**Handling categorical data in machine learning**

Categorical data only takes a limited number of values. Categorical data for the dependent and independent variables are handled differently. Create an independent matrix (X) and dependent vector (Y).

One-hot encoding introduces a new dummy column for each value of a certain category. Usually done when there is no natural ordering between the categories (nominal). Features with some order associated with them are called ordinal. Continuous features are numeric variables with an infinite number of values between any two values (numeric or datetime).

Other such encodings include ordinal, binary, Helmert contrast, sum contrast, polynomial contrast, backward difference contrast, hashing, baseN, leave-one-out, and target encoding.

Independent = response variable.

Transformation into numeric labels followed by encoding.

Issues:

* categorical features may have a very large number of levels (high cardinality) where levels appear in a relatively small number of instances
* input categories must be numeric
* categorical data doesn’t contain the same contextual information for machines

Relationships between a categorical feature and continuous feature can be investigated via boxplot (df\_flights.boxplot('dep\_time','origin',rot = 30,figsize=(5,6))

Should also check frequency distribution of categories within the feature using value\_counts(). Add .count() for the number of distinct categories.

Dummy encoding produces one less category than one-hot, represented by a row of all 0s.

Effect coding uses -1s instead of 1s.

Make sure to typecast categorical features to a category dtype rather than an object.

Label encoding can be done using cat.codes. Numerical values may be misinterpreted by an algorithm.

One-hot is not very useful when there’s lots of categories: the curse of dimensionality. As dimensionality increases, the volume of the space increases so fast that the available data become sparse.

Typical rule of thumb: there should be at least 5 training examples for each dimension in the representation.

Binary encoding encodes categories are ordinal, converts those to binary code, and then splits the binary digits into separate columns.

Backwards difference encoding is a contrast coding system. The mean of the dependent variable for a level is compared with the mean of the dependent variable for the prior level.

Bin-counting sounds like it might be

**Handling dates in machine learning**

* Use the Pandas datetime field. This allows a datetime index to be constructed, if we so desired.
* Common offset aliases <http://pandas.pydata.org/pandas-docs/stable/timeseries.html#offset-aliases>
* Can use resample to convert data to a new frequency is the index is a datetime index.
* Then we could aggregate things on an hourly frequency.

https://www.datacamp.com/community/tutorials/categorical-data

<https://towardsdatascience.com/basic-time-series-manipulation-with-pandas-4432afee64ea>

# This would be done better on paper.

Apply for MIDAS open.

Dataset comparison to the Chinese data.

Lit review categorical data, forecasting prediction, timestamp. Email to Noura.

**INTRODUCTION**

Train delays impose a huge cost on passengers and operators, contributing to the inefficiency of train operations (Van Oort, 2011). A preliminary analysis of historical delay attribution data[[1]](#footnote-1) released by Network Rail (NR) for the 2018-2019 financial year (1st April 2018 – 31st March 2019) shows that 7.2 million incidents caused 35 million minutes of delay. Extrapolation from figures[[2]](#footnote-2) released by the National Audit Office (NAO) in 2008 suggests this cost a total of £2.5 billion in terms of time lost to passengers.

# I haven’t actually found a good definition of what a train delay prediction system is.

Indeed, the validity of all levels of railway operations planning, such as creating feasible and sizeable timetables, predicting real-time traffic, predicting conflicts, and providing reliable passenger information, depends on the accurate estimation of train process times that are subject to delay incidents (Kecman, Corman, and Meng, 2015). Although the majority of current prediction systems utilise analytical models (Oneto et al, 2016), the focus of this paper will be data-driven models, and, in particular, machine learning techniques.

# Okay. 30 references. I should probably cut some down. And I want a meta-study on the value of including exogenous data in machine learning models as well. And I’d like to have actually read all of my papers, thoroughly. For now, I’m going to convert it into LaTeX and re-write it as I do. Wish me luck!

**DELAYS**

Though precise terminology differs, the literature agrees that there are two principal classes of delay (Olsson, 2004): primary (exogenous) and secondary (knock-on, consecutive).

A **primary delay** is “caused by external stochastic disturbances” (Oneto et al, 2016). The causes of primary delays are varied and numerous. Berger et al. (2011) lists the following causes: disruptions in the operations flow, accidents, malfunctioning or damaged equipment, construction work, repair work, and weather conditions. Network Rail[[3]](#footnote-3) additionally lists:

* Signals and points failures
* Severe heat (leading to buckled rails)
* Fatalities
* Flooding
* Landslips
* Leaves
* Snow and ice
* Vandalism and trespass

Milinkovic et al. (2013) add technical failures, lower-than-scheduled running speeds, prolonged alighting and boarding times of passengers, and bad weather conditions, citing Goverde (2010). The relative importance of these factors was explored by Olsson (2004), with the conclusion that, for double track lines, a key factor is the management of boarding and alighting passengers, and for single track lines, the management of train crossings. Goverde (2005) confirmed that non-negative arrival delays of passenger trains follow a negative exponential distribution. The distribution of train delays on the British railway network is accurately described by *q*-exponential functions (Briggs and Beck, 2007).

