# Rail transit delay forecasting with Causal Machine Learning

Nishtha Srivastava
Sardar Vallabhbhai National Institute of Technology
Surat, Gujarat, India
d20co005@coed.svnit.ac.in

Bhavesh N. Gohil
Sardar Vallabhbhai National Institute of Technology
Surat, Gujarat, India
bng@coed.svnit.ac.in

Suprio Ray
University of New Brunswick
Fredericton, Canada
sray@unb.ca

## Abstract

The rapid evolution of public transport and advances in analytics have significantly transformed the way we enhance transit services. Rail transit systems, celebrated for their comfort, speed, and minimal environmental impact, face ongoing challenges due to persistent delays. We introduce a novel approach that integrates causal inference with machine learning techniques to predict rail transit delays and uncover key causal factors. Utilizing the New Jersey Transit dataset, we apply uplift modeling and causal inference methods to enhance delay predictions. The study employs Individual Treatment Effect (ITE) and Average Treatment Effect (ATE) metrics to interpret and validate the predictions. Our research offers a comprehensive understanding of rail transit delays and provides actionable insights for policymakers, urban planners, and public health officials. By advancing causal analytical techniques, this work aims to improve transit reliability and efficiency on a global scale.

## CCS Concepts

• **Computing methodologies** → **Feature selection**; **Machine learning**; **Artificial intelligence**; • **Applied computing** → **Transportation**.

## Keywords

Rail transit, Public transport, Arrival time prediction, causal ML, Individual Treatment Effect, Average Treatment Effect

**ACM Reference Format:**
Nishtha Srivastava, Bhavesh N. Gohil, and Suprio Ray. 2024. Rail transit delay forecasting with Causal Machine Learning. In *1st ACM SIGSPATIAL International Workshop on Spatiotemporal Causal Analysis (STCausal'24), October 29-November 1, 2024, Atlanta, GA, USA*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3681778.3698784

## 1 Introduction

Due to increased demand and capacity utilization over recent decades, many rail transit systems are now operating at near maximum capacity. This high utilization heightens the risk of delays propagating from one train to others, causing extended disruptions throughout the network. Such delays can severely impact the efficiency and attractiveness of rail services. According to Spanninger et al. [21], train transit delay prediction methods can be divided into event-driven and data-driven categories. Event-driven models rely on a structured sequence of train events, such as departures and arrivals, to predict delays, using techniques like systems of equations [11], Bayesian Networks [4], Timed Event Graphs [9], Markov Chains [20], and Petri Nets [12, 28]. In contrast, data-driven methods do not focus on explicit train-event dependencies or traffic flow dynamics.

Historically, predicting public transit delays has been challenging due to limited access to real-time data and the constraints of traditional analytical methods. Recent advancements have dramatically improved both data collection and analysis. Many transit agencies now use GPS-based tracking systems to provide real-time updates on vehicle positions and arrival times. Additionally, the General Transit Feed Specification (GTFS) standard has facilitated the dissemination of transit updates. Despite these improvements, accurately predicting arrival times remains a challenge. Machine learning (ML) has shown promise in addressing these issues, often outperforming traditional methods [14, 22]. The effectiveness of ML models depends significantly on the quality of the data and features used. Identifying key features that affect transit delays is crucial for enhancing prediction accuracy.

This paper examines rail transit systems, which are favored for their comfort, speed, and low emissions but are often affected by delays, particularly during peak periods. Inaccurate delay forecasts can lead passengers to choose private cars over rail transport. Research on rail transit delays is less developed compared to bus systems [10]. Our objective is to improve rail transit delay predictions using causal ML techniques. The model proposed in this study is based on supervised learning. This approach typically involves training a model on a labeled dataset, where both the input features (such as train schedules, stop sequences, and historical delay data) and the corresponding target variable (the actual delays) are known.

The key contributions of this paper are:

- Applying causal inference and ML techniques to forecast rail transit delays.
- Using ITE and ATE to identify significant factors affecting rail transit delays.
- Re-training ML models with identified key features and comparing their performance to the original models.

The paper is organized as follows. Section 2 reviews relevant literature on causal inference and ML. Section 3 offers a detailed review of delay prediction methods, followed by problem definition in Section 4. Section 5 describes the datasets used, Section 6 presents the proposed methodology, and Section 7 outlines the experimental setup. Results and analysis are discussed in Section 8, with the paper concluding in Section 9, summarizing the findings and contributions.


*STCausal'24, October 29-November 1, 2024, Atlanta, GA, USA*

ACM ISBN 979-8-4007-1154-1/24/10
https://doi.org/10.1145/3681778.3698784

# 2 Background

## 2.1 Causal inference

Causal Machine Learning (ML) provides methods for estimating causal effects, such as the ATE [16] and the ITE [1], using both experimental and observational data. These techniques measure the impact of an intervention $W$ on an outcome $Y$, considering observed features $X$ of individuals, without requiring strict assumptions about the model's structure. An example of a causal graph that illustrates these concepts is shown in Figure 1.

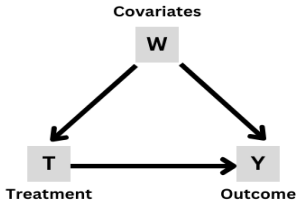


**Figure 1: Example of a causal graph [25]**

where,

- **Treatment effect (T):** The change in the outcome variable resulting from a modification in the treatment variable.
- **Covariates (W):** Factors that influence both the treatment and the outcome.
- **Outcome (Y):** The resulting variable or output.

