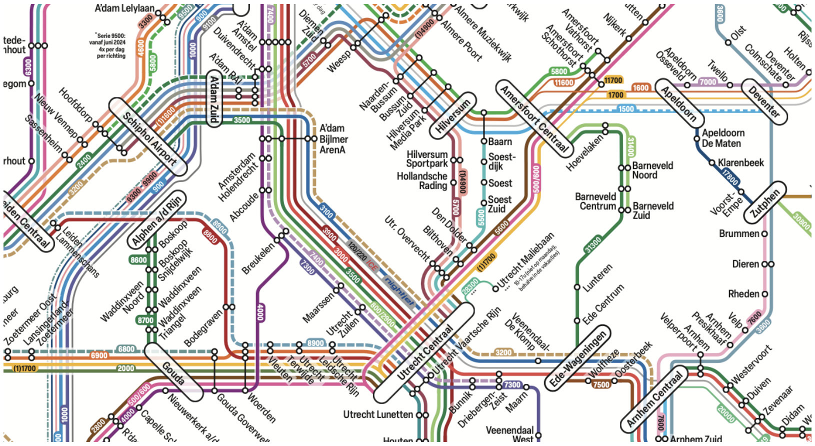
# Predicting Delayed Trajectories Using Network Features: A Study on the Dutch Railway Network

submitted in partial fulfillment for the degree of master of science

Merel Kämper
15137015

Master Information Studies
data science
Faculty of Science
University of Amsterdam

Submitted on 30.06.2024



| | UvA Supervisor | External Supervisor |
|---|---|---|
| **Title, Name** | Dr. Ali Alsahag PhD | Linda Lenkens MSc |
| **Affiliation** | University of Amsterdam | EY |
| **Email** | a.m.m.alsahag@uva.nl | Linda.Lenkens@nl.ey.com |

## ABSTRACT

The Dutch railway network is one of the busiest in the world, with delays being a prominent concern for the principal passenger railway operator NS. This research addresses a gap in delay prediction studies within the Dutch railway network by employing an XGBoost Classifier with a focus on topological features. Current research predominantly emphasizes short-term predictions and neglects the broader network-wide patterns essential for mitigating ripple effects. This research implements and improves an existing methodology, originally designed to forecast the evolution of the fast-changing US air network, to predict delays in the Dutch Railways. By integrating Node Centrality Measures and comparing multiple classifiers like RandomForest, DecisionTree, GradientBoosting, AdaBoost, and LogisticRegression, the goal is to predict delayed trajectories. However, the results reveal limited performance, especially in non-simultaneous testing scenarios, suggesting the necessity for more context-specific adaptations. Regardless, this research contributes to the understanding of transportation network evaluation and proposes future directions for developing more robust predictive models for delays.

## KEYWORDS

Machine Learning, Transportation Networks, Railway Delays, Nederlandse Spoorwegen (NS), Topological Features, Centrality Measures, Predictive Modeling

## GITHUB REPOSITORY

https://github.com/merelkaemper/MSc-Thesis-main

## 1 INTRODUCTION

Recently, the application of machine learning techniques to forecast delays [26, 49] and optimize transportation networks [50] has gained significant attention in academia and industry. These efforts are driven by the need to enhance transportation systems' efficiency, reliability, and sustainability [11, 20, 43]. With the power of data-driven approaches [42], researchers and stakeholders seek to understand the complex dynamics of transportation networks better and develop effective strategies for mitigating delays and improving overall network performance. Improving punctuality is also high on the agenda at the principal passenger railway operator in the Netherlands (NS) after it was announced that 1 in 10 journeys were delayed in 2023 [12].

Despite significant advancements in machine learning for predicting delays in transportation networks, recent studies have mostly focused on short-term predictions at the next station, typically within a 20-minute timeframe [26, 49] or optimization of train dispatching [33, 40, 51]. These approaches overlook the broader network-wide delay patterns, which are crucial for understanding and mitigating ripple effects throughout the system [16]. This research aims to address this gap by implementing and improving an existing machine-learning framework that focuses on the predictive power of topological features in forecasting the evolution of transportation networks to predict delays in the Dutch railway network. By emphasizing the significance of these network-related features in the prediction of delayed links, this research seeks to develop more comprehensive models that can capture complex interactions on the network level instead of short-term travels.

In transportation network forecasting, Lei et al. [25] have made a significant impact regarding the evolution of the fast-changing networks in the US air transport system and the Brazilian bus network. Their study primarily focuses on predicting removed links over a monthly time scale, a methodology that fits the dynamic nature of fast-changing networks. This research seeks to expand the applicability of their framework to the Dutch railway network operated by NS. Unlike the fast-paced nature of air and bus transport, the Dutch railway network exhibits a slower rate of change, with delays being a prominent concern. Therefore, this research challenges the notion that Lei et al.'s methodology is exclusively suitable for fast-changing networks and predicting removed links. By redefining 'removed links' in their study to 'delayed links' in this research, the goal is to implement their framework to better align with the operational dynamics of the Dutch railway network. This implementation and improvement will enable the investigation of the generalizability of their approach in the Dutch railway network and contribute to a broader understanding of transportation network forecasting methodologies. The redefinition of the missing link prediction is justified by the fact that, when examining delayed links, this can still be encountered as a binary classification problem. Here, the objective shifts from determining retained or removed links to discerning whether an edge is (significantly) delayed or not within a specific timeframe. Moreover, a delayed link can be seen as a missing link, as when a delay occurs, passengers must seek alternative routes to continue their travels.

Contrary to prior studies that primarily examine the impact of spatiotemporal features on train delays [18, 48, 51] or focus on predicting delays 20 minutes later [26, 49], this research emphasizes the identification of predictive network features that significantly influence delayed trajectories within the Dutch railway network. Inspired by Lei et al. [25], who highlighted the significance of topological features in predicting the evolution of transportation networks, this research shifts the focus from external variables to intrinsic network characteristics. By addressing this gap, the aim is to shed light on a less-explored aspect of delay prediction, offering valuable insights into vulnerable trajectories. When the location and cause of a delay are identified, potential new routes and stops can be analyzed. These insights could help in proactive decision-making processes and enable stakeholders to mitigate potential delays preemptively. According to B. van Zaalen (personal communication, June 4, 2024), head of Digitalisation Operations at NS, passenger punctuality is the most important operational KPI, next to seat probability. Punctuality for passengers on the main railway network was lower in 2023 than in 2022 with an average of 89.7% [2]. One of the reasons for the decline was an increase in train crowding after COVID. A lot of work is being done at NS to optimize operations and keep passenger punctuality as high as possible. Unfortunately, NS suffers a lot from a ripple effect: if one train is delayed or disrupted on a route, there is a big chance that other trains in that region will be as well. This research contributes to ongoing practices by investigating network-related features [43] as new predictors and emphasizing a scientific baseline.

Moreover, this research evaluates the effectiveness of various machine learning classifiers and feature sets by comparing them and enhancing a baseline model [25]. The ultimate goal is to test the generalizability of the baseline work and improve it to predict delayed trajectories within the Dutch railway network. By employing these methodological combinations, this research not only contributes to the broader field of transportation network optimization but also strives to enhance the efficiency and reliability of the Dutch railway network.

In the next section, related work will be outlined to understand the theoretical framework that is used in this research. After this, the methodology is extensively explained, starting with the datasets, followed by the model construction, and finishing with the validation and evaluation of this research. In the results section, the outcome of the methodology will be displayed through figures and tables. The limitations of this research and recommendations for future work will be described in the discussion. Finally, the research questions are answered in the conclusion.

### 1.1 Research Question

The main research question in this thesis is: *How can machine learning techniques, typically applied to forecast the evolution of fast-changing networks, be implemented and improved to predict delayed trajectories in the Dutch railway network operated by NS, considering the significance of network features?* To logically address this main research question, several sub-questions are outlined:

- To what extent can a well-known machine learning model like the XGBoost Classifier, used in existing methodologies for predicting removed links in fast-changing transportation networks, be implemented and improved?
- How can the improved models be constructed for predicting delayed trajectories in the Dutch railway network operated by NS?
- How do the enhanced models, applied to the Dutch railway dataset, compare to the implemented baseline models in terms of predictive performance?

## 2 RELATED WORK

The prediction of public transport delays is a much-researched topic in the data science domain. Many different countries have been subject to this investigation, for example, Belgium [41], China [17, 40], India [32], France [24], and Germany [14]. Besides that, different techniques have been used for network forecasts, of which many have been reviewed by making a distinction between event-driven and data-driven approaches [42]. Furthermore, there is a focus on climate change when it concerns studies on transportation networks [10, 20]. In recent work, the usefulness and usability of centrality measures in transportation networks in the face of climate change adaptation have been evaluated [43]. Centrality measures are a powerful tool in network theory that can be used to understand the importance of nodes in a network. The review highlights the need to reformulate these measures, because when this is done, they can properly be applied in transportation networks to expose the significance of their elements.

Numerous studies have also focused on predicting delays in the Dutch railway network, particularly concerning the primary provider NS. A significant incentive for this study stemmed from the Railroad Problem Solving Competition of 2018. During this event, a sizable dataset from NS and ProRail, comprising timetables, weather data, and delay records, was made available. The objective was to enhance the accuracy of train performance and delay forecasts about 20 minutes after a realization. Notably, the top three finalists employed distinct models for delay estimation. The winning approach used a neural network [13], the runner-up made use of a bi-level random forest method [34], and the third finalist employed non-homogeneous Markov chains [49]. At the primary level, the bi-level random forest predicts whether the current delay will decrease, increase, or remain unchanged within the next 20 minutes from the present time. At the secondary level, their model quantifies the amount of delay in minutes [34]. It is important to note that their model was significantly better at predicting the decrease of delay than the increase or equal delay. Other classifiers that were used in this study and had slightly lower accuracies than Random Forest were Gradient Boosting, SVM, Adaboost, Logistic Regression and Decision Tree. This Railroad Competition dataset has later been used by other researchers to predict near-term train delays [49] or discover influencing factors for delay propagation [26]. While these studies show better results than their baseline models, their results are not optimal, and the limitations include the failure to consider other features. The dataset of these studies is used as inspiration for this research because it contains historical data on departures, arrivals, and delays.

The methodologies of this research build upon the work of Lei et al [25]. Their study focuses on applying machine learning techniques to predict the evolution of dynamic transportation networks, specifically focusing on the US domestic air transport network and the domestic bus transport network in Brazil. A missing link prediction approach was used over a monthly time frame. The main steps in their work include comparing different classification models, testing if topological features are significant for removed edges compared to retained edges, studying the predictive potential of the topological features, testing resilience to external shocks such as the COVID-19 pandemic, and examining whether the forecast is stable over a longer period of time.

Lei et al. initially built graphs representing their transportation networks, with nodes denoting entities (airports or bus stations) and edges denoting connections (such as flights or bus routes). They extracted both weighted and unweighted topological features to capture the structural properties of their network. These features were assessed in terms of differences between retained edges and removed ones. The study tested 27 commonly used classification algorithms, including XGBoost, which emerged as the best performer based on balanced accuracy, F1 score, and ROC AUC. The resilience testing against external shocks, such as the impact of the COVID-19 pandemic, was incorporated by Lei et al. to understand the robustness of their models. They applied simultaneous and non-simultaneous testing to validate the models' ability to predict removed links both within the same time period and across different time periods. Furthermore, they used SHAP (SHapley Additive exPlanations) values to interpret the importance of features in their models, providing insights into which topological features were most influential in predicting link removals. Their results show that edge removal processes in transportation networks are not random and it is possible to make accurate predictions based on network structures. While simultaneous testing works well for both networks and the model can make accurate predictions in the same snapshot, a model trained on a single time snapshot is not able to correctly predict removed edges in different time snapshots for the Brazil Bus network.

In conclusion, this research integrates the extensively studied topic of train delay predictions with the methodologies proposed by Lei et al. Their focus on the network as a whole and the analysis of topological features addresses a critical research gap in the context of the Dutch railway network. Implementing Lei et al.'s work, which utilized XGBoost for predicting the removal of links in a fast-changing transportation network, serves as a baseline and an important validation step [35] for the application to the NS dataset. Additionally, this research leverages insights from Spanninger et al. [42] on various delay prediction methods, including event-driven and data-driven approaches. Specific classification models tailored to the NS data are evaluated for their effectiveness [34, 46]. Centrality measures are incorporated as indicators of network-wide importance, aiding in understanding delays [43]. Furthermore, this research adopts a similar approach for determining the arrival delay as a dependent variable compared to previous studies [33, 40, 51], emphasizing consistency towards and expansion of the already existing literature in this field.

# 3 METHODOLOGY

This section outlines the comprehensive methodology employed to predict delayed trajectories in the Dutch railway network using machine learning techniques from a baseline work where missing links are predicted in the US air network. The process begins with a description of the data collection and preparation stages of the NS dataset, followed by an exploratory data analysis to uncover patterns in both datasets. Subsequently, the model framework and improvements made to implement the work by Lei et al. are detailed. Finally, the validation and evaluation procedures are explained. Redoing the baseline work is an important part of the validation step of this research.

## 3.1 Datasets

Two datasets are used in this research. Firstly, the US domestic flights as used by Lei et al. [25]. This dataset is chosen over the Brazil Bus dataset because it showed better and more stable results in both testing scenarios. For this dataset, few cleaning and transformation steps needed to take place. Secondly, a dataset with train rides in the Netherlands is used. This dataset underwent more cleaning, preparation, and transformation steps to fit the baseline model and research approach. All of these efforts are described below, as well as the exploration of both datasets.