A **secondary delay** is “generated by operations conflicts” (i.e. primary delays) (Cerreto et al, 2016). As Berger et al. (2011) explain, primary delays induce a cascade of secondary delays of other trains which have to wait according to certain waiting policies between connecting trains.

Nor can these delay knock-ons be exactly forecast, as both Berger et al. (2011) and Milinkovic (2013) note, due to unpredictable influences: the length of primary delays, the timetable of the trains, and infrastructure (e.g. single or double track, station layouts, and interlocking). Additionally, drivers can attempt to reduce delays by driving faster than planned, or reducing their dwell time at stations, to make up for the delay. Daamen, Goverde, and Hansen (2009) further distinguish between two main classes of knock-on delay:

1. hindrance at conflicting track sections
2. waiting for scheduled connection in stations

Much of the literature focuses on one type or the other, or even on the problem of distinguishing between them, as in Lessan et al. (2019), where the key driver behind the implementation of a hybrid Bayesian network is this problem. Hansen et al. (2010) use historical train describer records to automatically classify delays into initial (primary) and consecutive (knock-on) ones. Wang et al. (2019) use a similar technique.

**TIMESCALE**

There are several classifications of timescale. Wang et al. (2019) divided predictions in short-term (i.e. predicted with real-time operating data) and long-time (a relatively long period; 3 days to 1 week in advance). Markovic et al. (2015) distinguish between the tactical level (in which models are applied to both timetabling and resource planning), the operational level (in which models are used for the real-time prediction of train delays), as reinforced by Corman and Goverde (2017). For the purposes of this project, *three* timescales will be defined: real-time, short-term, and long-term.

1. **Real-time**: typically modelled as a time series forecast problem (Oneto et al, 2016), real-time delay prediction models are *online*, i.e. updated as data on train movements becomes periodically available. Most data-driven models fall into this category. As Kecman, Corman, and Meng (2015) note, models for real-time traffic have so far focused on overcoming the great combinatorial complexity of train rescheduling, delay management, and rolling stock and crew scheduling.
2. **Short-term**: the long-term defined by Wang et al. (2019), this period is unusual in that it is bounded by exogenous data: the limited scope of weather forecasts. Wang et al. (2019) use forecasts accurate up to 10 days’ in advance: this project will use forecasts up to 5 days’ in advance. This is a relatively unexplored area, but much of methodology of both bounding timescales can be incorporated.
3. **Long-term**: in this timescale, models are applied to problems of timetabling and resource planning, and are typically tested for robustness using historical delay data (Markovic et al. 2015). For example, Yuan (2002, 2006) developed a delay prediction model that deals with stochastic behaviour, dependency of train delays, and delay propagation. to assess stability and punctuality of a published timetable against primary delays. According to Wang and Work (2015), the need for training data prevents the application of data-driven methods to scenario planning.

It is worth noting that it is possible to convert between timescales, though with diminished accuracy. In the case of converting to real-time to short-term, for example, the accumulated delay, and the run-time of the train, can be estimated.

**METRICS**

**Punctuality** is “a feature consisting in that a predefined vehicle arrives, departs, or passes at a predefined point at a predefined time” (Rudnicki, 1997). This definition, while laxer than that used by others (such as Gylee, 1994), allows greater flexibility in the measurement of punctuality. It also has the interesting effect that trains that arrive *early* cannot be considered punctual. However, the use of punctuality hides a lot of information, as Skagestad (2004) notes; consequently, Olsson (2004) uses reliability and variability. The Office of Road and Rail (ORR), the UK’s railway regulator, uses Public Performance Measure (PPM) to assess punctuality, and Cancellations and Significant Lateness (CaSL) to measure reliability.

**Reliability** has several measures. Rietveld et al. (2001) list the following:

* The probability that a train arrives *x* minutes late (punctuality, as defined above)
* The probability of an early departure
* The mean difference between the expected arrival and the scheduled arrival time
* The mean delay of an arrival given that one arrives late
* The mean delay of an arrival given that one arrives more than *x* minutes late
* The standard deviation of arrival times

**Variability** is a “measurement of the uncertainty of trip journey times in transportation” (Olsson, 2004). First introduced by Noland and Polak (2002), where it is related to the distribution of arrival times for a train. A train that arrives the same amount of minutes behind schedule every day has low variability, but would not be considered punctual.

