Deliverable D 4.2

WP4 Report on AI approaches and models

Project acronym:	RAILS
Starting date:	01/12/2019
Duration (in months):	36
Call (part) identifier:	H2020-S2R-OC-IPX-01-2019
Grant agreement no:	881782
Due date of deliverable:	Month 31
Actual submission date:	September 20^{th} 2022
Responsible/Author:	[UNIVLEEDS]
Dissemination level:	Public
Status:	Issued

Reviewed: yes

Document history		
Revision	*Date*	*Description*
1	July 31th 2022	First issue for internal review
2	September 20th 2022	Second issue after internal review
3	October 27th 2022	Third issue after AB internal review
4	March 1st 2023	Fourth issue after JU review

Report contributors		
Name	Beneficiary Short Name	Details of contribution
Ruifan Tang	LEEDS	Contributor
Zhiyuan Lin	LEEDS	Coordinator
Ronghui Liu	LEEDS	Contributor
Rob Goverde	TUDELFT	Reviewer
Nikola Besinovic	TUDELFT	Reviewer

Advisory Board Reviewer(s)	
Name	*Company or Institution*
Dennis D. Huisman	Netherlands Railways (NS)

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

The information and views set out in this document are those of the author(s) and do not necessarily reflect the official opinion of Shift2Rail Joint Undertaking. The JU does not guarantee the accuracy of the data included in this document. Neither the JU nor any person acting on the JU's behalf may be held responsible for the use which may be made of the information contained therein.

Contents

Executive Summary

This Deliverable leverages the results achieved in Deliverable D4.1 to address AI approaches customised to the rail sector. On the basis of the pilot cases studies selected in Deliverable D4.1, i.e., primary delay prediction and incident attribute analysis. Specific proof-of-concepts are provided to investigate the adoption of AI methods for smarter railway traffic planning and management.

The report specifically discusses the following topics for each case study: i) a description of the background to identify the main objectives and the open issues; ii) the identification of appropriate research questions; iii) the effective dimensions/measures that can be used as evaluation criteria in rail transport planning and management; iv) the analysis of technology and methodological alternatives; v) a high-level description of the selected AI approach and its possible adaptation to railway; vi) the identification of the principal technological and architectural issues to be considered.

The suggested proof-of-concepts are intended to provide a first glimpse into the defining of qualitative and technological roadmaps that might lead to the implementation of AI applications designed to improve rail traffic planning and management.

Abbreviations and acronyms

Abbreviations / Acronyms	Description
TOC	Train Operating Company
SDNE	Structural Deep Network Embedding
PCA	Principle Component Analysis
DT	Decision Tree
RF	Random Forest
SVD	Singular Value Decomposition
NLP	Natural Language Processing
KNN	K-Nearest Neighbour Algorithm
BRS	British Railway System
PPM	Public Performance Measure

1. Background

This document is complemented as the Deliverable D4.2 “WP4 Report on AI approaches and models” of the Shift2Rail JU project “Roadmaps for AI integration in the Rail Sector” (RAILS) - under 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 (EURail) established by Council Regulation (EU) 2021/2085 of 19 November 2021. The RAILS workpackage WP4 investigates the adoption of machine learning techniques and other AI methods for enhanced railway planning and management. The present Deliverable describes the work carried out in task 4.2 (Development of AI approaches and models) whose aim is to develop AI approaches according to the results and choices reported in D4.1:

• Directions of transferability, considering some identified research lines a) Integrating heuristic searching strategies with deep neural networks for vehicle routing; b) Alternative routes services/navigation for passengers based on Cognitive Internet of Things; c) Attributing Primary and Secondary delays in Railway networks using Explainable AI;

• Pilot case studies, where three different potential case studies have been proposed and introduced as future research research directions while not intended to be all addressed within this project horizon. They are: Train Delay Prediction for each train service, Disruptions Attribution from a network perspective, and Rescheduling after delays.

The case studies mentioned in D4.1 were aiming at providing the context for proofs-of- concept. However, the scope of this deliverable is to prepare the ground for the actual experimentation, which will be performed in the continuation of the project in order to highlight the main challenges, technical obstacles, opportunities, and artificial intelligence/machine learning frameworks that applied to railway delay/disruption management applications. Those results will be used to define the future roadmaps for effective AI integration in the railway planning and management sector.

2. Objective

The main goals for the RAILS workpackages (WP2, WP3, and WP4) are to establish pilot case studies and provide proof of concepts that will eventually lead to a technical roadmap for an efficient adoption of AI in the rail industry. With regard to the RAILS project's Objectives 4 and 5, more specifically, the project activities implemented in this document and the following Deliverable D4.3 are:

• Development of both Experimental and Methodological proof-of-concepts.
• Development of Benchmarks, Models, and Simulations tools.

This deliverable formalizes the pilot case studies identified in Deliverable D4.1 ("Report on case studies and analysis of transferability from other sectors"), and provides feasibility studies for the adoption of AI and related techniques in the area of rail planning and management. Deliverable D4.1 contains a report on case studies and an analysis of transferability from other sectors. To examine the real-world requirements, obstacles, and opportunities while using artificial intelligence and machine learning to railway traffic management issues, we particularly chose two pertinent case studies. It is worth to mention that not all the candidate case studies we proposed is going to expanded in detail in this deliverable, due to the scope of WP4 has been refined in line with the actual proof-of-concept requirements of RAILS project. Considering the gaps of objectives and implementation feasibility between the first two and the third case study, we skipped the third one such that it will remain on the concept introduction phase. That is, according to the opinion from the project board and Leeds advisors, the WP4 should be conducted from the perspective Railway service performance evaluation and analysis rather than focusing on tactical planning stage.

The first case-study addresses railway primary delays prediction - Enhancing the dependability and standard of train passenger services is crucial, as is minimizing financial losses for both customers and service providers. Regarding strategies, reliable train delay prediction can offer railway dispatchers and infrastructure management helpful insights for rescheduling train services and allocating railway resources in advance, allowing passengers to reschedule their journey before any delays arise. To predict the occurrence of primary and secondary delays, conventional methods, such as mathematical and stochastic models, have been widely used. However, as the volume of available data grows, it becomes less effective to use extracted historical operational data and incident records in numerical analysis models to predict the status of a delay for a particular train service because the mechanism underlying train delay status is complex and its determining factors collectively contribute to the occurrence of train delays. We reviewed the prior research and tried to find a suitable approach to coherently aggregate them together for investigating the problem of train delay prediction from a novel perspective.

The second case study relates to incident distribution analysis and how these incidents triggered cascading delays. A simulation of incidents that occurred at different links or railway stations to predict the traits/performance measures of passenger services affected by incidents will be conducted. On a higher level, a generic prediction model incorporating related affecting factors will show the possibility of fitting the distributional probability into each disruption type per station and per time period. During this, we would discover the linear or non-linear relations between presumed disruption predictors and the occurrence likelihood of different types of accidents for each station. In recent years, the primary delay in rail services has remained consistent, but the reactive delay has continuously increased and proved challenging to comprehend and manage. The consequences of a reactive delay depend on how different trains interact with one another and the track's network properties. Multiple trains may be affected by cascading reactionary chains where one reactionary delay leads to other reactionary delays. It is challenging to forecast these impacts and cascading patterns using conventional methods. An alternate approach is a better way to comprehend and obtain insights into the root causes of reactive delay and to assess the likely effect that changes will have on the delivery of services.

3. Introduction

This document offers a thorough explanation of the AI solutions and methods for dealing with the concerns and difficulties brought on by the reference applications, the associated models and metrics, as well as the technical and practical obstacles. As a result, this deliverable provides information on potential technological and methodological solutions, alternatives, and important issues that will need to be addressed by later implementations.