*3.1.1 Data collection.* The US air data is publicly available and can be found referenced in the baseline work [25] and in the online library created by authors of the baseline study. The original dataset contains the flight connections from January 2004 till March 2021, within the US (origin and destination details), the number of departures scheduled and performed, the number of seats, passengers, distance, carrier information on a monthly timescale. The original dataset can be reviewed in Table 5 in the Appendix.

The data utilized in this research is sourced from rijdendetreinen.nl and is also publicly accessible. This data is derived from the real-time data provided by NS, including live departure times, live arrival times, and service updates. It is also used in the NS app and Rijden de Treinen's website, highlighting its reliability and relevance. The dataset contains all passenger train services in the Netherlands since 2019. While this exact dataset has not been widely used in research, similar datasets have demonstrated reliability in various studies [5, 45, 52]. For instance, the disruptions dataset from the same website as the data used in this research has been employed in studies evaluating the resilience and vulnerability of transportation systems [5, 45], as well as in timetable rescheduling studies [52]. Furthermore, datasets used in the Railroad Problem Solving Competition from 2018, provided by ProRail—the organization responsible for maintaining and extending the national railway network—share many similar attributes with the NS dataset, underscoring its validity and usefulness for research purposes [26, 34, 49]. The similarities between the current dataset and the Railroad Competition dataset include historical train performance and infrastructure data. However, the Railroad Competition dataset is limited to four months in 2017, whereas the current dataset spans the past 5.5 years, providing a more extensive and recent data source. Additionally, the Railroad Competition studies focus on predicting delays in minutes at a station within a 20-minute timeframe, which is not a binary classification problem but a regression task. In contrast, the current research treats delay prediction as a binary classification problem, suitable for the specific objective of testing the generalizability of the models proposed by Lei et al [25] on the Dutch railway. A preview of the original NS dataset can be found in Table 7 in the Appendix.

*3.1.2 Transformation.* The NS dataset underwent a series of transformations to facilitate both exploratory analysis and model construction. Each row in the original data represents a stop at a station, with each service including at least a departure and an arrival at a station (i.e., two rows). For each stop, the dataset provides the station name, arrival and departure times, delays, and cancellations. The focus of this research is on services rather than individual rides due to the significant difference in their numbers and the nature of delays. Specifically, there are approximately ten times more rides than services, with many involving stops in close proximity (e.g., Amsterdam Zuid and Amsterdam Rai). Delays at such closely spaced stops are often not indicative of broader delay patterns; hence, they are aggregated into services to provide a more meaningful analysis of delay trends within the network. These services are defined as trajectories, referring to the source and target stations of a train ride. The efforts to fund trajectory optimization studies substantiate the use of trajectories in the research field [23]. An example of such trajectories can be found in Figure 1, where three trajectories [1] are displayed.

A critical metric calculated during the data transformation process is the final arrival delay. This metric, representing the delay at the last stop of each trajectory, is chosen because it directly impacts passengers' travel plans and serves as a measure of service reliability from an operational perspective [33, 40, 51]. The final arrival

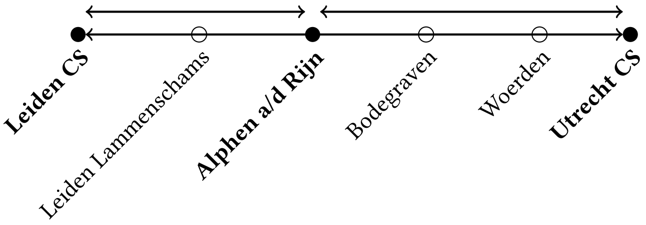


**Figure 1: Example of three trajectories: Leiden Centraal-Utrecht Centraal, Leiden Centraal-Alphen a/d Rijn, and Alphen a/d Rijn-Utrecht.**

delay provides a clear and stable target for predictive modeling, ensuring insights for improving the railway system's efficiency. Other potential metrics, such as maximum delay, mean delay, or departure delays, were considered but ultimately not selected. These alternatives do not capture the overall punctuality and introduce variability that may not accurately reflect the end-to-end travel experience. Consequently, the focus on the final arrival delay ensures a robust and consistent measure for the modeling purposes of trajectories.

To maintain the monthly timeframe, aligning with the baseline study [25], the dataset had to be further transformed. Grouping trajectories by month allowed for the evaluation of topological features on trajectories. However, this grouping introduced the challenge of handling multiple counted trains on a trajectory, which results in a non-binary environment (e.g., 5 out of 10 trains on a trajectory had a final arrival delay). To address this, a variable for the proportion of trains with delays on each trajectory was created. After that, a threshold was established to determine significant delays, ensuring the classification remained binary. The proportion of delayed trains was calculated with the following formula:

$$\text{proportion delayed} = \frac{\text{nr. of trajectories}}{\text{nr. of trajectories with final arrival delay}}$$

The 50th percentile was then used as a threshold to classify a trajectory as significantly delayed. This method provides a balanced distribution of delayed and non-delayed trajectories, making the dataset suitable for machine learning models and ensuring robustness for training and evaluation [15, 19]. Percentiles, including the 50th percentile, are commonly used in various fields to define anomalies or significant events [22, 36]. This method is based on the distribution of data and helps in setting a natural cutoff point [8]. The use of the 50th percentile means that over all the years, 50% of the delayed edges are classified as significantly delayed, and the other 50% as not significantly delayed. This calculation with the median sets the threshold for a significantly delayed edge at 21%, meaning that when a unique edge is delayed more than 21% of the time, it is classified as significantly delayed.

*3.1.3 Data Cleaning.* Data cleaning was performed after the data transformation process. For the US air dataset, data cleaning did not involve many steps. Most importantly, rows with identical source and target cities (indicating no flight) and rows with zero weight (indicating no flight) were removed after grouping the flights by year and month. This clean and transformed dataset can be reviewed in Table 6 in the Appendix.

For the transformed dataset of the Dutch railway network, the focus is on train rides provided by NS to make the research specific and because it is the leading provider of train rides in the Netherlands. All other data from providers like Arriva, Conexxion, and QBuzz were removed. Furthermore, NS has by far the most data (almost 70%), showing its important role in the Dutch railway network. Services by the NS that concern replacement buses or taxis are also removed from the dataset because the focus is on the railway network. In the original dataset, there are no NaN values to be deleted or outliers to handle. There are some NaN values present in the dataset but those have an important meaning, for example, if the arrival delay column is empty, this means that no arrival was planned. These kinds of data attributes were handled during the data transformation step. Handling NaN values in the dataset was crucial for ensuring accurate analysis. NaN values in the 'Arrival Delay of Last Stop' column were set to -1 to indicate unfinished trajectories, while NaN values in the 'Last Arrival Cancelled' column were set to True to signify cancellations. During data aggregation, NaN values in columns such as Final arrival delay and Intermediate arrival delays were filled with 0 to ensure consistency. This careful treatment of NaN values allowed for accurate calculations of the 'Proportion Delayed', which should not take completely canceled rides into consideration.

It is important to note that the dataset used is an archive and not a planned timeline. Furthermore, only trajectories with their source and target station inside the Netherlands were analyzed. This research aimed to analyze the topological features specific to the Dutch railway network. Including trajectories from other countries could introduce variations in network topology that might not be relevant or could confuse the analysis. Lastly, a threshold was established for the minimum number of rides counted per month to ensure data relevance and reliability. Many trajectories were recorded only once or twice a month, which likely indicates substitute transport or non-recurring routes. Including such infrequent data in the model would be impractical, as delays or the absence thereof on these routes could undermine the overall network performance. Therefore, a threshold of at least four rides per month was set, meaning that only routes with a minimum of one weekly train ride are included in the model. By establishing minimum frequency thresholds, routes with sporadic services that do not provide meaningful insight into the overall performance of the transportation network can be excluded [38]. The final NS dataset used for analysis and model implementation can be found in Table 8 in the Appendix and an overview of the data transformations and their descriptions is in Table 9.

*3.1.4 Exploratory Data Analysis.* To compare the distributions in both datasets, the number of data points per month (unique edges) and the proportion of those that are True for the dependent variable (e.g., removed or significantly delayed) were evaluated. For the US air data, the number of unique edges per month ranges between 6000 and 7000, representing unique flight connections. Of these unique edges, the proportion that is removed in the subsequent month is between 20% and 30% as shown in Figure 9 in the Appendix. Both of these counts are relatively stable, except for a downward peak when COVID-19 started in 2020. The proportion of edges that have been removed over the entire dataset's timeframe is 32.7%, showing an imbalanced dataset.

For the NS data, the number of unique edges per month ranges between 350 and 550, and the proportion of significantly delayed edges varies between 0.1 and 0.9. These counts are relatively unstable, as shown in Figure 10 in the Appendix. Furthermore, the fraction of significantly delayed edges is calculated based on the 50th percentile, as described in section 3.1.2 so over all years, 50% of the trajectories are labeled as significantly delayed. It is noticeable that the number of data points and the movement of the data are different in both cases. Furthermore, the increase in delayed links is visible for the NS data, signifying the relevance of this research.

While the initial phase of data exploration involved analyzing the number of trajectories counted per month for both datasets, it is also valuable to see the new NS network as a whole to gain a comprehensive understanding. Flights and trajectories are defined as connections grouped by source-target pairs. In the case of trajectories, this approach emphasizes network-wide patterns instead of isolated stopovers, while in the case of US flights, this decision was already made [25]. In Figure 11 in Appendix B.2, the trajectories and proportion of delays in the network for April 2024 are displayed.

## 3.2 Model Construction

The baseline work focuses on predicting removed links in fast-changing networks by leveraging a variety of machine-learning techniques and network topology features [25]. This approach is implemented for both the US air data to validate the work and identify its limitations as well as for the NS data to evaluate the generalizability of the baseline model and assess its effectiveness in predicting delayed trajectories in the Dutch railway network. Additionally, the model is improved by incorporating new classifiers to explore the possibility of improved model performance and robustness by testing new classifiers.

*3.2.1 Feature Extraction.* A comprehensive set of features was derived from both datasets for the prediction models. The primary model framework used by Lei et al. includes several key steps. After the data is cleaned and grouped by month, relevant features are extracted. For each time period, a graph is constructed where nodes represent airports and edges represent connections (flights), just like in Figure 11 but then for the US air network. These graphs of consecutive months are then compared to create the dependent variable, which is whether a link has been removed (e.g., the link is present in month x but not in month x + 1). For all graphs, the weight (number of flights) is provided to capture the strength of connections. Then, various topological features are calculated for each edge in the graph. These features include both weighted and unweighted variants of Common Neighbors, Resource Allocation, Preferential Attachment, Jaccard coefficient, Adamic-Adar index, Salton index, Sorensen index, Hub Promoted index, Hub Depressed index, Leicht-Holme-Newman index, and Local Path index. An overview of the weighted and unweighted topological features can be found in Appendix C. Eventually, four different models are considered in the baseline work using unweighted topological features, weighted topological features, edge weights, and unweighted topological features + edge weights as feature vectors for the best-performing classifier.

In the NS data, graphs are also created for each month; only in this way, it does not make sense to compare the graphs of consecutive months to look for the removed edges because the dependent variable, 'Significant Delay', is already present in the data. Nevertheless, the same topological features will be calculated for each graph.

A limitation in the baseline work is the small number of topological features that are tested [25]. While the Edge Betweenness Centrality and Edge Current Flow Betweenness Centrality were additionally evaluated, these features did not improve the model. A potential enhancement to the baseline model is the inclusion of Node Centrality Measures (NCM). It is argued that if critical nodes can be identified, they can expose the significance of their elements in transportation networks [43]. The motivation for including the node centrality measures is three-fold. Firstly, it is obvious that the edge weights cannot be used as a feature vector for the NS data since the dependent variable is counted using these edge weights and this would cause data leakage [21, 37]. Secondly, centrality measures are a powerful tool in network theory that can be used to understand the importance of nodes in a network [43], which can be a good addition for predicting removed links in the US air network. Lastly, the unique characteristics of the Dutch railway network, particularly the delayed links, benefit from this enhancement. By incorporating centrality measures, the analysis gains valuable insights into the significance and influence of individual stations within the network, thereby providing a deeper understanding of the network's dynamics and potential points of delays. Specifically, the following centrality measures were added: Degree Centrality, which indicates the number of direct connections a node has; Closeness Centrality, which reflects how close a node is to all other nodes in the network based on the shortest paths; and Node Strength, which represents the sum of weights of all edges connected to a node, indicating the total traffic or activity through the node. The definitions of the measures can be found in Table 1. The addition of these centrality measures enhances the model's ability to capture the importance of stations in the network. This addition aligns with the objective of leveraging machine learning techniques to predict delays, considering the significance of network features.

In conclusion, the feature sets that will be evaluated on the models for both datasets are the basic Topological Features (TF), the Weighted Topological Features (WTF), and the Node Centrality Measures (NCM).