**RAILWAY TERMINOLOGY**

**Capacity utilisation** is “the extent to which the maximum theoretical capacity (for a given scenario) is being used” (Armstrong and Preston, 2017). However, as the International Union of Railways (IUC) notes (2004, leaflet 406, capacity), for any given section of railway: “capacity as such does not exist [and] railway infrastructure capacity depends on the way it is utilised”. Achievable capacity depends upon infrastructure characteristics, the performance characteristics and mix of trains using the route, the timetable in operation, and the target levels of reliability and punctuality to be achieved by timetabled services (Armstrong and Preston, 2017). Capacity utilisation is widely assumed to be positively correlated with the propensity of delays to propagate in a rail network (Olsson, 2004).

**Dwell time** is the time a train stands at the platform, usually for the purpose of allowing passengers to board or alight (Douglas, 2012). Puong (2000) asserts that dwell time is a key parameter of system performance, service reliability, and quality. Passenger volume is considered to be the main factor influencing dwell time (San and Masirin, 2016). Magadelaga et al. (2017), though their work focused on freight trains, found that the main influencing factors were crew shortages, locomotive imbalance and defective perway (i.e. rails).

**Headway** is a measurement of the distance or time between vehicles in a transit system. The minimum headway is the shortest such distance or time achievable by a system without a reduction in the speed of the vehicles.[[4]](#footnote-4) The shorter the minimum headway, the greater the capacity of a transit system.

**Free motion time** is the time a train spends in transit between two scheduled stops.

**Slack time** **(timetable allowance, time supplement)**: the amount of time a train can be delayed without causing another train to be delayed. The greater the capacity utilisation, the shorter the slack time.

**CLASSIFYING MODELS**

There are many classifications of railway delay prediction models, based on scope, model type, and solution methods (Markovic et al, 2015). Wang and Work (2015) distinguish between analytical, simulation, and data-driven models; Markovic et al. (2015) between analytic, multiple regression, and machine learning; Corman and Goverde (2018) into static (offline), dynamic (online) and deterministic or stochastic, depending on “the timespan between the time they are run, and the operations they aim to predict; and on how they tackle uncertainty, respectively”.

For the purposes of this literature review, and simplicity, models will be classified into analytical and data-driven (incorporating multiple regression and machine learning), with a particular focus on data-driven models. The timescale dimension previously discussed will be disregarded.

**EXISTING METHODS – ANALYTICAL MODELS**

An **analytical model** is “primarily quantitative or computational in nature and represents the system in terms of a set of mathematical equations that specify parametric relationships and their associated parameter values as a function of time, space, and/or other system parameters” (Friedenthal and Steiner, 2015). Current state-of-the-art train delay prediction systems use analytical models (Oneto et al, 2016): static rules, built by experts on railway infrastructure, and based on classical univariate statistics.

The paper also notes that the estimation accuracy of secondary delays does not improve. They also note that the potential delay is captured by a single parameter, and advocate separately modelling other contributing factors, such as geometry or weather, which other papers discussed do. Other approaches such as non-linear regression models or calibration via robust least squares (as per Italy’s default model).

Simplistic early models, such as those developed by Frank (1966) make overly restrictive assumptions about railway operations, e.g. no overtakes are allowed, the departing times are uniformly distributed, and that the speed of each train is unique and constant.

Subsequent work in this area has largely relaxed these assumptions: Peterson (Over-the road transit time on a single line of railway, 1974), and Chen and Harker (Two moments estimation of the delay on single-track rail lines with scheduled traffic, 1994) included factors such as overtakes, different speeds, priority systems, and uncertainties associated with train departure time.

Further models have become increasingly advanced, incorporating stochastic approximation (Carey et al. Stochaistic approximation to the effects of headways on knock-on delays of trains, 1994), and the impacts of dispatching strategies on train delays and passenger waiting time.

Current state-of-the-art train delay prediction systems use analytical models (Oneto et al, 2016). Although new models are constantly being developed, a good example of a current *applicable* system is that developed by Berger et al. (2011), which is currently used in the German rail network.

**EXISTING METHODS – DATA DRIVEN (MACHINE LEARNING)**

Machine learning is “the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead”[[5]](#footnote-5). In the context of train delay prediction (and indeed, more generally), such models typically provide a better fit, but less straightforward interpretation, than statistical regression models (Markovic et al, 2015). The distinction between machine learning and data-driven models is increasingly becoming blurred.

Wang et al. (2015) acknowledges that the greatest difficulty to such data-driven approaches is data availability. However, the acceptance of data-driven approaches has greatly increased in the years since the paper was published, and the move towards open-source data feeds has increased both the availability and complexity of the data.