The two primary chapters that deal with the concerns listed above for the two pilot case studies make up the bulk of this document: Chapter 4 is devoted to Graph-embedding based Train Delay Prediction, and Chapter 5 is concerned with ML-based Disruption Probability Estimation on Railway Stations.

The two chapters share the same structure: a brief presentation of the scope and challenges of the selected case study (Sections 4.1 and 5.1), the specific objectives of related the proof-of-concept (Sections 4.3 and 5.3), a description of the proposed approach and exploitable tools (Sections 4.4 and 5.4), datasets (Sections 4.5 and 5.6), a presentation of the AI techniques being exploited in the development of the proof-of-concept (Sections 4.6 and 5.5), and a discussion about expected results and possible criticalities (Sections 4.7 and 5.7).

The implementation and experimentation activities are ongoing according to the objectives, approaches and techniques reported in this document. Details and results of these activities are the object of the next Deliverable D4.3 (“Report on experimentation, analysis and discussion of results”).

4. Primary Delay Prediction

4.1. Introduction

Delay prediction in the railway sector, according to [1], is a process of estimating probabilities that a train fail to arrive at a subsequent check point (i.e., stations or junctions) on the time shown on a pre-defined timetable, and we typically define the differences between scheduled arrival time and actual arrival time as 'arrival delays'.

Many researchers have looked into the various factors that influence railway punctuality. Delays can be caused by a number of different factors. Unanticipated disruptions, such as power and signal system failures, train malfunctions, and inclement weather, cause severe delays, while predictable restrictions, such as speed restrictions, overrunning engineering works, or crew shortages, usually cause minor delays, according to a comprehensive investigation by [2]. Apart from them, Smaller events that occur more frequently referred to as "incidents" include problems that the railway undertakings could not have anticipated, such as specific passenger actions or damage to trains by motor vehicles, or the forced closure of a station. Current research commonly use statistical distribution analysis and regression models to examine the distributional elements of railway delays, as well as the link between delays and their impacting factors. For example, [3] looked into the statistical distribution pattern for Chinese high-speed railway train delays by first assessing and analysing the reasons for delays, as well as the types of disturbances, and also their position and timing.

[4] suggested a delay prediction model for the Hong Kong subway system, which evaluates the interaction between external infrastructure defects and prospective congestions, to investigate the correlations between delays and various types of interference variables. [5] has begun a detailed investigation of the probable elements that influence the punctuality of a specific train on weekdays over the course of a month. The following four factors were discovered: With multiple regression coefficient estimation, station, train number, weekday, and season could be utilised to explain around 15% of the incidences of delays. [6] 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. [7] investigated the harmful effects of difficult winter weather conditions on rail operation systems in Norway, Sweden, Switzerland, and Poland throughout 2010, and applied their findings to a novel example involving train delays in Finland. Based on changes in met-states on the Finnish network as well as weather-related infrastructure damage, they assessed the likelihood of train delays.

4.2. Background and Description

Primary and secondary delays are the two types of train delays that are regularly seen in real-world circumstances. Primary delays are produced by an incident/disturbance on the tracks, and recorded as the differences between scheduled and actual timetable. One of its key characteristics is that they are only important when unexpected events occur, regardless of whether the train was running on time [8]. Primary delay is the most common type in a normal railway system in terms of quantity, and it has gotten a lot of attention in terms of delay management from both train operators and passengers.

Primary delays could, in some situations, propagate from the current place to adjacent delay check locations or even a network segment. As a result, the secondary delay, also known as knock-on delay, is created. They are usually triggered by primary delay rather than by the timetable. Delay propagation is an important factor to consider when it comes to rail service quality. The degree of robustness of the timetable and the reliability of train operations is reflected in the cascading of delays [8]. When one train comes to a halt, for example, subsequent trains must wait, causing a domino effect until the initial train resumes service. [9] distinguish between these two types of delays by looking into their causes. Primary delays, as previously stated, refer to the positive difference between the actual and planned departure/arrival time. It is caused by an occurrence, and a robust timetable can absorb most of these deviations to prevent them from spreading by setting margin time between stations and trains, but a few of them can spread randomly around the rail network and cause secondary delays if traffic is heavy.

Current applications for analyzing/simulating the incidence of knock-on delays include two key use-cases. The first is real-time operational prediction, which involves forecasting delays in order to improve TOC capabilities and compensate for unexpected delays. This type of software can provide additional information regarding probable travel delays. For example, a neural network-based delay prediction system was built to assist dispatchers in improving real-time delay monitoring in a variety of real train networks in Germany [10]. Traffic simulation and analysis are another application that is widely utilised in strategic planning. For example, the results of a knock-on delay study can be converted into monetary values for risk-averse strategic investment advice. In particular, the latter way of analysing primary and secondary delay correlations in order to comprehend the insights between them and the delay propagation mechanism. While real-time delay prediction paradigms have made significant headway, there has been minimal effort to dissect delays into their fundamental and secondary causes.

According to what we previously discussed, many external interference elements might impair the punctuality of passenger services. Furthermore, traffic levels in different regions of the network are geographically intertwined and linked. Meanwhile, they have a lot of variability and complexity, which makes it more difficult to estimate train delays accurately. As previously stated, incidents and the propagation of primary delays are technically driven by complicated nonlinear spatial-temporal correlations or interactions between variables, which are inherently difficult to foresee in real-world circumstances. To support such a claim, the following points from the literature are summarised:

- Because the delay of one train at the station would affect the regular arrival or departure of following trains, producing the spread of train delays, the train's delay would have a considerable time-domain dependence.
- The spatial dependency between the station and the nearby railway resources is considerable due to their geographical location and spatial distance. That is, adjacent railway stations share infrastructure resources such as sectors, tracks, and waypoints, and the status of these resources at one station may affect the status of these re-

sources at the next station.

- A significant spatial correlation exists between railway stations connected by a single route/corridor. In other words, each of the pre-scheduled stations in a line will be served by a train. The traffic flow on a busy line looks to be higher than on a less frequent service line. This means that the stations along this line will have more opportunity to engage (accepting trains from upstream stations and sending trains to downstream stations).

4.3. Objectives and Research Questions

The purpose of this case study's delay prediction is to estimate the average primary delay level of a certain time in the future using data from various periods in the railway's operating history. Taking into account the static characteristics of each railway station in the network, as well as the structural network characteristics (i.e., connectivity between these stations, route weight, and network density for various areas), etc. Because each railway station's delay status is influenced by other stations and is constantly changing, it must be in close accordance with not only the station's own delay status, but also the impact of other stations—the closer the geographical distance between railway stations, the greater the impact.

The Structural Deep Network Embedding (SDNE) graph embedding algorithm was first created as an effective dimensionality reduction tool in the computer science field by [11]. A similarity network is constructed for a set of D-dimensional nodes based on their neighbourhood 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. 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.

  We identified three research questions to explore in this study based on the horizon we discussed:

RQ1: What graph-based method can be applied into the process of encoding structural characteristics and temporal-spatial dependency among network elements? Will the generated node embedding representations effectively preserve the essential information of network?

RQ2: How can we compute the route embedding vectors by using the node embedding representations? Will the results obtained by implementing route embedding outperform that of only using node embedding in this case?

RQ3: Is possible to improve the prediction accuracy on future primary delay estimation by incorporating the SDNE framework with some popular ML-based prediction models, e.g., Decision Tree, Random Forest, and Neural Networks? To what extent it will be boosted?

RQ4: What are the potential correlations between primary delay and secondary delay and what are the most relevant prerequisites and influencing factors that determine the occurrence of secondary delays (by considering station features and network structure simultaneously)?

