled by β_{pos}:

$$\text{Penalty} = \beta_{\text{pos}} \cdot \mathbf{w}_{\text{pos}}.$$

After optional normalization, the final loss is:

$$\text{SPaCeLoss} = \frac{1}{B \times T} \sum_{i=1}^{B} \sum_{t=1}^{T} \text{FocalLoss}_{i,t} \times \text{Penalty}_t.$$

2.5 Evaluation Metrics

We evaluate the model with metrics prioritizing timely risk detection:

Precision Around First Failure. To quantify how early and accurately the model signals the first failure, we compute a distance-weighted precision defined as:

$$\text{Precision}_{\text{first-fail}} = \begin{cases} \dfrac{\sum_{i=1}^{N} s_i}{PP}, & \text{if } PP > 0, \\ 0, & \text{otherwise.} \end{cases}$$

where:

- PP = number of sequences with any predicted failure,
- s_i = score for sequence i reflecting how close the first predicted failure is to the first true failure.

Formally, let $\hat{\mathbf{y}}_i \in \{0,1\}^T$ denote thresholded predictions and $\mathbf{y}_i \in \{0,1\}^T$ the ground truth. Define:

$$\text{FirstFail}_i = \min\left\{ t \;\middle|\; y_{i,t} = 0 \right\}.$$

Similarly, define:

$$\text{FirstPred}_i = \min\left\{ t \;\middle|\; \hat{y}_{i,t} = 0 \right\}.$$

Then, for each sequence i where any failure is predicted, we compute:

$$s_i = \frac{1}{1 + \left|\text{FirstPred}_i - \text{FirstFail}_i\right|}.$$

This yields a precision score close to 1 when the predicted failure is very close to the true first failure, and decreasing as the prediction deviates further in time.

(a) Full model - Training Loss

(b) Full model - Validation Loss

(c) Baseline - Training Loss

(d) Baseline - Validation Loss

(e) Full model - Precision Around First Failure (Validation)

(f) Baseline - Precision Around First Failure (Validation)

Fig. 2: Training and validation loss curves for full and baseline models, along with Precision Around First Failure metric on validation dataset.

3 Experiments

3.1 Training Setup

All experiments were conducted on an Apple M1 Mac with 16 GB RAM, running macOS Sequoia 15.3.1. Our software environment comprised Python 3.10, PyTorch 1.13.1, and GeoPandas 0.13.2. We trained and evaluated models on real routing data from a logistics partner collected in 2025. To enable stratified splitting into training, validation, and test sets, we defined a helper variable indicating whether a route contained at least one failure (present in approximately 3.5% of routes across all splits). This variable was dropped after stratification and not used during model training or evaluation.

Model training, evaluation, and deployment were managed via containerized services using Docker, including PostgreSQL for data storage and MLflow for experiment tracking and model management. This environment ensured reproducibility and streamlined workflow management.

3.2 Hyperparameters

Input and Embedding Dimensions We use the following feature dimensions: GeoClip embeddings of size 512, LLaMA embeddings of size 2048, pattern features of size 10, and temporal features of size 12.

Sequence Mining Settings Failure patterns are mined with minimum frequency 5 and maximum length 5, whereas success patterns use a minimum frequency of 50 and maximum length 5.

Transformer Architecture The model employs a Transformer with hidden dimension $d_{\text{model}} = 16$, 2 attention heads, and 2 layers.

Training Hyperparameters We train with batch size 16, learning rate 5×10^{-4}, momentum 0.9, dropout rate 0.5, for 20 epochs using the Adam optimizer.

3.3 Baselines

We evaluate our model on two embedding configurations: (i) **Full**, which uses both GeoClip and LLaMA embeddings, and (ii) **RandomBoth**, which replaces both embeddings with random vectors as a baseline.