By considering each feature individually as a treatment variable, we can calculate two types of treatment effects:

(1) Average Treatment Effect (ATE)
(2) Individual Treatment Effect (ITE)

## 2.2 Individual Treatment Effect (ITE)

ITE [23] for a specific feature represents the effect of the treatment across all instances of that feature. As shown in Equation 1, ITE is defined as the change of $Y_0$ and $Y_1$, while keeping the covariates $X$ unchanged (i.e., condition on those covariates). For an instance $i$ with covariates $X_i$, its corresponding ITE is

$$ITE(X_i) = \mathbb{E}[Y_1|X_i] - \mathbb{E}[Y_0|X_i] \tag{1}$$

where:

- $X_i$: represents the covariates or features for the specific instance $i$. Covariates are the characteristics or attributes of the individual or unit being considered. They can include demographic information (like age, gender, income), historical data, or any other relevant factors that might influence the outcome. The covariates $X_i$ help control for external factors that could affect the outcome. By conditioning on these covariates, the analysis aims to isolate the effect of the treatment from other influences.
- $Y_1$: denotes the outcome variable when the individual receives the treatment. For example, if the treatment is a new medication, $Y_1$ would represent the health outcome (such as reduced symptoms or improved recovery time) for the individual if they take that medication. $Y_1$ is crucial for determining the positive effects of the treatment. It represents the scenario where the treatment is applied, allowing for an evaluation of its effectiveness.
- $Y_0$: represents the outcome variable when the individual does not receive the treatment. Using the medication example again, $Y_0$ would represent the health outcome for the individual if they do not take the medication. $Y_0$ is essential for understanding what happens without the treatment. Comparing $Y_1$ and $Y_0$ provides insights into the impact of the treatment.
- $\mathbb{E}[Y_1|X_i]$: This term represents the expected outcome $Y_1$ (the outcome if the individual receives the treatment) given the individual's covariates $X_i$. It answers the question: "What is the predicted outcome if the treatment is applied to this specific individual?"
- $\mathbb{E}[Y_0|X_i]$ : This term represents the expected outcome $Y_0$ (the outcome if the individual does not receive the treatment) given the same covariates $X_i$. It answers the question: "What is the predicted outcome if the treatment is not applied to this specific individual?"

## 2.3 Average Treatment Effect (ATE)

Since only a single potential outcome can be observed, estimating the effect at the individual level is exceedingly difficult. A more practical approach is to assess the treatment effect at the average level [23]. As shown in Equation 2, ATE for a given feature is determined by averaging the ITE values associated with that feature [23]. The feature with the highest ATE is likely to have the most significant causal impact for the specified treatment. The ATE for a feature can be computed as follows:

$$ATE = \mathbb{E}[Y_1 - Y_0] \tag{2}$$

where:

- ATE($x$) denotes the Average Treatment Effect for the feature $x$.
- $\mathbb{E}$: The expected value operator. It indicates that we are calculating the average of the treatment effect across all individuals in the population.
- $Y_1$: The potential outcome when the individual receives the treatment. This represents the outcome of interest (e.g., health improvement, performance increase) if the treatment is applied.
- $Y_0$: The potential outcome when the individual does not receive the treatment. This represents what would happen without the treatment.

## 2.4 Uplift tree classifier

An uplift tree classifier is a decision tree designed to predict the incremental impact of an intervention or treatment, such as whether showing an advertisement leads to higher conversions compared to no advertisement. Unlike traditional decision trees, uplift trees focus on maximizing the difference in outcomes between treatment and control groups. It helps to determine the causal effects of a treatment by comparing the potential responses of individuals if they receive the treatment versus, if they do not [19]. The

`UpliftTreeClassifier` is a component of the *CausalML* Python library. In our approach, `UpliftTreeClassifier` is used to identify the causal impact of various factors (such as, weather conditions, operational changes, or train schedules) on transit delays. The classifier can differentiate between conditions that genuinely cause delays and those that are merely associated with them [13].

`UpliftTreeClassifier` supports the estimation of ITE, which evaluates the effect of a treatment on specific instances, and ATE, which determines the overall treatment impact across the dataset. Utilizing these metrics allows for a more detailed analysis of the influence of various features on treatment outcomes, improving the accuracy of causal conclusions. The uplift tree classifier tries to determine the best split at each node by maximizing the difference in outcomes between the treatment and control groups. Here's the mathematical formulation [18]:

(1) **Divergence at a node:** At each node of the tree, divergence between the treatment and control group outcome is calculated. Two possible measures of divergence are Kullback-Leibler (KL) Divergence and Squared Euclidean Distance. The KL divergence between treatment and control distributions at node $j$ is defined as:

$$D_{\mathrm{KL}}(P^T(Y), P^C(Y)) = \sum_i P^T(Y_i) \log\left(\frac{P^T(Y_i)}{P^C(Y_i)}\right) \tag{3}$$

where $P^T(Y_i)$ and $P^C(Y_i)$ represent the probabilities of outcome $i$ in the treatment and control groups, respectively.

(2) **Conditional divergence after a split:** When a node is split into children nodes, the conditional divergence is calculated for each child node $a$. The conditional divergence is defined as:

$$D(T|A) = \sum_a \frac{N(a)}{N} D(P^T(Y|a), P^C(Y|a)) \tag{4}$$

where $N(a)$ is the number of instances of child node $a$, and $D$ is the divergence measure (either KL divergence or squared Euclidean distance).

(3) **Maximizing Gain:** To find the optimal split, the gain in divergence is maximized, which is , defined as the difference between the divergence at the parent node and the weighted sum of divergences at the children nodes:

$$\text{Gain} = D_{\mathrm{KL}}(P^T(Y), P^C(Y)) - D(T|A) \tag{5}$$

The split that maximizes this gain is chosen at each node of the tree.

These equations allow the uplift tree classifier to identify splits that best separate the treatment and control groups, highlighting where the intervention has the most significant impact.

## 3 Related work

Forecasting of train delays has seen considerable development through various methodologies, ranging from stochastic models to advanced machine learning techniques. This review provides an overview of significant contributions in this area.

### 3.1 Stochastic models for delay prediction

Carey and Kwiecinski's [2] stochastic approach for predicting carryover delays was one of the first attempts to model the knock-on effects of train delays. Their method laid the foundation for later works like Yuan J's enhanced probability analysis [26], which improved the accuracy of delay forecasts by incorporating blocking-time theory. These models, while effective in specific scenarios, often lack the flexibility to handle real-time data and dynamic network changes.

### 3.2 Advanced machine learning in rail delay forecasting

In the evolution towards more dynamic methods, Peters et al. [17] introduced an intelligent forecasting tool leveraging real-time delays, and Yaghini [24] explored artificial neural networks (ANNs) for passenger train delay prediction in Iran. Both studies underscore the value of machine learning, though their focus was primarily on prediction rather than understanding the causal relationships between variables.

### 3.3 Limitations with exisiting works

While several statistical and machine learning approaches have been employed [6, 7, 15], many existing methods fail to offer event-level insights necessary for actionable interventions. The growing body of research calls for innovative techniques that not only predict delays but also identify underlying causes, which remains under-explored.