4.4. Methodology and Tools

The acquisition of structural deep network representation is the most important aspect of the whole approach framework. To learn station representation in the railway network, a novel deep model named ”Structural Deep Network Embedding technique” would be proposed in order to capture the highly non-linear structure efficiently. This model is based on the recent successful implementation of deep learning techniques that were first derived from [11], which has been demonstrated by [12–15] to have powerful representation ability in dealing with various datatypes such as audio speech, image grids, and texts. However, there is no explicit proof that such an attempt on the public transportation network, particularly the train network, has been made. The goal of this module is to answer the RQ1 that we discussed in the previous section. The suggested SDNE framework is depicted in Figure 4.1 below. In this framework, characteristics of the original railway network as the input being fed into the Encoder-Decoder layers, whose detail structure will be introduced within the following paragraphs and equation 4.1 - including the identification of connectivity status between any of two nodes and the definition of 1st-order and 2nd-order proximity. Loss functions corresponding to each proximity in the output layer will convey the optimized parameters back to the Encoder-Decoder, for updating the generated embedding vector satisfying the minimum overall loss costs during training. And then the last-round low-dimensional embedding representation for each node in the network is obtained as the output of this framework.

As one of the gaps we identified in the D4.1, one of the expected outputs is to preserve both local and global structure effectively. In order to preserve essential structure information and sparsity as most as possible, we further give the definition of first-order and second-order proximity by referring the research statements from [16]. The primary intuition of applying SDNE to an existing network is to preserve the local network. That is, the local pairwise proximity between the vertices must be preserved. Therefore we introduce the first-order proximity as following:

First-order Proximity The first-order proximity describes the pairwise proximity between vertices. For any pair of vertices/stations u and v, if an edge is observed then the

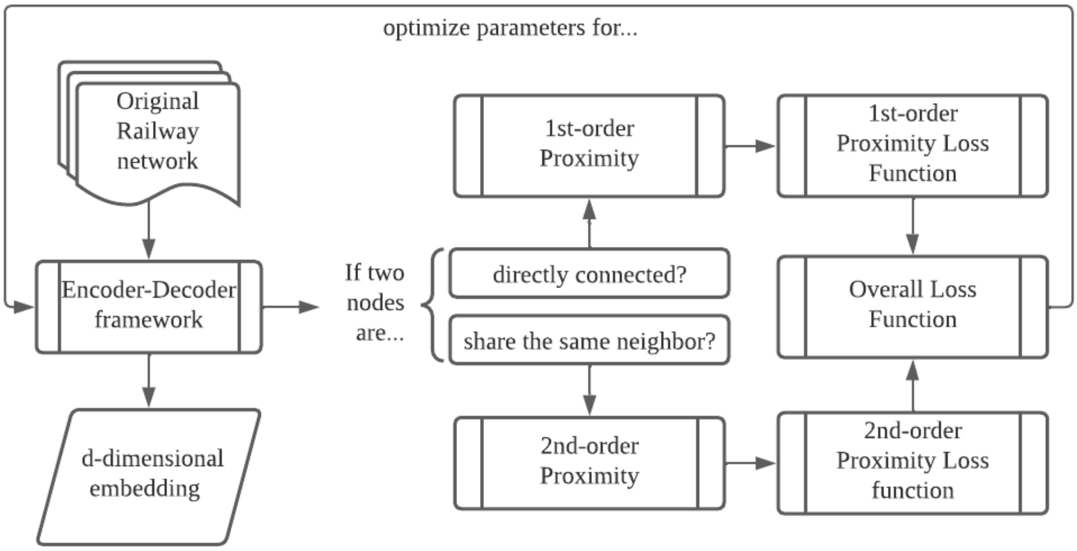

Fig. 4.1. The flow-charted SDNE framework

weight corresponding to that edge $w_{u,v}$ indicates the first-order proximity between u and v. Otherwise, their first-order proximity is 0.

This proximity implies that two nodes in real-world networks are highly similar if they are connected by a certain edge. For example, when one paper cites other papers, it means that they address similar topics. However, not all the vertexes are linked together – the real-life datasets are so sparse that observed edges only account for a small portion of all the combinations of vertexes. Therefore, capturing the first-order proximity alone is not sufficient to dig out all the substantial information in the graph and we then introduce second-order proximity as an alternative notion to explore the global network structure.

Second-order Proximity The second-order proximity between a pair of vertices u and v is the similarity describing their neighbourhood network structure. If we let $p_u = (w_{u,1}, w_{u,2}, ..., w_{u,k})\ (k \in V)$ denotes the first-order proximity of u with all the other remaining vertices in graph G. Then the second-order proximity between u and v is determined by the similarity between p_u and p_v.

A straightforward intuition is that if two vertexes share multiple common neighbours, they tend to be very similar even they are not directly connected to each other. For example, in social networks, people who share similar friends tend to have similar interests and characteristics thus tend to become friends. In another scenario, a certain word probably co-occur with the same sets of words that have similar meaning or shown as commonly used words collocation. This is the reason why we want to apply the second-order proximity to capture more relationships between nodes.

Before introducing the actual SDNE framework, we define a set of the terms and notations which would be used later. Note that symbol ˆ above the relevant parameters denotes the parameters of the decoder.

Notation	Explanation
n	Number of nodes in the network
p	Number of edges in the network
K	Number of layers in the model
V	The set of vertexes in the network, where $V = \{v_1, v_2, ..., v_n\}$
E	The set of edges in the network, where $E = \{e_1, e_2, ..., e_n\}$
$A_{u,v}$	The adjacency matrix for the network, where $u, v \in V$
x_d	The input data vector, where $d = \{1, 2, ..., n\}$
$\hat{x}_d$	The input data matrix, where $d = \{1, 2, ..., n\}$
X	The input data matrix, where $X = \{x_1, x_2, ..., x_n\}$
$\hat{X}$	The input data matrix for reconstruction layers, where $\hat{X} = \{\hat{x_1}, \hat{x_2}, ..., \hat{x_n},\}$
$Y^{(k)}$	The k-th layer hidden representations, where $Y^{(k)} = \left\{y_1^{(k)}, y_2^{(k)}, ..., y_n^{(k)},\right\}$
$W^{(k)}$	The k-th layer weight matrix
$\hat{W}^{(k)}$	The k-th layer weight matrix in decoder
$b^{(k)}$	The k-th layer biases
$\hat{b}^{(k)}$	The k-th layer biases in decoder
σ	The sigmoid function
L	Loss function
θ	The collection of parameters $\left\{W^{(k)}, \hat{W}^{(k)}, b^{(k)}, \hat{b}^{(k)}, \alpha, \gamma\right\}$

Given a network $G(V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is the set of railway stations in the network, and where $E = \{e_1, e_2, ..., e_n\}$ is the edge set composed of adjacent station nodes in the geographical space, that is, if there is an observed edge between station node 1 and station node 2, the second element of vector e_1 should be 1 (the first element shows the connectivity status of node 1 with itself). Therefore, we can obtain n adjacency matrices regarding each station node in the network $A_n = e_n$, which contains n instances $A_1, A_2, .., A_n$. For each elements $A_{u,v}$ in the instance A_n, if and only if there exists a link between node u and v. i.e. A_n provides the information of the neighbourhood structure of each vertex.

