Real-time Decision Making for Train Carriage Load Prediction via Multi-Stream Learning

Hang Yu[1], Anjin Liu[3], Bin Wang[1], Ruimin Li[2], Jie Lu[3][1111−2222−3333−4444], and Guangquan Zhang[3]

[1] Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, Australia
{Hang Yu,Bin Wang}@student.uts.edu.au
[2] Sydney Trains, Transport for NSW, Sydney, Australia
ruimin.li2@transport.nsw.gov.au
[3] Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, Australia
{Anjin Liu,Jie Lu,Guangquan Zhang}@uts.edu.au

Abstract. Real-time traffic planning and scheduling optimization are critical for developing low-cost, reliable, resilient, and efficient transport systems. In this paper, we present a real-world application that uses machine learning techniques to forecast the train carriage load when a train departure from a platform. With the predicted carriage load, crew can efficiently manage passenger flow, improving the efficiency of boarding and alighting, and thereby reducing the time trains spend at stations. We developed the application in collaboration with Sydney Trains. Most data are publicly available on Open Data Hub, which is supported by the Transport for NSW. We investigated the performance of different models, features, and measured their contributions to prediction accuracy. From this we propose a novel learning strategy, called Multi-Stream Learning, which merges similar streams to boost the training data size with the aim of achieving lower generalization errors. We have summarized our solutions and hope researchers and industrial users who might be facing similar problems will benefit from our findings.

Keywords: Multi-stream Learning · Machine Learning Application · Regression · Sydney Trains.

1 Introduction

Overcrowding in public transportation places a major strain on the quality of service and causes delays during peak hours [1]. Passenger load prediction for public transportation can benefit society from two perspectives. From the service provider perspective, an accurate load estimation can help traffic planning and scheduling, reducing the cost and increasing the reliability of transport systems. From the travelers' perspective, knowing the level of crowding and enjoying the trip could be more important than having the shortest travel time, or the least number of interchanges [2].

Being able to include reliable information on crowdedness, requires knowledge about current and future levels of passenger loads. The availability of an increasing amount and complexity of data describing public transport services allows us to explore the detection methods and analysis of different phenomena of public transport operations now better than ever [3, 4]. Transport NSW provides the possibility to use open data for occupancy information [5]. However, such information is only available for a trip at a current stop but cannot provide predictions for the next service, making it hard to incorporate it into real-time recommendation models. More importantly, if a disruption or a special event occurs, the occupancy information will become misleading. This is a typical data stream learning problem under a dynamic learning environment [6, 7].

Train carriage load prediction aims to estimate the number of passengers in a carriage when the train departs the platform. This information can be used by station crew to manage the flow of passengers while they are waiting at the platform, thereby improving boarding and alighting efficiency. Carriage load prediction also has the potential to provide the trip planner with a forecasting capability to advise passengers to avoid overcrowding.

Since there are eight carriages for a train, the prediction task can be considered as a multi-output problem. However, some of the carriages share the same passenger load patterns, which makes the problem more interesting. From the perspective of generalization error, that is, the more training data a model fits in, the lower the generalization error the model will have, we consider that vertically merging the similar streams and remove duplicated features would improve the overall performance. We label the task of merging similar streams and removing duplicated features as Multi-stream Learning.

In summary, the main contribution of this paper are as follows:

- A real-time train load prediction model to handle Waratah, Non-Waratah trains and be capable of handling disruption situations.
- A discussion of the findings in the model selection, and feature engineering during the development of this application.
- A discussion of the findings in the multi-stream learning strategy.

The rest of this paper is organized as follows. Section 2 details the background and objectives of the study. Section 3 describes the methodology proposed to solve the learning task. Section 4 summarizes the obtained results. Section 5 concludes the paper.