### 3.4 Need for causal machine learning

To bridge this gap, our study introduces causal machine learning techniques [3, 5, 8, 27], which aim to go beyond correlation and offer insights into the causal mechanisms behind train delays. This method allows for a deeper understanding of how different factors—such as route, operational status, and temporal variables—affect delays, providing a more robust tool for decision-makers in rail management.

By critically examining the current literature, we identify a significant opportunity to enhance rail delay prediction using causal models. This approach not only improves predictive accuracy but also offers actionable insights into the factors that drive delays, addressing a long-standing challenge in the field. Figure 2 shows the causal graph for train delay prediction. Here, $Y$ = *Outcome (delay)*, $X$ = Covariates (date_of_travel, route_of_train, stop_ sequence, operational status of train) and $T$ = *Treatment (spatial,temporal).*

## 4 Problem definition

The problem of rail transit delay prediction using causal machine learning (ML) involves estimating the impact of interventions (e.g., schedule changes, track maintenance) on delays. The primary objective is to predict delays by assessing the causal effect of these interventions on the outcome $Y$ (the delay), considering various covariates $X$ (e.g., date_of_travel, route_of_train, stop_ sequence, operational status of train). In mathematical terms, we define $Y_i$ as the delay for unit $i$, $T_i$ as the binary treatment indicator (whether an intervention was applied), and $X_i$ as a vector of covariates for

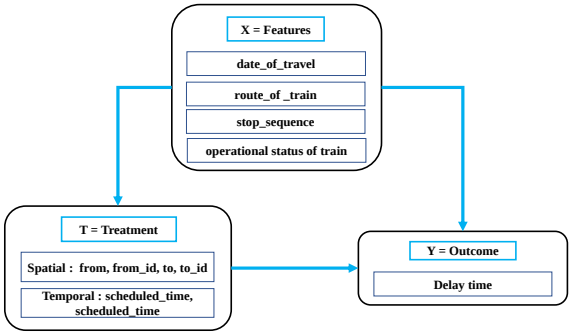


**Figure 2: Causal graph for delay prediction**

unit $i$. The key metrics to estimate are the ATE and ITE. ATE is the expected difference in outcomes between treated and untreated units, while ITE is the difference in outcomes for a specific unit. The approach involves several steps: preprocessing the dataset, sampling to ensure balanced representation, performing causal feature engineering to construct relevant features, and training models to estimate ATE and ITE. Feature analysis is then conducted to select the most significant features, followed by retraining the model to enhance delay prediction accuracy. This framework leverages causal ML to make informed predictions about rail transit delays.

## 5 Dataset

The dataset "NJ Transit Amtrak NEC Performance" is a trip transit dataset focusing on rail delays within the New Jersey Transit (NJ Transit) and Amtrak services along the Northeast Corridor (NEC). The dataset is sourced from publicly available data provided by the relevant transit authorities and is hosted on Kaggle[1]. It includes detailed records of train schedules, delays, and other performance metrics which are crucial for developing and validating predictive models in the context of rail transit.

The dataset spans from January 1, 2018, to December 31, 2020. This three-year period provides a comprehensive view of rail transit performance over a significant time frame, enabling robust analysis and forecasting. The dataset comprises approximately 1,500,000 rows. Each row represents an individual train operation, including records of scheduled and actual departure and arrival times, as well as calculated delays. Table 1 describes the key attributes of the dataset. Table 2 shows the snapshot of the New Jersey Transit Dataset.

## 6 Our proposed approach

Figure 3 shows our proposed framework. Let $Y$ represent the rail transit delay (the outcome we wish to predict) and let $T$ denote the treatment or intervention whose effect we want to estimate on $Y$. $X$ represents a set of covariates or features that may include factors like time of day, weather conditions, or train characteristics.

The goal is to predict rail transit delays by estimating the causal effect of various interventions (e.g., schedule changes, track maintenance) using causal machine learning techniques. Specifically, we aim to estimate ATE and ITE on the outcome $Y$ using the covariates $X$.

[1]https://www.kaggle.com/datasets/pranavbadami/nj-transit-amtrak-nec-performance

**Table 1: Feature description for NJ transit dataset**

| Feature | Description |
|---|---|
| Date | The date on which the train operation occurred. |
| train_id | A unique identifier for each train service. |
| stop_sequence | The sequence number indicating the order of stops for a train. |
| from | The station where the train originates or departs from. |
| from_id | A unique identifier for the departure station. |
| to | The destination station where the train is scheduled to arrive. |
| to_id | A unique identifier for the arrival station. |
| scheduled_time | The planned departure or arrival time according to the schedule. |
| actual_time | The actual recorded departure or arrival time. |
| delay_minutes | The time difference in minutes between the scheduled and actual time, representing the delay. |
| status | The operational status of the train |
| line | The train line or route that the train is . following |
| type | The type of train service |

**Table 2: A snapshot of the NJ Transit Amtrak NEC Performance dataset**

| Attribute | Record 1 | Record 2 |
|---|---|---|
| Date | 2024-09-01 | 2024-09-01 |
| train_id | 12345 | 63 |
| stop_sequence | 1 | 3 |
| from | New York | Philadelphia |
| from_id | 105 | 1 |
| to | Metropark | Newark Airport |
| to_id | 83 | 37953 |
| scheduled_time | 08:00 | 12:00 |
| actual_time | 08:05 | 12:10 |
| delay_minutes | 5 | 10 |
| status | Delayed | Cancelled |
| line | Northeast Corrdr | Northeast Corrdr |
| type | NJ Transit | NJ Transit |

### 6.1 Modeling framework

**The outcome variable** $Y_i \in \mathbb{R}$ represents the delay for rail transit unit $i$. The **treatment variable** $T_i \in \{0, 1\}$ indicates whether a particular intervention or treatment was applied, where $T_i = 1$ if the

intervention was applied and $T_i = 0$ otherwise. The **covariates** $X_i \in \mathbb{R}^p$ form a vector of $p$ observed features for rail transit unit $i$. When interventions are applied to features $X_i$ in a causal model, if $T_i = 1$ (indicating an intervention), the feature is adjusted to reflect the treatment's impact. For example, if $X_i$ represents the train schedule and $T_i = 1$ corresponds to a schedule change, $X_i$ would be replaced with the new schedule time. This adjustment helps estimate the treatment effect by comparing outcomes under treatment ($T_i = 1$) and no treatment ($T_i = 0$). This approach estimates the ITE and ATE, providing insights into how interventions impact outcomes like rail transit delays. The goal of causal inference is to calculate ATE and ITE as explained in Sections 2.3 and 2.2.