We next briefly review the key definition of deep auto-encoder. It is an unsupervised model which is composed of two basic components: the encoder and decoder. Given the input X, the hidden representations for each layer are shown as follow:

$$\begin{cases} y_i^{(1)} = \sigma\left(W^{(1)} x_i + b^{(1)}\right) \\ y_i^{(k)} = \sigma\left(W^{(k)} y_i^{(k-1)} + b^{(k)}\right) \end{cases} \tag{4.1}$$

The encoder consists of multiple non-linear functions that map the input data matrix x_i to the representation space and the decoder also contains some non-linear functions mapping the representation in embedding space to reconstruction space. It is worth to mention that we define the sigmoid function σ in our model as:

$$\sigma\left(W^{(k)}y_i^{(k-1)}+b^{(k)}\right)=\frac{1}{1+exp(-W^{(k)}y_i^{(k-1)}-b^{(k)})} \tag{4.2}$$

After obtaining $y_i^{(K)}$ we can then obtain the output $\hat{x}_i$ by reversing the calculation process of the encoder. The goal of the equation is to minimize the reconstruction error of the output and the input. The basic form of our proposed loss function can be defined as follows:

$$L=\sum_{i=1}^{n}\|\hat{x}_i-x_i\|_2^2 \tag{4.3}$$

Loss function for first-order proximity Besides preserving the global network structure, it is al so essential to capture the local structure charateristics. Next we would define the loss function for exploiting first-order proximity as below:

$$L_1=\sum_{u,v\in V}A_{u,v}\left\|y_i^{(K)}-y_j^{(K)}\right\|_2^2 \tag{4.4}$$

The loss function above would generate a penalty when similar vertexes are mapped far away in the embedding space. We incorporate this objective function to make sure the vertexes directly linked by an edge would be mapped not so far in the embedding space. Therefore our model can preserve the first-order proximity.

Loss function for second-order proximity If we exploit the adjacency matrix A as the input of our proposed auto-encoder, reconstruction process would make the vertexes with similar neighbourhood structures have similar latent representations. However, what we want to embed is not only the observed links but simultaneously some legitimate links that are not observed should be considered. That is to say, the connections between vertexes do indicate their similarity but no links between them do not necessarily imply their dissimilarity. Moreover, most of the real-world networks are so sparse that the number of non-zero elements in A is far less than that of zero elements. To address the problems above, we would impose more penalty to the reconstruction error of the non-zero elements than that of zero elements. So the revised loss function can be defined as below:

$$L_2=\sum_{i=1}^{n}\|(\hat{x}_i-x_i)*b_i\|_2^2=\left\|\left(\hat{X}-X\right)*B\right\|_F^2 \tag{4.5}$$

Where $*$ represents the operation of Hadamard product [17], if $A_{i,j}=0$ then $b_{i,j}=1$. Otherwise $b_{i,j}=\beta>1$. The Frobenius norm (F-norm), sometimes also called the Euclidean norm, is matrix norm of an matrix defined as the square root of the sum of the absolute squares of its elements [18]. Now by using the revised loss function for the second-order proximity, we can guarantee that the vertexes which have similar neighbourhood structure will be mapped near in the representation space, not only considering if they are connected or not.

The Objective function In order to preserve the first-order and second-order proximity simultaneously, we combine the equations together and joint minimizing the following objective function:

$$L_{mix} = L_2 + \alpha L_1 + \gamma L_0 = \left\| \left(\hat{X} - X \right) * B \right\|_F^2 + \alpha \sum_{u,v \in V}^{n} A_{u,v} \left\| y_u^K - y_v^K \right\|_2^2 + \gamma L_0 \tag{4.6}$$

Where L_0 is a L2-norm regularisation parameter used here for preventing overfitting, the definition of it can be given as below:

$$L_0 = \sum_{k=1}^{K} \left(\left\| W^{(k)} \right\|_F^2 + \left\| \hat{W}^{(k)} \right\|_F^2 \right) \tag{4.7}$$

So finally the objective function of the proposed model we want to minimize can be written as:

$$L_{mix} = \left\| \left(\hat{X} - X \right) * B \right\|_F^2 + \alpha \sum_{u,v \in V}^{n} A_{u,v} \left\| y_u^K - y_v^K \right\|_2^2 + \gamma \sum_{k=1}^{K} \left(\left\| W^{(k)} \right\|_F^2 + \left\| \hat{W}^{(k)} \right\|_F^2 \right) \tag{4.8}$$

The full algorithm for our proposed Structural Deep Network Embedding framework would be presented as figure 4.2:

Algorithm 2 Semi-supervised deep learning model of Structural Deep Network Embedding (SDNE)

Input: An input network $G(V, E)$, an adjacency matrix A, the hyper-parameters α and γ .
Output: An embedded network representation Y, the updated parameter θ

Assign the initialized parameters for each hidden layer: $\theta = \left\{ \theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(K)} \right\}$
$X = A$
for $k = 1,2, \ldots, K$ **do**
    Based on X and θ, apply equation (9.) to obtain $\hat{X}$ and $Y^{(K)}$
    $L_{mix}(X; \theta) = \left\| \left(\hat{X} - X \right) * B \right\|_F^2 + \alpha \sum_{u,v \in V}^{n} A_{u,v} \left\| y_u^K - y_v^K \right\|_2^2 + \gamma L_0$
    Based on equation (17.) and (18.), back-propagate through the entire network
    Then we can obtain the optimized parameters θ
Obtain the updated network representations $Y = Y^{(K)}$

Fig. 4.2. Algorithm description for the SDNE model

4.5. Artificial Intelligence and Machine Learning Models

We use the following five methods as the baselines. The first three are classical supervised machine-learning algorithms and we regard them as benchmarks for experimental analysis. PCA and the SDNE methods are two graph embedding strategies we want to make comparison regarding their capability on node embedding.

Decision Tree Classifier (DT)[19] is an algorithm that constructs a flowchart-like diagram for assisting the process of classification where specific feature of the dataset would be selected as the root node and each internal node represents a ‘test’ on an attribute.

Random Forest Classifier (RF)[20] is a classifier comprising a large number of individual decision trees, in which each decision tree generates a prediction of the label class and the predicted classification is determined by the major category output from the individual trees.

Multi-Layer Perceptron Classifier (MLP)[21] is a fully connected class of feedforward artificial neural network (ANN). A weight has been assigned to every artificial neuron for demonstrating the strength of each connection between this neuron next one. Each neuron will take the output of other neurons as input and conduct a mathematical computation to produce its own output.

Principle Component Analysis (PCA)[22] is defined as a technique for data analysis and pre-processing. It converts the original data into a set of linearly independent representations on each dimension linear transformation algorithm. It is commonly used to extract the main feature components of data vectors and thus reduce dimensionalities.

Structural Deep Network Embedding (SDNE)[11] The definition of this technique has been heavily introduced in the previous sections.

In this case study, we aim to perform the task of delay prediction (especially primary delays) for train services based on the historical operational data. For measuring the deviation between the predicted delay time and actual delay amount, we use the average cross validation score and standard deviation to evaluate the skill of proposed model on unseen data. It uses a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during training. In cross-validation, we partitioned our data set into a fixed number of folds (here is 5), run the analysis on each fold, and then averaged the overall error estimate.

4.6. Reference Datasets

The proposed dataset is provided by TransPennine Express[1], which is a famous train operating company in Britain. This data 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. Therefore, in this case study, the Leeds team only able to access to the dataset, but 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 Manchester area and major cities such as Glasgow, Liverpool, Leeds, and Newcastle are connected by the three main routes. The target network consists of 1348 train instances that operate in a medium-sized network with 177 stops/stations and 192 edges/links between these nodes. Figure 4.3 below shows the route map operated by TransPennine Express. Different colors represent different routes. The navy blue 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.

