Deliverable D 4.3

WP4 Report on experimentation, analysis, and discussion of results

Project acronym:	RAILS
Starting date:	01/12/2019
Duration (in months):	43
Call (part) identifier:	H2020-S2R-OC-IPX-01-2019
Grant agreement no:	881782
Due date of deliverable:	Month 37
Actual submission date:	June 30^{th} 2023
Responsible/Author:	UNIVLEEDS
Dissemination level:	Public
Status:	Issued

Reviewed: no

Document history		
Revision	*Date*	*Description*
1	Feb 15^{th} 2023	First issue for internal review
2	March 31^{st} 2023	Second issue for AB review
2	June 30^{th} 2023	Third issue for EU review

Report contributors		
Name	Beneficiary Short Name	Details of contribution
Ronghui Liu	UNIVLEEDS	Coordinator
Ruifan Tang	UNIVLEEDS	Contributor
Zhiyuan Lin	UNIVLEEDS	Contributor
Rob M.P. Goverde	TU DELFT	Reviewer

Advisory Board Reviewers	
Name	*Company or Institution*
He Wang	School of Computing, University of Leeds
Wilco Tielman	ProRail

Funding

This project has received funding from the Shift2Rail Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement n. 881782 Rails. The JU receives support from the European Union's Horizon 2020 research and innovation program and the Shift2Rail JU members other than the Union.

Disclaimer

Contents

Executive Summary

This deliverable proposes innovative AI approaches to address the problems and challenges emerged from the methodological proof-of-concepts reported in Deliverable D4.2, in order to investigate the adoption of learning techniques and other AI methods for enhanced rail traffic planning and management. On the basis of the objectives, research questions, and AI techniques identified in the previous deliverable for two pilot case studies, experimental proof-of-concepts are provided to explore the technical feasibility of specific railway functionalities through the use of AI approaches transferred from other transportation sectors.

To this aim, the document addresses the following themes for each case study: i) a technical description of the proposed innovative methods, highlighting the problem statement to solve and the learning techniques which are going to be leveraged; ii) the definition of the training phase for the learning approaches, which could be conducted exploiting selected datasets, as well as through the use of ad-hoc simulation platforms; iii) the description of the validation procedure to show the effectiveness of the proposed strategies in concrete operational scenarios; iv) a preliminary discussion of the results to highlight possible benefits and drawbacks of the innovative approaches.

A description of the background for each case study is addressed is Section 1; the main objectives of the deliverable are detailed in Section 2, while the content of the two experimental proof-of-concepts is reported in Section 3.

Abbreviations and acronyms

Abbreviations / Acronyms	Description
WP	Work Package
IPX	Innovation Programme X
AI	Artificial Intelligence
GDPR	General Data Protection Regulation
TPE	TransPennine Express
PCS	Polar Coordinates system
TOC	Train Operating Company
NR	Network Rail
SDNE	Structural Deep Network Embedding
PCA	Principle Component Analysis
ML	Machine Learning
DT	Decision Tree
RF	Random Forest
SVD	Singular Value Decomposition
MLP	Multilayer Perceptron
NLP	Natural Language Processing
KNN	K-Nearest Neighbour Algorithm
BRS	British Railway System
ORR	Office of Rail and Road
TRUST	Train Running Under System TOPS
TOPS	Total Operations Processing System
RSSB	Rail Safety and Standards Board
TDA	Train Delay Attributor
DA	Delay Attributor
API	Application Programming Interface
DAB	Delay Attribution Board
PSS	Performance System Strategy
SGD	Stochastic Gradient Descent

1. Background

The present document constitutes the Deliverable D4.3 “WP4 Report on experimentation, analysis, and discussion of results” of the Shift2Rail JU project “Roadmaps for AI integration in the Rail Sector” (RAILS). The project is in the framework of Shift2Rail’s Innovation Programme IPX. As such, RAILS does not focus on a specific domain, nor does it directly contribute to specific Technical Demonstrators but contributes to Disruptive Innovation and Exploratory Research in the field of Artificial Intelligence within the Shift2Rail Innovation Programme. The successor of the Shift2Rail Joint Undertaking is currently the Europe’s Rail Joint Undertaking (EU-Rail) established by Council Regulation (EU) 2021/2085 of 19 November 2021.

The RAILS Workpackage WP4 investigates the adoption of learning techniques and other AI methods for enhanced rail safety and automation. The present deliverable is consequent to the results reported in Deliverable D4.2, in which methodological proof-of-concepts have been provided for two selected case studies: “Graph Embedding based Primary Delay Prediction”, and “Big Data on Incident Attribution Analysis”. The first one aims to predict train primary delay times by optimised feature engineering, especially in how to represent train routes effectively using advanced approaches such as graph embedding. As to the second case study, the main goal is to provide proper impetus to the current delay attribution analysis, as well as proactively respond to the Steering Group’s promising view scope. The disadvantages of one-hot encoding in representing train routes are identified which have motivated our embedding based approach in the first case. For the second case study, we have chosen to focus on further automating the attribution process of cascading delays in order to explore the potential application of AI approaches to this process and to bridge the present communication gap between the corporate and industrial sectors. Our goal is to train a link prediction model, which, given two nodes, forecasts the existence or absence of a propagation connection between them. During the interactive delay attributing visualisation and possible propagation link prediction sub-tasks, respectively, Big Data and Graph Neural Network approaches are introduced.

The following step is therefore to support the theoretical proof-of-concepts proposed in Deliverable D4.2 with experimental results. This could give an answer to the research questions and the expected results emerged in the previous deliverable, and could represent a further step towards the definition of a benchmark for future research inspiration.

2. Objective

This document, in line with the previous deliverables, deals with the following objectives of the RAILS project:

- Objective 4: Development of methodological and experimental proof-of-concepts;
- Objective 5: Development of Benchmarks, Models and Simulations.

In particular, two pilot case studies have been selected, namely, “Graph Embedding based Primary Delay Prediction”, and “Big Data on Incident Attribution Analysis”; for each of them, methodological proof-of-concepts have been addressed to study the feasibility of AI methods in the railway field, and some learning approaches have been identified as a potential solution to develop their respective railway functionalities.

On the basis of the considerations made in Deliverable D4.2, the main goal of this deliverable is to define innovative AI models for the considered case studies to be evaluated via test and validation activities, in order to understand how and if the AI approaches identified during the previous tasks can support and enhance rail traffic planning and management. To this aim, it focuses on the following objectives:

- the definition of detailed AI models based on the selected learning approaches;
- the description of the learning process of the proposed methods through a training phase, which can be carried out exploiting specific datasets;
- the validation of the proposed models to show their effectiveness in synthetic and real-world operational scenarios;
- a preliminary analysis of the results, to highlight the possible benefits and drawbacks of the proposed techniques.

It is worth highlighting that this study *is not intended to give an exhaustive answer to a specific problem; it is instead meant to be a step towards the acquisition of the necessary knowledge to understand the potentiality of AI in railways, and to drive the rail sector towards a vision of AI-enabled traffic planning and management.* In this direction, the main object of the next deliverable will be an in-depth analysis to identify gaps and opportunities, weaknesses and strengths emerged from each case study, with the final aim of defining technology roadmaps towards the effective adoption of AI in the rail sector.

3. Introduction

Deliverable D4.3 reports the validation activities of the solutions and approaches described and deeply analysed in Deliverable D4.2. It focuses on the analyses and experiments conducted applying the selected AI techniques to the pilot case studies identified in Deliverable D4.1 based on both synthetic and real-world scenarios. Hence, this deliverable provides meaningful insights and information on the validity of the research results and the feasibility of the approaches in real settings. The heart of this document consists of two main chapters, addressing the above mentioned issues for the two selected pilot case studies: Chapter 4 is devoted to “Graph Embedding based Primary Delay Prediction” and Chapter 5 deals with “Big Data on Incident Attribution Analysis”.

The two chapters share the same structure: a brief introduction constitutes Sections 4.1 and 5.1; a detailed description of the proposed AI models is provided in Sections 4.2 and 5.2; the selection of specific datasets as well as the development of a simulation platform or experiment environment for training and validation purposes is described in Sections 4.3 and 5.3; the training phase together with the validation of the proposed approach in concrete operational scenarios are deeply analysed in Sections 4.4, 4.5 and 5.4; the results of the validation phase are shown in Sections 4.6 and 5.5; finally, a preliminary discussion of the potential advantages of the proposed approaches is addressed in Sections 4.6 and 5.5.

A critical examination of the work and of the results obtained in this Deliverable, also against the current state-of-the-art in railways, will be the object of the next Deliverable D4.4 (“Report on identification of future innovation needs and recommendations for improvements”). Specifically, the latter will report lessons learned, weaknesses and strengths shown by of each exploited technology, technical and implementation recommendations, unaddressed issues, innovation needs, with the aim of identifying technology roadmaps for AI integration in the rail sector.

4. Graph Embedding based Primary Delay Prediction

4.1. Introduction

As we introduced in the previous deliverables [1] [2], the purpose of this case study is to estimate the overall degree of primary delay level within a certain time in the future on an individual train service basis, based on data from various periods in the railway's operating history, and taking into account the static characteristics of each serving (pass-by or dwell) station, as well as the structural network characteristics (i.e., connectivity between these stations, route weight, and network density for various areas). The Structural Deep Network Embedding (SDNE) algorithm was created as an effective dimensionality reduction tool in computer science. This method has been refined and upgraded in this case study for interpreting station dependencies and structural correlations. That is, a similarity network is constructed for a set of D-dimensional nodes based on their neighborhood information, and each node of the graph is then embedded into a d-dimensional vector space, where $d \ll D$. The main idea behind embedding is to keep related nodes closer to one other in vector space so that the original network's structural relationships can be preserved. The main goal of this concept is to create an N-dimensional vector for each railway station, with each element representing the scalar value on a specific vector direction in Euclidean space. We also propose combining the obtained hypernode embedding vectors (i.e. nodes/stations distributed sequentially in a particular route) into a route embedding vector, which compresses and aggregates more structural information of the target railway network, reducing the dimensions of available features.

The varied aspects of railway punctuality have been the subject of various studies. A number of distinct circumstances can result in delays. According to a thorough investigation [3], unanticipated disruptions like power and signal system outages, rolling stock problems, and bad weather, result in significant delays, while imposed constraints such as temporary speed restrictions, lengthy engineering projects, or crew shortages typically result in minor delays. Sometimes, one train company's delayed services may cause delays for other trains either at the same station or at neighboring stations. Which is called delay propagation. Secondly, when delays propagate, they can cause operational conflicts between trains that are adjacent to each other. This can disrupt the train operation plan and threaten the safety of trains operated by other companies. These conflicts are known as timetable conflicts. Thirdly, there are minor disruptions that the system may not detect or record, normally known as "disturbances." These disturbances can often be resolved during train operations by providing a margin of time or running allowance. In this primary delay prediction case, we mainly investigate sub-threshold delays (disturbances) and those delays above the defined threshold but not caused by pre-planned disruptions (e.g. closed track due to maintenance works). Also, we limit our scope within the directly triggered delay however we do not intend to explore the delay propagation mechanism or occurrence of secondary delays. They will be left in the next chapter 5 for discussion. A primary delay prediction model for the Hong Kong subway system, which evaluates the interaction between external infrastructure defects and prospective congestion shown in the study of [4]. It statistically recovers the distribution of infrastructure-caused delays at each subway station using the fixed-parameter

maximum likelihood estimation. [5] also performed several statistical analysis towards arrival delays, departure delays, dwell time, and route section occupation time, by means of the S-plus software tool. The results show that delay times for trains often follow normal or negative-exponential distributions, which can be used to create better timetables and more accurate predictions of delays using historical data. However, subsequent studies are supposed to develop appropriate models for predicting how train delays spread throughout stations and networks. A timed event graph-based model was presented in [6] for predicting running times and arrival times of punctual and delayed trains as decision support for dispatchers to demonstrate the dependencies of running times and dwell times on current delays and periods of the day. [7] investigated the association between train delays and severe weather conditions by collecting and analysing a three-month dataset of weather along the Beijing-Guangzhou line, China's busiest train route. They applied a machine learning-based gradient tree boosting algorithm to predict the delay mins at each station for train services.