The steps involved in our framework are summarized below.

- Use causal feature engineering to identify and encode relevant features $X$.
- Train models to estimate ATE and ITE using observational data.
- Retrain the model with selected top features to improve predictive performance.

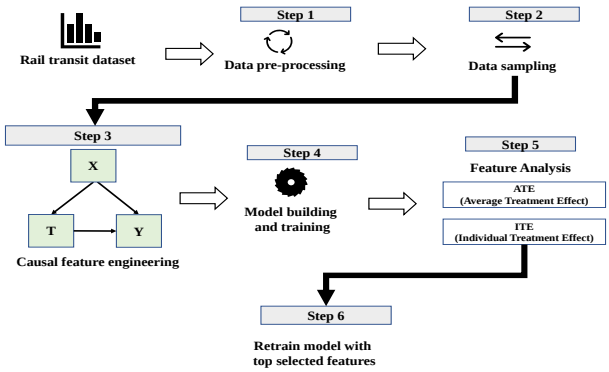


**Figure 3: Proposed framework**

## 6.2 Algorithm

Algorithm 1 involves several key steps for applying causal ML to the rail transit dataset. It begins with Step 1, by taking a rail transit dataset with covariates (e.g., train features), treatment (e.g., interventions like schedule changes), and outcome (e.g., delays) as input, aiming to output predicted delays and causal effects such as ATE and ITE. After data preprocessing to clean and encode the dataset, it samples the data to ensure a balanced representation of treated and untreated units in Step 4 and so on for each step of the algorithm. Relevant features influencing treatment and outcome are identified through causal feature engineering. The algorithm then trains a causal machine learning model to estimate ATE and ITE, providing insights into the causal impact of different features. Significant features are analyzed based on their ATE values, and the top ones are selected for model retraining to improve predictive accuracy. The retrained model is used to predict rail transit delays, enhancing accuracy by focusing on the most influential factors.

Detailed steps of the algorithm 1 are explained in Sections 6.3, 6.4, 6.5, 6.6, 6.7, and 6.8.

**Algorithm 1** Rail Transit Delay Prediction using Causal ML

1: **Input:** Rail transit dataset with covariates $X$, treatment $T$, and outcome $Y$
2: **Output:** Predicted rail transit delays and causal effects (ATE, ITE)
3: **Step 1: Data Preprocessing**: Clean and preprocess the dataset to ensure consistency and quality:
$D \leftarrow \text{load\_data}(\text{"}rail\_transit\_dataset.csv\text{"})$
4: Handle missing values and encode categorical features:
$D' \leftarrow \text{preprocess}(D)$
5: **Step 2: Data Sampling**: Perform data sampling to ensure a balanced representation of treated and untreated units:
$D_{\text{sampled}} \leftarrow \text{sample\_data}(D')$
6: **Step 3: Causal Feature Engineering**: Identify and encode relevant features $X$ that are hypothesized to influence the treatment $T$ and outcome $Y$:
$X_{\text{causal}} \leftarrow \text{causal\_feature\_engineering}(D_{\text{sampled}})$
7: **Step 4: Model Building and Training**: Train causal machine learning models to estimate ATE and ITE.
8: Train the model using $X_{\text{causal}}, Y, T$:
$\hat{f}(X, T) \leftarrow \text{CausalModel}(X_{\text{causal}}, Y, T)$
9: **Step 5: Feature Analysis**: Analyze and select the most significant features based on their estimated effects on $Y$:
$F_{\text{top}} \leftarrow \{f \mid \text{ATE}(f) > \tau\}$
10: **Step 6: Model Retraining**: Retrain the model using the top selected features to improve predictive performance:
$\hat{f}_{\text{retrained}} \leftarrow \text{CausalModel}(X_{F_{\text{top}}}, Y, T)$
11: Evaluate the retrained model's performance and predict rail transit delays.

## 6.3 Data pre-processing

Prior to model training, raw rail transit data undergoes several preprocessing steps to ensure the model's effectiveness and efficiency. The preprocessing pipeline includes the following stages:

*6.3.1 Outlier detection and removal.* Outliers in the dataset are identified and removed to prevent their disproportionate influence on the model. This step helps in maintaining the accuracy and reliability of the model's predictions.

*6.3.2 Handling missing values.* In the dataset, missing values were initially labeled as `"Unknown"` or left blank. To standardize the dataset, all missing values are categorized as `"Unknown"`. Specifically, since all variables in the dataset are categorical, missing values are replaced with the `"Unknown"` category. This approach ensures that the model can effectively handle these values without introducing bias.

*6.3.3 Categorical variable transformation.* The dataset, consisting entirely of categorical variables, requires transformation into a suitable format for model training. To facilitate this, the data is converted into a binary format using one-hot encoding. This process is implemented using Pandas' `get_dummies` function, which creates binary dummy variables for each category of the categorical variables. This transformation enables the model to interpret and process categorical data effectively.

*6.3.4 Data integration.* After handling missing values and converting categorical variables to binary format, the processed data is integrated into a unified dataset, ready for causal machine learning model training. This ensures that the data is in a consistent and machine-readable format, which is crucial for accurate model performance.

By following these preprocessing steps, the dataset is prepared for efficient and effective training of the causal ML model, enhancing its ability to accurately predict rail transit delays.

## 6.4 Data sampling

To ensure the relevance and quality of the data, the following criteria were applied during the sampling process:

- **Time frame:** Data from the dataset spanning the most recent two years were selected to ensure that the performance metrics reflect current operational conditions.
- **Data completeness:** Only records with complete and valid entries were included. Missing or corrupted data entries were excluded to maintain the integrity of the analysis.
- **Operational metrics:** Records including critical performance indicators such as delays, on-time performance, and train schedules were prioritized. Non-essential metrics were excluded to focus on key performance aspects.
- **Geographic scope:** Data from all relevant transit lines within the NJ Transit and Amtrak NEC networks were considered to provide a comprehensive view of the transit performance.

## 6.5 Causal feature engineering

To utilize causal machine learning techniques on the dataset, it is divided into three key elements: $X$, $Y$, and $T$. The $X$ element includes various covariates like weather conditions, address specifics, and vehicle attributes, offering essential contextual and environmental information. The $Y$ element represents the outcome of interest, specifically train delays, which the causal analysis aims to understand or predict. The $T$ element encompasses treatment variables, categorized into spatial and temporal factors, which are investigated for their impact on the outcome. This approach enhances prediction performance by focusing on the most relevant factors influencing rail transit delays. Specifically, this involves identifying features that have the highest ATE, such as "stop_sequence" and "actual_time," which show the strongest influence on delays. By filtering out less significant features and concentrating on these key factors, the model can more effectively capture the causal relationships that drive delays. This approach reduces noise and irrelevant data, leading to improved predictive accuracy.