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

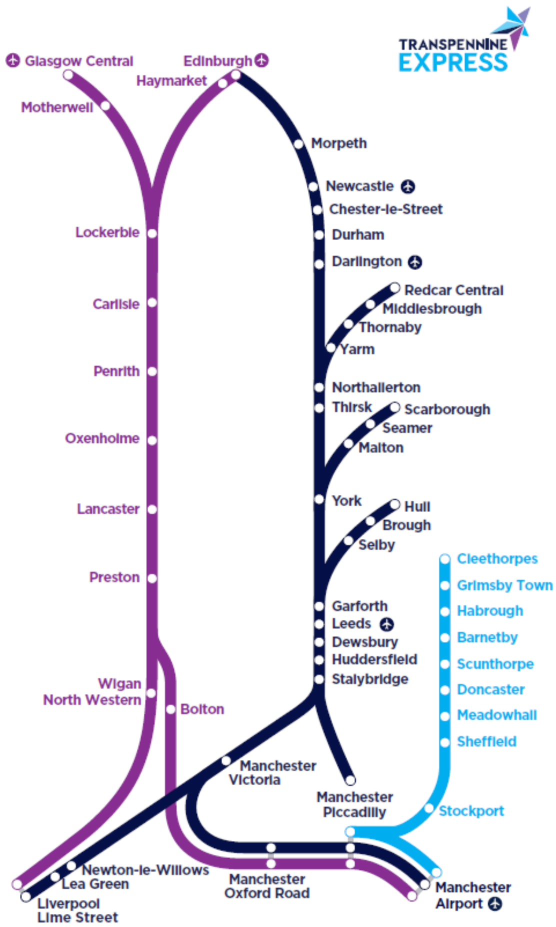

Fig. 4.3. Route map for the train services of TransPeninine Express (available from http://www.projectmapping.co.uk/Reviews/Resources/TPERouteMapDec2019nongeo.pdf)

4.7. Expected Results and Possible Criticalities

The codes implementation of SDNE model was based on “Keras”[2], which is a well-known deep learning API written in Python, running on the top of the machine learning platform ”TensorFlow”[3]. After applying the SDNE algorithm on the refered TransPennine network, we are expected to 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 can be given however if all the values 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).

After acquired the embedded representation we then start to validate the performance of the SDNE model. In a simple case, we obtain a continuous embedding vector for each element in a 11-node toy network (we suppose the output shown as figure 4.4). Euclidean distance between the corresponding node pairs can be effectively calculated. The lower

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

Euclidean distance value, the more close they are. In contrast, the lighter color the cell has, the farther the projected embedding expression distance is in the vector space. We can explicitly explain this heatmap from the following perspectives:

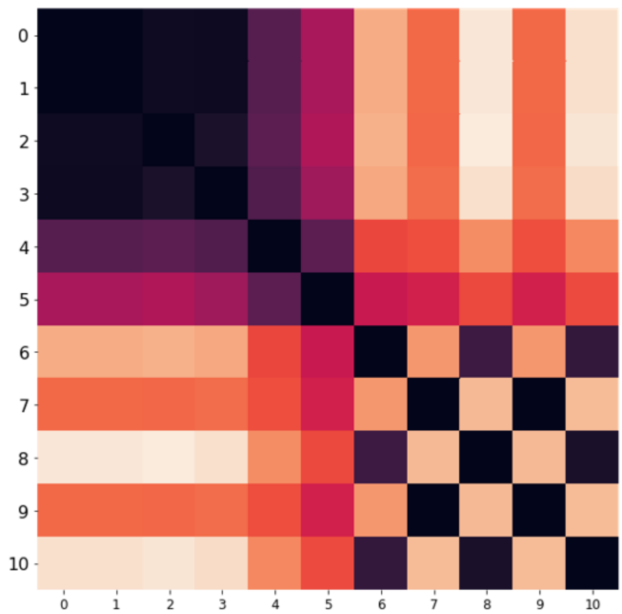

Fig. 4.4. heat map of Euclidean distance between station pairs

- The overall trend is, if the geographical distance between two stations is farther, the Euclidean distance between their embedded vectors are projected to be remote. We can easily identify this characteristic from the gradient color distribution.

- From the top-left corner heat map – the heat map between nodes 0, 1, 2 and 3, we can see the distances between them are similar and very small, which highly preserved the characteristics of clustering here since they situated in the same urban area (less than 2 miles) in real-world. In other words, in the real-life network, the location of these four stations are very close to each other and the spatial distribution density of stations in this area is much higher than other areas in this network.

- Node 8 and Node 10 are neither directly connected to each other, nor locating on the same line (route). Interestingly, they have a similar Euclidean distance in terms of each of the other nodes in this network which means they have been subjected as similar vectors in the Euclidean space. From this perspective, we can see the SDNE framework generates node embedding representations not only by first-order proximity, but on highly preserving the second-order proximity of the network as well. In other words, the model identified Node 8 and Node 10 hold a similar ”structural role” in the entire network (they are both the terminal stations in the railway network, they are both the station travels a long journey time to arrive the adjacent station).

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) [23] – is introduced for generating route vectors. To the best of author's knowledge, it is the first time to introduce such matrix decomposition technology into 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. (Shown in figure 4.5). 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 compute the sigma matrix for each of the routes, respectively. Based on this approach, the final route embedding for each service was generated accordingly.

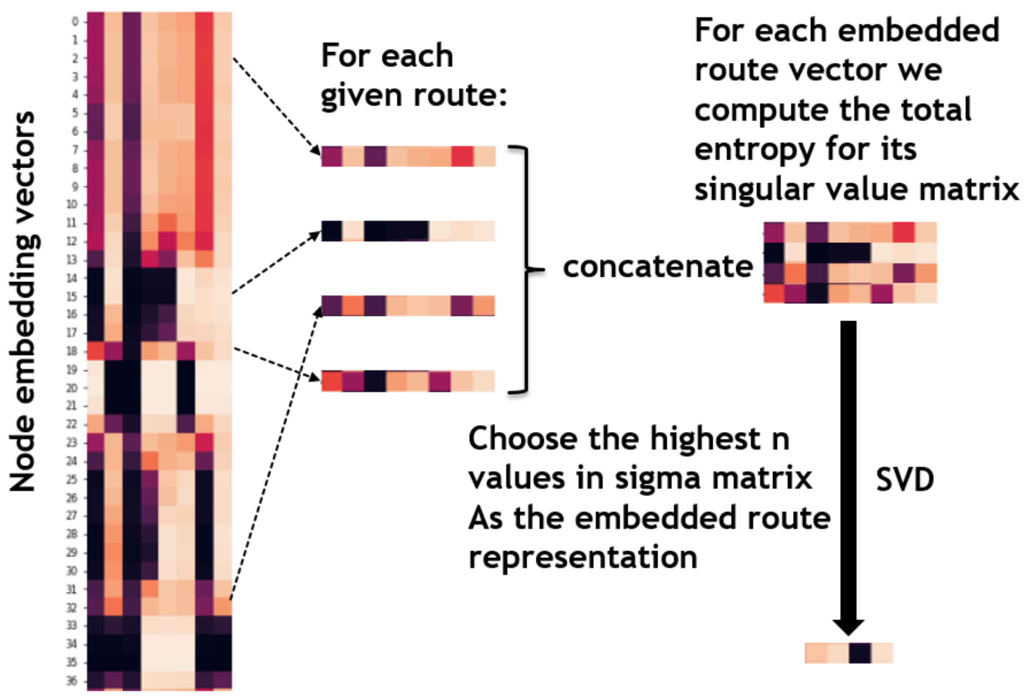

Fig. 4.5. Process of performing route embedding

The output of route-embedding method above was finally implemented within three classical supervised ML algorithms to predict primary train delays; the three ML algorithms are: Decision Tree (DT) [19]; Random Forest (RF) [20]; Multi-Layer Perceptron (MLP) [21]. In addition, we compare our Singular Value Decomposition (SVD) under SDNE framework, with Principle Component Analysis (PCA) strategy [24] - 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. Confusion matrix is another tool that will be used to visualize the performance of classification algorithms. For each prediction task perform in DT, RF and MLP, all predictable cases/instances are intend to be classified into five categories: none-delay (0), mild delay (1), moderate delay

(2), serious delay (3), and severe delay (4). Such that we can easily calculate the prediction accuracy of different predictors on various delay level respectively. Current the topological relations between each traffic elements have been captured and we have 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. A table 4.1 below has been given to summarize where we highlighted most promising candidate strategy for each module. 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 identified previously.