*3.2.2 Model Implementation.* After all features have been extracted for both networks, the focus will be on model training and testing. In the baseline work, 27 of the most common supervised classification algorithms [9, 39] are tested on the balanced training set to determine the best performer based on balanced accuracy, F1 score, and ROC AUC. Among the algorithms tested, XGBoost is identified as the best-performing classifier with the lowest variance among high-performing algorithms in terms of balanced accuracy. The best hyperparameters for the XGBoost model are already defined in the baseline code and can be reviewed in Appendix D. These hyperparameters will be used in the implementation of US air data

| Feature | Definition | Description |
|---|---|---|
| Degree Centrality | $k_i = \sum_{j=1}^{N} A_{ij}$ | Degree centrality of a node $i$ is the sum of the elements of the adjacency matrix $A$ at position $i, j$, where $N$ is the number of nodes. |
| Closeness Centrality | $Cl_i = \frac{1}{\sum_j d(i,j)}$ | Closeness centrality of a node $i$ is the reciprocal of the sum of the shortest path distances $d(i, j)$ from node $i$ to all other nodes $j$ in the graph. |
| Node Strength | $s_i = \sum_{j=1}^{N} w_{ij}$ | Node strength of a node $i$ is the sum of the weights $w_{ij}$ of the links connecting node $i$ to all other nodes $j$. |

**Table 1: The definitions and descriptions of the node centrality measures (NCM). These are calculated for the source and target nodes.**

as well as NS data using the XGBoost Classifier to ensure similar circumstances.

Due to the imbalance in the US air dataset, the RandomUnderSampler from the imblearn library is employed. This technique balances the dataset by randomly undersampling the majority class to ensure an equal representation of both classes during model training, which helps mitigate bias in the model predictions.

Model validation is performed using two primary methods: simultaneous and non-simultaneous testing. Simultaneous testing involves training and testing the model on data from the same period. This approach ensures that the model is evaluated within the same temporal context. Non-simultaneous testing, on the other hand, involves training the model on data from one period and testing it on data from subsequent periods to evaluate the model's ability to generalize over time.

Feature importance analysis is conducted using SHAP values [27, 28]. SHAP values are used to interpret the importance of features in the XGBoost model, providing a way to explain the output of the model by attributing the contribution of each feature to the prediction [31]. The SHAP function computes these values, allowing for a detailed understanding of which features most influence the model's predictions. By implementing this comprehensive framework, the results of the prediction can be evaluated. The combination of simultaneous and non-simultaneous testing ensures both immediate and long-term generalizability of the model, while SHAP values provide transparency into feature importance, enhancing the interpretability of the model's predictions.

The performance of the implemented baseline model on US air data is evaluated using balanced accuracy against the null-predictions. The results are presented in Figures 2 and 3, demonstrating the implementation's similarity to Lei et al.'s work [25].

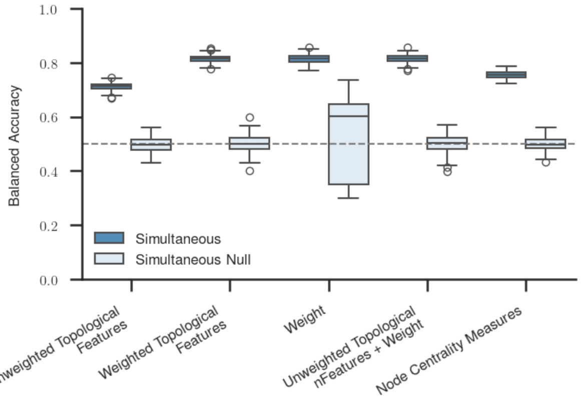


**Figure 2: The balanced accuracy boxplots for different feature sets during simultaneous testing on US air data.**

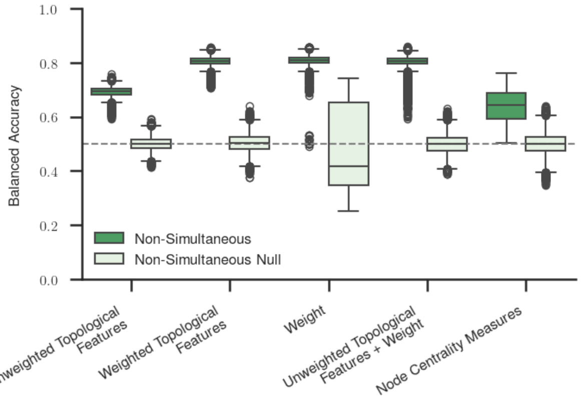


**Figure 3: The balanced accuracy boxplots for different feature sets during non-simultaneous testing on US air data.**

*3.2.3 Improvements.* In addition to incorporating node centrality measures, this research explores the effectiveness of several other classifiers beyond XGBoost on the NS dataset. The aim of this research, besides testing the generalizability of Lei's model, is to be able to predict delayed trajectories as accurately as possible. The new classifiers tested include Random Forest and Decision Tree, next to Gradient Boosting, Ada Boost, and Logistic Regression, which were the best-performing classifiers for the US air data [25]. Adding these two classifiers broadens the scope of the investigation to determine if performance can be further enhanced with alternative machine learning algorithms. The reason that these two algorithms were chosen is that, next to logistic regression, they are commonly used to perform binary classification tasks [46] with overall good results. Besides that, Random Forest [34] as well as Decision Trees [26] have been used in previous works on delay prediction.

Furthermore, hyperparameter optimization was conducted using RandomizedSearchCV with 10-fold cross-validation. Hyperparameter optimization is a crucial step in enhancing the performance of machine learning models. Traditional methods such as grid search, which exhaustively searches over specified parameter values, are computationally expensive and inefficient, especially with high-dimensional parameter spaces. RandomizedSearchCV addresses these limitations by sampling a fixed number of hyperparameter combinations from specified distributions, thus providing a more efficient and often equally effective method for hyperparameter tuning [4]. The decision to use 10-fold cross-validation ensures that the model is evaluated on different subsets of data, providing a more robust estimate of its performance and reducing the risk of overfitting. It is chosen over the default 5-fold cross-validation to explore a wider context and because it demonstrated better performance in previous studies [3, 30]. The specific hyperparameter grids used for each classifier in this research are detailed in Table 13 in the Appendix. Each grid is created to capture the most relevant parameters for the respective models, facilitating an effective and comprehensive search for optimal configurations.

All classifiers were subject to the same evaluation process, involving simultaneous and non-simultaneous testing. This approach ensures a fair comparison of their performance across different feature sets (TF, WTF, NCM). The results of these tests were evaluated based on balanced accuracy, F1 score, and ROC AUC, providing a comprehensive assessment of their effectiveness.

A visualization of the project flow and improvements to the baseline model can be seen in Figure 4. The components within the dashed line represent elements specific to this research, while the components within the continuous line represent the work by Lei et al. Additionally, bold text highlights additions made in this research compared to the baseline, and strikethrough text indicates elements from the baseline that were not considered.

## 3.3 Evaluation & Validation

The performance of the improved models trained on NS data, including the centrality measures as a feature set, is compared against the baseline models used by Lei et al. This comparison is based on balanced accuracy to ensure robustness when comparing the two datasets and to prevent bias in the classifier [7]. This metric avoids bias by providing a symmetric evaluation of performance across all classes, ensuring that the results are not distorted by class imbalances. Balanced accuracy is also the metric used for the evaluation of the performance of the added algorithms, next to F1 scores and the ROC AUC. On the one hand, these metrics were chosen because they are also used in baseline work, on the other hand; the F1 score is useful in situations where the costs of false positives and false negatives have to be balanced [29] and the ROC AUC is in general a good metric to measure a model's ability to distinguish between positive and negative classes [6, 35].

The implemented models undergo both simultaneous and non-simultaneous testing. The data is split into training and testing sets using a time-based split function. This function ensures that each month has data in both the training and testing sets, maintaining the temporal integrity of the data. Specifically, 70% of the data from each period is randomly sampled into the training set, and the remaining 30% into the testing set [25]. This approach mitigates potential temporal leakage and ensures that the model's performance is not wrongly based on future information in the training set.

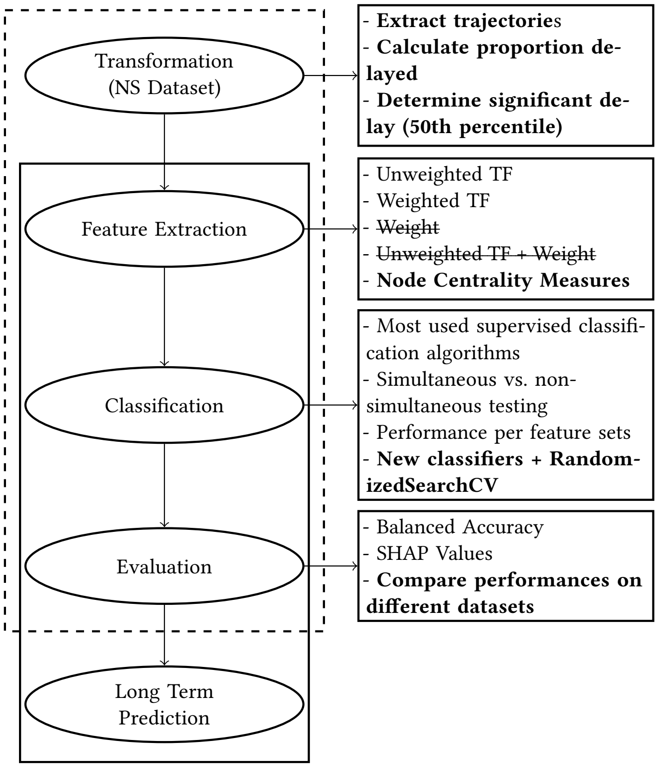

Transformation (NS Dataset)
Feature Extraction
Classification
Evaluation
Long Term Prediction
- **Extract trajectories**
- **Calculate proportion delayed**
- **Determine significant delay (50th percentile)**
- Unweighted TF
- Weighted TF
- ~~Weight~~
- ~~Unweighted TF + Weight~~
- **Node Centrality Measures**
- Most used supervised classification algorithms
- Simultaneous vs. non-simultaneous testing
- Performance per feature sets
- **New classifiers + RandomizedSearchCV**
- Balanced Accuracy
- SHAP Values
- **Compare performances on different datasets**

**Figure 4: Project flow highlighting the research scope and methodologies.**

The baseline model's performance on the US air data was evaluated to validate the accuracy of our implementation. Figures 2 and 3 demonstrate the balanced accuracy of all feature sets during simultaneous and non-simultaneous testing, respectively. The results closely match those reported by Lei et al., validating our implementation and allowing us to apply the model to the NS dataset confidently. By replicating Lei et al.'s methodology and rigorously applying it to the context of the Dutch railway network, we test the generalizability of their model for delay prediction instead of removed link prediction. This involved implementing their work to validate its correctness, identifying limitations, adding an additional feature set (node centrality measures) to overcome these limitations, and applying the exact same model to the NS dataset. The comparison of model performances on the NS data, particularly in terms of balanced accuracy, provides insights into the effectiveness of the implemented models in predicting delays within a different network. Furthermore, all results will be reflected against their null predictions to establish a baseline within each model, helping to assess performance and detect potential outliers [44, 47]. This comprehensive evaluation framework ensures that improvements are clearly demonstrated and performance validated against the baseline model.

# 4 RESULTS

In this section, the results of the methodology will be evaluated by following the structure of the research questions.

## 4.1 Implementation & Improvements Baseline

The first sub-research question focuses on how the machine learning models and methodologies used by Lei et al. can be implemented and improved for their effectiveness in predicting missing links in fast-changing networks. The models used by Lei et al. were successfully implemented using the US air dataset, as is already shown in section 3.2.2. The results confirm the effectiveness of their approach in predicting removed links when performing simultaneous testing as well as non-simultaneous testing for all feature sets. The centrality measures (degree centrality, closeness centrality, and node strength) that are added as an improvement to their models showed almost similar results as the other feature sets during simultaneous testing on the US air data, as can be seen in Figure 2. For non-simultaneous testing, the centrality measures perform a bit worse than the other feature sets and introduce more variance, as can be seen in Figure 3. The results prove that the models by Lei et al. have good performance for the different feature sets in both testing scenarios. This is an important first step for the validation of the models before applying them to the NS dataset. The confusion matrices 12 and 13 in Appendix F.1 also show that the centrality measures have performance comparable to the topological features.

Furthermore, when zooming in on the centrality measures and evaluating their importance using SHAP values in Figure 5, it is seen that node strength (source station) and degree centrality (source station) are particularly influential in predicting missing links. This indicates that source nodes with higher connectivity and interaction volumes are critical for network stability. Comparing these SHAP values to those of the unweighted topological features in combination with edge weights in Figure 6, it is observed that the centrality measures are less consistent in their importance and have less impact in general. Especially when looking at the impact of 'Curr Fweight', the weight of an edge in the current time frame, it is observed that this feature has a strong effect on the output. This is rather obvious since if this weight is high, there is little chance that an edge will be removed in the next month. For this reason, it is a good idea to add the centrality measures as a feature set. In addition, the weights of the connections cannot be used as a feature for the NS data anyway because this value is used to calculate the dependent variable, namely the significant delay.