## 6.6 Model building and training

After the data pre-processing stage, the dataset is trained using uplift tree classifier [19].

## 6.7 Feature analysis

The feature analysis is divided into two parts:

- ATE (Average Treatment Effect)
- ITE (Individual Treatment Effect)

*6.7.1 ATE (Average Treatment Effect).* The Table 5 presents the ATE estimates for various features affecting rail transit delays:

**Table 3: ATE values of the features in NJ transit dataset**

| Feature name | ATE values |
|---|---|
| stop_sequence | 0.98 |
| actual_time | 0.97 |
| scheduled_time | 0.88 |
| line | 0.85 |
| status | 0.84 |
| to | 0.77 |
| type | 0.71 |
| Date | 0.70 |
| to_id | 0.66 |
| from | 0.64 |
| from_id | 0.61 |
| train_id | 0.59 |

*6.7.2 Key Highlights.* The ATE estimates for various features impacting rail transit delays reveal the following key insights:

- **stop_sequence (ATE: 0.98):** This feature has the highest ATE value, indicating that the order in which a train stops significantly affects the outcome. This is likely due to the cumulative effect of stopping at multiple stations impacting delays or performance.
- **actual_time (ATE: 0.97):** The actual time at which a train departs or arrives is crucial in determining delays or performance outcomes, as deviations from the schedule are directly observed here.
- **scheduled_time (ATE: 0.88):** Scheduled time is also highly significant, reflecting the planned schedule's influence on the performance metrics. However, its effect is slightly less pronounced than the actual time.
- **line (ATE: 0.85):** Different train lines or routes have varying performance characteristics or constraints, making this feature important in explaining differences in outcomes.
- **status (ATE: 0.84):** The operational status directly impacts the outcome, with different statuses showing significant variations in performance metrics.
- **to (ATE: 0.77):** The destination station affects performance, possibly due to varying conditions or handling at different stations.
- **type (ATE: 0.71):** The type of service affects performance, with express services potentially having different performance characteristics compared to local services.
- **Date (ATE: 0.70):** The date of operation shows a moderate effect, possibly reflecting seasonal or time-based variations in performance.
- **to_id (ATE: 0.66):** The specific arrival station ID affects performance outcomes, though its impact is less than some other features.
- **from (ATE: 0.64):** The departure station also impacts performance, though less so than the destination station and other features.

- **from_id (ATE: 0.61):** The departure station ID has a lower effect compared to other features, indicating that while it is a factor, its impact is less pronounced.
- **train_id (ATE: 0.59):** The train ID has the lowest ATE among the features listed, suggesting it has the least direct effect on performance compared to other factors.

The features with the highest ATE values, such as stop_sequence and actual_time, have the most substantial impact on the outcomes in the NJ Transit dataset. Features like scheduled_time, line, and status also significantly influence performance. In contrast, features such as train_id have a relatively lower impact. This ordering helps prioritize the features that are most critical for understanding and predicting outcomes related to train performance.

*6.7.3 ITE for Rail Transit Features.* Table 4 presents the ITE of the top 5 features. These values represent the impact of changes in each feature on delay predictions for individual train trips.

| Sr. no. | Feature name | ITE key | ITE value |
|---|---|---|---|
| 1 | stop_sequence | 1 | 0.77 |
| | | 6 | 0.87 |
| | | 18 | 0.98 |
| | | 12 | 0.88 |
| 2 | actual_time | 02-04-2018 06:41 | 0.97 |
| | | 02-03-2018 01:21 | 0.88 |
| 3 | scheduled_time | 02-04-2018 06:41 | 0.87 |
| | | 07-04-2018 01:21 | 0.88 |
| 4 | line | Northeast Corrdr | 0.85 |
| | | Amtark | 0.86 |
| 5 | status | cancelled | 0.88 |
| | | departed | 0.84 |
| | | estimated | 0.83 |

**Table 4: ITE values for NJ transit dataset**

The Table 4 provides crucial insights into the factors affecting rail transit delays, specifically highlighting how different features in the NJ Transit dataset contribute to these delays through their ITE values. The important findings are as follows:

(1) The stop_sequence feature stands out as a significant predictor of delays, with ITE values ranging from 0.77 to 0.98. The highest ITE value of 0.98 is associated with the stop sequence key of 18, indicating that certain stop sequences have a strong impact on the likelihood or extent of delays. This suggests that the position of a stop within the sequence could influence delays, possibly due to congestion or scheduling issues.
(2) Actual_time is another critical factor influencing delays, with ITE values of 0.97 and 0.88 for specific timestamps. The higher ITE value of 0.97 for the timestamp "02-04-2018 06:41" highlights the importance of the exact time when a train operates in predicting delays. This suggests that delays could be more likely during specific times, possibly due to peak travel periods or operational challenges.
(3) Scheduled_time, while closely related to actual time, also shows significant ITE values (0.87 and 0.88), indicating that the planned schedule is an important factor in delay prediction. Consistent ITE values for different scheduled times suggest that discrepancies between planned and actual operations could be a key driver of delays.
(4) The train line, particularly "Northeast Corridor" and "Amtrak," shows ITE values of 0.85 and 0.86. This indicates that the specific route or line a train operates on can influence delays, potentially due to the varying infrastructure, traffic, or operational conditions associated with different lines.
(5) The status of the train, whether it is "cancelled," "departed," or "estimated," reveals ITE values ranging from 0.83 to 0.88. Cancellations have the highest ITE value of 0.88, underscoring that trains which are canceled are strongly associated with delay outcomes. Even trains that have departed or are estimated to depart show significant ITE values, indicating that operational status is a critical indicator of potential delays.