Several different machine learning techniques have been studied thus far in the literature (though this is by no means an exhaustive list): Bayesian networks (Lessan et al, 2019; Corman and Kecman, 2018), support vector regression (Markovic et al. 2015), random forests (Oneto et al, 2016), neural networks (Peters et al, 2005; Yaghini et al, 2013, Wang and Zhang, 2019), fuzzy Petri nets (Milinkovic et al. 2013), and extreme learning machines (Oneto et al. 2017). Attention will be given here to results; a thorough review of the exogenous data used follows.

**Peters et al. (2005)**

Perhaps the earliest work applying machine learning to delay prediction (though in the wider context of public transportation) is Peters et al. (2005), in which the authors develop a simple neural network for a small region of the Deutsche Bahn. Subsequent work on neural networks greatly expanded both the complexity of the model and the quantity of data, as in Yaghini et al. (2011).

**Yaghini et al. (2011)**

Yaghini et al. (2011) explore the implementation of a high-precision neural network to predict the delay of passenger trains in Iran. The authors used decision trees and multiple logistic regression models to evaluate the quality of the NN model and found it superior. The paper also tested three different encodings: normalised real number, binary, and binary set, and three different training methods: quick, dynamic, and multiple, and found the binary quick network to be the most accurate. The authors used a dataset of considerable size: roughly 180,000 trains over the course of four years, and a cumulative 5.4 million minutes of delay.

**Milinkovic et al. (2013)**

Milinkovic et al. (2013) used a novel model: a fuzzy Petri net (FPN). Petri nets are mathematical modelling tools used to analyse and simulate concurrent systems (Murata, 1989). The authors explored two separate FPNs. In the first, expert knowledge was used to define fuzzy sets and rules. In the second, an Adaptive Network Fuzzy Inference System (ANFIS) was trained on historical delay data and then replicated in an FPN.

Both were then tested with real data from a Belgrade station node. The ANFIS-defined FPN produced results within 5% of actual delay values for a subset of the data; slightly worse performance was observed for the expert-defined FPN.

**Markovic et al. (2015)**

Markovic et al. (2015) presented the first use of support vector regression (SVR, a close cousin of SVMs) to delay prediction problem. A comparison with artificial neural networks found that SVR outperformed the ANNs. Data for the analysis was again collected from Serbian Railways. The paper used the expert opinions of dispatchers as a key variable, which is discussed later. They noted that train category was an important factor, as higher-category passenger trains have right of way.

**Oneto et al. (2016)**

Oneto et al. (2016) proposed a train delay prediction system that considered both historical train running conditions and weather conditions. This inclusion is discussed later. The authors map the problem into a multi-variate regression problem and compare the performance of kernel methods, ensemble methods and feed-forward neural networks. They found it was possible to build a reliable and robust data-driven model based solely on historical data about train movements, with almost twice the performance of the existing system. The paper is also notable for its close co-operation with the Italian Railway Manager, which led to the development of a set of novel Key Performance Indicators (KPIs) to assess performance. The authors consider real-time prediction a time series forecast problem. They used Extreme Learning machines (ELM), Kernel Regularized Least Squares (KRLS) and Random Forests (RF) with the Nonparametric Bootstrap (NB) procedure to tune hyperparameters.

**Oneto et al. (2017)**

Oneto at al. (2017) generalised their earlier work to produce a dynamic data-driven train delay system. The performance of the system was tuned through the state-of-the-art thresholdout technique, which relies on differential privacy theory. They compare the performance of two implementations of shallow and deep extreme learning machines and find similar results to their previous paper. However, this scope of this paper must be emphasised: it lays the groundwork for the application of this system to a real-life rail network, rather than just academic work.

**Corman and Kecman (2018)**

Corman and Kecman (2018) present a stochastic model for predicting the propagation of train delays based on Bayesian networks (BNs). BNs allow the updating of probability distributions and reduce the uncertainty of future train delays in real-time as more data continuously comes available from the monitoring system. This authors extend this approach by modelling the interdependence between trains that share the same infrastructure or have a scheduled passenger train. The model is tested on historical train realisation data from a bus corridor in Sweden

**Lessan et al (2019)**

Lessan et al (2019) explore three different Bayesian network schemes: heuristic hill-climbing, primitive linear, and hybrid. Hybrid, incorporating domain knowledge and judgements of local experts, was found to outperform other models, with an accuracy of over 80% in predictions within a 60-minute horizon. The authors define a railway system as several interconnected subsystems: infrastructure, rolling stock, control and communication, and various operational rules and policies.

It was found that arrival and departure delays follow the same distribution, with a linear relationship (chain) with a high correlation between arrival and departure delays at the same station (at least 94%). This supports the hypothesis of Wang et al. (2019) of “key train delay stations” at which delays are considerably more likely to propagate, likely a result of an infrastructure bottleneck.

