BIRMINGHAM UK | 2022

Modelling traffic management decisions using a hybrid machine learning and simulation approach

Joanna KNIGHT[1], Andy KEANE[1], Ondrej HOVORKA[1]

[1]University of Southampton, Southampton, United Kingdom
Corresponding Author: Joanna Knight (jck1g19@soton.ac.uk)

Abstract

Railway simulation models can be used to assess the robustness of timetables by subjecting the simulated traffic to minor disruptions and analysing their impact. For the output of such models to be meaningful, signallers' actions in the event of disruptions need to be represented with a reasonable level of accuracy. This paper presents a hybrid modelling approach that combines simulation software with a model for making traffic management decisions. The construction of the traffic management model is flexible, and this paper considers different approaches. Each approach takes a pair of trains as input and predicts which one will have priority at a conflict location. Traffic management models created using conditional logic are compared with machine learning models built using years of historical data. Results are presented using a case study of six conflict locations: the models make predictions for a dataset of pairs of conflicting trains gathered over 90 days. The machine learning models demonstrate a higher level of agreement with the data than the programmatic models, although the gains for some conflict locations were more significant than others, indicating that each conflict location has its unique characteristics. The traffic management models were then integrated with the simulation software, and a week's worth of historical data was simulated. The machine learning approach for predicting traffic management actions again showed better agreement with the real data.

Keywords: traffic management, simulation, machine learning

1. Introduction

Railway simulation models have the potential to help planners deliver better services for passengers by modelling timetable changes ex-ante. One challenge of simulation modelling is the representation of traffic management when delays occur. The ability to model realistic decisions is necessary if using a simulation to assess the robustness of a new timetable. When trains deviate from their schedules, there is a risk of conflicts arising between train pathways, and it is a challenge to model the resolution of these conflicts realistically in a simulation model.

This paper compares traditional programmatic methods for deciding traffic management actions within a simulation model with machine learning approaches. The methodology uses historical data to evaluate the effectiveness of the different models and uses them to make traffic management decisions in a simulation.

The next section of this paper discusses simulation models and machine learning in the railway domain. Section 3 describes the methodology with the results presented in Section 4. The concluding section discusses future work.

2. Simulation and machine learning in railway applications

There are many different simulation modelling approaches; railway simulations that can replicate operations in a high level of detail are usually microscopic, synchronous and stochastic [1]. Microscopic models replicate a railway network to a low level of detail; for example, the track may be represented to the nearest metre. Synchronous railway simulations are capable of running trains simultaneously across the network. Stochastic models include random variables to account for uncertainty in, for example, run- or dwell-times.

Railway simulations typically require large amounts of data input to represent the network. They also require rules that control interlocking and the management of traffic. The output of such a simulation model may be metrics, such as total minutes delayed. All models employ some abstraction and simplification, but the rules governing how traffic is managed in a railway simulation need to be sufficiently realistic so that the outputs are meaningful.

Machine learning (ML) is a subfield of artificial intelligence. As with simulation modelling, there are many branches and types of ML, one of which is supervised ML, whereby the models are trained to learn rules that will map a set of input values to outputs. In a railway context, there are examples of supervised machine learning being used to predict the travel times of trains between locations [2]. Relevant to the work in this paper, a supervised machine learning model in [3] mimics the decisions of signallers. The model takes the details of two trains in conflict as input and outputs which train will be allowed to pass and which will be delayed. Using a set of 331 test data records, the model correctly predicted which train was given priority for 327 records, an accuracy of over 98%.

Simulations and ML are two different modelling approaches, but there are examples of hybrid models that combine the two techniques outside the railway domain. One example is in the healthcare domain, where ML models are used to predict the travel speed of ambulances; the learned ML models are then used in simulations to model the travel durations of ambulances to emergencies and hospitals [4].

3. Methodology

3.1 Aims

This work aims to build machine learning models that can resolve conflicts between pairs of trains on a railway network; that is, to determine which train will have priority at a conflict location. In this work the models seek to replicate the most likely decisions taken by signallers, rather than aiming to find the optimal decision. The ability to model other traffic management actions such as re-routing or cancelling services are out of scope. The accuracy of the ML models will be compared to more straightforward modelling techniques and programmatic rules. The ML models will be used to control traffic management within a railway simulation model.