## 6.8 Retraining model

The frequency and duration for retraining a model are important considerations for maintaining accuracy and relevance. Rail transit systems often experience frequent changes, such as schedule updates, new routes, and alterations in operational procedures. Therefore, it is essential to retrain the model with the most recent data to ensure it captures these changes effectively. Additionally, as the model encounters new data, it can refine its understanding of the relationships between different features and delay outcomes, thereby enhancing forecasting accuracy. Another important factor is the phenomenon known as concept drift, where the nature of delays and operational disruptions may evolve over time, leading to shifts in the underlying data distribution. Regular retraining allows the model to adjust to these changes, ensuring continued effectiveness. Moreover, incorporating new features into the model can further improve its predictions; as new data regarding train status or route-specific information becomes available, retraining is necessary. Overall, consistent model updates are vital in a dynamic environment like rail transit to improve predictive performance and adapt to evolving operational realities, ultimately ensuring more reliable service for passengers. The feature rankings based on the ATE values are shown in Table 5

**Table 5: Top ranked features of New Jersey transit dataset according to ATE score**

| Feature rank | Feature |
|---|---|
| 1 | stop_sequence |
| 2 | actual_time |
| 3 | scheduled_time |
| 4 | line |
| 5 | status |

Scheduled Departure Time is a crucial feature that denotes when a train is planned to leave a station, helping to understand delays relative to schedules and identify patterns based on departure times. Actual Departure Time indicates when the train actually departs, fundamental for calculating delays and assessing service performance. Scheduled Arrival Time represents when the train is expected to arrive, essential for evaluating reliability. Actual Arrival Time shows when the train reaches its destination, critical for

determining actual delay duration. Lastly, the Route identifies the train's path, valuable for analyzing route-specific issues affecting overall performance.

Incorporating features from Table 5 improves the model's delay forecasting accuracy and enhances operational efficiency in rail transit systems.

## 7 Experimental setup

The experiments were conducted using Python 3.12.3 on a server equipped with a 3.31 GHz Intel(R) Xeon(R) CPU and 16 GB of RAM. We employed the CausalML library [13, 27] for implementing causal machine learning techniques. This library offers various uplift modeling and causal inference methods, utilizing advanced algorithms based on contemporary research. Key features include uplift modeling for estimating causal impacts, causal inference methods for identifying relationships within the dataset, and integration of state-of-the-art techniques. This setup ensures a comprehensive analysis of train delays and causal relationships using modern causal ML methods.

## 8 Results

The analysis of ATE estimates for various features impacting rail transit delays highlights the key factors influencing train performance as shown in Figure 4. The ATE values indicate the relative importance of each feature in predicting delays or other performance outcomes. Stop feature (ATE: 0.98) emerges as the most influential

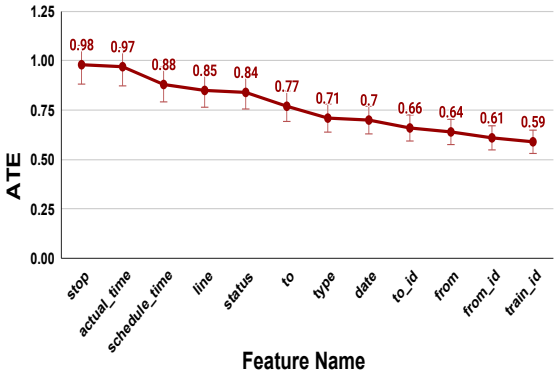


**Figure 4: Feature vs ATE scores**

factor. This suggests that the order of stops plays a crucial role, likely due to the cumulative impact of multiple stops on the train's schedule adherence. As trains make more stops, the likelihood of delays increases, making this a critical factor in performance analysis. Actual Time (ATE: 0.97) is another highly significant feature. This reflects the importance of when a train actually departs or arrives, as deviations from the schedule are directly observed. It highlights how real-time operations, rather than just planned schedules, are pivotal in determining train performance. Scheduled Time (ATE: 0.88), while still important, has a slightly lower impact than actual time. This shows that while the planned schedule is essential, the actual execution of the schedule (actual time) has a more significant effect. Line (ATE: 0.85) and Status (ATE: 0.84) are also critical, indicating that different train routes and operational statuses introduce varying performance dynamics. These features help explain the differences in delays or other performance metrics across different lines and operational conditions. Less influential factors include destination (ATE: 0.77) and Type of Service (ATE: 0.71), which still play roles in determining outcomes, but to a lesser extent. The date feature (ATE: 0.70) of operation also has a moderate impact, likely reflecting seasonal variations. Finally, features like Train ID (ATE: 0.59) have the least impact, indicating that while individual train characteristics matter, they are not as crucial as the operational and scheduling factors.

When comparing the graphs for the years 2018 (Figure 5), 2019 (Figure 6), and 2020 (Figure 7), a clear pattern emerges showing that as the number of stops increases, the average train delay time also tends to rise. However, there are some differences in how this trend evolves across the years.

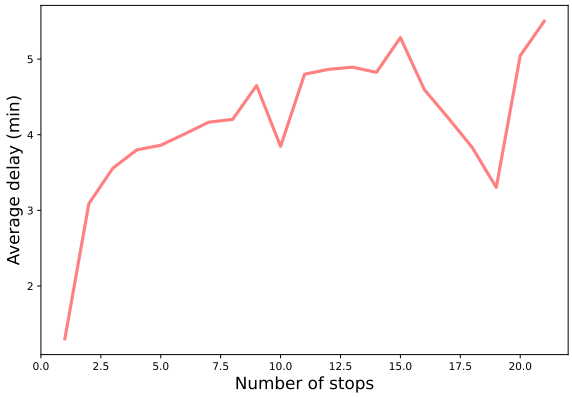


**Figure 5: Average train delay (min) vs, number of stops graph for the year 2018**

The graph in Figure 5 shows the relationship between average train delay time (in minutes) and the number of stops for the year 2018. The trends show an initial increase in delay time as the number of stops increases from 0 to approximately 10 stops. This suggests that adding stops generally contributes to longer delays, likely due to the accumulation of minor delays at each stop. After reaching a peak delay time of around 5 minutes at 10 to 12 stops, the trend becomes more variable. Interestingly, after 12 stops, the delay time slightly decreases until around 17 stops, where it drops to its lowest point, approximately 2 minutes. This drop might be due to the optimization of train schedules on longer routes or the presence of express services that skip certain stops, reducing overall delay. However, after 17 stops, there is a sharp increase in delay time, rising to over 5 minutes by the time the train reaches 20 stops. This indicates that beyond a certain threshold, adding more stops significantly impacts the overall delay, possibly due to compounded operational inefficiencies. The graph for the year 2019 (Figure 6) follows a similar initial pattern to 2018, with delays increasing

