

Weighted Ensemble Methods for Predicting Train Delays

Mostafa Al Ghamdi(✉), Gerard Parr, and Wenjia Wang

School of Computing Sciences, University of East Anglia, Norwich, UK
{m.al-ghamdi,G.Parr,Wenjia.Wang}@uea.ac.uk

Abstract. Train delays have become a serious and common problem in the rail services due to the increasing number of passengers and limited rail network capacity, so being able to predict train delays accurately is essential for train controllers to devise appropriate plans to prevent or reduce some delays. This paper presents a machine learning ensemble framework to improve the accuracy and consistency of train delay prediction. The basic idea is to train many different types of machine learning models for each station along a chosen journey of train service using historical data and relevant weather data, and then with certain criteria to choose some models to build an ensemble. It then combines the outputs from its member models with an aggregation function to produce the final prediction. Two aggregation functions were devised to combine the outputs of individual models: averaging and weighted averaging. These ensembles were implemented with a framework and their performance was tested with the data from an intercity train service as a case study. The accuracy was measured by the percentages of correct prediction of the arrival time for a train and correct prediction within one minute to the actual arrival time. The mean accuracies and standard deviations are 42.3%(±11.24) from the individual models, 57.8%(±3.56) from the averaging ensembles, and 72.8%(±0.99) from the weighted ensembles. For the predictions within one minute of the actual times, they are 86.4%(±14.05), 94.6%(±1.34) and 96.0%(±0.47) respectively. So overall, the ensembles significantly improved not only the prediction accuracies but also the consistency and the weighted ensembles are clearly the best.

Keywords: Ensemble · Train delays · Machine learning

1 Introduction

Train delays are a major problem for train operating companies and passengers. In the UK, in the last few years, the Public Performance Measure (PPM) for train services has been continuously declining - from over 91% in 2013–14 to 85.6% in Q3 of 2018–19 [18], although the entire rail industry has been working extensively in trying to improve their performance. There are many factors that can cause initial delays, which include signalling issues, bad weather, damaged equipment, breakdowns, construction works, accidents and disruptions to the

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12249, pp. 586–600, 2020.
https://doi.org/10.1007/978-3-030-58799-4_43

flow of operations [17]. Once an initial delay, called a primary delay, occurred, it can cause a chain of knock-on delays to other trains [11] and those delays are called reactionary delays. Given that the number of train passengers has been consistently increasing, the train operating companies have to run more train services on the current rail networks to meet the demand, which means that the rail networks are running at almost their full capacity, that is, the trains are closely scheduled to run one after another on a rail network with a minimum allowed interval. As a consequence, because there is a very little buffer space and time in the rail network for tolerating any disruption, then even a small primary delay can cause many reactionary delays and huge disruptions to the train services and a great deal of inconvenience to the passengers.

It is therefore essential for train controllers to have some systems to help predict train delays as earlier as possible so that they can take appropriate actions to either reduce or prevent further delays. This paper presents a machine learning ensemble method that combines different types of predictive models to improve the accuracy of delay prediction.

An ensemble is built with multiple models generated by using a variety of machine learning algorithms from the data of historical train services and weather data, and an aggregation function is employed to combine the predictions of individual models to produce a final prediction. In this study, we built heterogeneous ensembles and devised a weighted aggregation function for producing the final output of an ensemble. A framework has been built to implement these ensembles and the weighted aggregation function and tested in a case study on an intercity train service journey. The accuracy is measured with the percentage of correct predictions and predictions within 1 min of the actual arrival time.

The rest of this paper is organised as follows: Sect. 2 briefly reviews the related work, Sect. 3 describes in detail the methodology and construction of the ensembles. Section 4 presents the experiment design and results including a discussion of their implication. Finally, Sect. 5 draws the conclusions and gives suggestions for future work.

2 Related Work

Various methods have been developed for predicting train delays, including conventional regression methods and Stochastic approaches, and machine learning-based methods, such as Bayesian Networks, Support Vector Regression, Artificial Neural Networks and Deep Learning Neural Networks.

Bayesian Networks have been used by Lessan et al. [12] to predict train delays on the Chinese rail network, and also by Corman and Kecman [4] on the Swedish rail network with historical data.

Support Vector Machine (SVM) is another method studied for the prediction of arrival times when they are treated as a classification problem. Some researchers in China [24] used it to predict bus arrival times, while Markovic et al. [13] used SVM to identify connections between delays and railway network

qualities. This work focuses on avoiding and anticipating delays, particularly as finding any connections between the two could enable railway staff make use of learned choices to decrease delays. They also discussed two further potential methods - hybrid simulation and machine learning and multiple regression.