The Structural Deep Network Embedding (SDNE) graph embedding algorithm was first created as an effective dimensionality reduction tool in the computer science field by [8]. The main idea behind embedding is to keep related nodes closer to one other in vector space so that the original network's structural relationships can be preserved. The main goal of this concept is to create an N-dimensional vector for each railway station, with each element representing the scalar value on a specific vector direction in Euclidean space. Each value in the vector has no discernible significance, yet it does represent a characteristic of a certain station in part. When we wish to compare how similar two stations are, such a representation comes in handy. In this regard, the SDNE approach considerably compresses the fundamental information, making vector operations simpler and faster than traditional mathematical procedures. We also propose combining the obtained hyper-node embedding vectors (i.e., nodes/stations distributed sequentially on a specific route) into a route embedding vector, which compresses and aggregates more structural information of the target railway network, reducing the dimensions of available features. The following requirements must be met by the expected route embedding depictions:

- Regardless of the length of a specific route, the obtained route embedding vectors must be uniform in size – this makes them more convenient to use as input features for subsequent prediction tasks.
- The route representations can explicitly reflect the characteristics of the entire route, including the density of en-route station cluster, the sequence of these stations, and the degree of congestion on this route.
- Local and global characteristics can be effectively preserved by route embedding vectors.

4.2. Model Description

The main goal of the entire methodology framework is to acquire a structural deep network representation. In order to effectively capture the extremely non-linear structure, a unique deep model called the "Structural Deep Network Embedding approach" was proposed to learn the station representation in a railway network. This model is based on the most recent successful applications of deep learning methods that were originally derived from

[8] and which have been shown by [9–12] that it potentially has strong representational capabilities when dealing with a variety of data types.

There is, however, no concrete evidence showing that such an attempt on a public transit system, notably within a train system, has been made. In Figure 4.1, the suggested SDNE framework is shown. This framework uses the original railway network characteristics as input for the encoder-decoder layers, whose detailed structure includes the defining of first-order and second-order proximity and the identification of connectivity status between any two nodes. In order to update the resulting embedding vector to satisfy the lowest overall loss costs during training, loss functions corresponding to each proximity in the output layer will provide the encoder-decoder with the optimized parameters. As a result of this, each node in the network is given its final low-dimensional embedding representation.

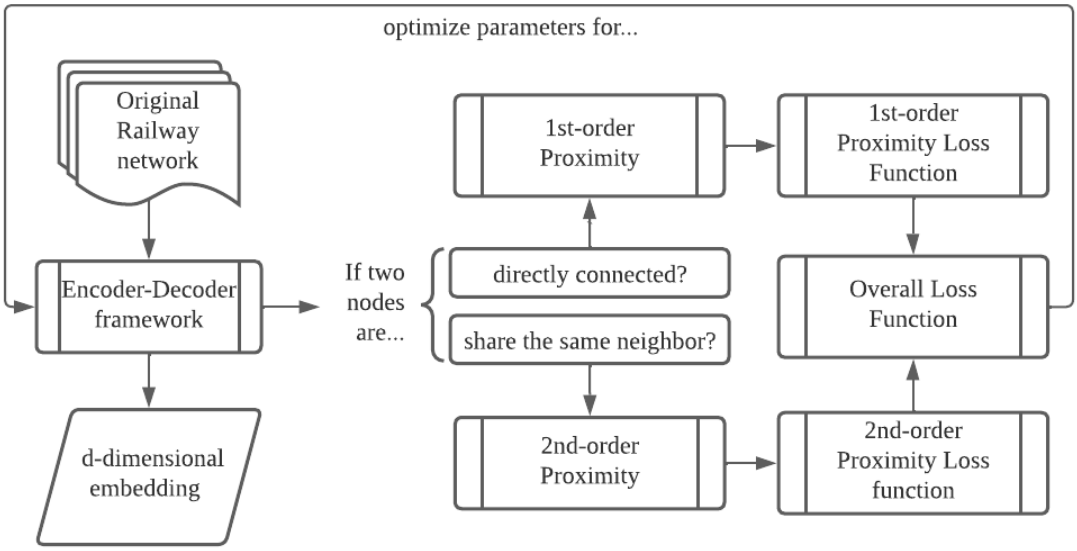


Fig. 4.1. The flow-charted SDNE framework

4.3. Dataset Generation

The proposed dataset is provided by TransPennine Express[1], which is a famous train operating company in Britain. This data source is only used for the research purpose, applied to cooperation activities conducted between the railway operating company and the University of Leeds Transport Research Institute. It was directly provided by the data management department of TPE and does not belong to open-source generic data under the scope of GDPR (General Data Protection Regulation). Therefore, in this case study, the Leeds team was only able to access the dataset, but was not enabled with ownership and sharing authorities. Its major business activities are providing regional and intercity passenger rail services between the cities of Northern England and Scotland. These services cover three regional routes around the Manchester area and major cities such as Glasgow, Liverpool, Leeds, and Newcastle are connected by the three main routes. The target network consists

[1] https://www.tpexpress.co.uk/

of 1348 train instances that operate in a medium-sized network with 177 stops/stations and 192 edges/links between these nodes. Figure 4.2 below shows the route map operated by TransPennine Express.

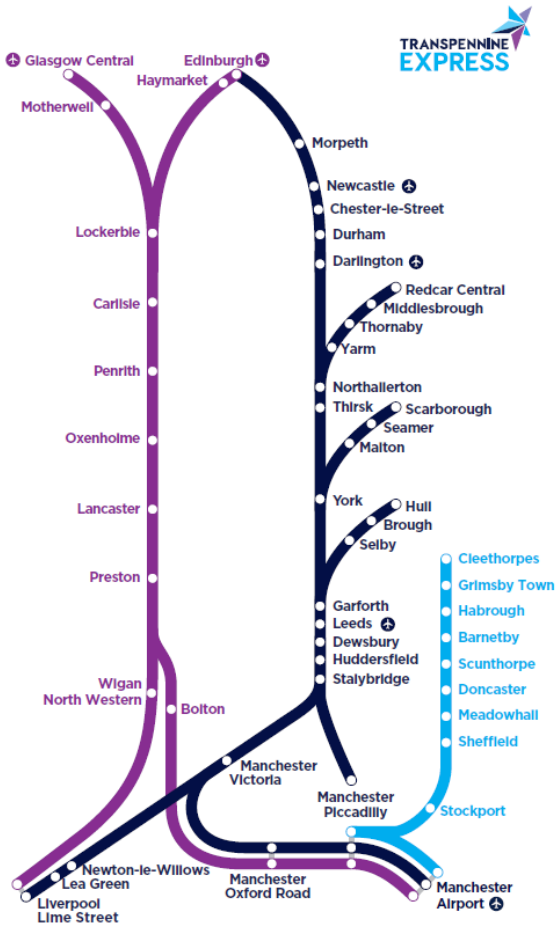


Fig. 4.2. Route map for the train services of TransPeninine Express (available from http://www.projectmapping.co.uk/Reviews/Resources/TPERouteMapDec2019nongeo.pdf)

Different colors represent different routes. The black one, which is “North TransPennine”, is the busiest one which passes through the core area of Manchester and Leeds. The purple one, we called “TransPennine North West”, mainly undertakes passenger traffic bound for Scotland. And the light blue one, “South TransPennine”, provides services to Sheffield and further south.

It’s a necessary and beneficial operation that conducting pre-processing during data generation. This phase is the most significant one compared to others due to the quality of its output directly determines how excellent the prediction accuracy will be and whether the experiment successful or not. Through the data analysis we identified several quality issues that may have negative impacts on improving prediction results. generally, the statistics provided by TPE are heavily imbalanced. Here we give an example - the electronic fraud dataset, in which fraudulent transactions are significantly lower than the normal healthy transactions. Similarly, the primary delay dataset has the same issue needs to be addressed: severe

delay (more than 10 minutes) samples account for around 10% of the total number of observations. However, as we mentioned in previous sections, the objective of this case study is to enhance identification capability on rare occurred and minority classes rather than just achieving greater overall accuracy on the test dataset. Additionally, some values in route stations such as the departure station are duplicated so we have to remove or refill the mistake field. There are other data quality issues such as inconsistent timestamp format and too many outliers exist in attributes such as ”passenger flow” and ”total margin time”. After the data pre-processing, we summarized the key attributions that have been used in the primary delay prediction paradigm in the table 4.1 below.

Table 4.1: All the features used in modelling

Feature Categories	Name of Features
Temporal Features	Date of Service; Weekday/Holiday; DepartureTime; Arrival Time
Numerical Features	Passenger Volumn; Total Margin; Speed Limit; Link Travel Time
Categorical Features	Rolling Stock Type; Train ID (headcode)
Label Feature	Primary Delay
Network Features	Origin Station; Terminal Station; Line-serving Stations

It is worth noting that a novel temporal feature engineering method is applied to temporal features (i.e., ”departure time” and ”arrival time”) since the ordinary pre-processing strategies for numerical characteristics may not work functionally on these features. For example, the train departure time and arrival time are two numerical variables where minimum time granularity is minute and the numbers count the temporal distance from the time point of 00:00 every midnight to the event time point. Some values of the temporal features are greater than 1440 due to these trains terminating in the early morning of the next day while departing from the previous day. An alternative processing strategy needs to be developed for addressing the encoding and interpreting effectiveness for such kind of cyclical temporal features. The core idea proposed here is from the mechanism of a popular mathematical tool - the Polar Coordinates system (PCS), which is a two-dimensional coordinate system where each point on a plane is determined by a distance from a reference point and an angle based on a reference direction. Such that the difficulties of featuring the temporal attributes can be properly accommodated. In figure 4.3 below, a circle with a radius unit of 1 has been created for mapping each cyclical variable into an evenly located point projection, such that 24 hours a day and 60 minutes an hour has been effectively projected to the corresponding position. That is the lowest value for that variable appears right next to the largest value. Then axis coordinates component on the x-axis and y-axis can be calculated by sin and cos trigonometric functions respectively. In summary, in the data generation of temporal features, we represented each minute/hour with its own polar coordinate components thus discrete variables can be converted to interpretable continuous numerical scalers.