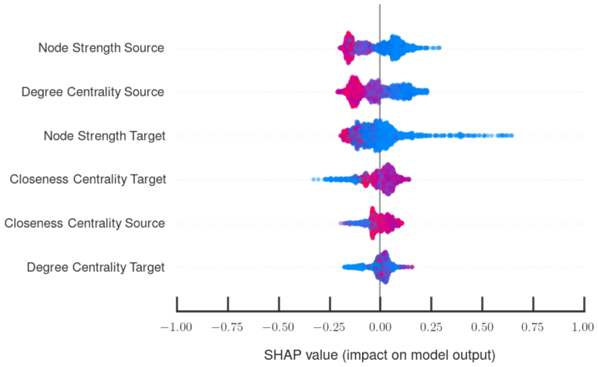


**Figure 5: SHAP values for the node centrality measures on US air Data.**

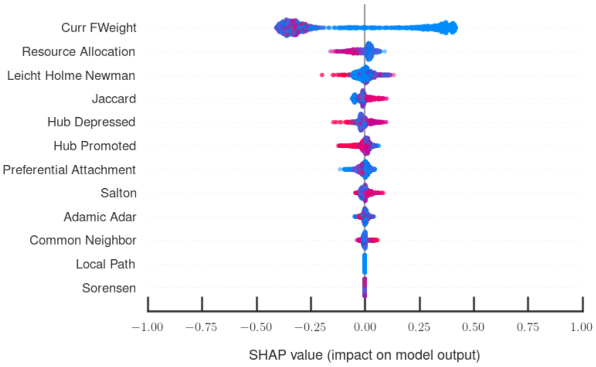


**Figure 6: SHAP values for the unweighted topological features + edge weights on US air Data.**

## 4.2 Implementation NS Data

The second sub-research question concerns the extent to which the improved methodologies and models can be adapted for predicting delays in the NS network. After the transformation of the data and the calculation of the dependent variable, namely, if a trajectory is significantly delayed (> 21%, see 3.1.2), it is possible to use the NS dataset in the implemented models. It is important to note that the edge weight and the unweighted topological features + weight will be removed from the model since the edge weight cannot be used as a variable. The node centrality measures will be added instead, next to the existing topological features (weighted and unweighted). When evaluating the obtained results in Figures 7 and 8 it is observed that the results are worse for the model when trained on the NS dataset with both simultaneous and non-simultaneous testing. While the balanced accuracies are above the null predictions, they are just above 50% meaning that they are either not good predictors for the delay of edges or the XGBoost Classifier is not suitable for this prediction.

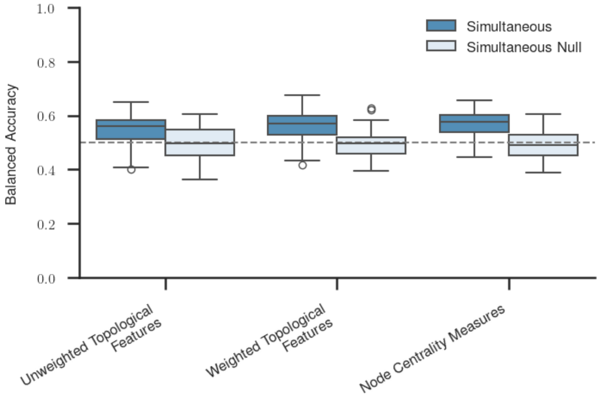


**Figure 7: The balanced accuracy boxplots for different feature sets during simultaneous testing on NS data.**

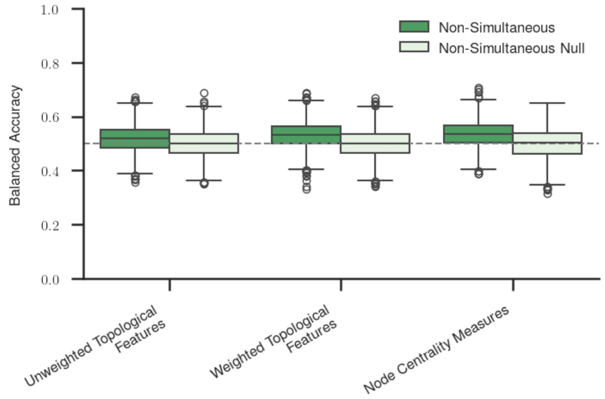


**Figure 8: The balanced accuracy boxplots for different feature sets during non-simultaneous testing on NS data.**

This is also the result observed in the confusion matrices in Appendix F.2. The accuracy scores seem to be unstable for this dataset. They predict right nearly as often as wrong. The SHAP figures in Appendix G show the same inconsistency.

For this reason, other classifiers were evaluated to compare their performances across different feature sets and test if they could result in higher balanced accuracies than the baseline model using the XGBoost Classifier. Only testing one model would not be sufficient to conclude that the baseline model is not generalizable for a new dataset. The results for these added classifiers, as shown in Table 2 for simultaneous testing and Table 3 for non-simultaneous testing, reveal that centrality measures consistently yielded better results than topological features across various classifiers. However, the overall performance remained poor, particularly in non-simultaneous testing scenarios. Furthermore, it becomes clear that the XGBoost Classifier performs best when simultaneous testing, with RandomForest as a close second. It is noticeable that both during simultaneous testing, and non-simultaneous testing, the results are very close to each other. This also applies to the F1 and ROC AUC scores which can be observed in Appendix H. These findings indicate that while the improved models, particularly those incorporating centrality measures, show some promise, they do not seem to be able to predict delayed trajectories in the Dutch railway network well.

| Classifier | TF | WTF | NCM |
|---|---|---|---|
| AdaBoost | 0.618 | 0.621 | **0.635** |
| DecisionTree | 0.602 | **0.604** | 0.591 |
| GradientBoosting | 0.617 | 0.620 | **0.623** |
| LogisticRegression | **0.636** | 0.633 | 0.632 |
| RandomForest | 0.620 | 0.626 | **0.645** |
| **XGBoost** | 0.635 | 0.633 | **0.646** |

**Table 2: Balanced accuracy scores of classifiers for every feature set during simultaneous testing.**

| Classifier | TF | WTF | NCM |
|---|---|---|---|
| AdaBoost | 0.527 | 0.532 | **0.538** |
| DecisionTree | 0.525 | 0.526 | **0.534** |
| GradientBoosting | 0.527 | 0.533 | **0.545** |
| LogisticRegression | 0.525 | 0.535 | **0.537** |
| **RandomForest** | 0.529 | 0.535 | **0.550** |
| XGBoost | 0.529 | 0.532 | **0.538** |

**Table 3: Balanced accuracy scores of classifiers for every feature set during non-simultaneous testing.**

## 4.3 Model Performance and Comparison

To answer the last sub-research question, the balanced accuracies for the XGBoost classifier are merged for both datasets. While the other classifiers showed similar balanced accuracy scores for the NS dataset, the XGBoost Classifier was used to enable the comparison with the baseline article. In Table 4 the performances for the different datasets can be compared and it is clear that the model performs better on the US air data when predicting removed links than on the NS data when predicting significantly delayed links, even when adding the additional centrality measures.

| Testing Type | Dataset | TF | WTF | NCM |
|---|---|---|---|---|
| Simultaneous | **US air Data** | 0.71 | **0.81** | 0.76 |
| | NS Data | 0.64 | 0.63 | **0.65** |
| Non-simultaneous | **US air Data** | 0.68 | **0.80** | 0.68 |
| | NS Data | 0.53 | 0.53 | **0.54** |

**Table 4: Balanced accuracy scores for the different datasets using the XGBoost Classifier in both testing scenarios.**

# 5 DISCUSSION

This research aimed to implement Lei et al.'s machine learning framework, initially designed for forecasting removed links in fast-changing networks, to predict delayed trajectories within the Dutch railway network. The baseline framework demonstrated high balanced accuracy in predicting removed links in the US air network, achieving scores of 0.71-0.81 in simultaneous testing and 0.68-0.80 in non-simultaneous testing. However, the improvements made to the model to implement the Dutch railway data yielded lower balanced accuracy scores, with simultaneous testing scores ranging from 0.63 to 0.64 and non-simultaneous testing scores between 0.52 and 0.54.

The lower performance on the Dutch railway data suggests that while Lei et al.'s approach is effective for predicting removed links in the US air network, it may not be directly applicable to networks with different dynamics and operational characteristics. The additional centrality measures, although theoretically promising, did not significantly enhance the model's predictive performance either. This observation is supported by the higher variance and lower balanced accuracy in the US context for non-simultaneous testing.

Several studies have highlighted the importance of adapting machine learning models and their features to the specific characteristics of the dataset and network being analyzed. For instance, research on public transportation networks in Europe has shown that incorporating domain-specific features such as scheduling information, passenger load, and infrastructure maintenance schedules can significantly improve prediction accuracy [5, 52]. The unique aspects of railway networks, such as fixed routes and schedules, contrast with the more dynamic nature of air transportation systems, potentially explaining the poorer performance observed in this research.

Even after incorporating other machine learning models commonly used in binary classification, the accuracy scores remained quite similar to those of the XGBoost Classifier. This suggests that these models might be too simplistic for the complex data, or that the issue lies within the data itself, not leading to performance improvements with new models.

## 5.1 Limitations & Future Work

The results were not as favorable as anticipated, particularly in non-simultaneous testing scenarios. Several factors can be considered to explain this discrepancy. First, certain topological features may be less predictive because the Dutch railway network has different operational characteristics than the US air network. The railway network in the Netherlands is very dense and therefore very sensitive to small timetable changes. As mentioned by the Head of Digitalization Operations at NS, B. van Zaalen (personal communication, June 4, 2024), the ripple effect is a big problem in the network. A small congestion somewhere on a route can cause many delays elsewhere. There can be many different causes for network congestion or delay, and perhaps topological and network features alone are not enough to predict it. The focus on topological features and centrality measures, while inspired by related work [25, 43], might not adequately capture other critical factors such as maintenance schedules, staffing issues, or external disruptions, which are known to affect railway network performance [26].

The scalability and generalizability of the adapted models also present challenges. While Lei et al.'s framework demonstrated robustness across different networks during simultaneous testing, its adaptation to the Dutch railway network indicates the need for more context-specific modifications. Additionally, the models' even poorer performance in non-simultaneous testing highlights potential issues with temporal generalizability, suggesting that further refinement is necessary to enhance predictive stability over time.

Following this, another possible limitation lies in the dataset. There is much less available data per month compared to the US air data (more than 10 times less), which could mean that there is not enough training data per month to find the relevant patterns. A suggestion would be to include other providers or look at all rides instead of the focus on trajectories. Furthermore, conducting a comparative analysis with other transportation networks, particularly those with similar operational characteristics to the US air network, could also provide valuable insights into model adaptability, generalizability, and performance.

Additionally, employing more sophisticated machine learning techniques, such as ensemble methods or deep learning architectures, might enhance predictive accuracy. However, it is noticeable that already various frequently used classifiers have been compared and they all show similar output results. This indicates that the model is not the limiting factor. In any case, more consideration will have to be given to the extreme complexity of the Dutch railway network. Future research should explore integrating operational, environmental, and temporal variables to develop a more holistic predictive model.

By addressing these limitations and exploring these future directions, it is possible to develop a more robust and accurate model for predicting delayed trajectories in the Dutch railway network. This research contributes to the broader understanding of the challenges and opportunities in applying machine learning to transportation networks, highlighting the need for context-specific adaptations.

# 6 CONCLUSION

The objective of this research was to evaluate the generalizability of Lei et al.'s machine learning framework, which predicts removed links in the fast-changing US Air network, by applying it to predict delayed trajectories in the Dutch railway network. To answer the first sub-research question, the XGBoost Classifier can be implemented to predict removed links in fast-changing transportation networks like the US Air data and can be improved by adding the node centrality measures as a feature set. The results show high balanced accuracies for this prediction. Furthermore, these improved models can be constructed to fit with another dataset, namely the NS dataset, as well. However, poor performance was observed when this dataset was used in the improved baseline model compared to the performance of the US air dataset. Even when other classifiers, like the Decision Tree and Random Forest, were evaluated it did not seem possible to predict delayed trajectories with the baseline framework.

The research findings highlight the challenges of directly transferring methodologies across different transportation networks with distinct operational dynamics. Significant effort was devoted to transforming the NS dataset to fit the implemented model, but these attempts did not yield the desired results. While the models achieved balanced accuracy scores above the null predictions, the overall performance was not good, indicating the need for more context-specific adaptations and the incorporation of diverse feature sets.

The limitations, especially the exclusion of non-topological factors, suggest areas for future research. Integrating operational, environmental, and temporal variables, alongside more advanced machine learning techniques, could enhance predictive accuracy.

In conclusion, while this research does not provide a definitive solution for predicting delayed trajectories in the Dutch railway network, it offers valuable insights into the complexities and challenges of transportation network forecasting. Future work should focus on developing more holistic models that account for the multifaceted and complex nature of railway operations, ultimately contributing to more reliable and efficient transportation systems.