Table 4.1: 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

5. Incident Attribution Analysis

5.1. Introduction

Several prior studies have considered the use of intelligent approaches and AI-based apps to forecast railway accidents. Extracting usable information from informative incident reports is one of the first steps in making such forecasts. The structural dataset can then be used to test the appropriate modelling methodologies.

Each incident takes place in a station or along a railroad route, and each has its own set of circumstances. These reasons might be individual or hybrid, and they must be well-documented once they occur. There are various techniques to determining what specific internal/external elements influence the probability of incident events occurring. Because it is based on real data rather than technical decisions, the text mining approach is the most extensively utilised. For example, [25] suggested an NLP model to extract insights from railway incident reports by disclosing the frequency, distribution, and co-occurrence of each incident term. Using text analysis tools, this study shows that it is possible to map distinct sequences of English text to concepts and entities in incident domains. [26] investigated the link between incident report texts and the real cause of adverse events using word embedding algorithms (GloVe and Word2Vec). The results showed that this approach can accurately diagnose incident causes based on report narratives while also detecting major anomalies between recorded reporting and real condition. The goal of the graph network representation in this part is to extract the topological graph's critical feature information.

The feasibility of using the DT approach in hazard categorization was investigated in [27]. An examination of accidents at railway stations was done in order to anticipate the characteristics of passengers who were involved in incidents. While [28] looked on the viability of predicting traffic incidents one month ahead of time on two high-speed train lines in Spain. Furthermore, [29] provided a general prediction model for weather-related incidents in the United Kingdom from the standpoint of meteorology, demonstrating the capability of adapting diverse formats of weather events in the incident prediction process.

5.2. Background and Description

As we have seen in section 5.1, incorporating empirical data into the statistical analysis/regression process has become the most significant feature in the tasks of abnormal events/disruptions prediction. Such data could include information regarding the frequency, location, and timing of various disturbances. However, we discovered from the literature that using empirical data to evaluate disruption frequencies and implications of various disruption is insufficient. Consider a medium-sized Railway transport network with 200 stations, where the goal is to forecast incident frequencies and impacts for 30 different disruption kinds during five different time periods during the day, and individually for each season. Then at least $200 \times 30 \times 5 \times 4 = 120{,}000$ empirical observations are necessarily required. To put it another way, each of these 120,000 cases requires enough empirical observations to fit a probability density function to. In practice, this means that assembling such a large amount of empirical evidence from historical disruption records is impossible (at least it is not economical). One of the outstanding advantages of ML-based methods is they have excellent performance on the limited dataset due to their hierarchical structural training system and intelligent inferring abilities. In additional, Data augmentation is another technique widely employed in machine learning, not just for limited data problems, to increase the variation in the dataset. Data augmentation exploits dynamics in the dataset to generate new examples from the available dataset. For larger datasets, simple transformations such as reflection, rotation and translation are effective. There are also more complex approaches to data augmentation such as Generative Adversarial Networks and Style Transfer Networks. In other words, The introduction of a supervised learning prediction model to predict disruption frequency and impact for each individual Railway transport network element is both promising and necessary, as it allows for disruption predictions to be conducted at any particular station for a specific time period without the need for sufficient empirical disruption observations for each location and time period. Furthermore, ML-based methods are able to improve the efficiency of the disruption impact prediction process by borrowing effective tools from Data Science sector, especially when there are not a large number of disruption instances. As a result, the supervised learning method has the benefit of overcoming the computing challenges posed by common traffic simulation models used in real-world Railway networks.

5.3. Objectives and Research Questions

There has been a few of work towards investigating incident frequencies under the background of railway traffic safety. One of the limitations of this objective is that most of these traffic researches primarily used descriptive and aggregate models to predict probabilities of various events. On the one hand, the complex nonlinear spatial-temporal interactions between different train vehicles (e.g. timetable conflicts) and operators (e.g. track accessible rights) comprehensively determine primary delays status for individual train services. They are difficult to be accurately predicted in real scenarios if we fail to preserve these relations. On the other hand, directly feeding all the observed relations/deterministic factors even graphical objects into the conventional statistical analysis function or descriptive model result in the computation space too vast thus computation efficiency will be undermined.

To this aim, we plan to develop a disaggregated modelling approach to predict disruption frequencies and to predict the corresponding impacts for each disruption type. Supervised learning approaches are proposed to perform these predictions, which allows the predictor to estimate disruptions at individual stations for each time period. Considering there is no sufficient empirical disruption observations available for each location and time slot, this approach will enable a fast and accurate prediction of disruption impacts for more unknown disruption instances without requiring a lot of historical disruption observations. According to the considerations above, we proposed several Research Questions (RQs) under the scope we proposed:

RQ1: How to develop a generic methodology to estimate the occurrence possibility of incidents/disturbances and their subsequent impacts on potential delays, for different causes, individual stations, and multiple time period, by incorporating the specific characteristics of the different stations?

RQ2: How can we properly extract useful information from incident report/log, which can be easily utilised by the model we developed in RQ1?

RQ3: Can we demonstrate possible answers to RQ1 and RQ2 through a simple proof-of-concept demonstrator in order to inspire future developments and a technology roadmap?

5.4. Methodology and Tools

Scientifically, commonly seen supervised machine learning model, such as Logistic regression, MLP, KNN and Random Forest are qualified to predict the frequency of occurrence for different kinds of accidents and their passenger delay impacts for individual railway stations, by incorporating the specific characteristics of each station. Practically, aim to provide dispatch agent with predicted disruption impacts for each individual station in the network, for each distinguished time period and disruption type, supporting the agent to prioritise locations where to put mitigation measures (rescheduling) in place. The expected output of this module is to identify the location and time period where and when accidents/disruptions are most likely to occur, which will be quantified by its corresponding occurrence probability.

At the very first step of our research, we need to conduct an empirical study about the different disruption types and distribution pattern among UK railway system. Next we will to adopt multiple supervised machine learning model mentioned before to analysis and fitting the distributional probability for each disruption type per station and per time period. We will also discover the linear or non-linear relations between presumed disruption predictors and the occurrence likelihood of different types of incidents for each station based on the empirical evidence we discovered in the first step.

5.5. Artificial Intelligence and Machine Learning Models

This section intends to elaborate how the proposed methodology will be used to predict disruption exposure and impact at different BRS stations. Firstly, we will introduce the proposed modelling framework. Then the supervised learning model used to predict disruptions will be explained, followed by the model for disruption impact predictions. Before the actual specification of the machine learning model we will start with the definition for variables and notations that are going to be used in this case study.