Treating the prediction of train delays as a regression problem, [2] used Support Vector Regression (SVR) for the American freight system and showed that the mean absolute error decreased by 14% from a baseline of historical running times, although no comparisons with other models were conducted. Gaurave and Srivastaz [8] provided the state of the zero shot Markov model for predicting the train delay process. This model was based on the train delays on the Indian network, which carried over eight billion passengers. The techniques were linked with the regression technique of the data modeling and it was reported an to be an efficient algorithm for the estimation of train delays and problems for the transport network. The main problem with modeling train delays is based on the volume of data and the algorithm required to mine the huge quantity of information. The algorithm can be applied with ensemble regression to solve the train delay problems.

Artificial Neural Networks (ANN) have also been used by Yaghini et al. [23] to predict delays on the Iranian passenger rail network, while Oneto et al. [16] tested the use of this model in Italy, and involved many detailed features, like displaying the weather conditions and the tracking ability of other trains on the network. Their work focused on the process of using machine learning algorithms and consists of kernels, ensembles and neural networks. Oneto et al. [16] use the Random Forest approach. They set the number of trees to 500 and used the ensemble technique, which will also be used in this research because, in comparison to classification/regression, they results show it performs better. A back-propagation Neural Network was used by Hu and Noch [9], who employed a genetic algorithm to improve the training of the model. Prediction performance improved in this case, but the training time was extended.

Wen et al. [22] also compared the Random-Forest Model to a simpler Multiple Linear Regression Model and found the Random Forest Model to have the better levels of prediction accuracy, compared to a simple Multi Linear Regression Model. Their finding is consistent with that from Oneto et al. [16], which is, that Random Forest outperforms other approaches, suggesting it to be the best algorithm for predicting delays.

Extreme Learning Machines (ELM), Shallow and Deep, were used by Oneto et al. [17] because ELM can learn faster than those which use traditional learning algorithms, which may not be fast enough, and they generalise well [10]. It is also more appropriate for use with big data than those which use univariate statistics because the model adapts and gets better when is fed with external data [17].

One recent study provide valuable insights into the use of combination of machine learning and other approaches to predict train delays in Germany. Nair et al. [14] developed a prediction model for forecasting train delays for the nationwide Deutsche Bahn passenger network, which runs approximately 25,000 trains per day. The data sources provided a rich characterisation of the network's

operational state, some of which was collected using track-side train passing messages to reconstruct current network states, including train position, delay information and key conflict indicators. The first model used was a statistical ensemble model comprised of two statistical models and a simulation based model. A context-aware Random Forest model was used as the first model. This had the capacity to account for current headways, weather events and information about work zones. Train-specific dynamics were accounted for by a second, kernel regression model. Thirdly, a mesoscopic simulation model was applied to account for differences in dwell time and conflicts in track occupation. Nair et al. [14] model demonstrated a 25% improvement in prediction accuracy and a reduction of root mean squared errors of 50% in comparison with the published timetable. The strength of their system was their use of ensembles which, as expected, showed was superior to constituent models. However, while the work improved the accuracy of a general prediction model, it did not consider state-dependent models which might have given further improvements. Their model was also sensitive to such hyper-parameters as outlier thresholds.

To sum up, these studies showed that machine learning models are capable of predicting train delays but they are generally limited to the rail networks where they were developed for. The motivation of this research is to develop a machine learning ensemble that will combine multiple models generated from different normal learning algorithms to enhance accuracy and reliability to help improve the performance of UK train services.

3 Weighted Ensembles for Predicting Train Delays

A train delay prediction problem can be formulated as follow. For a given train service journey, it should contain several stations, i.e. $J = \{S_1, S_2, ..., S_i, ..., S_{N-1}, S_N\}$ including the origin, i.e. the departure station S_1, the destination station S_N, and several stations S_2 to S_{N-1}, along the rail track where a train stops for passengers getting on or off the train. Along the rail track, there could be a set of n trains: $T = \{T_1, T_2, ..., T_j, ..., T_{n-1}, T_n\}$, running in accordance with their timetable.

Thus, the problem of train delay prediction is that for a given train T_i that has just departed from a Station S_i, we want to predict its arrival time at the next station S_j, and also at all the remaining stations of the journey. To complete this prediction task, we need to devise a suitable modelling scheme, which will be described in detail in the next subsection.