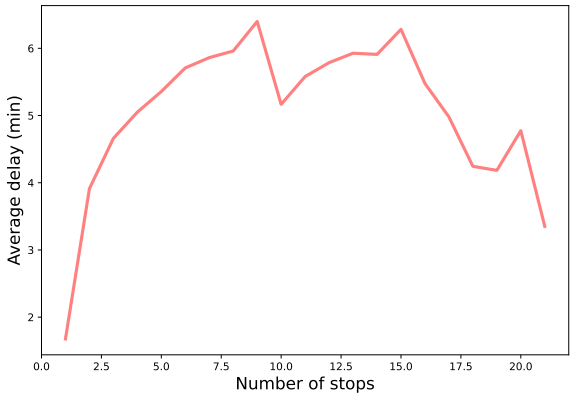


**Figure 6: Average train delay (min) vs number of stops graph for the year 2019**

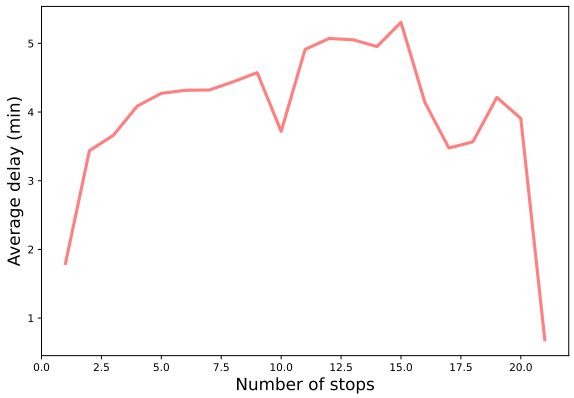


**Figure 7: Average train delay (min) vs number of stops graph for the year 2020**

as the number of stops rises to 10–12 stops, peaking at around 6 minutes. After 12 stops, the delay time fluctuates slightly, remaining above 3 minutes, before climbing again sharply after 17 stops. The delay time reaches approximately 6 minutes by 20 stops, showing a higher maximum delay than in 2018. In 2020 (Figure 7), there is a more gradual increase in delay time up to 10 stops, with a slower rise compared to the previous two years, peaking around 5 minutes. Beyond 12 stops, there is less fluctuation in delay time, remaining between 4 and 5 minutes up until 17 stops. The rise in delay time after 17 stops is more moderate than in 2019, reaching approximately 5 minutes at 20 stops.

All three years show an initial rise in delay time as the number of stops increases, peaking around 10 to 12 stops. The maximum delay in 2019 is higher than in both 2018 and 2020. After 12 stops, there are variations in how delay time behaves, with 2018 showing a sharper dip and rise, 2019 maintaining a higher delay, and 2020 showing more stability but still increasing.

Overall, the dependency of train delay on the number of stops reflects an initial increase, a mid-range optimization or efficiency, followed by a sharp increase as the number of stops becomes excessive.

## 8.1 Seasonal trends

The Figure 8 shows the percentage of delays each month, highlighting potential trends:

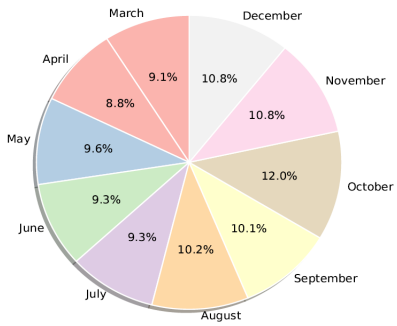


**Figure 8: Percentage rail transit delay seasonal trends for the year 2018**

October experiences the highest delay percentage at 12%, likely due to increased fall travel and adverse weather. August and September follow closely with 10.2% and 10.1% delays, possibly linked to summer's end, which sees more travel and maintenance activities. December and November show 10.8% delays, indicating potential disruptions from winter weather like snow and ice. Conversely, March and April have lower delay percentages of 9.1% and 8.8%, suggesting fewer disruptions due to favorable spring weather. Meanwhile, May, June, and July present moderate delay percentages around 9.3%, reflecting balanced impacts from summer challenges and maintenance.

This monthly breakdown allows the model to incorporate seasonal variation into its prediction of rail transit delays, enabling more accurate and context-sensitive forecasts for different times of the year. These insights help in understanding when delays are more likely, guiding both operational improvements and policy decisions.

## 8.2 Comparison with baselines

The Table 6 presents accuracy metrics of a model trained using different ML approaches like XGBoost (XGB), Random Forest (RF), Support Vector Machine (SVM), and UpliftTreeClassifier. The table compares the accuracy of various machine learning classifiers trained with and without feature selection in a causal ML context, specifically using an uplift tree classifier. It shows four accuracy values: 93.4% for XGBoost, 91.89% for Random Forest, and 90.56% for Support Vector Machine when not using selected features. In contrast, the uplift tree classifier achieved the highest accuracy of 95.6% when trained with selected features. This indicates that feature selection significantly improves model performance, particularly with the uplift tree classifier, highlighting its potential effectiveness in causal analysis applications.

**Table 6: Accuracy comparison of classifiers with and without feature selection**

| Accuracy without selecting features when trained using XGB | Accuracy without selected features when trained using RF | Accuracy without selecting features when trained using SVM | Accuracy with selected features when trained using UpliftTreeClassifier |
|---|---|---|---|
| 93.4% | 91.89% | 90.56% | **95.6%** |

## 9 Conclusion

In analyzing rail transit delays with causal machine learning, we identified the top five features from the NJ Transit dataset: stop_sequence, actual_time, scheduled_time, line, and status. This was achieved through data preprocessing, sampling, causal modeling, and feature analysis using an uplift tree classifier. Our analysis of ITE and ATE revealed that stop_sequence and actual_time are the most impactful on delays. Scheduled_time, line, and status also significantly influence delays, underscoring their importance in predictive modeling. Delays were notably higher during peak times, indicating a need for capacity adjustments or operational improvements. Variability in delays across routes suggests that targeted interventions could be more effective than system-wide changes. Seasonal and weather factors further contribute to delays, highlighting the need for better weather forecasting and contingency planning. Operational elements like train schedules and maintenance also affect delays. Retraining the model with these features enhances predictive accuracy, emphasizing the importance of feature selection and causal inference in improving transit performance. Future research could explore infrastructure or specific train characteristics to deepen understanding. Our approach not only advances methodological techniques but also provides actionable insights for improving rail transit performance. Future research could delve deeper into infrastructure issues or specific train characteristics to further understand the dynamics of transit delay performance.