2 Backgrounds and Objectives

2.1 Sydney Trains

As the operator of the suburban passenger rail network, Sydney Trains serve the city of Sydney, New South Wales, Australia. The network is a city-suburban hybrid railway system with a central underground core, 813 kilometers of track covering eight lines and 175 stations [8]. It has subway-comparable train frequencies of every three minutes or shorter at the underground core, 5-10 minutes off-peak at most major and inner-city stations, and 15 minutes off-peak at most minor stops. On weekdays, trains are more frequent during peak hours, while frequency decreases on weekends [8]. The network is managed by the Department of Transportation, NSW, and is part of its Opal ticketing system. In 2018-2019, the network carried out roughly 377.1 million passenger journeys.

As Sydney Train acknowledged, this study is the first to employ machine learning techniques to implement real-time passenger load forecasting. It will provide the public transport planning and management with a better tool, and play an important role for the development of Intelligent Transport Systems in Sydney in the near future.

2.2 Waratah Train Set

Reliance Rail designed and built the Waratah carriages. The initial order for series 1 Waratahs was the largest rolling stock order in Australia's history. These *A set* set carriages make up around half of the Sydney Trains fleet and replaced two-thirds of the *S set* carriages [9]. Waratah trains have many advanced features, including a portable boarding ramp, wheelchair-accessible entrances, symbols on accessible entrances, real-time carriage loading sensor, and much more. [9]. The specifications of Waratah trains are shown in Table 1.

Table 1: The specifications of Waratah train set [9]. The seating capacity is critical for noise and outlier identification. For example, if we assume the average weight of a passenger is 75kg, the maximum load of a carriage would be $75kg \times 118 = 8850kg$. If a carriage load exceeded this threshold, it might be due to a faulty sensor.

Specifications	Measurment
Seating Capacity	101-118 Passengers
Weight	48-51 tons (approx.)
Length	19393-20,000mm
Width	3,035mm
Height	4,410mm

At current stage, Sydney Trains has over 60% trains operated by Waratah Trains. This means there is around 40% trains cannot provide real-time carriage loading information for prediction. The coverage of Waratah trains is important to the prediction accuracy and the actual departure loading information is the only available true label for evaluations [10].

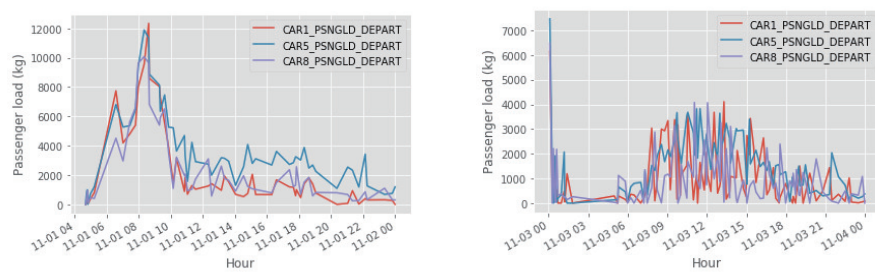
2.3 Data Source

Transport for NSW is opening up its data to provide Open Data Hub, a platform for developers, entrepreneurs, and data analysts to form a united community to create innovative solutions for smart transportation [5]. As one of the leaders in the open data space, the NSW Government has seen its related apps downloaded millions of times. With Open Data Hub, researchers and developers can create the next generation of real-time transportation management systems. They have now serviced an open data community of more than 30,000 users [5].

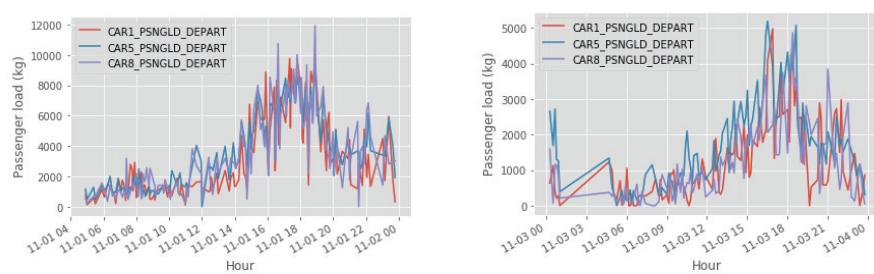
The data is well documented and access is via an application programming interface (API). Open Data Hub has an application management system for users to organize their API keys, which they can use to access real-time data an via HTTP GET request. An open source data such as this helps researchers to contextualize our solutions and will be beneficial to further improvement of related learning tasks.