3.1 Delay Prediction Modelling Scheme

Firstly, instead of predicting the actual arrival time of a given train T_i at its next station, we convert it to predict the difference between the planned and actual arrival time at a station. So let t_{pa} represent the planned arrival time on the timetable for train T_i at an intended station $S_j \in J$; t_{aa}, the actual arrival time of that train at that station. The time difference, Δt, between the timetabled

arrival time and the actual arrival time is calculated by the following equation, which is taken as the target variable y.

$$y = \Delta t(T_i, S_j) = t_{aa}(T_i, S_j) - t_{pa}(T_i, S_j) \tag{1}$$

The predicted arrival time for train T_i at Station S_j can then be derived by $S_j(T_i) = t_{pa} + \Delta t$. It should be noted that when Δt is positive, it means a train is delayed by this Δt amount of time, and when it is negative, it means that a train arrives early by Δt time. This predicted time will be taken, together with other variables, as the inputs to the next model for predicting the arrival time of a following station. Figure 1 shows the prediction modelling scheme for any train running on the rail track of the chosen journey.

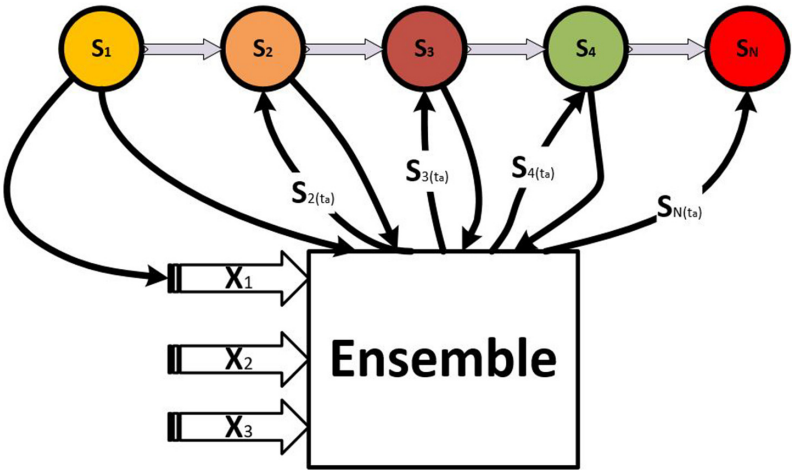

Fig. 1. Illustration of the delay prediction modelling scheme with machine learning ensembles.

The features used to make these predictions include the running time, the weather conditions and whether or not it is an off-peak service or during holiday periods. With these features, different models are trained as candidate models for building heterogeneous ensembles. Two different functions are used to combine the predictions of the individual models in an ensemble, which are averaging and weighted averaging, which are described in detail later.

3.2 Heterogeneous Ensembles

Two types of ensembles can be constructed in terms of types of models used as their member models: homogeneous and heterogeneous ensembles. A homogeneous ensemble is built with the models of same type, e.g. decision trees only or neural networks only; whilst a heterogeneous ensemble is built using models

of different types, e.g. a mixture of decision trees and neural nets. In this study, we choose to build heterogeneous ensembles because they have been shown to superior to the using either a single model or a homogenous ensemble [7,19].

Building Heterogeneous Ensembles: This paper presents a heterogeneous ensemble for train delay prediction because the complexity of the problem can overwhelm single models. Using ensemble methods has two advantages. The first is that, based on previous research [1] it is to be expected that an ensemble will outperform individual models [3,20]. Secondly, an ensemble offers a high level of reliability [21]. Furthermore, studies [1] comparing the effectiveness of heterogeneous and homogenous ensembles demonstrated that heterogeneous ensembles are generally more accurate and more reliable, which is particularly more important in a critical industrial application such as train delay prediction, where more consistent and robust predictions out-weight the absolute prediction accuracy as long as the accuracy is within a tolerable limit, such as within one minute. So, in this study, we chose to build heterogeneous ensembles for predicting train delay.

We construct a heterogeneous ensemble using more different types of models because they are likely to be more diverse than the models of the same types. We therefore chose sixteen different learning algorithms to generate candidate models. These were: Random Forest, Support Vector Regression, Linear Regression, Extra Trees Regressor, Multi-Layer Perceptron, Gaussian process regression, LassoLars, ElasticNet, Logistic Regression, Ridge, Gradient Boosting Regressor, Lasso Regression, Kernel Ridge Regression, Bayesian Ridge, Stochastic Gradient Descent, AdaBoost Regressor.

For a given training dataset each algorithm is used to generate a model. Thus 16 models are generated as the candidates to be selected by some rules for forming an ensemble. For comparison, we also build an ensemble by using all the candidates without any selection.

Then two different decision-making functions - simple averaging and weighted averaging (details given in the next subsection) are derived to combine the outputs of the member models in these two heterogeneous ensembles to produce a final output. Because of their different decision-making functions their ensembles are named as Averaging Ensemble (AE) and Weighted Ensemble (WE).

An Ensemble Framework: A framework for constructing heterogeneous ensembles has been developed, as shown in Fig. 2, with 5 components: Feature Extraction, Data Partition, Model Generation, Model Evaluation and Selection, and Decision Combination.