3.2 Case study area

This work will be demonstrated using a small section of the Havant train describer (TD) on the south coast of England, which is shown in Figure 1. The area comprises three junctions and six possible locations where conflicts may occur. The conflict locations are described as pairs of 'steps' from one signal to another and are listed in Table 1. There are three 'joining' conflicts (locations 1 – 3), where the conflicting routes are stepping into the same berths, and three 'crossing' conflicts (locations 4 – 6).

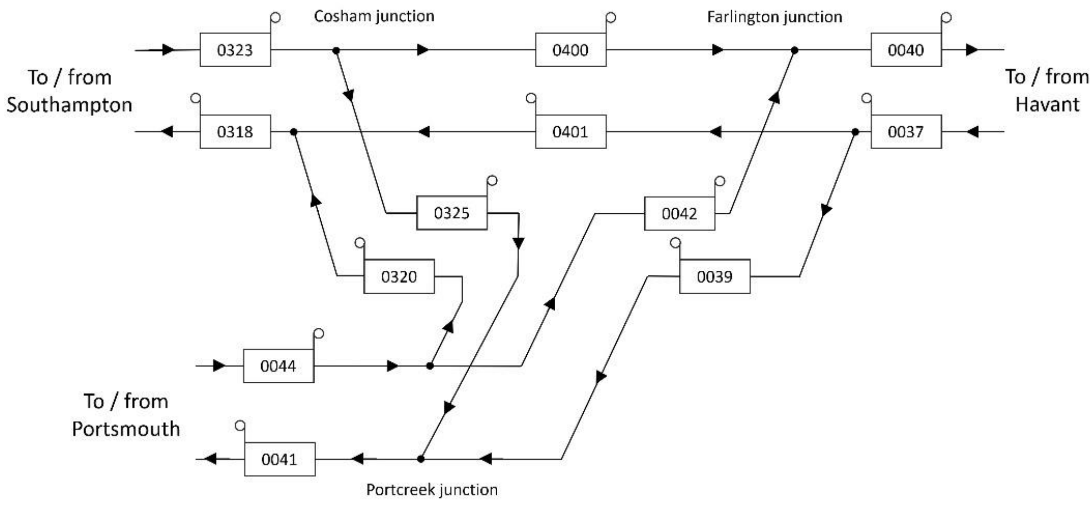

Figure 1: The junctions included in the case study

3.3 Data

The dataset for this work has been put together using Network Rail's open data feeds[1]. A process has been developed that creates a master dataset of train movements by matching historical train movements from Network Rail's train describer (TD) C-Class data to long- and short-term schedules. The TD data represent the times that trains move from and to berths, controlled by a main signal.

The dataset comprises passenger trains, freight trains and empty coaching stock (ECS). Data has been collected from 2018-04-06 to 2021-09-16 (inclusive) – a period of 1260 days. The data were mined to identify pairs of trains approaching the mutually exclusive steps listed in Table 1.

	Step 1		Step 2	
Conflict Location	**From berth**	**To berth**	**From berth**	**To berth**
1	HT_0042	HT_0040	HT_0400	HT_0040
2	HT_0320	HT_0318	HT_0401	HT_0318
3	HT_0325	HT_0041	HT_0039	HT_0041
4	HT_0037	HT_0401	HT_0042	HT_0040
5	HT_0044	HT_0042	HT_0325	HT_0041
6	HT_0323	HT_0325	HT_0401	HT_0318

Table 1: The conflict locations in the case study

3.4 Modelling process

Models are developed individually for each conflict location; they may be viewed as binary classification models where the output will be one or zero. For each conflict location, an output of one will correspond to the train moving towards Step 1 having priority. Conversely, an output of zero corresponds to the train moving towards Step 2 having priority.

The modelling process first consists of a period of exploration that experiments with a wide range of models. The records from the first 1170 dates in the dataset were used during this phase to create and validate different models and are referred to as the *training dataset*. Data from the last 90 days are referred to as the *testing dataset*. These data were held out during the exploration phase and were only used once the most promising machine learning models had been identified. These data were then used to evaluate the models. The number of records used for building and testing the models is shown in Table 2.