2.4 Application Objectives

The transport network can be divided into nine lines serving different areas. There are two types of trains, Waratah and non-Waratah, running on the network. One of the important differences between the two is whether or not a carriage is equipped with an occupancy weighting system. This means only the data collected from Waratah trains have carriages' loading weight when arriving or departing at a platform.



(a) Origin station: Central; Destination station: Schofields; Current station: Redfern; Platform: RD05; Date: 2019-11-01 and 2019-11-03.



(b) Origin station: Central; Destination station: Schofields; Current station: Central (the origin station); Platform: CE18; Date: 2019-11-01 and 2019-11-03.

Fig. 1: A demonstration of the passenger load patterns during different time periods with different locations.

Fig. 1a shows loading examples at platform RD05 of Redfern station on 2019-11-01 (Friday) and 2019-11-03 (Sunday). Note that each train has eight carriages, numbered 1-8 from head to end, and the weights of arrival and departure are recorded. We select the departure weights of Carriages 1, 5, and 8 as examples. In the left figure, there is a distinct peak phenomenon. During peak hours, around 8:00 am, the weight difference among three carriages is minimal, while for off-peak hours, Carriage 5 is much heavier. This may be attributed to the passengers' riding habits, preferring to choose the middle carriage to ride. In the right figure, there is no clear peak time on Sunday, but there is still a large passenger flow at midnight.

Fig. 1b shows the cases in Central station. As the point of origin, its peak times move from morning to afternoon. The difference in flow trend between Friday and Sunday is not very prominent, except that the passenger flow on Friday register as significantly higher.

In summary, several factors have a significant impact on passenger load, including calendar factors (hour, day of the week), position of the train carriage, and location of the platform, to mention only a few. This can generate a high variability in prediction.

3 Carriage Load Forecasting Methodology

3.1 Models Selection

We evaluate five models for train carriage load forecasting on all stations in Sydney regions. These range from the classical baseline model to advanced machine learning models. Since it is difficult to find in the literature what is the most suitable model for our task, we choose the most commonly used models for solving forecasting problems:

1. Last Value (Baseline): The simplest forecasting is to forward the last observed load on the train to the next one at the same station. More specifically, this model will predict the departure load of a train carriage as the same as its arrival load.
2. Linear Regression: This is the most intuitive regressor solution, able to optimize the weight of each features to link the feature to a target variable function. In our case, the features are the train status, and the target variable is the carriage load.
3. Gradient Boosting (LightGBM): LightGBM is an open source non-linear regressor model that produces prediction models in the form of an ensemble of weak decision trees and weighting average prediction by boosting. It can handle both numerical values and category features.
4. Random Forest: This regressor model produces a prediction model in the form of an ensemble of weak decision trees where the prediction is given by bagging. This model is like gradient boosting, but it does not iteratively update the cost for each base learner.

5. Neural Network: Our final model is a universal approximator, one that can capture any form of relationship in data [11, 12]. In this study, both a recurrent neural network and convolutional neural network are inapplicable. Therefore, we deploy a simple feed forward neural network to model the relationship between the input and output.

3.2 Feature Engineering

The forecasting of train passenger load when a train departs a station is an essential and challenging task, largely due to the influence of several factors [5] which are related to transport demand and supply and can be summarized as follows:

- Calendar factors, including the day of the week, time of the day, peak hours or non-peak hours, and the month of year. It is worth mentioning that public and school holidays are also influential. This information is included in the peak hour and non-peak hour features.
- Spatial factors such as the local information about the area where the station is located. For example, residential, employment, commercial, and leisure. Since this information will not change dramatically in the days of our life, we consider it is dependent with the station name, that is, for a given station, its residential information can be automatically considered by the model.
- Transport network factors, such as the train timetable which may vary slightly if there is an increase in local population or the development of a new public transport interchange hub. Track work and emergent service disruption might also contribute to the carriage load.
- External factors, such as the climatic conditions, special events like music concerts, Rugby League, or New Year's Eve events. Collecting this information is very demanding and could be just a minority record in the data. Therefore, we have not taken them into account.
- Sensors that collect the flow of passengers boarding or alighting the trains and which could provide data subject to different interpretations depending on the context. For example, null value has different meanings, such as no passengers, missing value, or sensor malfunction.