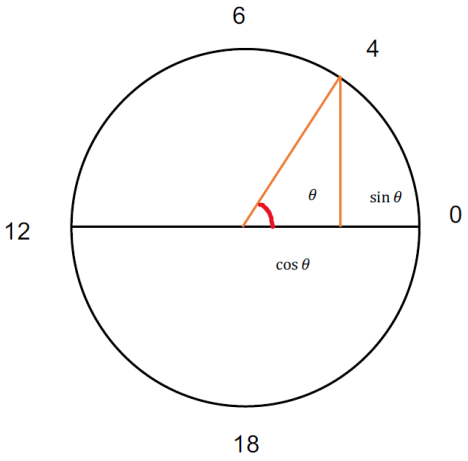


Fig. 4.3. Illustration for projecting cyclical variables into x-y axis components by PCS-based temporal feature engineering

As for the data preparation for numerical attributes, there was only one related feature in the original dataset: ”passenger flow”. After analysis we discovered that the attribute ”total margin time” can be treated as a continuous variable because it shows a distribution trend similar to Gaussian distribution. Generally speaking, our proposed machine learning models would benefit a lot from standardization, especially if some outliers are presented in the original dataset. Standardization has been commonly used at here for transforming the feature vector to a Gaussian distribution with zero mean and unit variance if this feature is less standard than normally distributed. After standardization, the negative impact of outliers on the update of hyperparameters will be offset. Correlation is a statistical measurement that indicates how many linear relationships two variables have. For example, we can say two variables are linearly dependent or non-linearly dependent. In figure 4.4 below, 0 represents the attribute “total margin time” and 1 represents “passenger flow”. While 2, the label feature, represents “primary delay level”. From the heatmap below We can conclude that “total margin time” is non-linearly dependent with both of two others features. Nevertheless, “passenger flow” and “primary delay level” are highly linearly dependent (The correlation Coefficient is around 0.5).

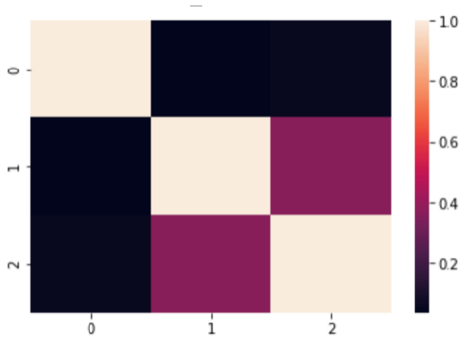


Fig. 4.4. Pearson correlation coefficients between numerical features

4.4. Experimental Design and Training

The experimental design was crafted to capture the specific variables that were described in the data preparation and generation section. For example, in the data description, it was noted that there was a high linearly dependency between the “passenger flow” and “primary delay level”. Therefore, the experimental design included controlled varying levels of passenger flow amount to test this relationship. Additionally, the data description mentioned if departure time falls into a particular time period that might be linked to delay susceptibility in trains. As a result, the experimental design included testing of the train instances to determine if there was a correlation between the presence of all the attributions and their susceptibility to the delay being studied. By integrating the insights gleaned from the data description into the experimental design, the study was able to more effectively test the hypotheses being explored.

4.4.1. Experimental Design

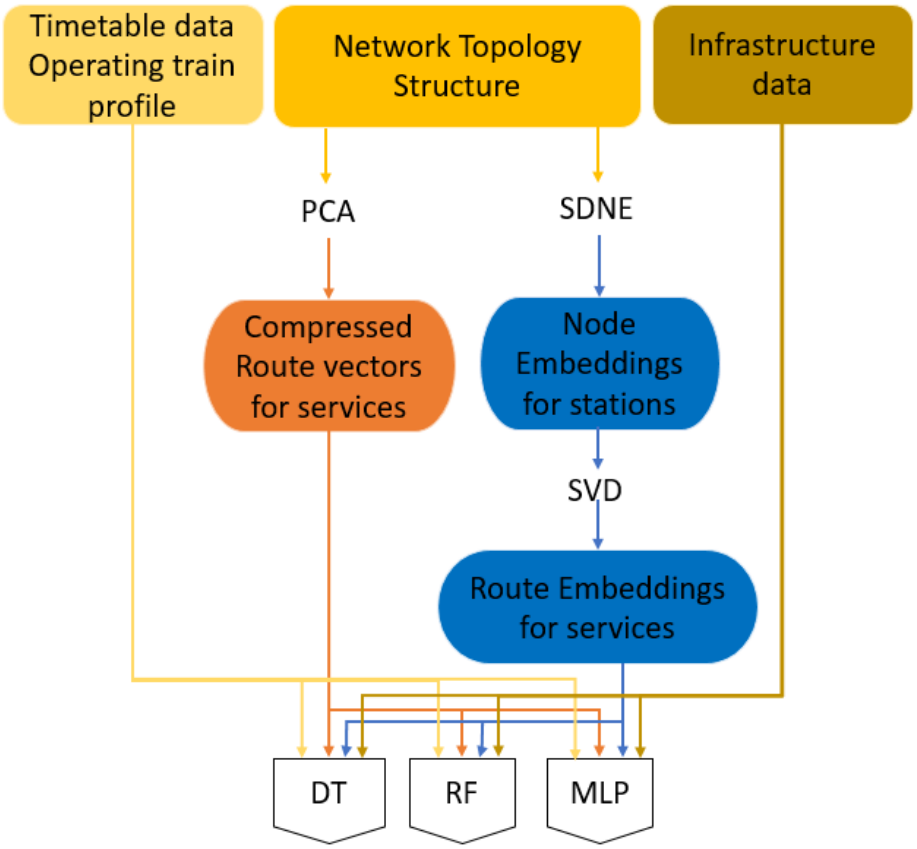


Fig. 4.5. Experimental design

The flowchart (shown in Figure 4.5) represents an experimental system for predicting delays in the railway network described in section 4.3. The designed experiment takes in data from a Network Topology Structure and splits it into two branches. The first branch uses the well-known Principle Component Analysis (PCA) algorithm [13] to compress the route vectors for services and then merges this data with timetable, operating train profile, and

infrastructure data. This merged data is fed into three different machine learning predictors: Decision Tree (DT) [14], Random Forest (RF) [15], and Multilayer Perceptron (MLP) [16] for train delay level prediction.

The second branch uses SDNE to generate node embeddings for stations, which are then processed by SVD to generate route embeddings for services. This branch also merges the timetable, operating train profile, and infrastructure data, and feeds the merged data into the same three machine learning predictors as the first branch. The reason why we choose these three benchmarks is that they are well-established algorithms that have been extensively tested and validated, and we want to obtain a standard of performance under the scope of defined research question and problem projectives in D4.2.

The machine learning predictors in both branches are used to predict delays for each train service. Overall, this system utilizes both PCA and SDNE to process data and combines the resulting data with timetable, operating train profile, and infrastructure data to make predictions using machine learning.

4.4.2. Training Process

The code implementation of the SDNE model was based on “Keras”[2], which is a well-known deep learning API written in Python, running on top of the machine learning platform ”TensorFlow”[3]. After applying the SDNE algorithm on the TransPennine network, we obtain the embedded node representation for each station. Where each row represents an embedding and each column gives information about the value of a specific position, Notably, if we consider each digit of the embedding vector individually there is no meaningful explanation that can be given however if all the values are compound together as a whole, the embedded information can be preserved effectively. We choose 8 here as the length of each embedding since $2^8 = 256 > 177$ (the number of stations in this network).

However, the SDNE model is only able to generate a node embedding vector for each station. We are also interested in further modelling the network topology from the route perspective as this will generate a more compressed vector for the subsequent prediction task. To this aim, Singular Value Decomposition (SVD) [17] – is introduced for generating route vectors. To the best of our knowledge, it is the first time to introduce such matrix decomposition technology into a train delay prediction procedure. SVD performs the matrix decomposition by extracting the most essential information (called ‘singular values’) in the original vector. By using SVD we are able to represent the original dataset by a much smaller dataset, such that the noise and redundant information are significantly reduced. From this perspective, SVD can be regarded as a process of extracting the most relevant features from a set of ordered node embeddings, see figure 4.6. Where we extracted the route information from the station vectors: 1191 routes representations corresponding to all available train service instances have been generated by orderly concatenating the en-route station embedding, and computing the sigma matrix for each of the routes, respectively. Based on this approach, the final route embedding for each service was generated accordingly. Here we simply attempt to predict delay times based on historic

[2] https://keras.io/
[3] https://www.tensorflow.org/

records which is also used for testing in the ML process.

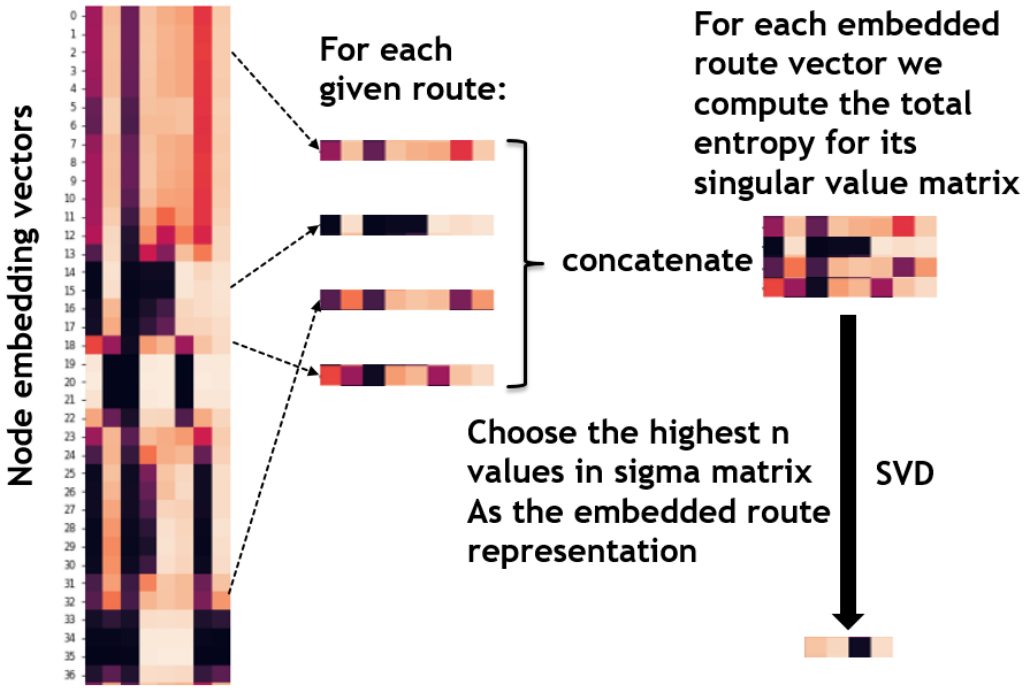


Fig. 4.6. Process of performing route embedding

The route-embedding method above was finally implemented within three classical supervised ML algorithms we introduced in the subsection 4.4.1 to predict primary train delays. In addition, we compare our Singular Value Decomposition (SVD) under the SDNE framework with the PCA strategy [13] - PCA is a technique used for extracting the main feature components and reduce dimensionalities. On the basis of these benchmarks, we intend to perform a five-fold cross-validation on the pre-processed dataset. Running the learning experiments on each fold and then inspect the overall prediction power. The confusion matrix is another tool that will be used to visualize the performance of classification algorithms. For each prediction task performed in DT, RF and MLP, all predictable cases/instances are intended to be classified into five categories:

no delay (0) There is no delay occurred on the specific train.

mild delay (1) Those delay minutes above 0 but under 6 minutes are mild delays.

moderate delay (2) Delays lasting between 6 minutes and 11 minutes are categorised as moderate delays.

serious delay (3) When a delay time falls into the gap of 11 minutes - 16 minutes, we classify it as serious delay.

severe delay (4) Those delays above 16 minutes are severe delays.

Such that we can properly convert the traditional delay distributional analysis problem to a ML-based delay level classification question. And it is easily to calculate the prediction accuracy for different predictors on various delay levels respectively. Currently the topological relations between each traffic elements have been captured and we plan to incorporate

more hidden correlations from the perspective of temporal and sequential-interactions. Our future work will focus on how to learn the correlations between primary delays and secondary delays and thus further estimate the occurrence of secondary delays. Table 4.2 highlights the most promising candidate strategy for each module. The column "Contributions" gives the information about what the main tasks/actions should be taken within this module. The description of the proposed solution including the applied AI components and suitable techniques we explained before.