	Conflict Location					
	1	**2**	**3**	**4**	**5**	**6**
Number of records in the training dataset	34203	21528	49585	34040	83533	21166
Number of records in the testing dataset	2674	1883	4402	2320	6432	1617

Table 2: Number of records in the training and testing datasets for each conflict location

The model exploration phase considers three different model types: baseline, programmatic and machine learning (ML). The baseline models are computationally inexpensive and straightforward to understand. Any

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

more complex model must, at the very least, be capable of producing a higher accuracy than the baseline models. A range of baseline models was developed, two of which are presented here:

- Baseline 1: This model makes a constant prediction for each conflict location based on the direction of travel that is most often given priority.
- Baseline 2: The records are grouped by the berths each train is in, their destinations, the types of trains (passenger, freight or ECS) and the train scheduled to go first. A constant prediction is made for each group, depending on which trains are most frequently given priority.

Programmatic models are defined by conditional statements (if… then… else…). Like the baseline models, their logic can be simple to interpret. As with the baseline models, a wide range of programmatic models were considered, three of which are presented here:

- Programmatic 1: The train scheduled to travel across the conflict first will have priority.
- Programmatic 2: A passenger train will have priority over a freight train or ECS if it is at least one signal closer to the conflict, and express passenger trains will have priority over non-express trains, again if it is at least one signal closer to the conflict. Otherwise, the trains will travel in scheduled order.
- Programmatic 3: A train that is stopping between its current location and the conflict location will give priority to a train that is not stopping. Otherwise, the trains will travel in their scheduled order.

A classification algorithm known as gradient boosting [5] was used to build ML models for each conflict location. The disadvantage of ML models is that they are more difficult to interpret than either the programmatic or baseline models. However, they can exploit more complex relationships in the data. The model exploration phase is more intensive for the ML models than the other modelling approaches, as many hyperparameters can be adjusted. A randomised search was used to explore the range of possible hyperparameters, and the hyperparameters that produced the highest accuracy may vary between conflict locations. The machine learning models output a value between zero and one, and the values are then rounded so that all the output predicts zero or one.

3.5 Simulation model

A discrete event simulation model has been developed using the Python programming language. It may be classed as a mesoscopic model, as it does not include the same level of detail as many commercial models. It is capable of modelling on a deterministic or stochastic basis. The simulation is designed to be integrated with the range of models described in the previous section. Therefore, it is possible to compare the impact of using baseline, programmatic and machine learning models within the simulation.

A week's worth of historical data between 2021-08-09 and 2021-08-15 was used to create input into the simulation model, referred to as the *simulation dataset*. The date range was selected as it contained good quality data and was a subset of the date range of the testing dataset. A small number of trains had to be removed due to incomplete data (for example, not being able to match a train's movements to a known schedule). The network area modelled in the simulation was larger than that shown in Figure 1 as it was necessary to simulate the trains as they approached the conflict locations from some distance away.

4. Results

4.1 Traffic management models

The agreement between the testing data set and the model output is shown in Table 3. The results show that the gradient boosting models can achieve the highest accuracies. However, the accuracies of many of the other modelling approaches were not significantly lower than the gradient boosting models. The gradient boosting models were sensitive to the values of their hyperparameters, and it was possible to create variations of the models that would produce poor results; this meant that the baseline and programmatic models were essential for judging the ML models.

The results show that each conflict location is unique, and some locations are more predictable than others. The so-called 'joining' locations (1, 2 and 3) achieve higher levels of agreement than the 'crossing' locations. This observation may indicate that deviating from the scheduled order of services can have a significant impact at joining locations. For example, for two trains approaching a joining location close in time, an express service would be scheduled ahead of slower services that stop regularly; reversing the order of such services at joining locations could have a detrimental impact on the remainder of the express train's journey.

	Conflict Location					
Model Name	**1**	**2**	**3**	**4**	**5**	**6**
Baseline 1	61.3%	92.4%	81.5%	51.7%	78.9%	87.1%
Baseline 2	**93.3%**	**96.0%**	**89.6%**	**79.9%**	**89.4%**	**87.2%**
Programmatic 1	<u>**95.0%**</u>	94.5%	86.7%	78.4%	86.5%	76.0%
Programmatic 2	92.3%	94.2%	76.3%	**80.7%**	79.0%	79.0%
Programmatic 3	85.7%	**94.8%**	**88.5%**	77.8%	**88.2%**	**83.9%**
Gradient Boosting	<u>**95.0%**</u>	<u>**96.7%**</u>	<u>**93.6%**</u>	<u>**83.9%**</u>	<u>**92.5%**</u>	<u>**90.9%**</u>