# REFERENCES

[1] [n. d.]. De populairste trajecten met de trein | NS. https://www.ns.nl/trajecten/

[2] [n. d.]. NS Annual Report 2023. https://www.nsannualreport.nl/annual-report-2023/our-activities-and-achievements-in-the-netherlands/operational-performance/punctuality

[3] Daniel Mesafint Belete and Manjaiah D. Huchaiah. 2021. Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. *International journal of computers applications* 44, 9 (9 2021), 875–886. https://doi.org/10.1080/1206212x.2021.1974663

[4] James Bergstra and Yoshua Bengio. 2012. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research* 13 (12 2012), 281–305. https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf

[5] Nikola Bešinović. 2020. Resilience in railway transport systems: a literature review and research agenda. *Transport reviews* 40, 4 (2 2020), 457–478. https://doi.org/10.1080/01441647.2020.1728419

[6] Kendrick Boyd, Kevin H. Eng, and C. David Page. 2013. *Area under the Precision-Recall Curve: Point Estimates and Confidence Intervals.* 451–466 pages. 29 https://doi.org/10.1007/978-3-642-40994-3\{_

[7] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. 2010. The Balanced Accuracy and Its Posterior Distribution. *IEEE* (8 2010). https://doi.org/10.1109/icpr.2010.764

[8] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection. *ACM computing surveys* 41, 3 (7 2009), 1–58. https://doi.org/10.1145/1541880.1541882

[9] Tianqi Chen and Carlos Guestrin. 2016. XGBoost. *KDD '16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (8 2016). https://doi.org/10.1145/2939672.2939785

[10] Stuart Coles. 2001. *An Introduction to Statistical Modeling of Extreme Values.* https://doi.org/10.1007/978-1-4471-3675-0

[11] Marie Colin, Fabien Palhol, and André Leuxe. 2016. Adaptation of Transport Infrastructures and Networks to Climate Change. *Transportation research procedia* 14 (1 2016), 86–95. https://doi.org/10.1016/j.trpro.2016.05.044

[12] Anp 22 Dec. 2023. NS belooft vaker op tijd te rijden, na zwakke prestatie dit jaar. https://www.businessinsider.nl/ns-belooft-treinen-2024-op-tijd-rijden-zwakke-prestatie-2023/

[13] Jørgen Thorlund Haahr, Erik Orm Hellsten, and Evelien van der Hurk. 2019. *Train Delay Prediction in the Netherlands through Neural Networks.*

[14] Florian Hauck and Natalia Kliewer. 2020. *Data Analytics in Railway Operations: Using Machine Learning to Predict Train Delays.* 741–747 pages. 90 https://doi.org/10.1007/978-3-030-48439-2\{_

[15] None Haibo He and E.A. Garcia. 2009. Learning from Imbalanced Data. *IEEE transactions on knowledge and data engineering* 21, 9 (9 2009), 1263–1284. https://doi.org/10.1109/tkde.2008.239

[16] Ping Huang, Jingwei Guo, Shu Liu, and Francesco Corman. 2024. Explainable train delay propagation: A graph attention network approach. *Transportation research. Part E, Logistics and transportation review* 184 (4 2024), 103457. https://doi.org/10.1016/j.tre.2024.103457

[17] Ping Huang, Javad Lessan, Chao Wen, Qiyuan Peng, Liping Fu, Li Li, and Xinyue Xu. 2020. A Bayesian network model to predict the effects of interruptions on train operations. *Transportation Research Part C: Emerging Technologies* 114 (5 2020), 338–358. https://doi.org/10.1016/j.trc.2020.02.021

[18] Ping Huang, Chao Wen, Liping Fu, Qiyuan Peng, and Yixiong Tang. 2020. A deep learning approach for multi-attribute data: A study of train delay prediction in railway systems. *Information sciences* 516 (4 2020), 234–253. https://doi.org/10.1016/j.ins.2019.12.053

[19] Nathalie Japkowicz and Shaju Stephen. 2002. The class imbalance problem: A systematic study. *Intelligent data analysis* (10 2002). https://doi.org/10.5555/1293951.1293954

[20] Tao Ji, Yanhong Yao, Yue Dou, Shejun Deng, Shijun Yu, Yunqiang Zhu, and Huajun Liao. 2022. The Impact of Climate Change on Urban Transportation Resilience to Compound Extreme Events. *Sustainability* 14, 7 (3 2022), 3880. https://doi.org/10.3390/su14073880

[21] Shachar Kaufman, Saharon Rosset, Claudia Perlich, and Ori Stitelman. 2012. Leakage in data mining. *ACM transactions on knowledge discovery from data* 6, 4 (12 2012), 1–21. https://doi.org/10.1145/2382577.2382579

[22] Jaehoon Kim. 2018. Using Median as a Threshold in Determining Anomaly in Back-End Authentication. *IEEE* (7 2018). https://doi.org/10.1109/iccia.2018.00055

[23] Dimitris Kouzoupis, Ishan Pendharkar, and Francesco Corman. 2023. TTO-Bench—an Open-Source Library for Train Trajectory Optimization. *SN Operations Research Forum* 4, 4 (9 2023). https://doi.org/10.1007/s43069-023-00248-x

[24] Sara Lbazri, Soumaya Ounacer, Houda Jihal, and Azah Mohamed. 2020. Predict France trains delays using visualization and machine learning techniques. *Procedia Computer Science* 175 (1 2020), 700–705. https://doi.org/10.1016/j.procs.2020.07.103

[25] Weihua Lei, Luiz G. A. Alves, and Luıs A. Nunes Amaral. 2022. Forecasting the evolution of fast-changing transportation networks using machine learning. *Nature Communications* 13, 1 (7 2022). https://doi.org/10.1038/s41467-022-31911-2

[26] Zhongcan Li, Chao Wen, Rui Hu, Chuanlin Xu, Ping Huang, and Jiang Xi. 2020. Near-term train delay prediction in the Dutch railways network. *International Journal of Rail Transportation* 9, 6 (11 2020), 520–539. https://doi.org/10.1080/23248378.2020.1843194

[27] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. *arXiv (Cornell University)* (1 2017). https://doi.org/10.48550/arxiv.1705.07874

[28] Scott M. Lundberg, Bala Nair, Monica S. Vavilala, Mayumi Horibe, Michael J. Eisses, Trevor Adams, David E. Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim, and Su-In Lee. 2018. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature biomedical engineering* 2, 10 (10 2018), 749–760. https://doi.org/10.1038/s41551-018-0304-0

[29] Hossin M and Sulaiman MN. 2015. A Review on Evaluation Metrics for Data Classification Evaluations. *International journal of data mining and knowledge management process* 5, 2 (3 2015), 01–11. https://doi.org/10.5121/ijdkp.2015.5201

[30] Seyed Matin Malakouti. 2023. Babysitting hyperparameter optimization and 10-fold-cross-validation to enhance the performance of ML methods in predicting wind speed and energy generation. *Intelligent systems with applications* 19 (9 2023), 200248. https://doi.org/10.1016/j.iswa.2023.200248

[31] Wilson E. Marcilio and Danilo M. Eler. 2020. From explanations to feature selection: assessing SHAP values as feature selection mechanism. *IEEE* (11 2020). https://doi.org/10.1109/sibgrapi51738.2020.00053

[32] Arshad Mohd and Muqeem Ahmed. 2021. Train Delay Estimation in Indian Railways by Including Weather Factors Through Machine Learning Techniques. *Recent advances in computer science and communications* 14, 4 (7 2021), 1300–1307. https://doi.org/10.2174/2666255813666190912095739

[33] Weiwei Mou, Zhaolan Cheng, Chao Wen, National Engineering Laboratory of Integrated Transportation Big Data Application Technology, National United Engineering Laboratory of Integrated, and Intelligent Transportation. 2019. Predictive Model of Train Delays in a Railway System. *8th International Conference on Railway Operations Modelling and Analysis - RailNorrköping 2019* (2019), 913–914. https://ep.liu.se/ecp/069/059/ecp19069059.pdf

[34] Mohammad Amin Nabian, Negin Alemazkoor, and Hadi Meidani. 2019. Predicting Near-Term Train Schedule Performance and Delay Using Bi-Level Random Forests. *Transportation Research Record* 2673, 5 (4 2019), 564–573. https://doi.org/10.1177/0361198119840339

[35] Gireen Naidu, Tranos Zuva, and Elias Mmbongeni Sibanda. 2023. A Review of Evaluation Metrics in Machine Learning Algorithms. *Lecture notes in networks and systems* (1 2023), 15–25. 2 https://doi.org/10.1007/978-3-031-35314-7\{_

[36] Arvind Narayanan, Elaine Shi, and Benjamin I. P. Rubinstein. 2011. Link prediction by de-anonymization: How We Won the Kaggle Social Network Challenge. *IEEE* (7 2011). https://doi.org/10.1109/ijcnn.2011.6033446

[37] Odai Nassar. 2023. Data Leakage in Machine Learning. (03 2023). https://doi.org/10.13140/RG.2.2.27468.59528

[38] Benjamin Otto. 2018. Aggregation techniques for frequency assignment in public transportation. *Public transport* 11, 1 (12 2018), 51–87. https://doi.org/10.1007/s12469-018-0177-3

[39] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, INRIA Saclay Parietal, CEA Saclay 91191 Gif sur Yvette – France Neurospin, Bât 145, Olivier Grisel, Nuxeo, 75 020 Paris – France 20 rue Soleillet, Mathieu Blondel, Nada Kobe 657-8501 – Japan Kobe University, 1-1 Rokkodai, Peter Prettenhofer, 99421 Weimar – Germany Bauhaus-Universität Weimar, Bauhausstr. 11, Ron Weiss, New York NY 10011 – USA Google Inc, 76 Ninth Avenue, Vincent Dubourg, EA 3867 LaMI BP 10448 63000 Clermont-Ferrand – France Clermont Université, IFMA, Jake Vanderplas, Box 351580 Seattle WA 98195 – USA Astronomy Department, University of Washington, Alexandre Passos, Amherst MA 01002 – USA IESL Lab, UMass Amherst, David Cournapeau, Cambridge CB3 0FA – UK Enthought, 21 J.J. Thompson Avenue, Matthieu Brucher, avenue Larribau 64000 Pau – France Total SA, CSTJF, Matthieu Perrot, Édouard Duchesnay, and Bât 145 CEA Saclay 91191 Gif sur Yvette – France LNAO, Neurospin. 2011. *SciKit-Learn: Machine Learning in Python.* Technical Report. 2825–2830 pages. https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf?ref=https:/

[40] Rui Shi, Xinyue Xu, Jianmin Li, and Yanqiu Li. 2021. Prediction and analysis of train arrival delay based on XGBoost and Bayesian optimization. *Applied Soft Computing* 109 (9 2021), 107538. https://doi.org/10.1016/j.asoc.2021.107538

[41] Léon Sobrie, Marijn Verschelde, Veerle Hennebel, and Bart Roets. 2023. Capturing complexity over space and time via deep learning: An application to real-time delay prediction in railways. *European Journal of Operational Research* 310, 3 (11 2023), 1201–1217. https://doi.org/10.1016/j.ejor.2023.03.040

[42] Thomas Spanninger, Alessio Trivella, Beda Büchel, and Francesco Corman. 2022. A review of train delay prediction approaches. *Journal of Rail Transport Planning Management* 22 (6 2022), 100312. https://doi.org/10.1016/j.jrtpm.2022.100312

[43] Iraklis Stamos. 2023. Transportation Networks in the Face of Climate Change Adaptation: A Review of Centrality Measures. *Future transportation* 3, 3 (7 2023), 878–900. https://doi.org/10.3390/futuretransp3030049

[44] Zhoujian Sun, Wei Dong, Hanrui Shi, Hong Ma, Lechao Cheng, and Zhengxing Huang. 2022. Comparing Machine Learning Models and Statistical Models for Predicting Heart Failure Events: A Systematic Review and Meta-Analysis. *Frontiers in cardiovascular medicine* 9 (4 2022). https://doi.org/10.3389/fcvm.2022.812276

[45] Christopher Szymula and Nikola Bešinović. 2020. Passenger-centered vulnerability assessment of railway networks. *Transportation research. Part B: methodological/Transportation research. Part B, Methodological* 136 (6 2020), 30–61. https://doi.org/10.1016/j.trb.2020.03.008

[46] Etienne van de Bijl, Jan Klein, Joris Pries, Sandjai Bhulai, Mark Hoogendoorn, and Rob van der Mei. 2022. The Dutch Draw: Constructing a Universal Baseline for Binary Prediction Models. arXiv:2203.13084

[47] W. Patrick Walters. 2021. Comparing classification models—a practical tutorial. *Journal of computer-aided molecular design* 36, 5 (9 2021), 381–389. https://doi.org/10.1007/s10822-021-00417-2

[48] Dawei Wang, Jingwei Guo, and Chunyang Zhang. 2024. A Novel Hybrid Deep Learning Model for Complex Systems: A Case of Train Delay Prediction. *Advances in civil engineering* 2024 (5 2024), 1–14. https://doi.org/10.1155/2024/8163062

[49] Jing Xu, Weiqi Wang, Zheming Gao, Haochen Luo, and Wei Qian. 2022. A Novel Markov Model for Near-Term Railway Delay Prediction. *arXiv (Cornell University)* (5 2022). https://doi.org/10.48550/arxiv.2205.10682

[50] Jiateng Yin, Miao Wang, Andrea D'Ariano, Jinlei Zhang, and Lixing Yang. 2023. Synchronization of train timetables in an urban rail network: A bi-objective optimization approach. *Transportation research. Part E, Logistics and transportation review* 174 (6 2023), 103142. https://doi.org/10.1016/j.tre.2023.103142

[51] Dalin Zhang, Yunjuan Peng, Yumei Zhang, Daohua Wu, Hongwei Wang, and Hailong Zhang. 2022. Train Time Delay Prediction for High-Speed Train Dispatching Based on Spatio-Temporal Graph Convolutional Network. *IEEE transactions on intelligent transportation systems* 23, 3 (3 2022), 2434–2444. https://doi.org/10.1109/tits.2021.3097064

[52] Y. Zhu, R.M.P. Goverde, and E. Quaglietta. 2018. Railway timetable rescheduling for multiple simultaneous disruptions. *CASPT* (1 2018). https://www.narcis.nl/publication/RecordID/oai%3Atudelft.nl%3Auuid%3A1664b528-4e12-4745-bf87-a798a7e878ef