Table 4.2: Suitable AI approaches for Primary Delay Prediction

Module	Contributions	Suitable Approaches
Node Embedding Module	Capturing structural information for TPE network	SDNE
Route Embedding Module	Concatenating the generated node embedding into meaningful route vectors	SVD
Delay Prediction Module	Predicting the overall delay level for each train service	Decision Tree Random Forest Multi-Layer Perceptron

4.5. Numerical Simulations

Route information from the station vectors: 1191 routes representations corresponding to all available train service instances departure/passby/dwell/terminate at stations, have been generated by orderly concatenating the en-route station embedding and computing the sigma matrix for each of them. In this chapter, the aims is to give some validation results for demonstrating the effectiveness of our SVD method in generating route embedding vectors. We select 4 different routes from the Network Features:

Route1: Newcastle - Liverpool Lime Street: Newcastle, Chester-le-Street, Durham, Darlington, Northallerton, York, Leeds, Huddersfield, Manchester Victoria, Liverpool Lime Street

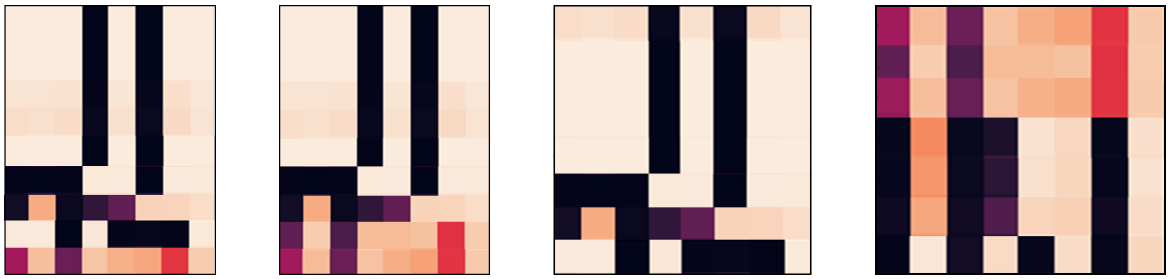
Route2: Newcastle - Manchester Airport: Newcastle, Chester-le-Street, Durham, Darlington, Northallerton, York, Leeds, Huddersfield, Manchester Piccadilly, Manchester Airport

Route3: Newcastle - Manchester Victoria: Newcastle, Durham, Darlington, Northallerton, York, Leeds, Huddersfield, Manchester Victoria

Route4: Manchester Airport - Doncaster: Manchester Airport, Manchester Piccadilly, Stockport, Dore Totley, sheffield, Meadowhall, Doncaster

One of the train services launched from Newcastle on the black line has its route information defined by the stop stations in Route 1 (see figure 4.2). Another service that travels from Newcastle to Manchester Airport is referred as route 2 above, which is largely similar to route 1. Despite having somewhat different last two calling stations, these two itineraries share the majority of the same dwell stations. Route 3 is a subset of route 1 and they share all of their calling stations, while route 3 skips some of the intermediate stops. At last, route 4 offers a service that is entirely distinct from route 1 and is located in a different region (i.e., light blue route in figure 4.2). For the convenience of the next-step processing, we sequentially concatenate the node embedding representations of those four routes' calling stations (i.e.,

departure station, dwell stations, and terminal stations) into four route vectors in figure 4.7 below.



(a) Route1 nodes representation

(b) Route2 nodes representation

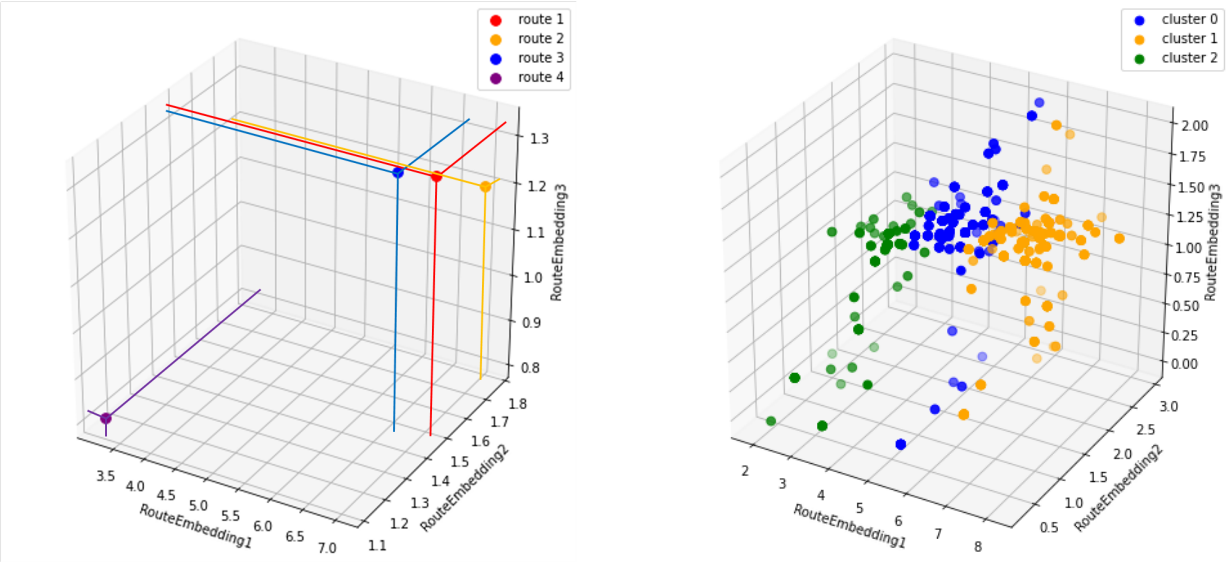
(c) Route3 nodes representation

(d) Route4 nodes representation

Fig. 4.7. Concatenated node embedding vectors for 4 different train services

In a machine learning context where embeddings are being used, the distance between two nodes' embeddings that have been projected far or close in Euclidean space indicates the similarity or dissimilarity between those nodes in the original high-dimensional space: If the distance between two nodes' embeddings in the projected space is small, then these nodes are likely to be similar in the original high-dimensional space. On the other hand, if the distance between two nodes' embeddings in the projected space is large, then these nodes are likely to be dissimilar in the original high-dimensional space.

The distance between two nodes' embeddings in the projected space can be used to measure the similarity or dissimilarity between the corresponding nodes in the original high-dimensional space. This can be used as a powerful validation tool for our proposed method on the considered dataset in many cases, such as clustering and classification. Projections of Route 1 to Route 4 in euclidean space are shown in Figure 4.8(a). As we can see, Routes 1 and 3 are projected very closely together in the embedding space, as can be seen. Route 2 comes next and exhibits a pattern that is similar to that of route 1 in the embedding space. It is also reasonably close to route 1 in Euclidean distance. When compared to the other routes, route 4 is embedded far away. The characteristic/proximity of the actual topology relationships in real-world topologies are largely preserved in the SVD-based route embedding representations, as is readily apparent. In addition, we used k-means methods [18] to cluster every route that was available, as shown in figure 4.8(b).The result shows us that all the route embedding vectors have been automatically clustered into 3 different classes and each of them highly correspond to the three-colored routes shown in figure 4.2.



(a) Projections of the 4 selected routes in embedding space

(b) Results of k-means clustering on all routes

Fig. 4.8. Concatenated node embedding vectors for 4 different train services

The three axes shown in figure 4.8 are the generated route embedding (1-3), which means there are three dimensions in each of the routes representation after applying SDNE + SVD strategy. There are many heuristic strategies for determining the number of dimensions to keep. In this study we keep 90% of the energy/entropy information from the original matrix, whose effectiveness has been successfully approved in the previous study [19]. The result turns out that the first three elements in sigma matrix already properly preserve over 90% of the essential information of the whole node embedding vector. Therefore, we set the route embedding dimensions as 3 in our experiments.

4.6. Discussion of Results

The output of the route-embedding implementation will feed into three classical supervised ML algorithms we introduced in section 4.4.1, for predicting primary train delays in this section; the three ML algorithms are DT, RF, and MLP. In addition, we compare our Singular Value Decomposition (SVD) under SDNE framework with the Principle Component Analysis (PCA) strategy [13]. To assess the effectiveness of the SDNE + SVD model, we use the cross-validation provided in (Table 4.3), confusion matrix (shown in figure 4.9).

5-fold cross-validation is a technique used to evaluate the performance of a machine learning model [20]. It involves randomly dividing our dataset into 5 equal parts (folds), training the model on 4 of the folds and testing it on the remaining fold, and then repeating this process for each fold, so that each fold is used as the test set once. We did not take factors such as seasonal effects into account as the dataset was rather limited in the time horizon. This aspect could be further considered in future work.

The results of 5-fold cross-validation (Shown in 4.3) can be explained in terms of the average performance metrics across the 5 folds. Here we use prediction precision as the

Table 4.3: Comparison between PCA and SDNE+SVD method: 5-fold cross validation results

Strategy	Algorithm	1-fold	2-fold	3-fold	4-fold	5-fold	Average Score	Standard Deviation
PCA	DT	0.7198	0.7564	0.7347	0.7113	0.7381	0.7321	0.0227
	RF	0.7773	0.8190	0.8084	0.7916	0.8239	0.8040	0.0174
	MLP	0.8138	0.8378	0.8393	0.8150	0.8511	0.8314	**0.0146**
SDNE + SVD	DT	0.7443	0.7537	0.7035	0.7249	0.7132	0.7279	0.0187
	RF	0.8362	0.8338	0.8138	0.8196	0.7941	0.8195	0.0152
	MLP	0.8436	0.8181	0.8421	0.8346	0.8313	0.8339	**0.0091**

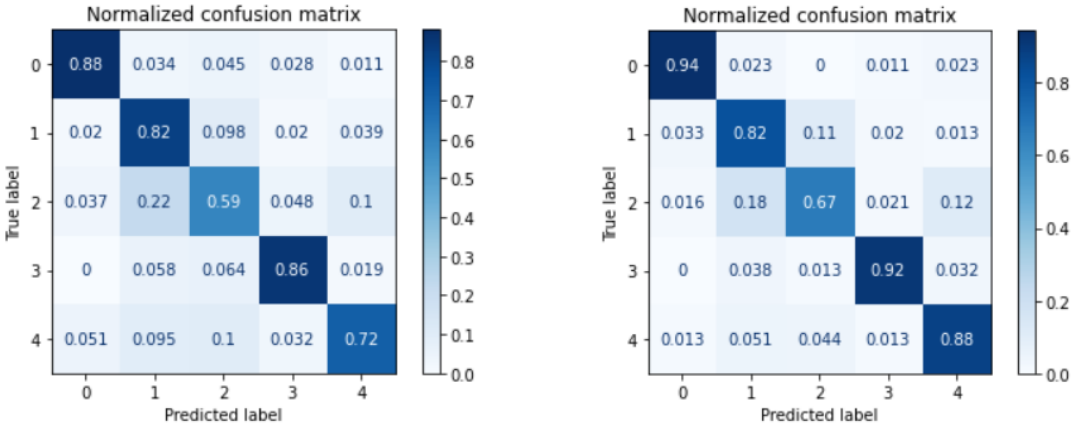
performance measure. These figures provide an estimate of how well the model (i.e., PCA and SDNE + SVD) will perform on new, unseen data. Additionally, the variance of the performance metrics across the 5 folds can be used to assess the stability of the model's performance. A high variance suggests that the model may be overfitting to the training data, while a low variance indicates that the model's performance is consistent across different subsets of the data.