Feature Extraction: Two types of input data, i.e. train running data and weather data, were collected and used in this study. The basic form of the train running data is provided in a format likes a train timetable, that is, for each train, it lists the planned departure and actual departure times, and the planned arrival and the actual arrival times for each stop station along the service journey. From this format of raw data, we need to transform it into a structured representation by extracting some features, which will be described in the later section in detail.

Weather data was provided with over 20 features and we selected some of them based on some prior experience from domain experts. In total, we generated three groups of features.

Data Partition: The data was then partitioned into training and testing, the ratios being 80% and 20% respectively. The training data could be further partitioned with k-fold cross-validation mechanism for preventing over-training in some algorithms.

Model Generation: The next stage of the process is to generate a pool of predictive models by using above listed learning algorithms as the candidates to be selected for building an ensemble.

Model Evaluation and Selection: The generated models are then evaluated for their level of accuracy with training and validation data. The metrics used for this were percentage correct classification and the percentage of correct prediction with within one minute, see details in the next section.

Combination: This stage is designed to combine the results of the individual models selected from the pool of models at the previous stage, to produce the final prediction for a station. For this process, two functions have been used: the averaging and weighted averaging.

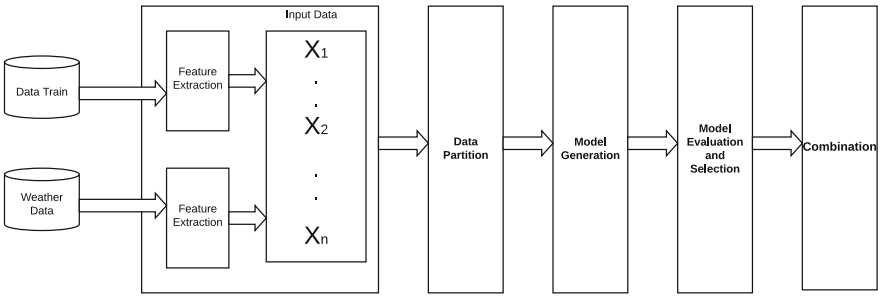

Fig. 2. A process for building heterogeneous ensemble.

This framework was implemented using Python, based on Scikit-learn and other libraries.

Methods for Combining the Models: In any ensemble, it is essential to have a decision making function to combine the outputs of individual models to produce the final output. This function plays a critical role in determining the performance of an ensemble [21]. We devised two functions: averaging and weighted averaging.

– Averaging Ensemble
For a given ensemble with N models, $\Phi = \{m_1, m_2, ..., m_i, ..., m_N\}$, an averaging decision function simply calculates the mean of the outputs from all the N models: $\{\Delta t_1, \Delta t_2, \ldots, \Delta t_i, \ldots, \Delta t_N\}$, that is,

$$\Delta t = \frac{1}{N} \sum_{i=1}^{N} \Delta t_i \tag{2}$$

The result is taken as the final prediction for the difference of the arrival time of a given train at a station.
– Weighted Ensemble
In a weighted ensemble, we employ another decision-making function - the weighted averaging method. The motivation is based on the fact that some models perform better than others, so logically their decisions should have more weight than those models whose performances are lower, when making a final decision. Thus we devised a weighted decision function as follows to investigate whether it can improve the accuracy of an ensemble.

In an ensemble of N models, $\Phi = \{m_1, m_2, ..., m_i, ..., m_N\}$, each model will be evaluated with a validation dataset and their accuracies *Accuracy* $\{A_1, A_2, ..., A_i, ..., A_N\}$, will be used to calculate their weight for making a final output. Specifically, for a model m_i, its weight W_i is computed by the following equation.

$$W_i = \frac{A_i}{\sum_{i=1}^{N} A_i} \tag{3}$$

As can been seen, the weight is normalised so that $\sum_{i=1}^{N} W_i = 1$. The final predication Δt can be calculated as follows.

$$\Delta t = 1/N \sum_{i=1}^{N} W_i \Delta t_i \tag{4}$$

3.3 Evaluation Metrics and Methods

For comparison, we chose Random Forest, which is also an ensemble-based method, as a competitive target because this method has been demonstrated to the best by some studies as reviewed earlier. All the individual models were also used as the baselines in the evaluation.

The time unit used for train timetable is based on the *minute*, not the second. So, to follow the practice of rail service, we devised the following measures to evaluate the accuracy of predictions.

Percentage of Correct Prediction (*CP*) After Rounding: The continuous output produced by regression models is unlikely to be whole integers. Output values are therefore rounded to the nearest integer. Then the rounded predicted time difference value is compared with the actual outputs in the test dataset.