3.4 Performance Analysis

Table 1 reports validation metrics from the best checkpoints under two embedding configurations: Full embeddings (GeoClip + LLaMA) and RandomBoth (random embeddings). The Full model clearly outperforms the RandomBoth baseline on all key metrics during validation. The *PrecisionFirstFail* metric, measuring the precision of detecting the very first failure event, is particularly

Table 1: Validation performance comparison of embedding configurations.

Embedding Setting	PosWeightedAcc	PrecisionFirstFail	Validation Loss
Full (GeoClip + LLaMA 3.2 1B)	0.97	0.88	0.0259
RandomBoth (Random embeddings)	0.49	0.20	0.0470

important operationally. The Full model achieves high precision (0.88), indicating it reliably flags early failures with relatively few false positives. In contrast, the RandomBoth baseline struggles to detect failures accurately, reflected by low precision (0.20), despite a lower overall positional accuracy. The validation loss curves (Figure 2b) corroborate these findings, showing that the Full model converges to a lower loss, indicating stronger learning of failure signals.

3.5 Test Set Performance

To assess generalization, we evaluated both models on a held-out test set with the same metrics. Table 2 summarizes the results. On the test set, both models

Table 2: Test performance comparison of embedding configurations.

Embedding Setting	PosWeightedAcc	PrecisionFirstFail	Test Loss
Full (GeoClip + LLaMA 3.2 1B)	0.60	0.21	0.0312
RandomBoth (Random embeddings)	0.62	0.21	0.0502

see a marked drop in *PrecisionFirstFail* compared to validation, with the Full model's precision falling from 0.88 to 0.21. This drop suggests the Full model is overfitting to the training and validation data and struggles to detect early failures reliably on new, unseen routes. The RandomBoth baseline maintains low precision consistent with its validation performance, indicating it relies largely on predicting the majority no-failure class. These results highlight the gap between validation and real-world generalization, underscoring the challenge of developing robust models in highly imbalanced, complex operational settings.

3.6 Discussion

Our emphasis on *PrecisionFirstFail* reflects the practical need to minimize false alarms in early failure detection. High precision ensures that flagged failures are credible, enabling planners to trust and act on predictions without unnecessary operational disruptions. Traditional metrics such as recall or F1 score are less aligned with this objective, as missing some failures (lower recall) is less critical than avoiding false positives that trigger costly interventions. The Full model's strong validation precision confirms that multimodal embeddings (GeoClip + LLaMA) enhance the model's ability to capture meaningful failure signals. These

findings indicate that while frozen embeddings provide strong representations for initial training, fine-tuning may be required to capture task-specific signals and ensure consistent generalization to unseen delivery routes.

3.7 Overfitting and Generalization

The decline in test precision around first failure suggests overfitting by the Full model, potentially because pretrained GeoClip and LLaMA embeddings were used without fine-tuning for our specific delivery failure detection task. These embeddings may encode features that do not generalize well across different routes, customers, or operational conditions. Future work will address this by fine-tuning embeddings, augmenting data, applying regularization, and incorporating contextual features such as weather, traffic, and driver-specific data to improve robustness on unseen data.

4 Conclusion and Future Work

We proposed a privacy-aware multimodal framework for early detection of delivery failures in last-mile logistics. Our model integrates spatial-temporal patterns, semantic embeddings from large language models applied to customer notes, and sequential dependencies across delivery routes. Unlike prior work focusing on individual delivery points, our route-level modeling captures localized operational risks informed by delivery sequence context. Empirical evaluation on industrial data shows that incorporating GeoClip and LLaMA embeddings improves precision of early failure detection on validation data compared to random baselines. However, using these embeddings out-of-the-box leads to overfitting, limiting generalization to unseen routes. Future work includes fine-tuning pretrained LLM and geospatial embeddings for logistics-specific patterns; employing oversampling or synthetic data to better capture rare failures; applying regularization and architectural constraints to mitigate overfitting; and integrating contextual signals (e.g., weather, traffic, driver data) to enrich risk modeling. These efforts aim to improve model robustness and enable earlier, more reliable detection of delivery failures in real-world last-mile operations.