Across all five sets of data samples, the MLP method had the highest overall prediction precision, followed by RF, while DT had the lowest average precision scores. Although the average scores of the two strategies, PCA and SDNE+SVD, were very similar across all three ML algorithms, the combination of SDNE and SVD demonstrated better generalization capability and superior expected fitting results on all three models. Specifically, the SDNE + SVD method showed promise in achieving more prominent performance on unseen data instances with less overfitting during training. Additionally, the standard deviations using the SDNE+SVD strategy were consistently lower than those with PCA strategy, suggesting that the prediction performances of our proposed method are more stable and reliable as the complexity of the prediction model increases, regardless of the dataset used.

Confusion matrix is another commonly used tool to visualize the performance of classification algorithms. The confusion matrices from our experiments are shown in Figure 4.9, where 4.9(a), 4.9(b) and 4.9(c) are the confusion matrices generated by predictor DT, RF and MLP, respectively.

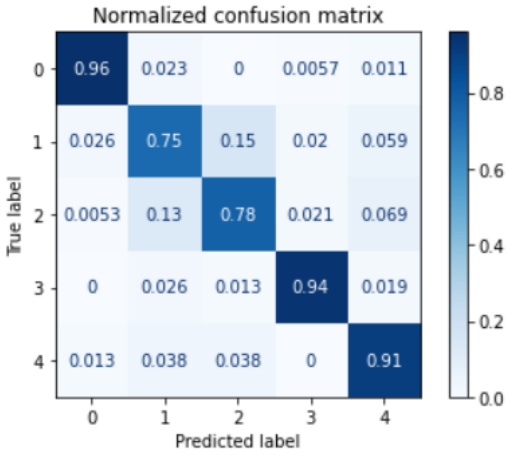
By examining the decision tree matrix in Figure 4.9(a), we can see that this classifier performed well in cases with no delay (0) and serious delay (3), achieving 88% and 86% accuracy, respectively. However, about 41% of the moderate delay samples were classified incorrectly as other categories. The confusion matrix of the random forest classifier in Figure 4.9(b) showed a significant improvement in accuracy across all delay categories, particularly in non-delay (0), serious (3), and severe delay (4) samples, achieving 94%, 92%, and 88% accuracy on the test dataset, respectively. The multi-layer perceptron classifier in Figure 4.9(c) further boosted prediction accuracy on each sub-category, achieving a higher level of accuracy (96% and 94%, respectively) in non-delay and serious delay cases. Nevertheless, although mild delay (1) and moderate delay (2) are the two categories with lowest prediction accuracy, MLP classifier seems tend to reach a balance between them, and managed to make the cases of mild delay (1) wrongly classified as the moderate delay (2) as less as

possible, and vice versa.



(a) Decision Tree Classifier

(b) Random Forest Classifier



(c) Multi-Layer Perceptron Classifier

Fig. 4.9. Confusion matrices on three different baselines

Table 4.4 shows that our method achieved an improvement in the average accuracy over all delay classes compared with the baselines performed on the dataset, reaching 86.80% on the best-performed model. In addition, the second row demonstrates that the overall training time of our method is significantly lower than that of competitive methods on every benchmark. Specifically, the model running time efficiency on the most accurate predictor (MLP) improved by 490% (as shown by the green line in Figure 4.10), which means the delay status of each unknown train service can be calculated promisingly within 0.24s, even if we migrate it to a brand new dataset and the model needs to be retrained. Figure 4.10 illustrates the significant decline in necessary computational efforts, indicating that our method has a prominent achievement in saving model training time and great potential for short-term, even real-time delay prediction.

Table 4.4: Comparison between PCA and SDNE + SVD method: Overall accuracy and Training time

Strategy	PCA			SDNE+SVD		
Algorithm	DT	RF	MLP	DT	RF	MLP
Overall Accuracy	76.68%	82.09%	83.89%	77.40%	84.59%	**86.80%**
Overall Training time (s)	374	1133	1417	96	235	**289**

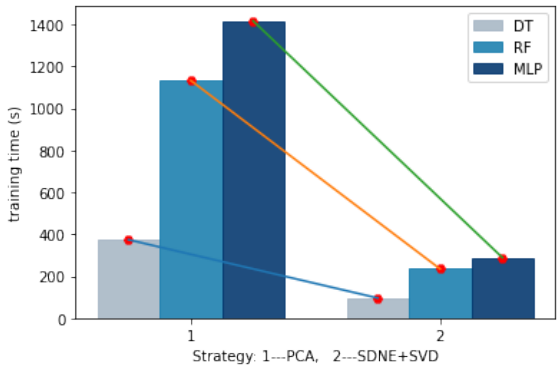


Fig. 4.10. Comparison between PCA and SDNE+SVD: training time by percentage

Finally, it is noticed that our work aligns with a subfield of AI known as “representation learning/self-supervised learning”. The proposed approach benefits from a broader selection of models within that subfield. However, the overall model appears capable of accurately predicting the primary delay. It would be intriguing to explore the generalizability of the learned model, preferably on a separate dataset.

5. Big Data on Incident Attribution Analysis

5.1. Introduction

According to the Delay Attribution Report by Rail Delivery Group to ORR (Office of Rail and Road)[1], the scoping stage report in July 2019 [2] pointed out 10 different recommendations regarding those areas that could be improved and need to be examined further to ensure the system performs better quality, higher reliability, more comprehensive understanding towards the input information. It discussed and allocated various responsible owners to take forward. Although there has been some progress in developing a number of the recommendations, very little work was carried out on the remainder. Therefore, in order to provide proper impetus to the current delay attribution analysis, as well as proactively respond to the Steering Group's promising view scope, we have decided to put this case study scope on furtherly automating the attribution process of cascading delays, with the aim of exploring the possible road map of applying AI techniques into this process.

Both the infrastructure provider and train operators need comprehensive insights to better understand how a delay at a certain location affects the wider network. Under this circumstance, the TRUST system (delay attribution data provider) only reviews trains that have been delayed by at least 3 minutes, however, those delays of less than 3 minutes are automatically attributed by the system to the railway company responsible and Network Rail responsible without further investigation into what caused the delays. On the one hand, the complex nonlinear spatial-temporal interactions between different train vehicles (e.g. timetable conflicts) and operators (e.g. track access rights) comprehensively determine how long the delay will last, and the range of its propagation in the network. They are difficult to be accurately predicted using the traditional approaches if we fail to preserve these relations and analyze/derive these delays from their original small disturbances. On the other hand, different railway disturbances and abnormal events are triggered by various determining factors. Some of them share the same root causes but others not. directly feeding all the observed relations/deterministic factors into the conventional statistical analysis function or descriptive model may not help us to figure out the correct delay propagation chain. Which may result in the computation space being too vast thus computation efficiency will be undermined. There are two major tasks we need to address in this study. How can we use Big Data techniques to interactively visualize historic train delay records to reproduce how these delays were triggered by small disturbances/disruptions/unexpected events is the first step we will perform. Second, by deriving and learning how these disturbances develop to observed primary delay and then propagate along a specific line/route of the network, we are able to generate meaningful prediction insights of whether a delay will occur or propagate between particular locations, timepoints, and train services. We aim to train a link prediction model (i.e., given two nodes, predict whether a propagation link between these two nodes should exist or not) in this step. Big Data and Graph Neural Network techniques

[1] https://www.orr.gov.uk/sites/default/files/2020-09/rdg-delay-attribution-review-report-2020-09-28.pdf
[2] https://www.orr.gov.uk/sites/default/files/2021-06/delay-attribution-review-scoping-stage-report.pdf

are introduced during interactive delay attributing visualization and potential propagation link prediction sub-tasks, respectively.

5.2. Model Description

5.2.1. Motivation and relevant works

This case study is developed with the purpose of seeking industry-corporate agreement on the best way to accommodate the unaddressed recommendation, considering a proposal for greater automation of the secondary delay attribution process. Regarding this, Network Rail and several operators recommended that some elements of the attribution of secondary delay can be replaced by 'hard-coded rules', rather than on the previous 'case-by-case' basis. This proposal would improve the consistency of the attribution of secondary delay and, by implementing a more intelligent hard-coded rule-based attribution process, would reduce the scope for potential disputes. For example, [21] presents a machine learning-based framework for predicting key performance indicators related to secondary delays in British railways with greater accuracy than existing systems. [22] proposes a novel model for studying train delays, which simulates delay propagation through a diffusion-like process, and applies it to the Belgian railway system, finding that spatial aggregation significantly increases the model's performance and showing the potential of this type of modelling to understand large-scale properties of railway systems. Recent progress on this was developed by the Rail Safety and Standards Board (RSSB) [23], where an approach of automating and visualizing the attribution of secondary delay has been proposed. The contributions of that study can be summarised as two separate parts:

- To investigate the causes and effects of secondary delay, a set of modeling tools for railway performance on a route basis was developed, and it will be shown how train operating firms and Network Rail can use them.
- Explain how the tools can be used to test a variety of performance-improving interventions and anticipate the level of service performance that can be attained through each intervention.

The model proposed by RSSB [23] pays much attention on understanding secondary delay in trains using a set of Monte-Carlo style Agent-Based Model runs. The visualizations help identify the locations, types, and mechanisms of primary and secondary delay and assist in designing interventions. However, it failed to consider the whole problem from the perspective of railway network and how the incidents triggered the initial delays.

In our case study, we got more interest on analyzing the delay attribution data, due to it is possible to predict the propagation of delays and the occurrence of secondary delays. For example, if a delay event is caused by infrastructure issues, it may result in train bunching, which can cause delays to other trains on the same route. By analyzing historical delay data, it is possible to identify these patterns and predict the likelihood of secondary delays occurring in the future. This information can then be used to develop strategies to minimize the impact of delays on train services.

5.2.2. Proposed Delay Attribution framework

Understanding the root causes of performance issues is not that easy, due to the fact that railway system has complex interactions and dependencies between individual components

(i.e., passengers, trains, staff, stations, timetables, junction, weather). Secondly, the propagation of delays is sensitive to small variations in inputs that can cause an escalating chain of events, such as, cascading delays across the network. In addition, an observed delay can be affected or determined by rare combinations of events.

Our proposed tools consist of a set of interactive visualizations to explore the complex interactions between modeled train services and events. Based on this, a GraphSAGE-based model has been developed to estimate the potential primary/secondary delay resulting from the existing incidents/train service event across the network of TPE routes [3]. In addition, a pilot intervention simulation will be performed with several supervised machine learning techniques, with the purpose of improving overall service quality. (See Fig 5.1)

Fig. 5.1. High-Level Architecture for Big Data on Incident Attribution Analysis.

3-D interactive visualizations This module aims to simulate how the sequential chain reaction is triggered between different incidents and trains, as well as between trains themselves, in an informative space of a hybrid spatial-temporal scale. We will inspect the evolutionary process of how a ”significant” delay develops from small disturbances to an observable primary delay and then secondary delays, in a more intuitive and clear way. Consequently, how these delays subsequently affect the punctuality of other train services. Multiple essential information will be effectively illustrated such as the length of delay minutes, scale of incident/delay, the cause of the incident, triggering relationships between delays, and the significance of dependencies between services.

With the use of such informative visualization, over thousands of statistical values, such as places, trains or times, can be easily displayed and understood. Interactions allow the user to find out more information or compute on-demand statistics for particularly interesting scenarios. Our visual summaries not only provide the insights of problematic train services and locations, but also enable users to delve further into the information to comprehend the causes of these delays and aid in the planning of intervention policies.