**Wang and Zhang (2019)**

Wang and Zhang (2019) have an unusual time constraint: that of the maximum acceptable resolution of a weather forecast, as discussed early. As discussed previously, their work does not fit in Markovic et al. (2019)’s real-time / operational / tactical trichotomy: rather, it lies between the first two. This is the region that this project intends to explore, so significant attention is paid to this paper in particular in the following section. The authors place emphasis on explaining the necessity of predicting delays in this window. The rationale is simple: passengers cannot easily compensate for delays in real-time. If a passenger can be informed that a train is *likely* to be delayed, they can make an informed choice on catching that train or altering their plans.

The authors present a relatively simple, but effective, gradient-boosted regression trees (GBRT) model. They note that a number of errors are likely due to stochastic errors and the limited timespan of their dataset – only 3 months’ worth of data was collected, which precludes the incorporation of the seasonality of weather.

**INCLUDING EXOGENOUS DATA**

It is widely accepted amongst machine learning practitioners that the greater the quantity of information available for the creation of a model, the greater the performance of the model will be (Oneto et al, 2016). Features can be engineered from existing features or exogenous data can be incorporated.

Data is **exogenous** if it is independent of other input data and the output data depends on it. The scope for inclusion is essentially limitless: any source of data which may affect railway dispatching operations is a viable candidate. In the context of delay prediction, the more of the causes of primary delay that can be incorporated into a model, the better it will perform. Prediction models tend to use either infrastructure (Markovic et al, 2015; Milinkovic et al, 2013) or weather (Oneto et al, 2016; Wang and Zhang, 2018; Oneto et al, 2017), though Oneto et al. (2017) also recommend using information about passenger flows and about railway asset conditions. This project plans to do all of the above.

Primary delay data analyses, such as those performed by Harris (1992), Gorman (2009), and Wen et al. (2017) reveal, in a rudimentary way, the use of proxies for variables. Harris (1992), investigating train punctuality in the UK and the Netherlands, used:

* train length (as a proxy for the number of doors to manage, and passenger demand)
* distance covered (as a proxy for the likelihood of encountering track defects and other technical / operational problems)
* the previous number of stops (as a proxy for cumulative delay resulting from passengers alighting and boarding)
* the age of the motive power unit (as a proxy for reliability; it is industry-held fact that a motive unit’s reliability declines after 20 years)
* track occupation (as a proxy for capacity utilisation the railway, and thus the likelihood of delays propagating)

Gorman (2009) investigated freight rail congestion delay. Factors were broken down into free running time (horsepower per ton, track topography, and slow orders) and congestion-related factors (meets, passes, overtakes, prior time periods’ train counts, total train hours, train spacing variability, and train departure headway). Primary congestion predictive factors were found to have the largest effect on congestion delay.

Wen et al. (2017) found that…

Wang et al. (2019) use three datasets: train schedule data, train delay data, and weather data, all collected from a three-month period between 1st January and 31st March 2018. The schedule data comprised 7172 trains and 2761 stations; each station was additionally geo-located to enable cross-referencing with weather data. Approximately 2.7 million delays were observed in this period, of which 37.4% involved high-speed trains.

Weather data was collected from 344 cities along the route in question, Beijing to Guangzhou. It is worth noting that the two are approximately 2200km apart, and so delays are of a magnitude not often seen in datasets with smaller geographic coverage. The fields were relatively simple, comprising the lowest and highest daily temperature, a categorical weather type, a categorical wind ranking (the Beaufort scale), and an air quality index.

Oneto et al. (2016) found that their random forest model, without the inclusion of weather data, roughly doubled the predictive capability of the original system. Including weather data increased this accuracy by approximately 10%, with the caveat that, the further ahead in the future the forecast was for (and thus the less accurate), the smaller this increase was.

Milinkovic et al (2013) defined their Fuzzy Petri Net (FPN) model by three input parameters: train category (freight, regional, or passenger), timetable influence, and distance travelled by the train. Timetable influence was used as a catch-all of sorts. It included the influence of infrastructure parameters, timetable characteristics, operation time, the type of locomotive, local conditions, technological solutions, principles for safety and signalling, and weather conditions.

Milinkovic et al (2013) defined FPN model by three input parameters:

* Train category (freight, regional, or passenger)
* Timetable influence
* Distance travelled by the train

Timetable influence was used as a catch-all of sorts. It included the influence of infrastructure parameters, timetable characteristics, operation time, the type of locomotive, local conditions, technological solutions, principles for safety and signalling, and weather conditions.

The study used train delay collected in July 2010 over several stations in Belgrade. 3710 trains were included in all categories: international passenger, domestic passenger, suburban and regional passenger, international freight, direct freight, pick-up freight, and other trains. In this period, 826 freight trains and 427 other trains were not registered in the predefined timetable.

The ANFIS model this project plans to replicate functioned best with the following input parameters: train category, arrival time at station, the travelled distance, and the infrastructure influence.