In addition to these factors, generated when a train stopped at a station, we also manually create two features to reflect the trend of the carriage load in the context of a trip.

- Contextual moving average consists of using the average load of trains committed on the same day type and time slice in the history. The moving average shows the passenger flow's historical pattern, and is an important indicator in reflecting the increasing or decreasing of the carriage load.
- Last stopped Waratah train information is adopted to provide information regarding to the most recent traffic condition.

The records of these factors are the features of the data. The actual carriage loads measured during operation is the target variable. The purpose of the features is to understand the context of the forecasting problem more readily and then to model it.

3.3 Multi-stream Learning

This section introduces a novel concept that classifies the forecasting task as a multi-stream learning problem. The fundamental idea of multi-stream learning is to merge similar streams to increase the training set size, thereby reducing the model's generalization error.

Some researchers translate the multi-stream learning into multiple-output learning [13]. The point of departure in multi-output learning is the idea that the multiple outputs might have a structure representing the relationship between outputs [14]. Traditionally, based on the structure of outputs, multi-output regression can be divided into global and local methods [15]. Global methods use the idea of classical learning algorithms for single output, i.e., to predict the multiple outputs as a whole. In contrast, the idea of local methods is simpler, i.e., to decompose the multi-outputs into multiple single outputs, then use classical learning algorithms for single output to predict each output [16].

However, in our opinion, the structure of outputs might be neither local nor global. For example, assuming there are three data streams and each stream with one output attribute $\{Y_1(t), Y_2(t), Y_3(t)\}$, where the attribute Y_1 has a relationship with Y_2 attribute, but the attribute Y_3 is independent of Y_1 and Y_2, the structure of these three output attributes should be presented as $\{(Y_1, Y_2), (Y_3)\}$. Hence, we use the modularity property of data streams to distinguish them from the local and global methods. In our proposed method, we explore the similarity between each carriage load, then carriage loads with high similarity are built as a modularity. For example, if we observed the passenger load of carriage 1 is similar to carriage 8, and the rest of the carriage loads are similar, there would be two modularities {Car1, Car8} and {Car2, Car3, Car4, Car5, Car6, Car7}. Then, for each modularity, we learn a prediction model to boost the overall performance.

4 Experiment Evaluations

To evaluate the learning performance of different models, and to analyze the importance of different features, we use the Waratah train operation records of Nov 2019 and Dec 2019. Most of the data is available from Open Data Hub. Only the occupancy feature is different. The carriage loading data provided by Sydney Trains is in kilograms but not in levels as provided in Open Data Hub. Therefore, we use mean absolute error (MAE) as the evaluation metric.

4.1 Experiment 1. Model Selection

The first thing we need to confirm is the base model. As mentioned in Section 3.1, in this experiment, we search out the models on the basic features with the lowest MAE. Then we finetune the model's parameters and evaluate it with different cost functions.

Experiment Settings The implementation of the models is as follows.

- Last Value (Baseline): The last value regressor has no specific settings. For a non-Waratah train, we use the mean of the training set as the prediction.
- Linear Regression: Linear regressor is implemented by scikit-learn LinearRegression module with intercept fitting and normalization.
- LightGBM: The parameters for LightGBM are Objective←Poisson; $num_{leaves} \leftarrow 20$, $learning_{rate} \leftarrow 0.05$; $num_{estimators} \leftarrow 1000$. The resets are set as default values.
- RandomForest: Random forest regressor is implemented by scikit-learn Random Forest module with $num_{estimators} \leftarrow 20$; $\max\_depth \leftarrow 10$.
- NeuralNetwork: Neural network regressor is implemented by scikit-learn MLPRegressor module with $hidden\_layer\_sizes \leftarrow 32$; *relu* activation function; *adam* optimizer; *adaptive* learning rate.