Intervention simulation Once the potential reasons for the reactive delay have been determined, interventions that aim to shorten these delays might be suggested. The

[3] https://www.tpexpress.co.uk/

modeling and visualization tools can then be used to recreate the sequential occurrence process of events with a set of input data that describes what the interventions are intended to accomplish in order to validate the efficacy of these interventions, such as reducing or preventing the causes of significant delays that the railway stakeholders might be interested in resolving. For example, reducing the number of track-based primary incidents, or reducing a range of incident durations.

GraphSAGE-based model In this module, our task is to learn if an edge exists between a provided node (service) and the existing nodes (services) we represented in the first module. In other words, exploring the possible responsible train and the potential reacted train for a newly introduced train service in the network that is characterized by an analysis of Network Rail attributed delay data. We use our implementation of the GraphSAGE algorithm [24] to build a model that predicts propagation links in our proposed TPE-Network Rail hybrid dataset, see the description in section 5.3.1. This problem is treated as a supervised link prediction problem on a heterogeneous delay propagation network with nodes representing incident and delay cases for train services.

This case study lays the groundwork of modelling and visualization of multiple services with various performance information (such as delay scale and impact of delays), over longer periods of time.

5.3. Dataset Generation

5.3.1. Data Sources

Steering group We held regular group discussions with the WP4 partners and the project steering manager. These discussions initiated potential data resources, innovative assumptions and experimental designs. The valuable opportunities to test our proposed framework, the emerging finding, and guide our obtained resources to the most valuable developments.

Data providers We've searched online and the result turns out that no suitable pre-built datasets were exactly qualified for the investigation purpose. Therefore, we leveraged some existing data repositories provided by our cooperating railway undertaking, see the section 4.3 for more details. Apart from this, we also collected a number of corresponding historic delay attribution data (including incident/delay locations, type of incidents and affected train service group) from the open data feeds provided by Network Rail [4].

TPE provided historical operation records including the possible factors that affect delays (i.e., passenger loading volume, total margin time, service speed limit, rolling stock type), the real-time train timing (i.e. date of service, departure/arrival time), train route information (i.e. origin/terminal station, line-serving stations), and delay data.

[4] https://www.networkrail.co.uk/who-we-are/transparency-and-ethics/transparency/open-data-feeds/

Expert opinion TUDelft provided expert opinion, particularly on identifying type of delays caused, explaning sub-threshold delay and its corresponding processing strategy, understanding and distinguishing the concepts of disturbances, incidents, and disruptions in the dataset.

5.3.2. Data Explanation

The task requires the corresponding Train Delay Attributor (TDA) at the relevant Network Rail route to give its judgment on whether a train service is committed with delay or not. Please note that the whole dataset contains primary delay cases and reactionary delay records, including the chain reaction of knock-on delays - there is not necessarily one layer between the occurred incident and the knock-on delay, several hidden layers can be found. In other words, in this case, only incidents directly result in primary delays, while secondary delays can be triggered either by a primary delay or another secondary delay. There are two different categories regarding how a disturbance triggers delays and how its negative consequences cascade along the network:

- causing secondary delay to another train or a set of services in the network (the reaction sequence denotes as 'incident $\rightarrow$ primary delay $\rightarrow$ 1st layer knock-on delay $\rightarrow$ 2nd layer knock-on delay $\rightarrow$ ... $\rightarrow$ last layer knock-on delay)
- not causing any secondary delay to another service in the network (the reaction sequence shown as 'incident $\rightarrow$ primary delay')

In the first step, the 'TRUST DA' - Delay Attributor of Train Running Under System TOPS (Total Operating Processing System) [5] was used for delay attribution and using a feed from TRUST [6]. When there is an observed delay or another event that caused a service to experience an unplanned disruption, the TRUST DA identifies where delays have occurred and those above a certain threshold.

It is worth noting that although all delays on the network are captured and recorded by the TRUST system, they are not all attributed, investigated, or explained. For example, a Delay of 3 minutes or less deemed not to cause any reactionary delay of 3 minutes or more is not attributed. For this sub-threshold delay, no action is taken to understand the detailed cause. That is, the responsible parties for these **sub-threshold** delays have been automatically and equally split between the Network Rail and the specific Passenger Train Operating Company. According to the statistics of Network Rail [7], about 35% of all delay minutes on the network are sub-threshold. Secondly, there are instances where Network Rail is unable to investigate the causes of delays as they do not have sufficient resources to investigate all the delays in a timely manner under significant disruptions. Therefore, the number of **uninvestigated** delays surges. In some extreme circumstances, Network Rail does investigate the delay, but no cause can be found by either of the parties involved, in which case the delay is classified as **unexplained**. Figure 5.2 shows the possible outcomes

[5] The 'TRUST (Train Running Under System TOPS)' is a system recording historical abnormal events of train movements when a train arrives at a location later than specified in the Plan of the Day, including scheduled and actual departure/arrival times, as well as cancellations or fail to run part of the journey.

[6] https://www.orr.gov.uk/sites/default/files/2021-06/delay-attribution-review-scoping-stage-report.pdf

[7] https://www.networkrail.co.uk/who-we-are/transparency-and-ethics/transparency/open-data-feeds/

of the delay attribution process when a delay occurs on the network and captured by the TRUST system.

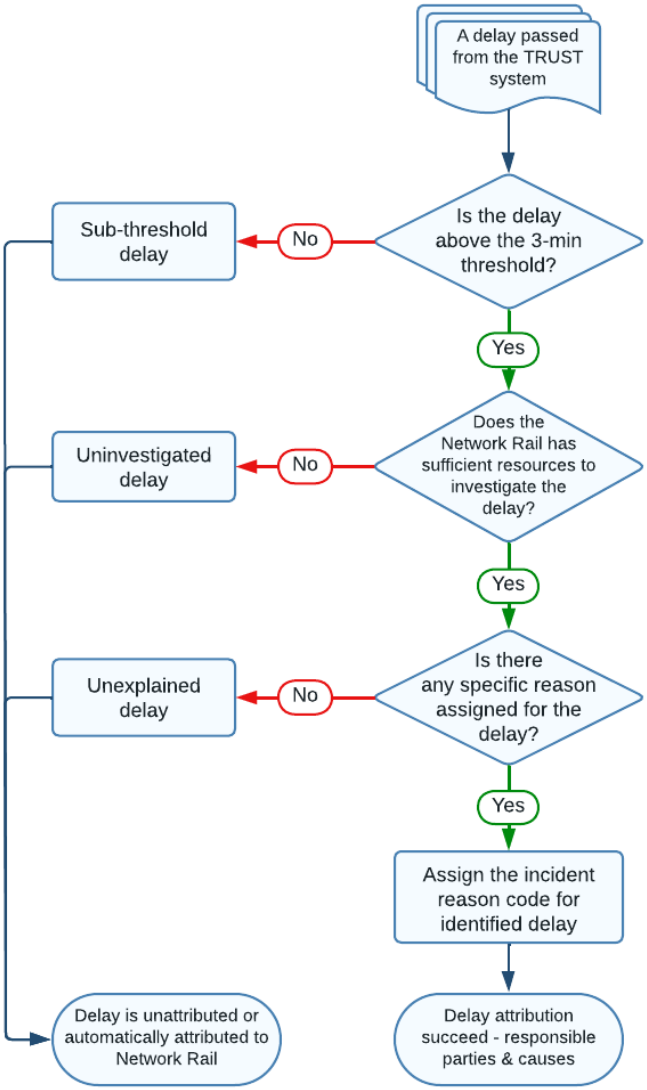


Fig. 5.2. Illustration of Unattributed, Uninvestigated and Unexplained delays in TRUST

At the last step, if the received record is deemed to be linked to an existing delay, the delay is attributed to an existing incident as a secondary delay with its secondary delay code. Alternatively, if it is thought to be an operator-caused delay, the Network Rail DA (Delay Attributor) will assign a proper incident reason code to it and pass it to the relevant operator through the TRUST system.

5.3.3. Data Preparation

We have explored graph networks to explain the occurrence and propagation patterns in train delay management. One of the main challenging tasks that deals with graphs is how

to properly visualize the network, the node properties, and edge dependencies in a neat, visually appealing way. Normally the default plotting options are built on Matplotlib [8], which is acceptable however it is sub-optimal for the interactive visualization we proposed in the last section.

3-D interactive visualizations In the first module, instead of adapting the Matplotlib axis API [9], manually modifying the graph, or creating a one-off function, something more robust, easier to use straight out of the box, aesthetically beautiful, and neat. Which is more acceptable by those who work in railway disruption management team or delay attribution institute. The straightforward output that being displayed depending on the real-world network is significantly important to those who has expertise in train dispatching but little experiences on data visualization.

This module produces delay results for considering train services in a detailed manner, at several levels of information aggregation: individual incidents, zero-loading testing trains, passenger train services, which train initially caused the incident (if any), who is responsible for the delay (industry/operating or other possible responsible parties), affected railway business route and service groups, the cause code for the incident, where the incident/propagation took place, time stamps, etc. The most important aspect of these visualizations is that they take into account all railway service interactions, not only the explainable ones that are attributed to delays longer than three minutes. As a result, a comprehensive picture of what causes reactionary delay is shown, capturing the mix of events and other factors that have influenced these modeled delays:

- any train conflicts that naturally emerge from the normal timetable scheduling or running, without any sign of incident triggering.
- train delays below the assigned threshold (those that are under three minutes and not attributed to a cause, however all the delays under 3 mins (2*1.5) has been equally split into the Network Rail responsible and the corresponding Train Undertaking responsible)
- trains delayed as a result of any existing incident with a known cause (delays over three minutes)

Intervention simulation This module is data driven, using the output of the interactive visualization as input to characterise any part of the railway system during any segment of the time period of a specific day. This means this module is capable to be fine-tuned for customizing any part of the UK railway network by changing the data input.

A brief data preparation procedure is needed to identify the rail network segment that will be modelled and to automatically retrieve the pertinent train services from the national timetable. Our original plan was for this procedure to be totally automated and to take advantage of any data already available on the visualized interaction network, notably ”3-D interactive visualizations”. After making an effort to accommodate the dataset that characterises the stations, depots, and lines and link the timetabled services (provided by TPE) with the Network Model, we were unable to locate any data defining the necessary relationships between them, in particular, the relationship between TIPLOC locations used in the timetable and the Network Rail Network Model.

[8] https://matplotlib.org/
[9] https://matplotlib.org/stable/api/axes_api.html

Automating the entire data preparation process is a crucial next step, but doing so would involve spending more time establishing the connections between the various data sources. Once this is accomplished, Network Rail and Train Operating Companies may be ready to utilise the uniform railway performance tools and quickly apply them to other areas of the UK rail network.

The final step in data preparation is defining the artificial incidents that could be incorporated in model runs, causing train services to be delayed and resulting in reactive delay.

GraphSAGE-based model This task required the annotators (Delay Attribution Board) to give their judgments on whether a train service is delayed or not. There are 3 steps to complete this task:

- **[Sub-task A]** In the first step, the Delay Attribution Board (DAB) marks the train service as being delayed or not delayed.

  In sub-task A, we are interested in the identification of those delayed services that are above the threshold of 3 minutes, fully investigated, well explained, and have not been canceled at any part of the serving segment. There are 2 categories in which the train service can be classified at the end of this task.

- **[Sub-task B]** If the train service is delayed then we need to figure out if the delay is targeted towards a particular train service (primary delay) or a bunch of subsequent relevant services (cascading reactionary delay), or it will not propagate.