If these two values are equal, then the prediction is correct, otherwise is wrong. The percentage of correct predictions is calculated over the number of journeys that the model accurately predicted to the minute after the rounding.

Percentage of Predictions Within 1 min After Rounding, $|P| < 1$: This is the same as the above evaluation, using comparison to assess whether the model predicted exactly or within 1 min either way of the actual time.

Statistical Tests for Comparing the Results: The performance of the proposed ensemble method is evaluated using statistical significance tests, which compare one method against multiple other methods. The selection of the test is determined by the experimental design. In our study, the Friedman test was used to compare all the methods used and the results presented in the critical difference diagram. The Friedman test is a nonparametric test designed for comparison of multiple learning algorithms. Its results show whether or not there is a statistical difference in performance between the algorithms. The critical difference diagram provides a graphic representation of overall performance. A slide bar is used to indicate algorithms which are statistically similar [6]. We used the critical distance diagram to show the ranking differences between single models, Random Forest, Averaging Ensembles and Weighted Ensembles.

4 Experiments and Results

To test our ensemble methods, a case study was carried out on an intercity train service between Norwich and London. Table 1 lists the stops between a pair of stations for this journey. For this service, trains run every half of an hour and stop at different stations.

Table 1. List of pair of adjacent stations between Norwich and London Liverpool Street for all of the journey variations.

Pair No.	Station1	Station2	Pair No.	Station1	Station2
1	NRW	DIS	4	IPS	MNG
2	DIS	SMK	5	MNG	COL
3	SMK	IPS	6	COL	CHM

4.1 Data and Features

We collected the train running data in seven months between 2017–2018, from the HSP - Historic Service Performance (NRE) [15] data repository. It contains large quantities of historic data and has a filter which allows for data to be requested in particular time frames between specific stops (OpenRailData) [5].

Weather data was also collected from the weather stations nearest to the railway stations in question.

The data requires pre-processing before it could be used for analysis or modelling because each journey can have a different number of stops. These are represented by separate rows which needed to be brought together to avoid the necessity of searching and joining the rows every time they were used. A custom class in programming language was used for this purpose. The class represents a journey and can include such elements as arrival, departure and stop times, as well as other variables which are helpful for modelling. Scripts also contain code carrying information stored about each journey object. This allows model features to be created and the addition of further information, such as weather, to be added. Rows of model inputs are then ready for the learning process. The features which were thus derived consisted of three groups. The first group includes running time information, the date and day of the week, timing, and whether a train is running at off-peak period or peak period. The second group of features derived from the weather data, taken from the weather station closest to the stop. The third group used information about the journeys in front of the modelled train and any deviation from the timetable.

4.2 Experiment Design and Results

Three sets of experiments were conducted, using the chosen journey, by using all features. Each set of the experiment was run 5 times with different data partitions to test the consistency of prediction. Data partition 80% for training and 20% for testing. Some cross validation is used later during optimization of the hyper-parameter of some models. This partition procedure was repeated a number of times and the mean values and the standard deviations were computed.

Table 2 lists the mean accuracies and standard deviation of three types of models: the individual models, the averaging ensembles and the weighted ensembles. It also gives the mean correction prediction rate within one minute of the targets, $|P| < 1$ and their standard deviations.

Figure 3 shows the percentage of correct prediction between pairs of stations and Fig. 4 depicts the correct predictions within one minute.

As can be seen, the average of correct prediction accuracies of single models is never higher than 50% and falls to as low as 38%. However, with the averaging ensembles, the prediction accuracy was clearly improved between 5% to 25%, with the lowest average correct prediction of 43% and the highest of 71%. The best results were produced by the weighted ensembles.

In general, the single models had an overall mean accuracy level of only 41.6%, with a mean standard deviation of 11.24 for correct predictions, which are low and also varied considerably, and 86.42% for correction prediction within one minute of the actual time. Averaging ensembles improved on average by 15% to 56.78% for a correct prediction with a smaller standard deviation, and 8% to 94.64% for correct prediction within one minute. The weighted ensembles outperformed significantly both the individual models and the averaging ensembles,

Table 2. The results for single models, averaging ensemble, and weighted averaging ensemble for each individual stop station between Norwich and London Liverpool Street.