There is no specific assumption in these settings. We choose the configurations which are the most commonly used settings.

Evaluation Metric To quantitatively evaluate the performance, we use the Mean Absolute Error (MAE) of the carriage load as the evaluation metric, where the MAE is calculated as

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y - \hat{y}|$$

Since the carriage load information is given in kilograms, the MAE in our experiments indicates the difference between the estimated and the true values in kilograms.

Findings and Discussions From Fig. 2, we can glean that LightGBM had the best performance on all four tasks, followed by RandomForest, NeuralNetwork, LinearRegression and Baseline. Since we only used a very shallow neural network structure, future study on this regressor should be undertaken. But for the Baseline, LinearRegression, LightGBM and RandomForest, we are confident to conclude that LightGMB is the best one for our learning tasks, and it follows that we focus on evaluating it for the rest of our experiments.

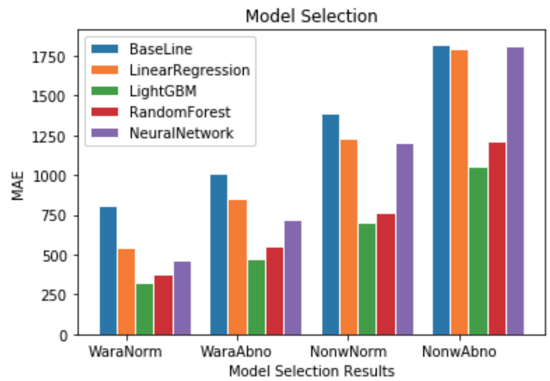


Fig. 2: Model selection result on the four tasks. WaraNorm: Waratah trains operate normally without disuptions; WaraAbno: Waratah trains operate abnormally with disruptions; NonwNorm: Non-Waratah trains operate without disruptions; NonwAbno: Non-Waratah train operate with disruptions.

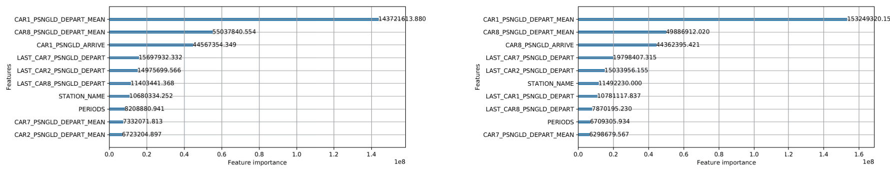
4.2 Experiment 2. Feature Engineering

In this experiment, we evaluate the contribution of the base features and our new proposed features (Historical Mean and Last Available Waratah Information). We also plotted the total gain of the top 10 features to represent their overall contribution. The experiment settings and evaluation metric are the same as we used in Experiment 1.

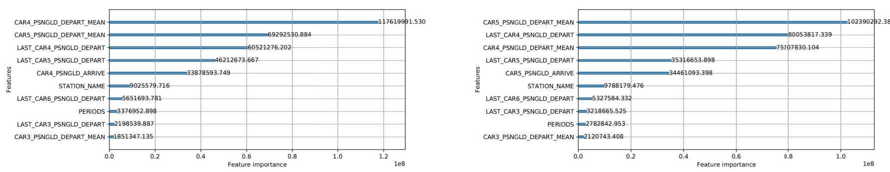
Table 2: The results of LightGBM model with different features. The historical mean is calculated based on the average departure loads in the previous week. The last available Waratah train information is searched based on the timestamp closest to current trains. There were 116,112 sampled instances for WaraNorm, 11,864 for WaraAbno, 174,440 for NonwNorm, 17552 for NonwAbno.