  In this sub-task, we are interested in categorizing delays. The input only contains cases that were qualified assessed in task A. We need to label and distinguish each case from the following categories: (1) Incident triggered individual primary delay, and it ends there. (2) Incident triggered multiple primary delays but failed to initiate any subsequent impacts on other train services. (3) A primary delay triggered individual/a group of reactionary delay(s).

- **[Sub-task C]** If the delay is targeted as reactionary delay then we also need to tell which reactionary delay reason is applied.

  In this sub-task, we are interested in the specific category of secondary delays. Only train delays that are either derived from another primary delay or another secondary delay are included in this sub-task. Each instance needs to label from one of the following categories as shown in Table 5.1.

Table 5.1: Reactionary delay reason and corresponding explanation

Reason code	Explanation
YA	Lost path - regulated for train running less late
YB	Lost path - regulated for another later running train
YC	Lost path - following train running less late
YD	Lost path - following another later running train
YE	Waiting path onto/from single line
YF	Waiting for late running train off single line
YG	Regulated in accordance with Regulation Policy
YH	Late arrival of inward loco
YI	Late arrival of booked inward stock
YJ	Late arrival of booked inward train-crew
YK	Waiting connecting Freight or Res traffic to attach
YL	Waiting passenger connections within Connection Policy
YM	Special stop orders within the contingency plan or agreed by NR/TOC
YN	Service Recovery-booked train crew, not available
YO	Waiting platform/station congestion/platform change
YP	Delays due to diversions from booked route or line
YQ	Passenger overcrowding caused by a train being of short-formation
YR	Tactical cancellation for service recovery not caused by late running
YT	Reactionary delay by a train that is leaving the network
YU	Service Recovery-booked rolling stock, not available
YV	Tactical hold of train at origin or at a strategic location en-route
YX	Passenger overcrowding caused by delay or cancellation of another train

5.4. Training and Validation

This section discusses the Big-data visualization techniques and GraphSAGE-based algorithm we adopted to implement the modules depicted in Fig. 4.1. Notably, the module of 'Intervention simulation' does not involve any AI, therefore, its details are not elaborated in this section but the corresponding discussion is postponed to the remaining sections.

5.4.1. 3-D interactive visualizations

The available dataset after data preparation in section 5.4 consists of 20829 historic attributed delays from 27th May 2018 to 23th June 2018 (in line with the timeline provided by TPE delay dataset described in section 4.3, which was extracted from the Performance System Strategy (PSS) database. According to guidelines provided by the Delay Attribution Board (DAB), the information in the file includes any delays that have been "attributed" to passenger rail services. The tabulated series data includes important information such as what caused the incident, the train that directly caused the "target train" to be delayed, the train that caused the initial delay which set off a chain of delays, the reactionary reason why

the delay is propagated to the next train service, etc. Since there is no strong correlation between the train services in two different days - few trains are operating during the period of 2am to 5am, and an incident occurs during these time slots may not propagate to a wide range of the network, we manually divide the dataset on the daily basis. It is worth to note that the fundamental data feeds we utilized in the GraphSAGE-based delay attribution model is based on what we developed in this stage.

This module simulates a multitude of potential delay propagation outcomes that can occur on a given day. It is capable to explore the range of possible conflicts and subsequent reactionary delays that can cascade from a timetabled day of train services, no matter scoping on an individual train operating company or a part of collaborated railway networks that include a set of railway undertakings involved. Such detailed output data describe the many ways where/when/how train services can be delayed.

As an innovative advancement and the most integral component adopted in this case study. The results of interactive visualisation after numerous model runs include thousands of primary and reactionary delays. To validate, analyse, and act upon the model's illustrations, considerations must be taken into account from various angles and degrees of abstraction (i.e., by train service, location, time-slice, incident reasons, etc.). Although the model's high-level statistical summaries are informative, they exclude redundant and inexplicable information. Information visualisation is capable of displaying hundreds of statistical values for numerous locations, trains, or times in an understandable manner. For instance, a map of delays is easier to understand than a table of figures for each site. This is due to the fact that humans have an innate ability to discern values and create comparisons, for instance utilising circle size or colour lightness. The user can learn more information or generate statistics for certain scenarios of interest by interacting with visualisations. Figure 5.4 is an illustration of the visualisation on the day of 27th May 2018. For example, once we touch any of the leaf node (or intermediate node) in the interactive visualization, a hover text will pop up for giving the information about what this node represents (at what time which train service has been detected with how many mins of delays, and the responsible train which caused this delay). More than that, we are still working more costomized visualization - when a particular node has been selected, the visualization will highlight the whole delay propagation chain until its root causes to be found (normally they are the big-sized nodes shown at the core area of the figure). Such that the railway delay distributing staff can easily identify most observed delays with their root events and find a general pattern between the occurrence of severe delay and these abnormal events.

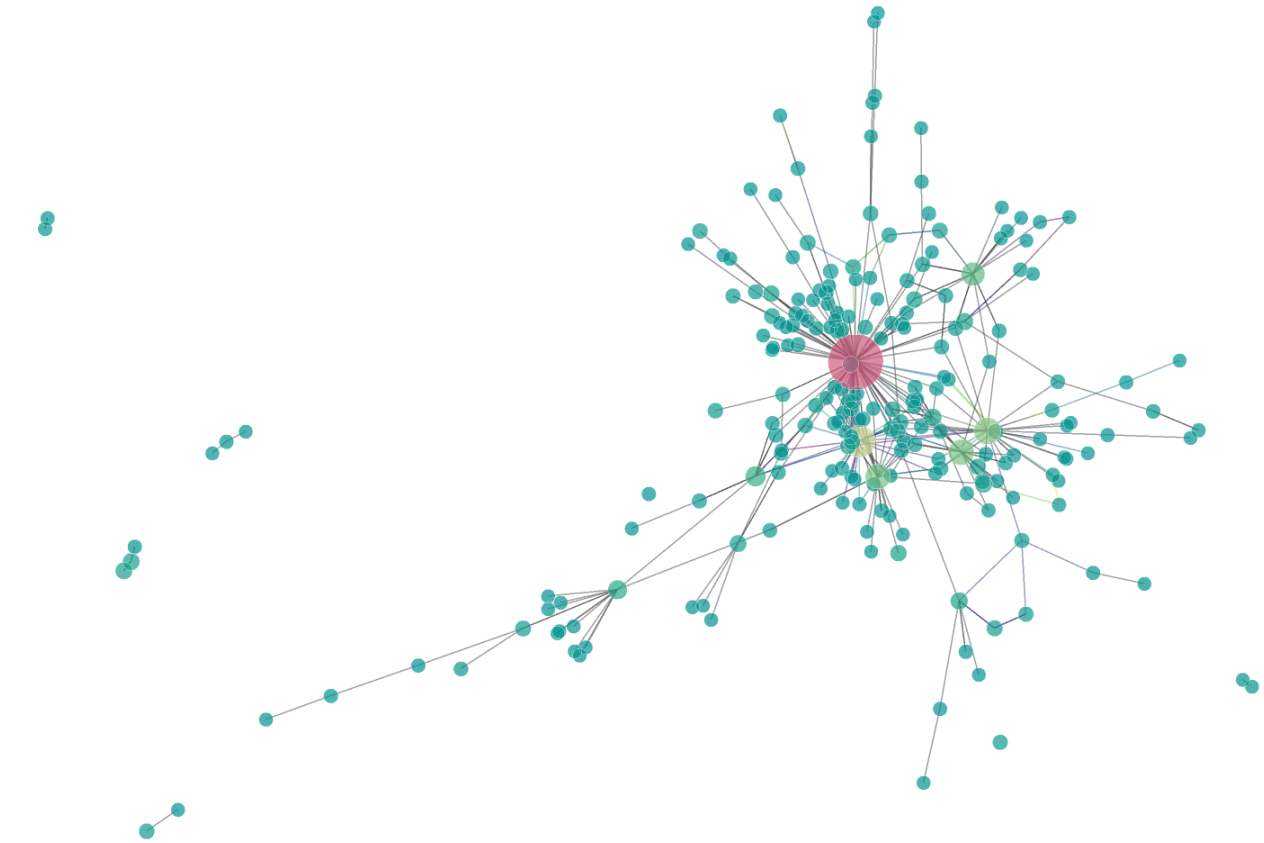


Fig. 5.3. Screenshot from the 3D interactive visualization

5.4.2. GraphSAGE-based model

To create a link prediction model in this module, we must first construct the train and test sets of connections as well as the matching graphs without those links. Using the EdgeSplitter class in the library of stellargraph.data [10], we will divide our input graph into a train and a test graph. The model (a binary classifier that, given two nodes, predicts whether a link between these two nodes should exist or not) will be trained using the train graph, and its performance on hold-out data will be assessed using the test graph. Each of these graphs will include the same number of nodes as the input graph, but their number of links will vary (and probably be lowered) since certain links will be eliminated after each split and utilized as positive samples for developing and testing the link prediction classifier, extract a selection of test edges (true and false citation linkages) at random from the original graph G, then create the reduced graph G test by removing the positive test edges.

Specifically, we build a model with the following architecture. First, we build a multi-layer GraphSAGE model that takes labeled node pairs (responsible-incident/train-delay $\rightarrow$ affected-train-delay) corresponding to possible propagation links, and outputs a pair of node embeddings for the pair of 'responsible-incident/train-delay' and 'affected-train-delay' nodes. These embeddings are then fed into a link classification layer, which first applies a binary operator to those node embeddings (e.g., concatenating them) to construct the embedding of the potential link. Thus obtained link embeddings are passed through the subsequent

[10] https://stellargraph.readthedocs.io/en/stable/README.html

dense link classification layer to obtain link predictions - probability for these candidate links to actually exist in the network. The entire model is trained end-to-end by minimizing the loss function of choice (e.g., binary cross-entropy between predicted link probabilities and true link labels, with true/false citation links having labels 1/0) using stochastic gradient descent (SGD) [25] updates of the model parameters, with mini-batches of 'training' links fed into the model.

The generators for the testing and training links are then built into the model. where they essentially "map" pairs of nodes (responsible-incident/train-delay $\rightarrow$ affected-train-delay) to the input of GraphSAGE: they take minibatches of node pairs, sample 2-hop subgraphs with head nodes extracted from those pairs (responsible-incident $\rightarrow$ affected-train-delay or responsible train-delay $\rightarrow$ affected-train-delay), and feed them, along with the corresponding binary labels indicating whether those pairs represent true or false propagation links, to the input layer of the GraphSAGE model, for SGD updates of the model parameters.

5.5. Findings and Discusssion

5.5.1. 3-D interactive visualizations

The proposed interactive visualization tool has generated the following interesting discoveries: Our 3D visualization tool identifies the locations that cause the most reactionary delayed segments of network, time slots in a day, and the certain categories of incidents. Specifically, it summarises the relative lengths of the primary delays by the primary delay types, and how these length varies between different times of the day. Also, it demonstrated the top incidents that triggered the most significant amount of primary delay and then secondary delays.

In this 3D interactive visualization diagram, the viewer can use various tools and controls to manipulate the position and orientation of the camera, zoom in or out, and interact with the objects in the scene. This allows for a more detailed and comprehensive exploration of the object or scene being represented and can provide a more intuitive understanding of its features and properties. Some common functions of it include: 1) Three-dimensional objects or scenes that can be rotated, panned, or zoomed in and out. 2) Realistic lighting and shading to give the objects a sense of depth and dimensionality. 3) The ability to select and highlight specific objects or parts of the scene. 4) Annotations or labels to provide additional information about the objects or features in the scene. 5) Interactive controls and tools for manipulating the view or performing specific actions on the objects or scene. Overall, this 3D interactive visualization diagram can provide a powerful tool for exploring complex data or designs in a more intuitive and immersive way, allowing users to gain a deeper understanding of the objects or scenes being represented.