Table 3: Accuracy of each model (given to one decimal place) when applied to the testing data set; for each conflict location, the highest result for each model type is in bold, and the highest overall result is underlined

Traffic management model	**Number of trains with inaccurate travel times**	**Percentage of all trains**
Baseline 2	112	4.3%
Programmatic 1	135	5.2%
Best Programmatic	134	5.2%
Gradient Boosting	105	4.1%

Table 4: Simulation results by traffic management model

4.2 Simulation model

The simulation model was run deterministically in a series of experiments using different models for determining traffic management actions. The simulation used historical run-times of the trains to determine when the trains moved between berths apart from when an adverse signal prevented a train from moving, in which case the train would take longer to travel in the simulation than it did in reality. Therefore, any experiment that modelled the signalling actions with 100% accuracy would also find perfect agreement between the travel times of the trains in the simulation with the historical travel times. Table 4 shows how many trains in each experiment had an inaccurate travel time compared to reality: a lower number represents more agreement with the historical data. The 'Best Programmatic' model followed the programmatic logic that provided the highest accuracy for each conflict location.

The gradient boosting traffic management model provided the best agreement with the historical data. However, none of the experiments produced significantly poor results. This is likely due to the selection of the particular junctions and the date range chosen to model.

5. Conclusions and further work

This paper has presented various approaches to modelling the resolution of conflicts in railway traffic and has shown that these can be used to take signalling actions in a simulation model. Machine learning models were shown to have higher levels of agreement with reality than programmatic logic. However, no modelling approach was entirely accurate, and, for some conflict pairs, uncertainty remains about the action a signaller would actually take.

The uncertainty may be due to a lack of information: additional data not identified in this work may improve the decision-making of the models. Alternatively, there may be some uncertainty that cannot be easily modelled. For example, signallers with different levels of experience may take alternative actions when faced with the same scenario. Similarly, there may be some conflicts where the action taken is of little consequence, for example, if an amount of buffer time has been added to schedules at a conflict location. Human signallers may also make mistakes and schedule trains in non-optimal fashions.

Future work will consider whether supplementary data can improve the model accuracy and explore probabilistic versions of these models that will capture this uncertainty in stochastic simulations. Machine learning models can be calibrated so that the quality of their probabilistic output is improved, and being able to model the uncertainty in traffic management decisions may provide an additional benefit compared to programmatic models. Similar models will also be created for other traffic management decisions, such as re-routing trains, and the case study area will be expanded to a more extensive section of the network.

Furthermore, the work will be carry on to generate traffic management rule bases using optimisation techniques. For this, some objective function would be required, such as total minutes delayed. The simulation model presented in this study could then be used to compare the signalling actions learned by the machine learning algorithms with an optimised rule base.

Acknowledgement

This research was funded by Network Rail and EPSRC through an Industrial CASE research studentship. The authors acknowledge the helpful discussions with Ben Ford and Lloyd Barson from Network Rail, and the use of the IRIDIS High Performance Computing Facility at the University of Southampton.

References

[1] Medeossi, G. and de Fabris, S., "Simulation of Rail Operations", Handbook of Optimization in the Railway Industry, pp 1-24, 2018, https://doi.org/10.1007/978-3-319-72153-8_1

[2] Kecman, P. and Goverde, R. M. P., "Adaptive, data-driven, online prediction of train event times", International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), 2013, https://doi.org/10.1109/ITSC.2013.6728330

[3] Dündar, S. and Şahin, İ., "Train re-scheduling with genetic algorithms and artificial neural networks for single-track railways", Transportation Research Part C: Emerging Technologies, 2013, https://doi.org/10.1016/j.trc.2012.11.001

[4] Olave-Rojas, D. and Nickel, S., "Modeling a pre-hospital emergency medical service using hybrid simulation and a machine learning approach", Simulation Modelling Practice and Theory, 2021, https://doi.org/10.1016/j.simpat.2021.102302

[5] Hastie, T., Tibshirani, R. and Friedman, J., "Boosting and Additive Trees", The Elements of Statistical Learning: Data Mining, Inference, and Prediction, pp 337-387, 2009, https://doi.org/10.1007/978-0-387-84858-7_10