Let us define each station $s \in S$ with $|S|$ indicating the total number of stations in the considered network. i and j represent the departure station and terminal station for a certain journey (route), respectively. The type of disruptions was defined by d, and $|D|$ is the number of disruptions types. In the modelling we assume that time t is not continuous but is split into distinguished time periods and we have a set of time slices called T. In addition, we also define the disruption frequency f and the disruption impact w. To calculate the measurement of station significance, we define its formulation as below.

$$C_s = \sum_{t \in T} \sum_{d \in D} (f_{d,t,s} \times w_{d,t,s})$$

Where the significance of a station is defined as the sum of all time periods during which the predicted frequency of each disruption f (the total number of disruptions per year) at station s multiplied by the predicted impact w. In practice, the impacts of a disruption are generally more than passenger delay only (e.g. the costs of crew rescheduling and passenger delay compensation, etc.), and more than the stations themselves (e.g. incidents also occur at the links between stations), yet we limit our scope within the nominal travel time impact a

Notation	Explanation
s, S	Railway station, The set of the stations
i	Departure station for a journey
j	Terminal station for a journey
e, E	Edge of graph G, The set of edges
d, D	Disruption type, The set of disruption types
t, T	Time period, The set of time periods
C_s	Station significance for station s
f	Disruption frequency
L_t	The travel time for a journey
p	Disruption probability
q	Passenger demand
N	Network vulnerability
w	Passenger delay
y, Y	Label for samples, the set of label categories
$\hat{y}$	Predicted label for classification models
x	Dummy variable
$l_{i,j}$	The length of the shortest path between i and j
$n_{i,j}$	The number of simple paths between i and j
Δh	Increased passenger-weighted travel time

disruption has inflicted on passengers at specific stations. In other words, the more frequent with different kinds of disruptions occur, and the more yearly passenger delay hours these disruptions result in, the more significant the station is. Further, we define the considered network vulnerability N as the sum of predicted station significance just as the equation 10.

$$N = \sum_{s \in S} C_s$$

It is worth to note that these two equations do not consider the interactions between different disruptions occurring at the same time, as this may result in inter-dependencies that enlarge the impact of independent disruption event. In this module we propose an integrated modelling framework which incorporates the calculation process of C_s and N simultaneously in which the supervised machine learning models is used to predict disruptions and passenger delay impacts, while the unsupervised learning model (clustering) is applied to categorise different stations. The modelling framework is explained in the remainder of this section.

Next a major task is to employ supervised learning approach to predict the exposure in terms of different disruptions $d \in D$ at station $s \in S$ during each time period $t \in T$. Due to the fact that each disruption type occurs relatively infrequently at a particular station within a specific time period, what we want to do technically implies predicting the occurrence of relatively rare events. For this reason we cannot regard $f_{d,t,s}$ as the objective function since the model always predicting zero for it. Which will further lead to a low MSE (Mean

Squared Error) and high F1 score (meaning the model has a poor ability on fitting the target dataset and hard to be transferred to the unseen dataset). Alternatively, we consider the disruption probability $p_{d,t,s}$ of every disruption type detected during each considered time period (i.e. each AM, PM, Peak hours and evening period). Therefore the amount of available samples in this model equals $|S| \times |T|$. Specifically, we apply a classification algorithm which will calculate disruption probabilities for each $d \in D$ and then assigns each sample to one of the disruption category d based on the highest probability. Thus the dimension of the target vector equals ($|S| \times |T|$, 1), where the column position can take $|D|+1$ different values. A matrix with the size of ($|S| \times |T|$, $|D|+1$) will be the final output of this part.

One-hot encoding method will be applied for the categorical features Time of Day, Seasons and Lines, in order to capture the potential influence of different slots, i.e. peak, off-peak, evenings and weekends even the switch of season, on disruption probabilities. A line refer to the specific train service serving each station. An intuition behind it is that different stock types or different state/age of infrastructure potentially influence the occurrence of accidents. The features of Start station and Transfer station are regarded as binary attributes for prediction. Being equal to 1 if the station is a departure/arrival or route-to-route transfer station, respectively. It is believed that the occurrence of some disruptions is related to a station is departure/arrival or due to malfunctioning train or the late/absent of a train driver often arise here. Similarly, it is reported that transfer stations might be more sensitive to get involved in disruptions as more complex infrastructure such as switches, large passenger volumes, and longer passenger in-station time.

Given the purpose of predicting the probability of various disruption types at a certain station within specific time period, we implement two different machine learning algorithms suitable here: Logistic Regression classifier and Multi-Layer Perceptron classifier. The whole dataset will be split into a 75% training set and a 25% test set. There is no validation set because we will apply a randomised five-fold cross validation here. Next we will propose the formula of cross entropy loss to generate the accuracy matric. As the equation shown below, it calculates the log-likelihood of the true label y, given the predicted probability that an instance equals this true label.

$$\log p_{(y|\hat{y})} = y \times \log p_{\hat{y}} + (1-y) \times \log\,(1-p_{\hat{y}})$$

5.6. Reference Datasets

Network Rail will regard primary delay as a scaler that mainly contribute to the performance for train services – PPM, There are various possible reasons for a particular event. For example, in the section of ‘External events – Train Operator Company Responsibility’ in the list below.

And according to Network Rail, disruption responsibility corresponds to British Railway Transport System can be attributed to the following causes:

In the orange part of pie chart, issues including operational issues, damage to or failure of the infrastructure of the railway such as tracks, signalling or points, including where bad, but not extreme, weather causes delays to the rail network (Network Rail responsibility).

- Network Rail External here are the delays caused by external factors such as weather,

- **A*** - Freight Terminal Operations Causes
- **D*** - Holding Codes
- **F*** - Freight Operating Causes
- **I* and J*** - Infrastructure causes
- **M* and N*** - Mechanical or Fleet Engineer Causes
- **O*** - Network Rail Operating causes
- **P*** - Planned or excluded delays or cancellations
- **Q*** - Network Rail Non-Operating causes
- **R*** - Station Operating Causes
- **T*** - Passenger Operating causes
- **V*** - External events – TOC Responsibility
- **X*** - External events - Network Rail
- **Y*** - Reactionary Delays
- **Z*** - Unexplained delays and cancellations

Fig. 5.1. A category for the potential reason of events

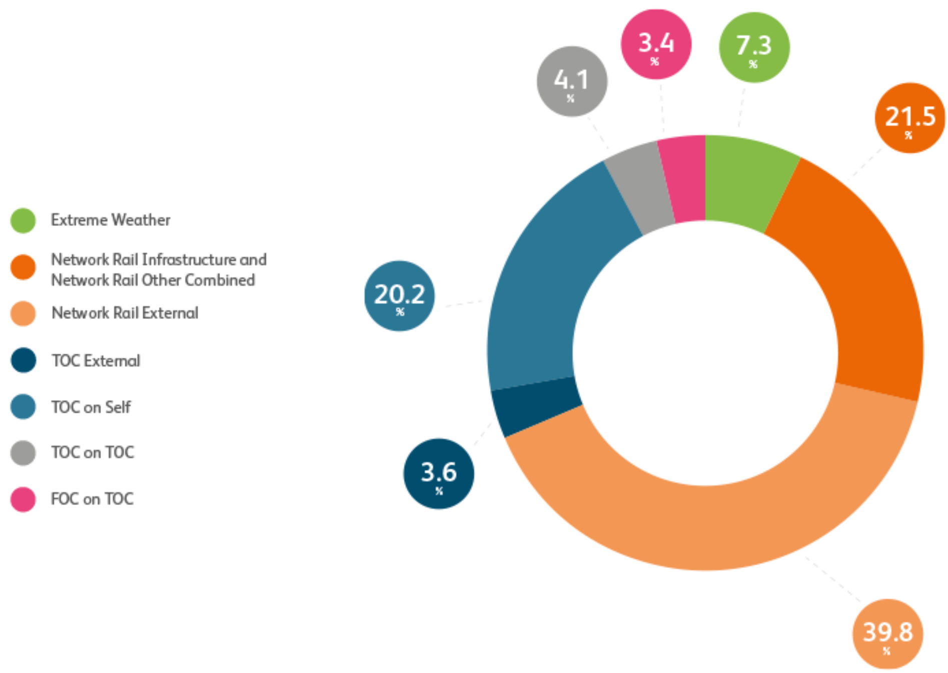

Fig. 5.2. A category for the potential reason of events

trespass, vandalism, cable theft and fatalities, these causes normally result in serious consequences such as hours-long suspension of a specific route/stations, which we called as ”disruptions”.

- TOC External: includes issues which the train company couldn’t have foreseen such as passenger action or illness on trains, damage to trains by road vehicles or the forced closure of a station as they were managed, commonly known as ”incidents”.