	Average single models				Average ensemble				Weighted ensemble			
	% CP		$\%\|P\| < 1$		% CP		$\%\|P\| < 1$		% CP		$\%\|P\| < 1$	
Section	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
NRW - DIS	39.93	11.71	82.23	20.06	54.79	1.47	93.25	0.66	71.69	1.41	94.35	0.79
DIS - SMK	48.78	10.90	92.28	5.35	71.34	0.65	95.41	0.43	75.41	1.08	96.21	0.38
SMK - IPS	37.87	9.39	84.85	17.34	42.81	2.61	93.69	1.99	65.14	0.81	95.89	0.66
IPS - MNG	43.69	10.71	90.94	11.47	67.47	7.26	97.58	0.95	81.85	1.13	98.36	0.40
MNG - COL	44.96	10.76	92.86	11.02	60.49	4.88	97.55	0.63	78.10	0.79	98.38	0.26
COL - CHM	32.34	13.97	75.36	19.03	43.76	4.48	90.35	3.35	64.52	0.70	92.82	0.34
Mean	41.26	11.24	86.42	14.05	56.78	3.56	94.64	1.34	72.79	0.99	96.00	0.47

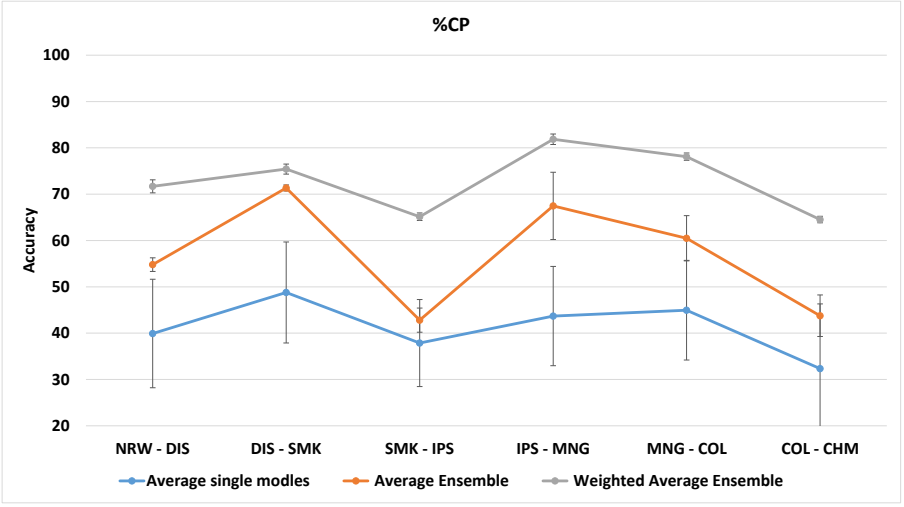

Fig. 3. The results (mean accuracies and standard deviations) of predicting the delays between all stations with single models, averaging ensembles and weighted average ensembles. The Weighted ensembles are most accurate and also most reliable.

with a mean accuracy of 72.79% for correct predictions and 96.0% for correction predictions within one minute of the actual time.

In addition, it should be particularly noted that ensembles are not only more accurate but more importantly more consistent and reliable, because they have very low standard deviations. For individual models, their predictions varied considerably with a mean standard deviation of 11.24%, which implies that individual models are not consistent. On the other hand, the averaging ensembles are more consistent because they have a much smaller mean standard deviation

(3.56), whilst the weighted ensembles are very consistent with a mean standard deviation as low as 0.99.

Another phenomenon can be observed from the results is that for some stations on the same journey, the accuracies of either individual models or ensembles performed poorer than other stations. This suggests that the underlying prediction problems are specific to journeys between particular pairs of stations, where there may be some uncertainties that were not represented by the data. For example, for the pair of SMK-IPS, it was found later that there were some freight trains going through IPS station but not recorded in this dataset.

Moreover, the accuracy of predictions was also affected by the amount of available train data, which was reflected by a dip at CHM station because some trains did not stop here and hence there was not the same amount of the training data as for other stations.

Figure 4 shows a considerable improvement in accuracy when ensembles were used to predict the arrival time within one minute of the targets. From a practical operational point of view, the departure and arrival times are shown in a unit of minute to passengers, So, our weighted ensembles achieved the accuracy levels as high as 98% with a standard deviation of 0.47%.