# APPENDIX

# A DATA PREVIEWS

| DEP_SCHEDULED | DEP_PERFORMED | SEATS | PASSENGERS | DISTANCE | UNIQUE_CARRIER | UNIQUE_CARRIER_NAME | ORIGIN_AIRPORT_ID | ORIGIN | ORIGIN_CITY_NAME | ORIGIN_STATE_ABR | DEST_AIRPORT_ID | DEST | DEST_CITY_NAME | DEST_STATE_ABR | YEAR | MONTH |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.00 | 17.00 | 0.00 | 0.00 | 2846.00 | "9S" | Southern Air Inc. | 13930 | ORD | Chicago, IL | IL | 10299 | ANC | Anchorage, AK | AK | 2008 | 10 |
| 0.00 | 1.00 | 262.00 | 0.00 | 408.00 | "DL" | Delta Air Lines Inc. | 14112 | PIE | St. Petersburg, FL | FL | 10397 | ATL | Atlanta, GA | GA | 2008 | 10 |
| 0.00 | 1.00 | 68.00 | 24.00 | 321.00 | "09Q" | Swift Air, LLC d/b/a Eastern Air Lines d/b/a Eastern | 12264 | IAD | Washington, DC | VA | 11057 | CLT | Charlotte, NC | NC | 2008 | 11 |
| 0.00 | 1.00 | 4.00 | 0.00 | 53.00 | "2O" | Island Air Service | 10324 | AOS | Amook Bay, AK | AK | 10170 | ADQ | Kodiak, AK | AK | 2008 | 11 |
| 0.00 | 1.00 | 0.00 | 0.00 | 1321.00 | "8C" | Air Transport International | 12206 | HRL | Harlingen/San Benito, TX | TX | 15295 | TOL | Toledo, OH | OH | 2008 | 5 |
| 0.00 | 1.00 | 0.00 | 0.00 | 961.00 | "AMQ" | Ameristar Air Cargo | 12206 | HRL | Harlingen/San Benito, TX | TX | 15016 | STL | St. Louis, MO | MO | 2008 | 5 |
| 0.00 | 1.00 | 6.00 | 2.00 | 30.00 | "GV" | Grant Aviation | 11336 | DLG | Dillingham, AK | AK | 14037 | PCA | Portage Creek, AK | AK | 2008 | 5 |
| 0.00 | 1.00 | 3.00 | 1.00 | 107.00 | "KAH" | Kenmore Air Harbor | 11762 | FRD | Friday Harbor, WA | WA | 13865 | OLM | Olympia, WA | WA | 2008 | 5 |
| 0.00 | 1.00 | 5.00 | 1.00 | 30.00 | "KS" | Peninsula Airways Inc. | 11336 | DLG | Dillingham, AK | AK | 14037 | PCA | Portage Creek, AK | AK | 2008 | 5 |

**Table 5: A preview of the original US data from the online library created by Lei et al. [25]. Each row represents a flight with various attributes such as the number departures scheduled, and performed, number of seats, passengers, distance, carrier information, origin, and destination details.**

| YEAR | MONTH | Source | Target | Passengers | Weight |
|---|---|---|---|---|---|
| 2004 | 1 | aberdeen_sd | jamestown_nd | 45.0 | 25.0 |
| 2004 | 1 | aberdeen_sd | minneapolis_mn | 2782.0 | 141.0 |
| 2004 | 1 | aberdeen_sd | pierre_sd | 505.0 | 52.0 |
| 2004 | 1 | aberdeen_sd | sioux_falls_sd | 0.0 | 35.0 |
| 2004 | 1 | aberdeen_sd | watertown_sd | 139.0 | 29.0 |
| 2004 | 1 | abilene_tx | dallasfort_worth_tx | 3727.0 | 186.0 |
| 2004 | 1 | abilene_tx | elko_nv | 44.0 | 1.0 |
| 2004 | 1 | abilene_tx | houston_tx | 835.0 | 81.0 |
| 2004 | 1 | abilene_tx | lubbock_tx | 0.0 | 44.0 |

**Table 6: A preview of the cleaned US data. Each row represents a flight connection with various attributes such as year, month, source, target, passengers, and weight showing the number of departures performed.**

| RDT-ID | Date | Type | Company | Completely cancelled | Partly cancelled | Maximum delay | Station name | Arrival time | Arrival delay | Arrival cancelled | Departure time | Departure delay | Departure cancelled |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 738804 | 01-01-2019 | Intercity | NS | False | False | 1 | Rotterdam Centraal | NaN | NaN | NaN | 2019-01-01T02:0 | 1.0 | False |
| 738804 | 01-01-2019 | Intercity | NS | False | False | 0 | Delft | 2019-01-01T01:0 | 1.0 | False | 2019-01-01T02:1 | 0.0 | False |
| 738804 | 01-01-2019 | Intercity | NS | False | False | 0 | Den Haag HS | 2019-01-01T01:1 | 0.0 | False | 2019-01-01T02:2 | 1.0 | False |
| 738804 | 01-01-2019 | Intercity | NS | False | False | 0 | Leiden Centraal | 2019-01-01T01:2 | 0.0 | False | 2019-01-01T02:4 | 0.0 | False |
| 738804 | 01-01-2019 | Intercity | NS | False | False | 0 | Schiphol Airport | 2019-01-01T01:3 | 0.0 | False | 2019-01-01T03:0 | 0.0 | False |
| 13090236 | 29-02-2024 | Intercity | NS | False | False | 0 | Amsterdam Zuid | 2024-03-01T00:4 | 8.0 | False | 2024-03-01T00:4 | 9.0 | False |
| 13090236 | 29-02-2024 | Intercity | NS | False | False | 0 | Schiphol Airport | 2024-03-01T00:5 | 9.0 | False | 2024-03-01T00:5 | 8.0 | False |
| 13090236 | 29-02-2024 | Intercity | NS | False | False | 0 | Leiden Centraal | 2024-03-01T00:5 | 8.0 | False | 2024-03-01T00:7 | 8.0 | False |
| 13092579 | 29-02-2024 | Extra trein | NS | False | False | 0 | Meppel | NaN | NaN | NaN | NaN | NaN | NaN |

**Table 7: A preview of the original NS dataset from rijdendetreinen.nl. A unique RDT-ID defines a train service/trajectory and every row defines a stop at a station.**

| YearMonth | source | target | Rides planned | Final arrival delay | Final arrival cancelled | Completely cancelled | Intermediate arrival delays | Proportion delayed | Significant Delay |
|---|---|---|---|---|---|---|---|---|---|
| 2019-01 | 's-Hertogenbosch | Arnhem Centraal | 57 | 20 | 4 | 0 | 36 | 0.3509 | True |
| 2019-01 | 's-Hertogenbosch | Den Haag Centraal | 1106 | 99 | 20 | 5 | 853 | 0.0899 | False |
| 2019-01 | 's-Hertogenbosch | Deurne | 757 | 247 | 8 | 2 | 503 | 0.3272 | True |
| 2019-01 | 's-Hertogenbosch | Dordrecht | 100 | 23 | 3 | 0 | 44 | 0.2300 | True |
| 2019-01 | 's-Hertogenbosch | Eindhoven Centraal | 257 | 81 | 1 | 1 | 138 | 0.3164 | True |
| 2019-01 | 's-Hertogenbosch | Roosendaal | 34 | 9 | 3 | 1 | 15 | 0.2727 | True |
| 2019-01 | Bergen op Zoom | Amsterdam Centraal | 22 | 14 | 1 | 0 | 22 | 0.6364 | True |
| 2019-01 | Bergen op Zoom | Roosendaal | 44 | 2 | 4 | 4 | 2 | 0.0500 | False |
| 2019-01 | Breda | Amsterdam Centraal | 1257 | 727 | 131 | 70 | 948 | 0.6125 | True |

**Table 8: A preview of the cleaned and transformed NS dataset displaying the unique trajectories per month, their counts, and the dependent variable (most-right column) being the binary value showing if a trajectory was significantly delayed (50th percentile, > 21%) within that month.**

| Data | Number of rows | Description |
|---|---|---|
| Original | 115.266.904 | Service archive from 2019 till April 2024 |
| Cleaned | 75.620.094 | Filtered on NS and trains (e.g., no replacement busses or taxis) |
| Grouped | 7.344.087 | Grouped by RDT-ID (e.g., all trajectory rides) |
| Daily | 559.668 | Daily Aggregation on source - target (all unique trajectories per day) |
| Monthly | 28.557 | Monthly aggregation on source - target (all unique trajectories per month); Trajectories outside NL removed; threshold for min. 4 rides |

**Table 9: Overview of the different data transformations on the NS dataset, the number of rows for each dataset, and a description of what the dataset entails or how it has been transformed.**

# B EXPLORATORY DATA ANALYSIS

## B.1 Dataset size comparison

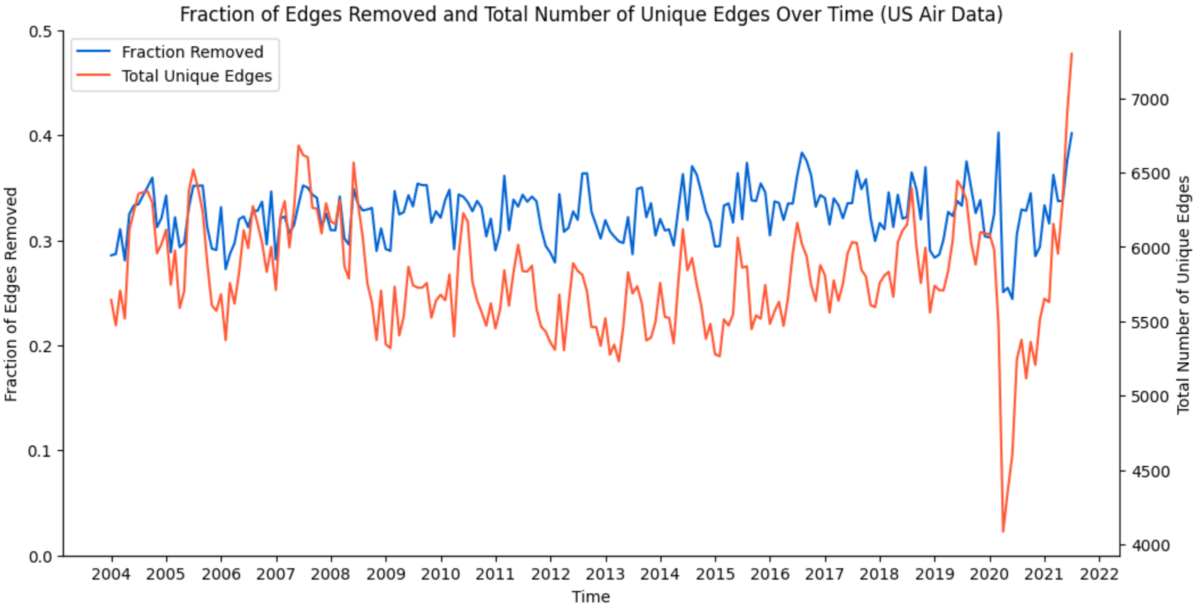


**Figure 9: The proportions of removed edges (left axis) and the number of unique edges (right axis) per month for the US air data.**

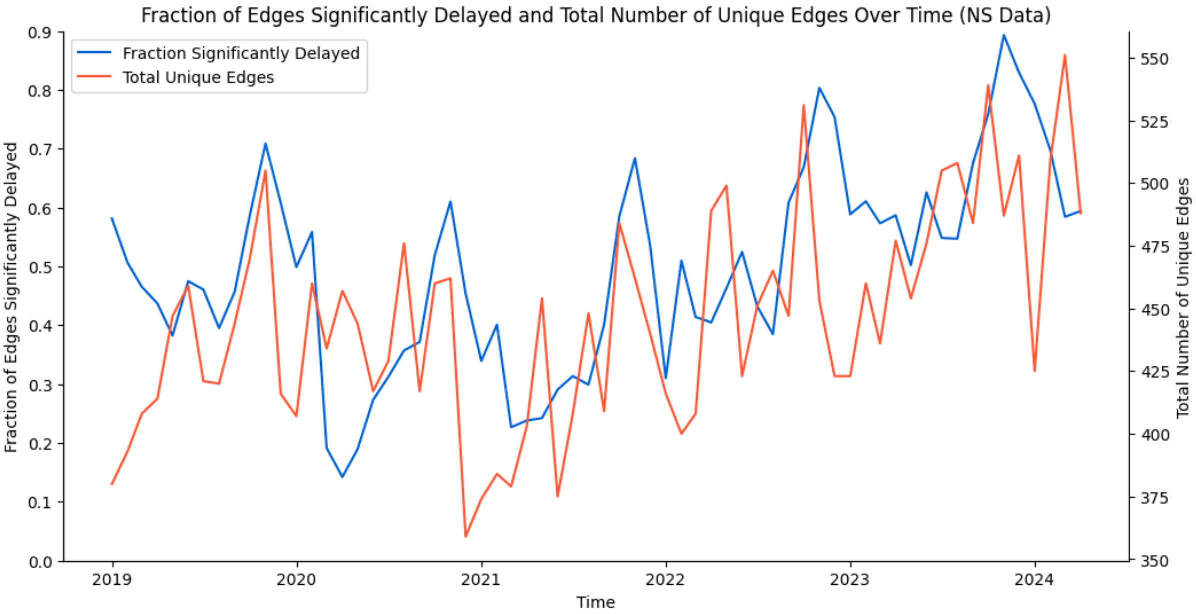


**Figure 10: The proportion of significantly delayed edges (left axis) and the number of unique edges (right axis) per month for the NS data.**