- TOC on TOC: Several situations under this category: Firstly, train operation company has its services delayed by the actions/impacts of another train company such as a delayed train from one company causing other connection trains (either at the same station or the neighboring station) to be delayed (i.e., delay propagation). Secondly, As a result of delay propagation, delays may turn into operational conflicts between adjacent trains, which may disturb the arrangement of the train operation plan and threaten the operational safety of the trains from other train operators (i.e., timetable conflicts). Thirdly, the small disturbances that barely can be noticed or recorded by the system, known as ”disturbances”. And likely, these ”disturbances” will be recovered during train running by giving it the ”Margin time” or ”Running Allowance”.
- FOC on TOC: Train companies having their services delayed because of the actions of a freight train on the network. (We do not incorporate freight trains in this case)

5.7. Expected Results and Possible Criticalities

In this sub-task we aim to adopt multiple supervised machine learning model, such as regression tree, Multi-layer Perceptron, K-nearest Neighbours and Random Forest to analyse and fit the distributional probability for each disruption type per station and per time period. During which we will discover the linear or non-linear relations between presumed disruption predictors and the occurrence likelihood of different types of accidents for each station. To sum up, this case study demonstrates a proof-of-concept about how supervised-ML approaches/algorithms could be leveraged to deal with the tasks of predicting disruption exposure and potential impact at different stations. It is worth to note that our proposed models do not consider the interactions between different disruptions occurring at the same time, as this may result in inter-dependencies that enlarge the impact of independent disruption event. This may cause the performance of result less convinced especially in a very large-scale network. However, it will minimise the scope of our research question into an ideal environment and benefits us to focus on the proof-of-concept itself.

Data is most essentially required to perform an high-quality disruption analysis throughout the entire chapter. In the case of interactive visualization, it would be possible to exploit already existing collections of delay attribution data (i.e., Open data feeds from Network Rail [1] and leverage existing train operating timetable provided by TPE (i.e., The main data resource we use in last case study) to build a suitable dataset. A supervised-ML model must be paired with relevant data for it to become a reliable regression/classification model - A model without ground-truth data is unconvincing and meaningless, while the data, which is going to be collected from the previous incident reports and disruption records. Collecting sufficient data is challenging and unrealistic since it is often a lack of good quality disruption log data especially some severe disruption events are not that commonly occur in the real-life. As the requirement of input data scale and limited computing resources of supervised-ML models, the incorporated incident attribution analysis will become increasingly reliable and reference valuable.

[1] https://publicdatafeeds.networkrail.co.uk/ntrod/welcome

From a practical standpoint, the supervised-ML based incident attribution analysis framework we suggest in this case study can not totally replace the traditional statistical methods and linear regression algorithms. As an novel complement to traditional applied methods, it is expected to provide more predictive insights for the occurrence of different kinds of disruptions before it is actually detected. A summary in the table 5.1 has been provided where the most suitable solution for each module has been highlighted. To investigate the complicated relationships between modeled train services and events, our suggested modules in this case study include user-friendly interactive visualizations with appealing interfaces and look-up functions. The potential primary/secondary delay resulting from the current incidents/train service event across the TPE network is then estimated using a GraphSAGE-based model based on the visualized event-delay interactions. In order to enhance overall service quality, a pilot intervention simulation will also be carried out using a number of supervised machine learning approaches. The information about the key contributions that should be completed inside each module is provided in the column named ”Contributions.” A list of the suggested solution, which includes the relevant techniques and AI components, shown in the last column.

Table 5.1: Suitable AI approaches for Incident and Secondary delay Analysis

Module	Contributions	Suitable Approaches
Interactive Visualization Module	Visually illustrating how the sequential chain reaction is triggered between different incidents and train services.	Big Data Analysis NetworkX[a] Plotly[b]
Intervention Simulation Module	Simulating the effects of interventions to improve service performance.	Regression Tree Multi-layer Perceptron K-nearest Neighbours Random Forest
GraphSAGE-based Module	Predicting if there is a trigger reaction between a provided service and the existing train/events with any cause-respond effects.	GraphSAGE[c]

a. https://networkx.org/documentation/stable/index.html
b. https://plotly.com/
c. https://snap.stanford.edu/graphsage/

6. Conclusions

The definition of two pertinent case studies in the context of AI Train Delay Prediction and Incident Distribution Analysis is covered in this deliverable. The case studies will be created as the project moves forward in order to gain some insightful information that will help define future roadmaps for the usage of AI methods in linked railway applications. At this point, the case studies have primarily been defined in terms of their history and motivation, aims and research questions, intended AI methodology and tools, available datasets, the AI/ML models that will be utilized, expected outcomes, and potential criticalities. In order to fully utilize the capabilities of AI and ML, researchers and engineers must handle a number of opportunities and difficulties that will become clearer during the proof-of-concept stage. Due to the project's scope, a number of pertinent scenarios had to be chosen while taking into account the available resources, the goals that could be achieved, and the outcomes that could be obtained. Despite this, we have highlighted a wide range of potential research avenues and angles that could be pursued in addition to the two case studies that are covered in this deliverable.

Bibliography

[1] C. Wen, W. Mou, P. Huang, and Z. Li, "A predictive model of train delays on a railway line," *Journal of Forecasting*, vol. 39, no. 3, pp. 470–488, 2020.

[2] 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.

[3] Y. Yang, P. Huang, Q. Peng, J. Li, and C. Wen, "Statistical delay distribution analysis on high-speed railway trains," *Journal of Modern Transportation*, vol. 27, no. 3, pp. 188–197, 2019.

[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] E. V. Andersson, "Assessment of robustness in railway traffic timetables," Ph.D. dissertation, Linköping University Electronic Press, 2014.

[6] 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.

[7] J. Ludvigsen and R. Klæboe, "Extreme weather impacts on freight railways in europe," *Natural hazards*, vol. 70, no. 1, pp. 767–787, 2014.

[8] D. Rößler, J. Reisch, F. Hauck, and N. Kliewer, "Discerning primary and secondary delays in railway networks using explainable ai," *Transportation Research Procedia*, vol. 52, pp. 171–178, 2021.

[9] W. Daamen, R. M. Goverde, and I. A. Hansen, "Non-discriminatory automatic registration of knock-on train delays," *Networks and Spatial Economics*, vol. 9, no. 1, pp. 47–61, 2009.

[10] J. Peters, B. Emig, M. Jung, and S. Schmidt, "Prediction of delays in public transportation using neural networks," in *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, vol. 2. IEEE, 2005, pp. 92–97.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1067–1077.

[17] R. A. Horn, "The hadamard product," in *Proc. Symp. Appl. Math*, vol. 40, 1990, pp. 87–169.

[18] C. F. Van Loan and G. Golub, "Matrix computations (johns hopkins studies in mathematical sciences)," *Matrix Computations*, 1996.

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

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

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

[22] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

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

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

[25] K. N. Syeda, S. N. Shirazi, S. A. A. Naqvi, H. J. Parkinson, and G. Bamford, "Big data and natural language processing for analysing railway safety: Analysis of railway incident reports," in *Human Performance Technology: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2019, pp. 781–809.

[26] M. Heidarysafa, K. Kowsari, L. Barnes, and D. Brown, "Analysis of railway accidents' narratives using deep learning," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 1446–1453.

[27] H. Alawad, S. Kaewunruen, and M. An, "Learning from accidents: Machine learning for safety at railway stations," *IEEE Access*, vol. 8, pp. 633–648, 2019.

[28] C. Bergmeir, G. Sáinz, C. Martínez Bertrand, and J. M. Benítez, "A study on the use of machine learning methods for incidence prediction in high-speed train tracks," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 2013, pp. 674–683.

[29] Q. Fu and J. M. Easton, "Prediction of weather-related incidents on the rail network: prototype data model for wind-related delays in great britain," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, vol. 4, no. 3, p. 04018027, 2018.