Markovic et al. (2015) used the expert opinion of dispatchers to estimate the likelihood of multiple factors along a rail line (single-tracking, reduced speeds, characteristics of block and interlocking systems, numbers of stations, stops, loops, road-rail level crossings, and junctions) causing a delay. The final estimate was obtained using the Delphi method. A higher score denoted better infrastructure conditions. Their input variables were:

* Passenger train category
* Scheduled time of arrival at station
* Infrastructure influenced defined by expert opinions
* Percent of journey completed distance-wise
* Distance travelled
* Time travelled
* Headway

They found a strong correlation between expert opinions and train delays. As Martin (2016) notes, in real-world train operations, delay prediction relies heavily on the experience and intuition of a local dispatcher, rather than a network-work computational instrument.

Markovic et al. (2015) used the expert opinion of dispatchers to estimate the likelihood of multiple factors along a rail line (single-tracking, reduced speeds, characteristics of block and interlocking systems, numbers of stations, stops, loops, road-rail level crossings, and junctions) causing a delay. A higher score denoted better infrastructure conditions. They found a strong correlation between expert opinions and train delays. As Martin (2016) notes, in real-world train operations, delay prediction relies heavily on the experience and intuition of a local dispatcher, rather than a network-work computational instrument.

The proposed primary dataset consists of historic delay attribution data gathered by Network Rail for the 2018-2019 financial year (1st April 2018 – 31st March 2019). The proposed delay dataset consists of all TRUST messages for the same period. The proposed weather dataset consists of Met Office MIDAS data, across the entirety of the UK, for the same period. Additionally, passenger volume statistics released by the ORR will be incorporated. This is, to the best of the author’s knowledge, the first time such a dataset has been collected in the UK, and likely the most comprehensive gathered to-date in the area of train delay prediction.

Much of the challenge of this dissertation lay in the construction of a dataset amenable to machine learning, a task commonly referred to as ETL (Extract-Transform-Load). The author had little knowledge, and no experience of the details of such a task.

# <https://networkrail.opendata.opentraintimes.com/>

All open data is stored here.

It’s a nicer mirror, for sure.

But if his is there, why does mine need to be? It doesn’t. We’ll have to continue with the ML. Fuck!

**Extraction**

This is the simplest stage. It is the process of collecting data, often from multiple and different sources.

MIDAS data is available under the Open Government License. This allows anyone to copy, publish, distribute, transmit, and adapt the licensed work, and to exploit it both commercially and non-commercially. The user of the licensed work has to acknowledge the source of the work and, if possible, provide a link to the OGL.

Network Rail data is ‘open’. I cannot find an actual license.

National Rail encourage the use and re-use of our feeds in their services. It is available under the Open Government License.

Would it be complete *enough*? Yes.

I am not allowed to alter the content of the feeds themselves, but I do not plan on doing so, so that’s okay.

Much of the relevant material is found in the appendices, but an overview is presented here to provide an understanding of the vagaries of the source data.

Extraction is the process of collecting data, often from multiple and different sources.

Darwin messages switch to a new schema in early 2019, version 16.

Should, for instance, a train be included if the first Darwin message regarding it is of type PASS, rather than DEPARTURE, as should be the case for a departing tra

The first stage was *downloading* the data. Peter Hicks, the founder of RealTimeTrains, and without whom this dissertation would not have been possible, has archived several feeds – SCHEDULE, TD, TRUST, and Darwin – for a number of years.

TD, TRUST, and Darwin all start on 2017-01-01, albeit with missing files several days (2017-01-03, 2017-01-04, 2017-06-11 for TD, 2017-01-03, 2017-01-04 for TRUST).

SCHEDULE starts earlier. However, it is very important with the SCHEDULE feed that no days are missed. Full SCHEDULE extracts are released every Friday at 01:00; on every other day, an *update* extract is provided, which must be applied the previous full extract each day. SCHEDULE only allows inserts up to 48 hours’ before a train is due to run – after which VSTP takes over – so a missing update file on a Saturday could invalidate schedules from Monday to the next full schedule extract on Friday.

SCHEDULE logs start on 2014-10-23. Full extracts are hosted only intermittently. However, this date, naturally, is of no interest to us.

There is a large range of missing schedule data between 2017-07-26 and 2018-03-29 (an update). The next full extract is on 2018-03-30, and so this provides a lower bound of our starting point. Conveniently, this is the start of the UK financial year, which corresponds to a dataset released by ORR pertaining to historical delay attribution, which at the time of writing was thought to be potential avenue of exploration, though this proved infeasible.

However, this range invalidated an ORR dataset, which captures how busy stations are, likely a useful factor.

A year’s worth of data was deemed necessary to capture the seasonality of train delays.