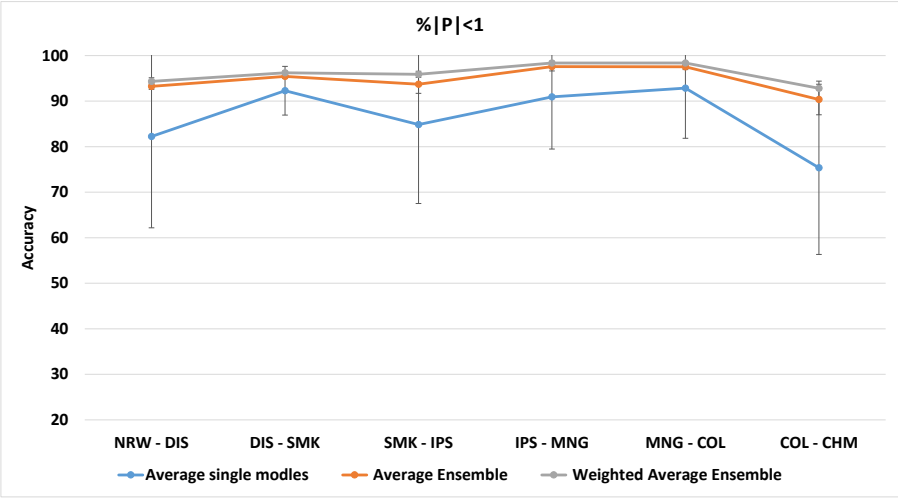

Fig. 4. The results for predicting the delays between all stations within one minute with single models, averaging ensembles and weighted ensembles.

The accuracies of these three types of models are statistically compared in the following section.

In summary, our results demonstrated that using ensembles improved the accuracy of train delay predictions and that the weighted ensembles consistently and significantly out-performed other methods and were the best for this purpose.

4.3 Critical Comparisons

The statistical tests described in the earlier section were conducted to compare the accuracies of individual models, the averaging ensembles and the weighted ensembles and also Random Forest, which was chosen as a comparison baseline because it was already considered to be one of the most accurate methods for predicting train delays [16].

On average over all the experiments for the entire train service journey, our weighted ensembles were ranked in the first place, with the averaging ensembles the second place, Random Forest the third and the individual models the fourth. The critical distance among these four methods are represented by the critical distance diagrams Fig. 5 and Fig. 6. The interpretation is that the methods linked with a thick bar are not statistically different from each other.

As can be seen, the methods are linked with two thick bars: the first one includes the WE (Weighted Ensembles), AE (Averaging Ensembles) and RF, and the second group includes, AE, RF and single models. They show that our weighted ensembles are statistically and significantly better than the individual models, and better than other two ensemble methods although not significantly.

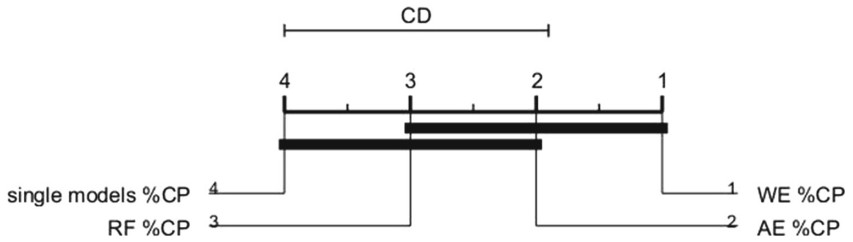

Fig. 5. Comparison between our ensembles with single models and the Random Forest ensembles, using correct prediction measure %CP.

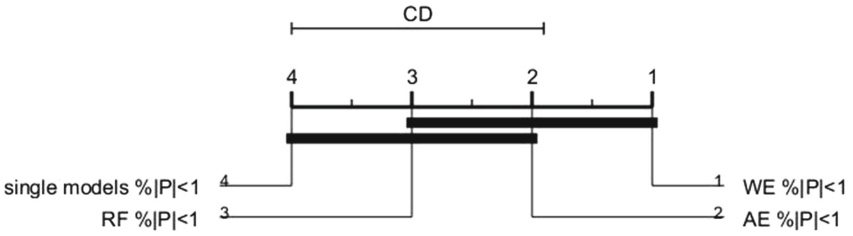

Fig. 6. Comparison between our ensembles with single models and the Random Forest ensembles within the measure of correction prediction within one minute.

5 Conclusion and Future Work

In this paper, we presented a framework for developing machine-learning ensembles to improve the accuracy and reliability of predicting train delays. We built two types of ensembles with two decision fusion strategies: averaging and weighted averaging. We tested our methods with a case study on an intercity train service between Norwich to London Liverpool Street. We extracted a wide range of features from train running data and weather data. The results of our experiments show that our ensembles built using both the averaging and the weighted averaging are always not only better than single models, but also more reliable. Furthermore, our ensembles out-performed the Random Forest ensembles, which were considered as state of the art methods for train prediction.

However, our study was limited by the amount of the data available to us at time, and because of this limit, our results should be interpreted with caution. In addition, we only used 16 types of models so the pool of candidate models is relatively small. These two issues will be addressed in our future study. We want to use more data, which we have already collected, more features and more types of models, such as deep learning models, to test our methods and also try to improve the prediction accuracies as much as possible.