## References


[1] Jason Abrevaya, Yu-Chin Hsu, and Robert P Lieli. 2015. Estimating conditional average treatment effects. *Journal of Business & Economic Statistics* 33, 4 (2015), 485–505.

[2] Malachy Carey and Andrzej Kwieciński. 1994. Stochastic approximation to the effects of headways on knock-on delays of trains. *Transportation Research Part B: Methodological* 28, 4 (1994), 251–267.

[3] Victor Chernozhukov, Christian Hansen, Nathan Kallus, Martin Spindler, and Vasilis Syrgkanis. 2024. Applied causal inference powered by ML and AI. *arXiv preprint arXiv:2403.02467* (2024).

[4] Francesco Corman and Pavle Kecman. 2018. Stochastic prediction of train delays in real-time using Bayesian networks. *Transportation Research Part C: Emerging Technologies* 95 (2018), 599–615.

[5] Stefan Feuerriegel, Dennis Frauen, Valentyn Melnychuk, Jonas Schweisthal, Konstantin Hess, Alicia Curth, Stefan Bauer, Niki Kilbertus, Isaac S Kohane, and Mihaela van der Schaar. 2024. Causal machine learning for predicting treatment outcomes. *Nature Medicine* 30, 4 (2024), 958–968.

[6] Holger Flier, T Graffagnino, and M Nunkesser. 2009. Finding robust train paths in dense corridors. In *3rd International seminar on railway operations modelling and analysis*.

[7] Tijs Huisman, Richard J Boucherie, and Nico M Van Dijk. 2002. A solvable queueing network model for railway networks and its validation and applications for the Netherlands. *European Journal of Operational Research* 142, 1 (2002), 30–51.

[8] Jean Kaddour, Aengus Lynch, Qi Liu, Matt J Kusner, and Ricardo Silva. 2022. Causal machine learning: A survey and open problems. *arXiv preprint arXiv:2206.15475* (2022).

[9] Pavle Kecman and Rob MP Goverde. 2014. Online data-driven adaptive prediction of train event times. *IEEE Transactions on Intelligent Transportation Systems* 16, 1 (2014), 465–474.

[10] Yong-Hong Kuo, Janny MY Leung, and Yimo Yan. 2023. Public transport for smart cities: Recent innovations and future challenges. *European Journal of Operational Research* 306, 3 (2023), 1001–1026.

[11] Giorgio Medeossi, Giovanni Longo, and Stefano de Fabris. 2011. A method for using stochastic blocking times to improve timetable planning. *Journal of Rail Transport Planning & Management* 1, 1 (2011), 1–13.

[12] Sanjin Milinković, Milan Marković, Slavko Vesković, Miloš Ivić, and Norbert Pavlović. 2013. A fuzzy Petri net model to estimate train delays. *Simulation Modelling Practice and Theory* 33 (2013), 144–157.

[13] Aleksander Molak. 2023. *Causal Inference and Discovery in Python: Unlock the secrets of modern causal machine learning with DoWhy, EconML, PyTorch and more* (1. ed.). Packt Publishing, Birmingham. https://amzn.to/3RebWzn.

[14] Rahul Nair, Thanh Lam Hoang, Marco Laumanns, Bei Chen, Randall Cogill, Jácint Szabó, and Thomas Walter. 2019. An ensemble prediction model for train delays. *Transportation Research Part C: Emerging Technologies* 104 (2019), 196–209.

[15] Nils OE Olsson and Hans Haugland. 2004. Influencing factors on train punctuality—results from some Norwegian studies. *Transport policy* 11, 4 (2004), 387–397.

[16] Judea Pearl. 2010. An introduction to causal inference. *The international journal of biostatistics* 6, 2 (2010).

[17] Jan Peters, Bastian Emig, Marten Jung, and Stefan Schmidt. 2005. Prediction of delays in public transportation using neural networks. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, Vol. 2. IEEE, 92–97.

[18] Piotr Rzepakowski and Szymon Jaroszewicz. 2010. Decision trees for uplift modeling. In *2010 IEEE International Conference on Data Mining*. IEEE, 441–450.

[19] Piotr Rzepakowski and Szymon Jaroszewicz. 2012. Decision trees for uplift modeling with single and multiple treatments. *Knowledge and Information Systems* 32 (2012), 303–327.

[20] Malte Schmidt, Norman Weik, Stephan Zieger, Anke Schmeink, and Nils Nießen. 2019. A generalized stochastic Petri net model for performance analysis of trackside infrastructure in Railway Station Areas under uncertainty. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 3732–3737.

[21] Thomas Spanninger, Alessio Trivella, Beda Büchel, and Francesco Corman. 2022. A review of train delay prediction approaches. *Journal of Rail Transport Planning & Management* 22 (2022), 100312.

[22] Jinjun Tang, Lanlan Zheng, Chunyang Han, Weiqi Yin, Yue Zhang, Yajie Zou, and Helai Huang. 2020. Statistical and machine-learning methods for clearance time prediction of road incidents: A methodology review. *Analytic methods in accident research* 27 (2020), 100123.

[23] Guandong Xu, Tri Dung Duong, Qian Li, Shaowu Liu, and Xianzhi Wang. 2020. Causality learning: A new perspective for interpretable machine learning. *arXiv preprint arXiv:2006.16789* (2020).

[24] Masoud Yaghini, Mohammad M Khoshraftar, and Masoud Seyedabadi. 2013. Railway passenger train delay prediction via neural network model. *Journal of advanced transportation* 47, 3 (2013), 355–368.

[25] Liuyi Yao, Sheng Li, Yaliang Li, Hongfei Xue, Jing Gao, and Aidong Zhang. 2019. On the estimation of treatment effect with text covariates. In *International Joint Conference on Artificial Intelligence*.

[26] Jianxin Yuan. 2006. *Stochastic modelling of train delays and delay propagation in stations*. Vol. 2006. Eburon Uitgeverij BV.

[27] Yang Zhao and Qing Liu. 2023. Causal ML: Python package for causal inference machine learning. *SoftwareX* 21 (2023), 101294.

[28] He Zhuang, Liping Feng, Chao Wen, Qiyuan Peng, and Qizhi Tang. 2016. High-speed railway train timetable conflict prediction based on fuzzy temporal knowledge reasoning. *Engineering* 2, 3 (2016), 366–373.