VSTP – Very Short Term Planning – trains are scheduled at most 48 hours in advance. They do NOT appear in the SCHEDULE feed. Instead, Darwin publishes their schedules when the trains are ‘activated’. A corresponding TRUST message may or may not also be published. Their schedules can, therefore, be reconstructed by parsing the Darwin log files. However, much of the metadata included in the SCHEDULE feed is missing, and it seemed unreasonable to attempt to coerce it into this format.

So those trains were discarded. This led to a problem parsing Darwin messages.

Each line in a Darwin log file is simply a Darwin message, of which there are eight types: Train Order, Forecast, Schedule.

**SCHEDULE**

SCHEDULE provides daily extracts and updates of train schedules from the Integrated Train Planning System (ITPS) in CIF format.

The ITPS was introduced in 2010; a report was conducted by the ORR into the resulting disaster (<https://orr.gov.uk/__data/assets/pdf_file/0004/5746/ITPS-ORR-report.pdf>).

Schedules can be uniquely identified by a UID, the schedule start date (SSD), and STP indicator, which defines which schedule should be given priority on a given day.

The creation of a high-quality, open-source dataset could itself have been a valuable dissertation contribution.

Wouldn’t that be cool? I know the dataset very well, after all, I wouldn’t even need to do anything else.

To retrieve the correct schedule for a given UID on a given day, all schedules with that UID must be retrieved. Those not active on the *day* in question – as defined by the *days\_run* field – can be discarded, as well as those with a start date in the future. From those remaining, the schedule with the lowest STP indicator is the correct one, in order `C`, `O`, `V`, `P`. `C` is a planned cancellation. `O` is an overlay from STP. `V` is a variation from LTP. `P` is a permanent, base schedule from

An existing repository (<https://github.com/jhumphry/ukraildata_etl>) was used as an inspiration for a custom CIF parser configured for Peter Hick’s update files.

There are several feeds which were unfortunately not logged, such as the VSTP, and, more interestingly, the TSR (Temporary Speed Restrictions) feed, which details temporary reductions in permissible speeds across the rail network. Fortunately, this data is captured in field in SCHEDULE, namely engineering\_allowance.

Trains from VSTP were therefore disregarded. This, in itself, likely biased the dataset. It seems reasonable to assume that these ‘rushed’ trains are prone to delay. The opposite could also be argued: by their special nature, they are given priority – but we cannot be sure. Any UID in Darwin NOT found in the day’s schedule was discarded.

To ameliorate these concerns, a daily database was defined. For each day, only the ‘winning’ schedule was stored.

Despite my best efforts, there is some data missing. I routinely found that my days (generated by applying updates to full extracts) were smaller in size than ‘full’ database extracts generated for the same day.

Eventually, it was decided to simplify the problem.

Network Rail defines x routes:

* Anglia
* East Midland
* Kent
* London North Eastern
* London North Western
* London Northwestern
* London Overground
* Midland and Continental
* Scotland
* South East
* South West Main Line
* Sussex
* Wales
* Wessex
* Western

Much of the challenge of this dissertation lay in the construction of a dataset amenable to machine learning, a task commonly referred to as ETL (Extract-Transform-Load). The author had little knowledge, and no experience of the details of such a task.

# <https://networkrail.opendata.opentraintimes.com/>

All open data is stored here.

It’s a nicer mirror, for sure.

But if his is there, why does mine need to be? It doesn’t. We’ll have to continue with the ML. Fuck!

**Extraction**

This is the simplest stage. It is the process of collecting data, often from multiple and different sources.

MIDAS data is available under the Open Government License. This allows anyone to copy, publish, distribute, transmit, and adapt the licensed work, and to exploit it both commercially and non-commercially. The user of the licensed work has to acknowledge the source of the work and, if possible, provide a link to the OGL.

Network Rail data is ‘open’. I cannot find an actual license.

National Rail encourage the use and re-use of our feeds in their services. It is available under the Open Government License.

Would it be complete *enough*? Yes.

I am not allowed to alter the content of the feeds themselves, but I do not plan on doing so, so that’s okay.

Much of the relevant material is found in the appendices, but an overview is presented here to provide an understanding of the vagaries of the source data.

Extraction is the process of collecting data, often from multiple and different sources.

Darwin messages switch to a new schema in early 2019, version 16.

Should, for instance, a train be included if the first Darwin message regarding it is of type PASS, rather than DEPARTURE, as should be the case for a departing tra

The first stage was *downloading* the data. Peter Hicks, the founder of RealTimeTrains, and without whom this dissertation would not have been possible, has archived several feeds – SCHEDULE, TD, TRUST, and Darwin – for a number of years.

TD, TRUST, and Darwin all start on 2017-01-01, albeit with missing files several days (2017-01-03, 2017-01-04, 2017-06-11 for TD, 2017-01-03, 2017-01-04 for TRUST).