	WaraNorm	WaraAbno	NonwNorm	NonwAbno	Average
BasicFeatures	321.29	469.86	701.64	1057.39	574.54
HistorcialMean	317.12	498.57	601.88	964.48	514.62
LastAvailableWaratah	305.75	468.51	624.23	981.62	522.48
Mixture	**299.68**	**489.49**	**550.33**	**882.1**	**475.37**

Findings and Discussions Table 2 summarizes the average results of LightGBM with different features. Fig. 4. shows the importance of the top 10 features in terms of total gain for Carriages 1, 8, 4, and 5. The feature importance of Carriages 2, 7, 3, and 6 have similar patterns. In summary, both the historical mean and last available Waratah train contribute to the final predictions. An interesting finding of the feature importance is that for Carriages 1 and 8, the historical means are the top two most important features. We found a similar phenomenon in car 2-7, 3-6 and 4-5. Our interpretation of this is that it's due to



(a) Feature importance of the LightGBM for carriage 1 & 8 prediction.



(b) Feature importance of the LightGBM for carriage 4 & 5 prediction.

Fig. 3: The feature importance of the learning models on each carriage. The historical values of carriage 8 is beneficial to the load prediction on carriage 1. Similar phenomena happened on carriage 8.

the head and tail of the trains having similar passenger loading patterns along with the middle carriages. These findings inspired us to think about the multi-stream learning issue. That is, the possibility that merging Carriages 1 and 8 might achieve a better performance than considering them individually.

4.3 Experiment 3. Multi-stream Learning VS. Multi-output Regression

In this experiment, we evaluate the concept of multi-stream learning compared with multi-output learning. Since LightGBM does not support multi-output regression, we adopt the multi-output strategy in sklearn - fitting one regressor per target. The experiment settings were slightly changed, but the evaluation metric is the same as in the previous experiments.

Experiment Settings For multi-output regression, we built an independent regressor for each carriage with the same model configuration as introduced in Experiment 1. For multi-stream learning, we concatenate the data of Carriages 1 and 8 vertically and removed the duplicated carriage load information. We applied the same multi-stream learning strategy to Carriages 2-7, 3-6 and 4-5. As a result, we now have four independent regressors instead of eight, and the training data for each regressor is doubled. Since the fundamental idea of multi-stream learning is to merge similar streams to increase the training set size, thereby reducing the model's generalization error, we chose a different sample size to evaluate the benefits of multi-stream learning.

Findings and Discussions Table 3 shows the results of multi-stream learning and multi-output. Multi-stream learning always outperforms multi-output, and the average improvement increases as the size of available training data decreases. For sample size 25000, the improvement is 8.87kg; for 20000, the improvement is 11.03kg; for 15000, the improvement is 12.18kg, and for 10000, the improvement is 13.55kg. This finding strongly supports our assumption that multi-stream learning can boost the number of training data, thereby reducing the generalization error.

Table 3: The result of multi-stream learning with different sample size. The average improvement shows how the multi-stream learning enhanced the result compared to multi-output regressor. In this experiment, the multi-output regressor fits one regressor per target. This is a simple strategy for extending LightGMB to support multi-target regression.

Sample Size	25000		20000		15000		10000	
Learning Method	MulOutput	MulStream	MulOutput	MulStream	MulOutput	MulStream	MulOutput	MulStream
Car 1-8	386.90	382.14	392.52	386.15	395.32	386.98	397.46	391.44
Car 2-7	492.63	486.58	498.25	489.57	501.72	493.28	510.84	502.23
Car 3-6	513.15	505.01	518.86	510.44	520.38	512.51	531.32	518.40
Car 4-5	487.22	470.69	492.00	471.34	496.92	472.85	507.69	481.03
Average	469.98	461.10	475.41	464.37	478.58	466.40	486.83	473.28
Improvement	8.87		11.03		12.18		13.55	

5 Conclusion and Further Study

In this paper, we investigated train load forecasting with advanced machine learning models and the importance of the features that are part of the construction of these models. The obtained results have shown that machine learning models, and more particularly ensemble learning approaches such as LightGBM or RandomForest can address train load forecasting by using a combination between short- and long-term features that translate influencing factors. Furthermore, the multi-stream learning strategy of merging similar streams has proven its ability to deal with the issue of training sample augmentation. Future work will focus on experiments to enhance multi-stream learning performances through redesigning the merging indicator in terms of the training sample size.