For example, in the four figures shown below (figure 5.4), the interactive visualization gives the information on what a particular node in this visualized network represents and the color/size of such node also denotes the overall delay minutes caused by this specific service. In other words, the bigger the node is, the more severe the delays caused by this service are, and vice versa. when moving inspection perspective to a specific node '6C79', the viewer is able to identify how many subsequent cascading delays/events have been triggered by this train service. More important, it also gives the information about which train

service has been affected by this origin train service and what time the delay propagated along the network.

Fig. 5.4. An example of the 3D interactive visualization

We would also like to adapt this tool to work with the real-time attributed delay data (if there is any) and train movement data (not only on a service basis but also make the investigation scope into a more detailed level, which is capable to visualize the delay attribution results within the serving station basis), such that the mechanisms of delays propagation between services/stations in a short previous period can be explored.

It would take some effort to construct the uniform visualizations and interactions model described above, but Train Operating Companies (TOCs) who intend to build interactions with their own needs are likely to find them useful. That is, these graphs are hopefully to be input into other automated tools that provide practical information from it. It is worth noting that the practical tools used several recent studies such as time-distance diagram [26], as well as colored railway network maps [27]. However, what we aim to investigate in this interactive visualizations part concerning a different aspects: this study concentrates on how the referred disturbances/events trigger delays among train services, instead of inspecting the propagation affects from the railway network perspective. In other words, such visualization aiming to figure out the root cuase-effect relations rather than simulating the propagation process. Once the problematic areas and trains have been located, each model run can be investigated by gathering information on the types of delays that they were causing, the individual trains that initially resulted in the delay, and those that are picking up the subsequent reactionary delays. To determine which facts, visuals, and interactions will best complement

interaction design, we would like to work more intently and closely with a broader TOCs scope, for gaining more insights and then reflect them in our visualization.

5.5.2. Intervention simulation

An end user-friendly interface identifies the most problematic train services, locations and time slices, at which interventions or other following policies might be appropriate to be supplied. The variation between model runs indicates the range of possible consequences of delays. This allows the users to directly and swiftly identify the most problematic locations, where the registered incident events cause the most severe reactionary delays. The locations where the reactionary delays derived from are therefore to be identified and these locations might be worth trying to apply artificial interventions to simulate different effects of the reduce delay impacts, as well the propagation pattern of them.

Once possible interventions have been identified and coded, the model produces delay results for this intervention. For example, we characterise three different sets of interventions, and based on which explore to what extent service performance improvement can be achieved.

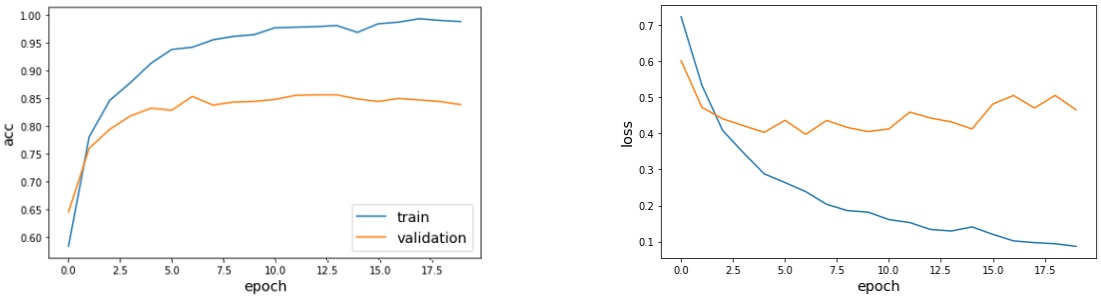
- **Intervention 1:** Reduce the number and duration time for all primary incidents, i.e., halve the probability of incidents occurring, halve the incident duration, and halve the delay duration/lasting time.
- **Intervention 2:** Reduce the number and duration for those primary incidents that only causing the most significant amounts of reactionary delay, which include - network management incidents, non-track incidents, track incidents, as well as off-network incidents.
- **Intervention 3:** Reduce the duration time for all incidents by 50%. This intervention was designed to demonstrate the effect of focusing efforts to significantly improve recovery times from incidents.

In practice, to reach this level of impact, a large amount of efforts would be needed. However, we sought to illustrate what a major performance improvement would resemble in a model and visualisation. Interactive visualization can be used to understand the impact of these interventions and the details of this by location, train, and model run. It is likely that the effect of the intervention will vary by model runs, so some interventions may have more consistent effects than others. Our visualization tool allows side-by-side direct comparison but does not currently directly encode the difference between model runs. We would like to develop tools to compare different interventions.

The second part generates delay effects for this intervention once potential interventions have been discovered and coded. The impact of these actions and the specifics of this by location, train, and model run can be understood via interactive visualization. It is likely that the intervention's impact will differ depending on the model run, meaning that certain interventions may have more predictable results than others. The difference between model runs is presently not directly encoded by our visualization tool, but it does allow side-by-side direct comparison. Tools can be created to evaluate the proposed approaches.

5.5.3. GraphSAGE-based model

This report presents the findings of our study on the use of a GraphSAGE-based link prediction model for predicting delay propagation in a railway system. Our study involved implementing the model on a dataset of railway timetable information, and evaluating its performance using standard evaluation metrics. The results of our study suggest that the GraphSAGE-based model is an effective tool for predicting delay propagation in railway systems.



(a) Prediction accuracy curves on training set and validation set

(b) Training loss curves on training set and validation set

Fig. 5.5. Training accuracy and loss on GraphSAGE-based model

The training accuracy and loss curve diagram shown in Figure 5.5. It is a common way to visualize the performance of a machine learning model during the training process. This diagram gives information about how the model's accuracy and loss change over time as it is trained on a set of input data. The horizontal axis of the diagram represents the number of training iterations, or epochs, and the vertical axis represents the accuracy or loss value. The accuracy is typically expressed as a percentage, while the loss is a measure of how well the model's predictions match the actual output values. During training, the model tries to minimize the loss function by adjusting its parameters to fit the training data as closely as possible. As a result, we expect to see the loss curve decrease over time, indicating that the model is getting better at predicting the correct outputs. At the same time, the accuracy curve should increase as the model becomes more accurate in its predictions. However, it is possible that the accuracy could plateau or even decrease if the model overfits the training data and starts to perform poorly on new, unseen data. Finally, it is noted that the testing loss remained relatively stable throughout the training process. This indicates that the initial guess was already fairly effective compared to the final trained model. It suggests the potential for further enhancements in terms of the model's generalization ability.

Overall, the training accuracy and loss curve diagram provides a helpful visual representation of the model's performance during training and can be used to diagnose issues such as overfitting, underfitting, or convergence problems.

6. Conclusions

In this deliverable, the potential of AI solutions towards a vision of traffic planning and management in the railway sector have been investigated through experimental proof-of concepts. These last represents the continuation of the methodological analyses provided in the previous deliverable for two selected case studies, namely, “Graph Embedding based Primary Delay Prediction”, and “Big Data on Incident Attribution Analysis”.

In accordance to the objectives, techniques, and research questions identified in the previous deliverable, for each case study, an innovative approach which exploits AI techniques has been provided. The effectiveness of the proposed strategies have been evaluated via an experimental validation in concrete operational scenarios. Results showed that AI can represent a valuable solution for enhancing rail traffic planning and management. These proof-of-concepts are meant to be a first step to inspire future developments and a technology roadmap. A detailed analysis of the results from each case study to identify opportunities, gaps, strengths, and weaknesses will be indeed the main object of the next deliverable.

Bibliography

[1] RAILS, "Deliverable 4.1 – WP4 Report on case studies and analysis of transferability from other sectors ," 2022. [Online]. Available: https://rails-project.eu/downloads/deliverables

[2] ——, "Deliverable 4.2 – WP4 Report on AI approaches and models," 2022. [Online]. Available: https://rails-project.eu/downloads/deliverables

[3] J. Preston, G. Wall, R. Batley, J. N. Ibáñez, and J. Shires, "Impact of delays on passenger train services: Evidence from great britain," *Transportation research record*, vol. 2117, no. 1, pp. 14–23, 2009.

[4] J. Weng, Y. Zheng, X. Yan, and Q. Meng, "Development of a subway operation incident delay model using accelerated failure time approaches," *Accident Analysis & Prevention*, vol. 73, pp. 12–19, 2014.

[5] R. M. Goverde, I. Hansen, G. Hooghiemstra, and H. Lopuhaa, "Delay distributions in railway stations," in *9th World Conference on Transport Research, Seoul, Korea, July 22-27, 2001*. WCTRS, 2001.

[6] I. A. Hansen, R. M. Goverde, and D. J. van der Meer, "Online train delay recognition and running time prediction," in *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2010, pp. 1783–1788.

[7] P. Wang and Q.-p. Zhang, "Train delay analysis and prediction based on big data fusion," *Transportation Safety and Environment*, vol. 1, no. 1, pp. 79–88, 2019.

[8] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 1225–1234.

[9] Y. Bengio *et al.*, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[10] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[12] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[13] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[14] J. R. Quinlan, "Simplifying decision trees," *International journal of man-machine studies*, vol. 27, no. 3, pp. 221–234, 1987.

[15] T. K. Ho, “Random decision forests,” in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1. IEEE, 1995, pp. 278–282.

[16] L. V. Shavinina, *The international handbook on innovation*. Elsevier, 2003.

[17] D. Yang, Z. Ma, and A. Buja, “A sparse svd method for high-dimensional data,” *arXiv preprint arXiv:1112.2433*, 2011.

[18] A. Likas, N. Vlassis, and J. J. Verbeek, “The global k-means clustering algorithm,” *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[19] M. Banerjee and N. R. Pal, “Feature selection with svd entropy: Some modification and extension,” *Information Sciences*, vol. 264, pp. 118–134, 2014.

[20] T.-T. Wong, “Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation,” *Pattern Recognition*, vol. 48, no. 9, pp. 2839–2846, 2015.

[21] P. Taleongpong, S. Hu, Z. Jiang, C. Wu, S. Popo-Ola, and K. Han, “Machine learning techniques to predict reactionary delays and other associated key performance indicators on british railway network,” *Journal of Intelligent Transportation Systems*, vol. 26, no. 3, pp. 311–329, 2022.

[22] M. M. Dekker, A. N. Medvedev, J. Rombouts, G. Siudem, and L. Tupikina, “Modelling railway delay propagation as diffusion-like spreading,” *EPJ Data Science*, vol. 11, no. 1, p. 44, 2022.

[23] A. Slingsby, J. Hyde, and C. Turkay, “Visual Analysis of Reactionary Train Delay from an Agent Based Model,” in *EuroVis 2019 - Posters*, J. Madeiras Pereira and R. G. Raidou, Eds. The Eurographics Association, 2019.

[24] W. L. Hamilton, R. Ying, and J. Leskovec, “Representation learning on graphs: Methods and applications,” *arXiv preprint arXiv:1709.05584*, 2017.

[25] L. Bottou, “Stochastic gradient descent tricks,” *Neural Networks: Tricks of the Trade: Second Edition*, pp. 421–436, 2012.

[26] W. Daamen, R. M. Goverde, and I. A. Hansen, “Non-discriminatory automatic registration of knock-on train delays,” *Networks and Spatial Economics*, vol. 9, pp. 47–61, 2009.

[27] R. M. Goverde, “A delay propagation algorithm for large-scale railway traffic networks,” *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 3, pp. 269–287, 2010, 11th IFAC Symposium: The Role of Control. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X10000124