SCHEDULE starts earlier. However, it is very important with the SCHEDULE feed that no days are missed. Full SCHEDULE extracts are released every Friday at 01:00; on every other day, an *update* extract is provided, which must be applied the previous full extract each day. SCHEDULE only allows inserts up to 48 hours’ before a train is due to run – after which VSTP takes over – so a missing update file on a Saturday could invalidate schedules from Monday to the next full schedule extract on Friday.

SCHEDULE logs start on 2014-10-23. Full extracts are hosted only intermittently. However, this date, naturally, is of no interest to us.

There is a large range of missing schedule data between 2017-07-26 and 2018-03-29 (an update). The next full extract is on 2018-03-30, and so this provides a lower bound of our starting point. Conveniently, this is the start of the UK financial year, which corresponds to a dataset released by ORR pertaining to historical delay attribution, which at the time of writing was thought to be potential avenue of exploration, though this proved infeasible.

However, this range invalidated an ORR dataset, which captures how busy stations are, likely a useful factor.

A year’s worth of data was deemed necessary to capture the seasonality of train delays.

VSTP – Very Short Term Planning – trains are scheduled at most 48 hours in advance. They do NOT appear in the SCHEDULE feed. Instead, Darwin publishes their schedules when the trains are ‘activated’. A corresponding TRUST message may or may not also be published. Their schedules can, therefore, be reconstructed by parsing the Darwin log files. However, much of the metadata included in the SCHEDULE feed is missing, and it seemed unreasonable to attempt to coerce it into this format.

So those trains were discarded. This led to a problem parsing Darwin messages.

Each line in a Darwin log file is simply a Darwin message, of which there are eight types: Train Order, Forecast, Schedule.

**SCHEDULE**

SCHEDULE provides daily extracts and updates of train schedules from the Integrated Train Planning System (ITPS) in CIF format.

The ITPS was introduced in 2010; a report was conducted by the ORR into the resulting disaster (<https://orr.gov.uk/__data/assets/pdf_file/0004/5746/ITPS-ORR-report.pdf>).

Schedules can be uniquely identified by a UID, the schedule start date (SSD), and STP indicator, which defines which schedule should be given priority on a given day.

The creation of a high-quality, open-source dataset could itself have been a valuable dissertation contribution.

Wouldn’t that be cool? I know the dataset very well, after all, I wouldn’t even need to do anything else.

To retrieve the correct schedule for a given UID on a given day, all schedules with that UID must be retrieved. Those not active on the *day* in question – as defined by the *days\_run* field – can be discarded, as well as those with a start date in the future. From those remaining, the schedule with the lowest STP indicator is the correct one, in order `C`, `O`, `V`, `P`. `C` is a planned cancellation. `O` is an overlay from STP. `V` is a variation from LTP. `P` is a permanent, base schedule from

An existing repository (<https://github.com/jhumphry/ukraildata_etl>) was used as an inspiration for a custom CIF parser configured for Peter Hick’s update files.

There are several feeds which were unfortunately not logged, such as the VSTP, and, more interestingly, the TSR (Temporary Speed Restrictions) feed, which details temporary reductions in permissible speeds across the rail network. Fortunately, this data is captured in field in SCHEDULE, namely engineering\_allowance.

Trains from VSTP were therefore disregarded. This, in itself, likely biased the dataset. It seems reasonable to assume that these ‘rushed’ trains are prone to delay. The opposite could also be argued: by their special nature, they are given priority – but we cannot be sure. Any UID in Darwin NOT found in the day’s schedule was discarded.

To ameliorate these concerns, a daily database was defined. For each day, only the ‘winning’ schedule was stored.

Despite my best efforts, there is some data missing. I routinely found that my days (generated by applying updates to full extracts) were smaller in size than ‘full’ database extracts generated for the same day.

Eventually, it was decided to simplify the problem.

Network Rail defines x routes:

* Anglia
* East Midland
* Kent
* London North Eastern
* London North Western
* London Northwestern
* London Overground
* Midland and Continental
* Scotland
* South East
* South West Main Line
* Sussex
* Wales
* Wessex
* Western

1. <https://www.networkrail.co.uk/who-we-are/transparency-and-ethics/transparency/our-information-and-data/> [↑](#footnote-ref-1)
2. <https://www.nao.org.uk/report/reducing-passenger-rail-delays-by-better-management-of-incidents/> [↑](#footnote-ref-2)
3. <https://www.networkrail.co.uk/running-the-railway/looking-after-the-railway/delays-explained/> [↑](#footnote-ref-3)
4. <https://en.wikipedia.org/wiki/Headway> [↑](#footnote-ref-4)
5. <https://en.wikipedia.org/wiki/Machine_learning> [↑](#footnote-ref-5)