### B.2 Map

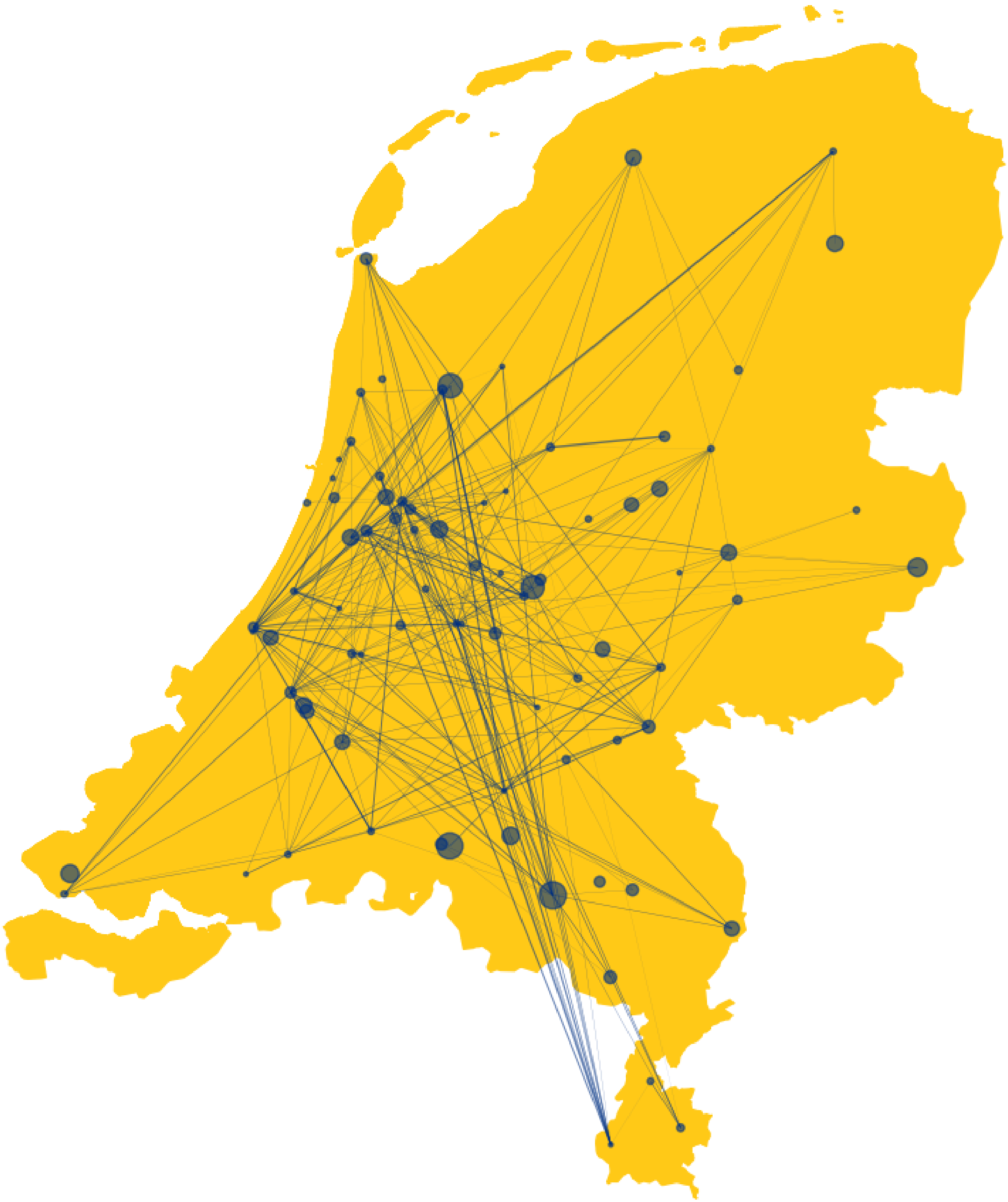


**Figure 11: Trajectories in April 2024 with the size of the nodes representing the degree of the nodes in the graph, e.g., the connections (or edges) it has to other nodes. The thickness of the edges displays the proportion of delayed edges.**

# C FEATURE SETS

## C.1 Unweighted Topological Features (TF)

| Feature | Definition | Description |
|---|---|---|
| Common Neighbors (CN) | $\lvert\Gamma_i \cap \Gamma_j\rvert$ | The number of common neighbors of nodes $i$ and $j$ |
| Salton Index (SA) | $\frac{\lvert\Gamma_i\cap\Gamma_j\rvert}{\sqrt{k_i\times k_j}}$ | The number of common neighbors normalized by the geometric average degree of both nodes |
| Jaccard Index (JA) | $\frac{\lvert\Gamma_i\cap\Gamma_j\rvert}{\lvert\Gamma_i\cup\Gamma_j\rvert}$ | The number of common neighbors normalized by the union of neighbors of both nodes |
| Sørensen Index (SO) | $\frac{2\lvert\Gamma_i\cap\Gamma_j\rvert}{k_i+k_j}$ | The number of common neighbors normalized by the average degree of the two nodes |
| Hub Promoted Index (HPI) | $\frac{\lvert\Gamma_i\cap\Gamma_j\rvert}{\min(k_i,k_j)}$ | The number of common neighbors normalized by the smaller degree of the two nodes |
| Hub Depressed Index (HDI) | $\frac{\lvert\Gamma_i\cap\Gamma_j\rvert}{\max(k_i,k_j)}$ | The number of common neighbors normalized by the larger degree of the two nodes |
| Leicht-Holme-Newman Index (LHNI) | $\frac{\lvert\Gamma_i\cap\Gamma_j\rvert}{k_i\times k_j}$ | The number of common neighbors normalized by the product of degrees of the two nodes |
| Preferential Attachment Index (PA) | $k_i \times k_j$ | The product of the degrees of the two nodes |
| Adamic-Adar Index (AA) | $\sum_{n\in\Gamma_i\cap\Gamma_j}\frac{1}{\log k_n}$ | The number of common neighbors with each of them normalized by the logarithm of their degree |
| Resource Allocation Index (RA) | $\sum_{n\in\Gamma_i\cap\Gamma_j}\frac{1}{k_n}$ | The number of common neighbors with each of them normalized by their degree |
| Local Path Index (LPI) | $S_{ij,2} + \epsilon S_{ij,3}$ | The first term represents the number of paths of length equal to 2 between the node $i$ and $j$. The second term is the number of paths of length equal to 3 between the node $i$ and $j$ damped by parameter $\epsilon$. We set $\epsilon = 0.01$. |

**Table 10: The definitions and descriptions of the unweighted topological features (TF).**

### C.2 Weighted Topological Features (WTF)

| Feature | Definition | Description |
|---|---|---|
| Weighted Common Neighbors (WCN) | $\sum_{w_n \in \Gamma_i \cap \Gamma_j} \min(w_{in}, w_{jn})$ | The number of common neighbors weighted by the minimum weight of their connections |
| Weighted Salton Index (WSA) | $\frac{\text{WCN}}{\sqrt{w_i \times w_j}}$ | The weighted common neighbors normalized by the geometric average weight of both nodes |
| Weighted Jaccard Index (WJA) | $\frac{\text{WCN}}{w_i + w_j - \text{WCN}}$ | The weighted common neighbors normalized by the union of neighbors of both nodes |
| Weighted Sørensen Index (WSO) | $\frac{2 \times \text{WCN}}{w_i + w_j}$ | The weighted common neighbors normalized by the average weight of the two nodes |
| Weighted Hub Promoted Index (WHPI) | $\frac{\text{WCN}}{\min(w_i, w_j)}$ | The weighted common neighbors normalized by the smaller weight of the two nodes |
| Weighted Hub Depressed Index (WHDI) | $\frac{\text{WCN}}{\max(w_i, w_j)}$ | The weighted common neighbors normalized by the larger weight of the two nodes |
| Weighted Leicht-Holme-Newman Index (WLHNI) | $\frac{\text{WCN}}{w_i \times w_j}$ | The weighted common neighbors normalized by the product of weights of the two nodes |
| Weighted Preferential Attachment Index (WPA) | $w_i \times w_j$ | The product of the weights of the two nodes |
| Weighted Adamic-Adar Index (WAA) | $\sum_{w_n \in \Gamma_i \cap \Gamma_j} \frac{1}{\log w_n}$ | The weighted common neighbors with each of them normalized by the logarithm of their weight |
| Weighted Resource Allocation Index (WRA) | $\sum_{w_n \in \Gamma_i \cap \Gamma_j} \frac{1}{w_n}$ | The weighted common neighbors with each of them normalized by their weight |
| Weighted Local Path Index (WLPI) | $W_{ij,2} + \epsilon W_{ij,3}$ | The first term represents the number of paths of length equal to 2 between the node $i$ and $j$. The second term is the number of paths of length equal to 3 between the node $i$ and $j$ damped by parameter $\epsilon$. We set $\epsilon = 0.01$. |

**Table 11: The definitions and descriptions of the weighted topological features (WTF).**

## D HYPERPARAMETERS USED FOR XGBOOST CLASSIFIER

| Hyperparameter | Value | Explanation |
|---|---|---|
| lambda | 0.5650701862593042 | L2 regularization term on weights. Increasing this value will make model more conservative. Range: $[0, \infty]$ |
| alpha | 0.0016650896783581535 | L1 regularization term on weights. Increasing this value will make model more conservative. Range: $[0, \infty]$ |
| colsample_bytree | 1.0 | The subsample ratio of columns when constructing each tree. Subsampling occurs once for every tree constructed. |
| subsample | 0.5 | Denotes the fraction of observations to be random samples for each tree. Typically set to 0.5. |
| learning_rate | 0.009 | Step size shrinkage used in update to prevent overfitting. After each boosting step, we can directly get the weights of new features, and $learning_rate$ shrinks the feature weights to make the boosting process more conservative. Range: $[0, 1]$ |
| n_estimators | 625 | Number of boosting stages to be run. More stages can improve accuracy but may lead to overfitting and higher computation time. |
| objective | reg:squarederror | Regression with squared loss. |
| max_depth | 5 | Maximum depth of a tree. Increasing this value will make the model more complex and more likely to overfit. 0 indicates no limit on depth. Range: $[0, \infty]$ |
| min_child_weight | 6 | Minimum sum of instance weight (hessian) needed in a child. Range: $[0, \infty]$ |

**Table 12: Hyperparameters provided in baseline [25] for the XGBoost Classifier. Explanations are from the official XGBoost Documentation Website.**

## E HYPERPARAMETER GRIDS FOR RADOMIZEDSEARCHCV

| Classifier | Hyperparameter | Values | Explanation |
|---|---|---|---|
| GradientBoosting | n_estimators | [50, 100, 200, 300] | Number of boosting stages to be run. More stages can improve accuracy but may lead to overfitting and higher computation time. |
| | learning_rate | [0.01, 0.05, 0.1, 0.2] | Shrinks the contribution of each tree. Lower values require more trees to achieve the same training error, balancing between speed and accuracy. |
| | max_depth | [3, 5, 7, 9] | Maximum depth of the individual trees. Deeper trees can capture more patterns but are more prone to overfitting. |
| | min_samples_split | [2, 5, 10] | Minimum number of samples required to split an internal node. Prevents overfitting by ensuring splits are based on a sufficient number of samples. |
| | min_samples_leaf | [1, 2, 4] | Minimum number of samples required to be at a leaf node. Prevents overfitting by ensuring leaves have enough samples. |
| | max_features | ['sqrt', 'log2'] | Number of features to consider when looking for the best split. Helps in reducing overfitting. |
| AdaBoost | n_estimators | [50, 100, 200, 400] | Number of boosting stages to be run. More stages can improve performance but also increase computation time. |
| | learning_rate | [0.01, 0.1, 0.5, 1.0] | Shrinks the contribution of each classifier. Smaller values require more stages to achieve the same training error. |
| LogisticRegression | C | [0.001, 0.01, 0.1, 1, 10, 100] | Inverse of regularization strength. Smaller values specify stronger regularization. |
| | penalty | ['l1', 'l2', 'elasticnet'] | Norm used in the penalization. Helps in controlling overfitting by penalizing large coefficients. |
| | solver | ['liblinear', 'saga'] | Algorithm to use in the optimization problem. Different solvers have different strengths and computational efficiencies. |
| RandomForest | n_estimators | [100, 200, 500, 1000] | Number of trees in the forest. More trees can improve performance but also increase computation time. |
| | max_features | ['auto', 'sqrt', 'log2'] | Number of features to consider when looking for the best split. Helps in reducing overfitting by ensuring diverse trees. |
| | max_depth | [10, 20, 30, None] | Maximum depth of the tree. Limits the number of nodes in the tree to prevent overfitting. |
| | min_samples_split | [2, 5, 10] | Minimum number of samples required to split an internal node. Ensures splits are meaningful by requiring a sufficient number of samples. |
| | min_samples_leaf | [1, 2, 4] | Minimum number of samples required to be at a leaf node. Ensures leaves have enough samples to be statistically meaningful. |
| | bootstrap | [True, False] | Whether bootstrap samples are used when building trees. Affects the variance and bias of the model. |
| DecisionTree | max_features | ['auto', 'sqrt', 'log2'] | Number of features to consider when looking for the best split. Helps in reducing overfitting. |
| | max_depth | [10, 20, 30, None] | Maximum depth of the tree. Limits the number of nodes in the tree to prevent overfitting. |
| | min_samples_split | [2, 5, 10] | Minimum number of samples required to split an internal node. Ensures splits are meaningful by requiring a sufficient number of samples. |
| | min_samples_leaf | [1, 2, 4] | Minimum number of samples required to be at a leaf node. Ensures leaves have enough samples to be statistically meaningful. |
| | criterion | ['gini', 'entropy'] | Function to measure the quality of a split. Different criteria can lead to different splits and tree structures. |