Acknowledgement

This study was undertaken as part of the carriage load prediction project coordinated by Sydney Trains.

References

1. Z. Li and D. A. Hensher, "Crowding in public transport: a review of objective and subjective measures," *Journal of Public Transportation,* vol. 16, no. 2, p. 6, 2013.
2. P. Campigotto, C. Rudloff, M. Leodolter, and D. Bauer, "Personalized and situation-aware multimodal route recommendations: the FAVOUR algorithm," *IEEE Transactions on Intelligent Transportation Systems,* vol. 18, no. 1, pp. 92-102, 2016.
3. L. Heydenrijk-Ottens, V. Degeler, D. Luo, N. van Oort, and J. van Lint, "Supervised learning: Predicting passenger load in public transport," in *CASPT Conference on Advanced Systems in Public Transport and TransitData,* 2018.
4. J. Lu, A. Liu, Y. Song, and G. Zhang, "Data-driven decision support under concept drift in streamed big data," *Complex & Intelligent Systems,* vol. 6, no. 1, pp. 157-163, 2020.
5. TfNSW. (2020, 16 July). *Tfnsw Open Data Hub And Developer Portal.* Available: https://opendata.transport.nsw.gov.au/user-guide
6. A. Liu, J. Lu, and G. Zhang, "Diverse Instance-Weighting Ensemble Based on Region Drift Disagreement for Concept Drift Adaptation," *IEEE Transactions on Neural Networks and Learning Systems,* 2020.
7. A. Liu, J. Lu, and G. Zhang, "Concept Drift Detection via Equal Intensity k-Means Space Partitioning," *IEEE Transactions on Cybernetics,* 2020.
8. TrainWiki. (2020, 16 July). *Sydney Trains.* Available: https://nswtrains.fandom.com/wiki/Sydney_Trains
9. TfNSW. (2020, 16 July). *Waratah Trains.* Available: https://transportnsw.info/travel-info/ways-to-get-around/train/fleet-facilities/waratah-trains
10. A. Abadi, T. Rajabioun, and P. A. Ioannou, "Traffic flow prediction for road transportation networks with limited traffic data," *IEEE transactions on intelligent transportation systems,* vol. 16, no. 2, pp. 653-662, 2014.
11. J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for city-wide crowd flows prediction," in *Thirty-First AAAI Conference on Artificial Intelligence,* 2017.
12. F. Toqué, E. Côme, M. K. El Mahrsi, and L. Oukhellou, "Forecasting dynamic public transport origin-destination matrices with long-short term memory recurrent neural networks," in *2016 IEEE 19th international conference on intelligent transportation systems (ITSC),* 2016, pp. 1071-1076: IEEE.
13. D. Chow, A. Liu, G. Zhang, and J. Lu, "Knowledge graph-based entity importance learning for multi-stream regression on Australian fuel price forecasting," in *2019 International Joint Conference on Neural Networks (IJCNN),* 2019, pp. 1-8: IEEE.
14. W. Chung, J. Kim, H. Lee, and E. Kim, "General dimensional multiple-output support vector regressions and their multiple kernel learning," *IEEE transactions on cybernetics,* vol. 45, no. 11, pp. 2572-2584, 2014.
15. D. Kocev, C. Vens, J. Struyf, and S. Džeroski, "Tree ensembles for predicting structured outputs," *Pattern Recognition,* vol. 46, no. 3, pp. 817-833, 2013.
16. H. Borchani, G. Varando, C. Bielza, and P. Larra?aga, "A survey on multi-output regression," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,* vol. 5, no. 5, pp. 216-233, 2015.