Acknowledgments

This work was supported by a collaboration between the University of Luxembourg and and `Gulliver Luxembourg s.a.r.l.` We thank `Danilo d'Aversa` for their support.

Competing Interests

The authors have no competing interests to declare that are relevant to the content of this article.

References

1. de Araujo, A.C., Etemad, A.: End-to-End Prediction of Parcel Delivery Time With Deep Learning for Smart-City Applications. IEEE Internet of Things Journal **8**(23), 17043–17056 (Dec 2021). https://doi.org/10.1109/JIOT.2021.3077007
2. Dettenbach, A.M., Ubber, S.: Managing Disruptions in Last Mile Distribution. In: 2015 48th Hawaii International Conference on System Sciences. pp. 1078–1087 (Jan 2015). https://doi.org/10.1109/HICSS.2015.132, iSSN: 1530-1605
3. Florio, A.M.: A Machine Learning Framework for Last-Mile Delivery Optimization (2021)
4. Gao, C., Zhang, F., Wu, G., Hu, Q., Ru, Q., Hao, J., He, R., Sun, Z.: A Deep Learning Method for Route and Time Prediction in Food Delivery Service. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 2879–2889. KDD '21, Association for Computing Machinery, New York, NY, USA (Aug 2021). https://doi.org/10.1145/3447548.3467068
5. Garg, A., Ayaan, M., Parekh, S., Udandarao, V.: Food Delivery Time Prediction in Indian Cities Using Machine Learning Models (Mar 2025). https://doi.org/10.48550/arXiv.2503.15177, arXiv:2503.15177 [cs]
6. Giuffrida, N., Fajardo-Calderin, J., Masegosa, A.D., Werner, F., Steudter, M., Pilla, F.: Optimization and Machine Learning Applied to Last-Mile Logistics: A Review. Sustainability **14**(9), 5329 (Jan 2022). https://doi.org/10.3390/su14095329, https://www.mdpi.com/2071-1050/14/9/5329, number: 9 Publisher: Multidisciplinary Digital Publishing Institute
7. Khiari, J., Olaverri-Monreal, C.: Boosting Algorithms for Delivery Time Prediction in Transportation Logistics. In: 2020 International Conference on Data Mining Workshops (ICDMW). pp. 251–258 (Nov 2020). https://doi.org/10.1109/ICDMW51313.2020.00043, https://ieeexplore.ieee.org/abstract/document/9346325, iSSN: 2375-9259
8. Leeuw, J.d., Bukhsh, Z., Zhang, Y.: Parcel loss prediction in last-mile delivery: deep and non-deep approaches with insights from Explainable AI (Oct 2023). https://doi.org/10.48550/arXiv.2310.16602, arXiv:2310.16602 [cs]
9. Oršič, J., Jereb, B., Obrecht, M.: Sustainable Operations of Last Mile Logistics Based on Machine Learning Processes. Processes **10**(12), 2524 (Dec 2022). https://doi.org/10.3390/pr10122524, https://www.mdpi.com/2227-9717/10/12/2524, number: 12 Publisher: Multidisciplinary Digital Publishing Institute
10. Ren, S., Guo, B., Cao, L., Li, K., Liu, J., Yu, Z.: DeepExpress: Heterogeneous and Coupled Sequence Modeling for Express Delivery Prediction (Aug 2021). https://doi.org/10.48550/arXiv.2108.08170, arXiv:2108.08170 [cs]
11. Sharma, M., Glatard, T., Gelinas, E., Tagmouti, M., Jaumard, B.: Data models for service failure prediction in supply-chain networks (Oct 2018). https://doi.org/10.48550/arXiv.1810.09944, arXiv:1810.09944 [cs]
12. Ye, T., Hijazi, A., Hentenryck, P.V.: Conformal Predictive Distributions for Order Fulfillment Time Forecasting (May 2025). https://doi.org/10.48550/arXiv.2505.17340, arXiv:2505.17340 [cs]