**Table 13: Hyperparameter grids for RandomizedSearchCV used for the classifier comparison on the NS dataset. Explanations are from the Scikit Learn Website.**

# F CONFUSION MATRICES

## F.1 US Air data

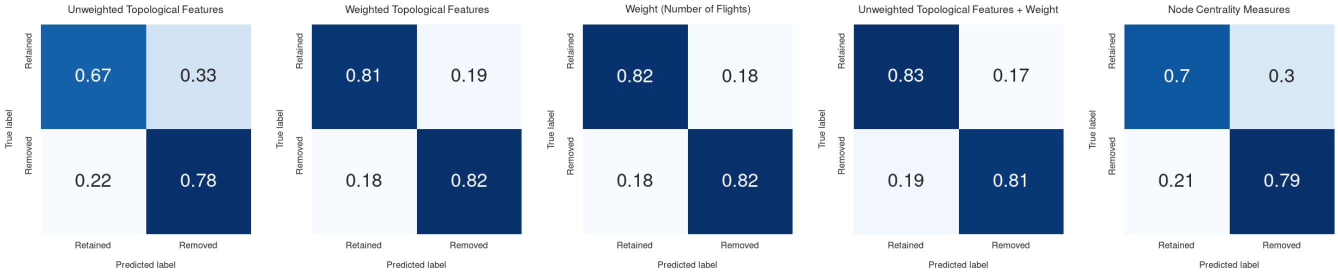


**Figure 12: Confusion matrices for simultaneous testing on the US air data with different feature sets.**

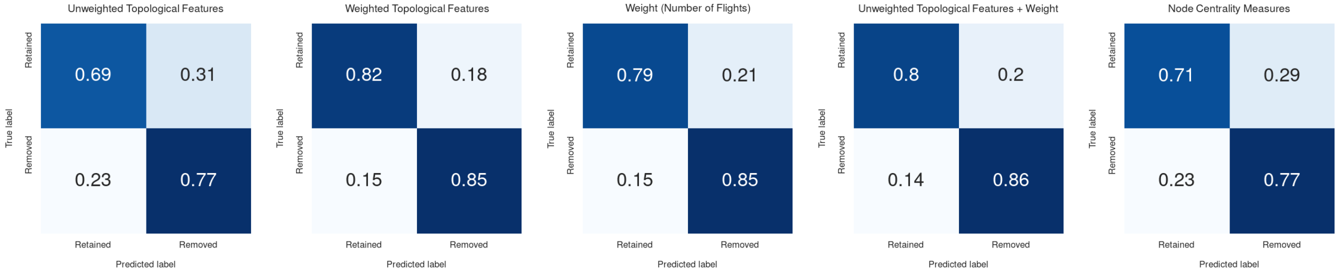


**Figure 13: Confusion matrices for non-simultaneous testing on the US air data with different feature sets.**

## F.2 NS data

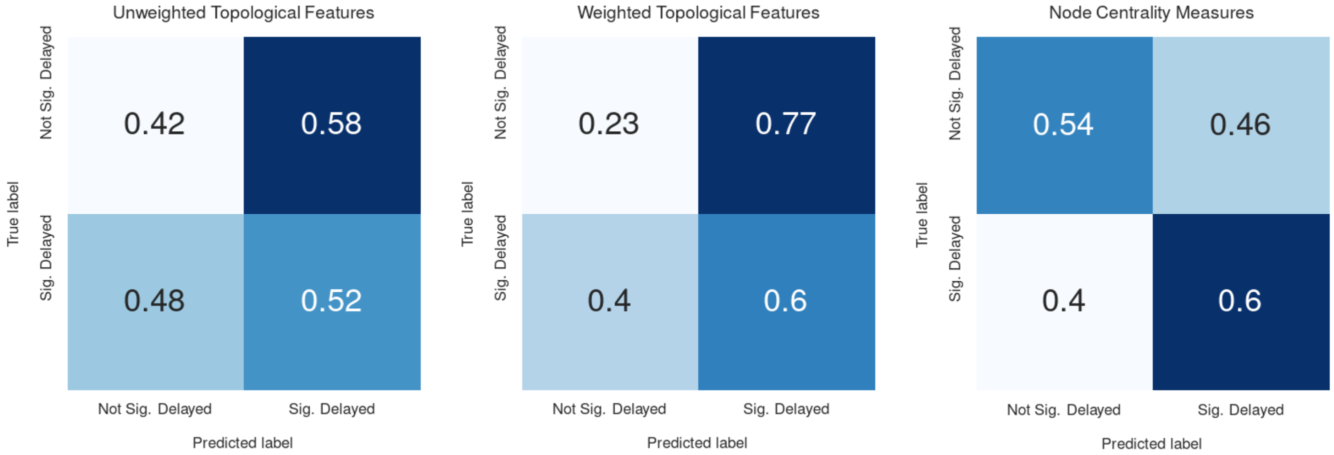


**Figure 14: Confusion matrices for simultaneous testing on the NS data with different feature sets.**

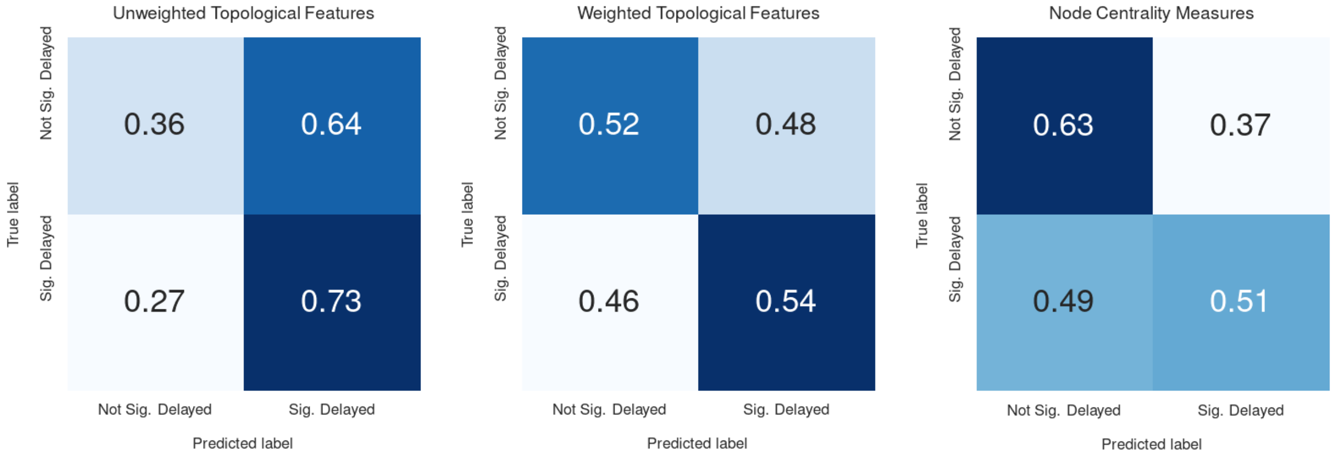


**Figure 15: Confusion matrices for non-simultaneous testing on the NS data with different feature sets.**

# G SHAP VALUES FEATURES FOR NS DATA

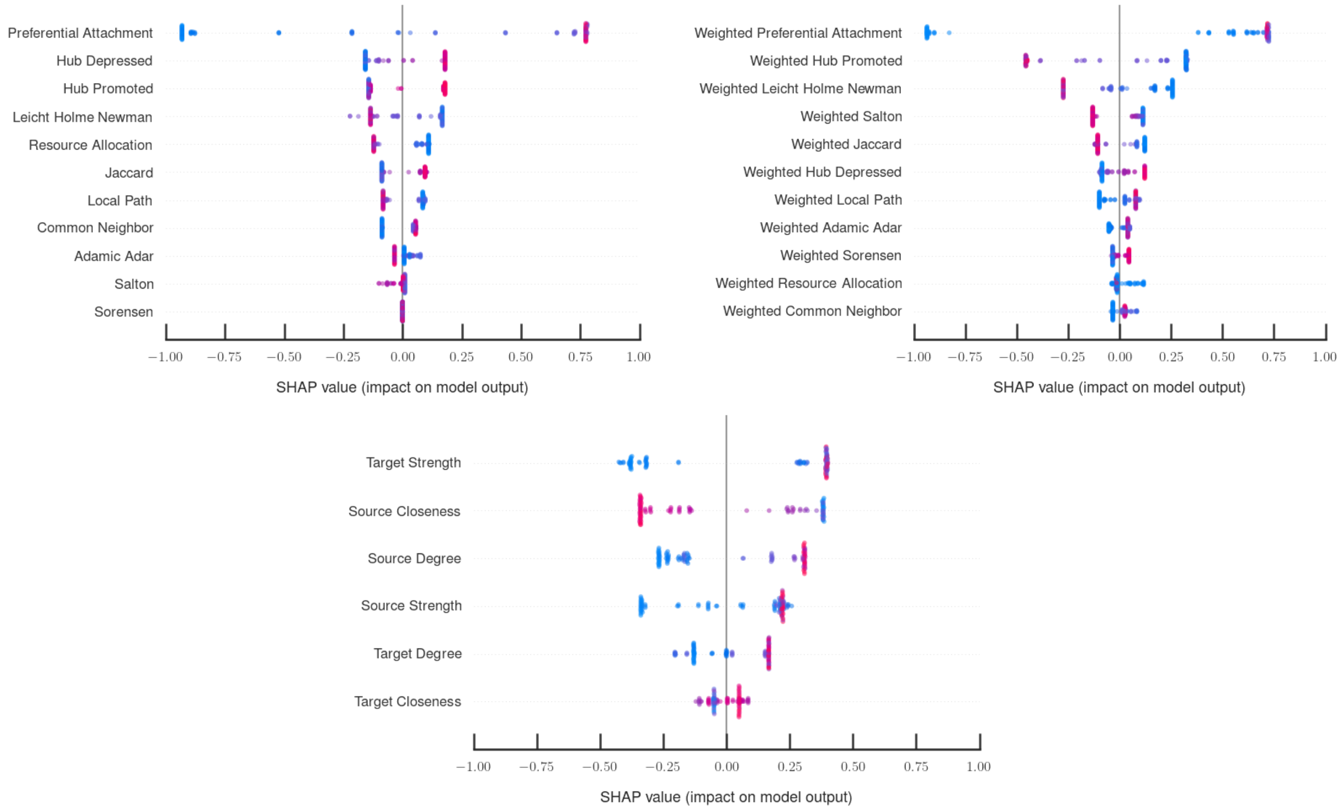


**Figure 16: SHAP figures for the three different feature sets used as predictors in the NS data.**

# H METRICS

## H.1 Simultaneous Testing

| Classifier | TF | WTF | NCM |
|---|---|---|---|
| AdaBoost | 0.627 | 0.627 | 0.639 |
| DecisionTree | 0.597 | 0.601 | 0.597 |
| GradientBoosting | 0.625 | 0.623 | 0.629 |
| LogisticRegression | 0.637 | **0.640** | 0.637 |
| RandomForest | 0.625 | 0.629 | **0.648** |
| XGBoost | **0.638** | 0.632 | 0.647 |

**Table 14: F1 scores of classifiers for every feature set when simultaneous testing.**

| Classifier | TF | WTF | NCM |
|---|---|---|---|
| AdaBoost | 0.646 | 0.653 | 0.661 |
| DecisionTree | 0.615 | 0.614 | 0.595 |
| GradientBoosting | 0.645 | 0.649 | 0.665 |
| LogisticRegression | **0.688** | 0.685 | 0.691 |
| RandomForest | 0.666 | 0.671 | 0.695 |
| XGBoost | 0.684 | **0.686** | **0.701** |

**Table 15: ROC AUC scores of classifiers for every feature set when simultaneous testing.**

## H.2 Non-Simultaneous Testing

| Classifier | TF | WTF | NCM |
|---|---|---|---|
| AdaBoost | **0.501** | 0.510 | 0.513 |
| DecisionTree | 0.481 | 0.485 | 0.522 |
| GradientBoosting | 0.506 | 0.513 | **0.525** |
| LogisticRegression | 0.478 | **0.520** | 0.502 |
| RandomForest | 0.497 | 0.510 | 0.519 |
| XGBoost | 0.484 | 0.500 | 0.494 |

**Table 16: F1 scores of classifiers for every feature set when non-simultaneous testing.**

| Classifier | TF | WTF | NCM |
|---|---|---|---|
| AdaBoost | 0.533 | 0.537 | 0.548 |
| DecisionTree | 0.522 | 0.524 | 0.537 |
| GradientBoosting | 0.529 | 0.537 | 0.559 |
| LogisticRegression | 0.535 | 0.552 | 0.552 |
| RandomForest | **0.538** | **0.548** | **0.569** |
| XGBoost | **0.538** | 0.543 | 0.551 |

**Table 17: ROC AUC scores of classifiers for every feature set when non-simultaneous testing.**