Acknowledgement. The authors would like to thank Mr. Douglas Fraser in particular for his important work in gathering the data and the advice given in this research and also the WeatherQuest for providing the weather data for this project. We acknowledge the foundational works carried out by two MSc students at time, Mr. Bradley Thompson and Ms Mary Symons. In addition, we really appreciate the support and advice given by the people from the Train Operating Company - Greater Anglia, the Network Rail, the Rail Delivery Group, and the Rail Standards and Safety Board (RSSB) for the grant awarded through the rail big data sandbox competition in 2017. Specifically, we would also like to thank the Albaha University for providing a studentship for Mr Mostafa Al Ghamdi to do his PhD at the University of East Anglia.

References

1. Alyahyan, S., Farrash, M., Wang, W.: Heterogeneous ensemble for imaginary scene classification. In: KDIR, pp. 197–204 (2016)
2. Barbour, W., Mori, J.C.M., Kuppa, S., Work, D.B.: Prediction of arrival times of freight traffic on us railroads using support vector regression. Transp. Res. Part C Emerg. Technol. **93**, 211–227 (2018)
3. Brown, G., Wyatt, J., Harris, R., Yao, X.: Diversity creation methods: a survey and categorisation. Inform. Fusion **6**(1), 5–20 (2005)
4. Corman, F., Kecman, P.: Stochastic prediction of train delays in real-time using Bayesian networks. Transp. Res. Part C Emerg. Technol. **95**, 599–615 (2018)
5. Open Rail Data: HSP (2016). https://wiki.openraildata.com/index.php/HSP. Accessed 13 Nov 2019
6. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)

7. Gashler, M., Giraud-Carrier, C., Martinez, T.: Decision tree ensemble: small heterogeneous is better than large homogeneous. In: 2008 Seventh International Conference on Machine Learning and Applications, pp. 900–905. IEEE (2008)
8. Gaurav, R., Srivastava, B.: Estimating train delays in a large rail network using a zero shot Markov model. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 1221–1226. IEEE (2018)
9. Hu, J., Noche, B.: Application of artificial neuron network in analysis of railway delays. Open J. Soc. Sci. **4**(11), 59 (2016)
10. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. Neurocomputing **70**(1), 489–501 (2006). https://doi.org/10.1016/j.neucom.2005.12.126. http://www.sciencedirect.com/science/article/pii/S0925231206000385, Neural Networks
11. Lee, W.H., Yen, L.H., Chou, C.M.: A delay root cause discovery and timetable adjustment model for enhancing the punctuality of railway services. Transp. Res. Part C Emerg. Technol. **73**, 49–64 (2016)
12. Lessan, J., Fu, L., Wen, C.: A hybrid Bayesian network model for predicting delays in train operations. Comput. Ind. Eng. **127**, 1214–1222 (2019)
13. Marković, N., Milinković, S., Tikhonov, K.S., Schonfeld, P.: Analyzing passenger train arrival delays with support vector regression. Transp. Res. Part C Emerg. Technol. **56**, 251–262 (2015)
14. Nair, R., et al.: An ensemble prediction model for train delays. Transp. Res. Part C Emerg. Technol. **104**, 196–209 (2019)
15. NRE: Darwin data feeds (2018). http://www.nationalrail.co.uk/100296.aspx. Accessed Oct 2019
16. Oneto, L., et al.: Advanced analytics for train delay prediction systems by including exogenous weather data. In: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 458–467. IEEE (2016)
17. Oneto, L., et al.: Train delay prediction systems: a big data analytics perspective. Big Data Res. **11**, 54–64 (2018)
18. Office of Rail and Road: Passenger and freight rail performance 2018-19 q3 statistical release on 21/02/2019. https://dataportal.orr.gov.uk/media/1210/passenger-rail-usage-2018-19-q3.pdf. Accessed 05 Apr 2019
19. Smetek, M., Trawiński, B.: Selection of heterogeneous fuzzy model ensembles using self-adaptive genetic algorithms. New Gener. Comput. **29**(3), 309 (2011). https://doi.org/10.1007/s00354-010-0305-3
20. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 226–235. ACM (2003)
21. Wang, W.: Some fundamental issues in ensemble methods. In: IEEE World Congress on Computational Intelligence, pp. 2243–2250, July 2008. https://doi.org/10.1109/IJCNN.2008.4634108
22. Wen, C., Lessan, J., Fu, L., Huang, P., Jiang, C.: Data-driven models for predicting delay recovery in high-speed rail. In: 2017 4th International Conference on Transportation Information and Safety (ICTIS), pp. 144–151. IEEE (2017)
23. Yaghini, M., Khoshraftar, M.M., Seyedabadi, M.: Railway passenger train delay prediction via neural network model. J. Adv. Transport. **47**(3), 355–368 (2013)
24. Yu, B., Yang, Z.Z., Chen, K., Yu, B.: Hybrid model for prediction of bus arrival times at next station. J. Adv. Transport. **44**(3), 193–204 (2010)