Heterogeneous Machine Learning Ensembles for Predicting Train Delays

Mostafa Al Ghamdi, Gerard Parr, and Wenjia Wang

***Abstract*— Train delays have been a serious persisting problem in the UK and also many other countries. Due to increasing demand, rail networks are running close to their full capacity. As a consequence, an initial delay can cause many knock-on delays to other trains, and this is the main reason for the overall deterioration in the performance of the rail networks. Therefore, it is really useful to have an AI-based method that can predict delays accurately and reliably, to help train controllers to make and apply alternative plans in time to reduce or prevent further delays, when a delay occurs. However, existing machine learning models are not only inaccurate but more importantly unreliable. In this study, we have proposed a new approach to build heterogeneous ensembles with two novel model selection methods based on accuracy and diversity. We tested our heterogeneous ensembles using the real-world data and the results indicated that they are more accurate and robust than single models and state-of-the-art homogeneous ensembles, e.g. Random Forest and XGBoost. We then verified their performances with an independent dataset from a different train operating company and found that they achieved the consistent and accurate results.**

***Index Terms*— Train delay prediction, heterogeneous ensemble, random forest, diversity.**

I. Introduction

DESPITE significant efforts made by train operating companies (TOCs) in the UK to improve the performance of train services, the Public Performance Measure (PPM)[1] [1] decreased from 91% in 2013-14 to 82.8% and On-Time measure to 62.3% in December of 2022 [2]. In general, train delays can be classified into two types: primary and reactionary. A primary delay is an initial delay that can be caused by a variety of factors, such as accidents, equipment or signal failures, construction works, bad and hot weather, flooding, vandalism, trespass, etc. [3]. A primary delay can then initiate a series of consequential reactionary delays on other trains running on the same or related rail networks [4].

Over the decades, the number of train passengers has been steadily increasing, except at the peak of the Covid-19 pandemic, so the number of train services has had to be increased accordingly. But the increased train services put more pressure on the rail networks to be running close to their full capacity, and hence leave very little buffer to absorb disturbance of train operations. As a consequence, one small primary delay can cause many reactionary delays cascading through the rail network. This can result in major disruptions to the network and significant inconvenience to the passengers. Whilst the Covid-19 pandemic was very bad for many things, it provided an unprecedented opportunity to verify the impact of rail networks running at their capacity. Due to the significant drop of passenger numbers, fewer trains were running on the rail networks, so most trains were running on time and the PPM improved considerably. But as the pandemic eased off, more people started travelling again, and the train services were almost back to their normal schedule. Since then, more trains have been delayed, as indicated by the most recent PPM figure of 83.9% for 26 June to 24 July 2022, which is much lower than the 90.3% for the equivalent period during the high time of COVID-19 in 2021 [1]. This highlights once again the need for predicting delays and then producing alternative decisions to reduce the impact of delays as much as possible.

Nevertheless, due to the high complexity of rail networks and service operations, it is very difficult for train controllers to foresee all the reactionary delays in long sequences and then come up with some evidence-based alternative plans to manage or mitigate the delays and disruptions. Thus, it is essential to develop some new methods or systems to predict train delays at a sufficiently early stage to assist train controllers to take appropriate actions to minimise the number and impacts of consequent delays.

This paper proposes a machine learning ensemble method that combines different types of predictive models to improve the accuracy of delay prediction. The ensemble approach works by generating several models in an analogous way to a committee of human experts. The outputs of member models in an ensemble are then combined using an aggregation function to generate a hopefully improved final output. But an ensemble may not produce a better output than that of individual models, and it all depends on how an ensemble is constructed and what aggregation function is used [5]. Simply speaking, for example, if an ensemble is built with some identical models, i.e. there is no difference or diversity among these models, then such an ensemble will not produce any better results, but the same as that of the individual models,

Manuscript received 16 November 2022; revised 13 March 2023; accepted 1 November 2023. The Associate Editor for this article was D. Pelusi. *(Corresponding author: Mostafa Al Ghamdi.)*

Mostafa Al Ghamdi is with the School of Computing and Information, Al-Baha University, Al Baha 65779, Saudi Arabia (e-mail: malsager@bu.edu.sa).

Gerard Parr and Wenjia Wang are with the School of Computing Sciences, University of East Anglia, NR4 7TJ Norwich, U.K.

Digital Object Identifier 10.1109/TITS.2023.3337858

[1] PPM has been the performance indicator of train services in the UK. It was replaced by an enhanced metric—Control Period 6 (CP6) in April 2019, but PPM is still a useful indication and as our data was up to 2019 before the Covid-19 Pandemic, we used it in this study.

because they do not have different ability to compensate the weakness of each other. Therefore, an ensemble should ideally be built with diverse models.

However, most current ensemble methods generate homogeneous ensembles, where the models are all of the same type, e.g. decision trees, hence they are highly likely to be similar to each other and make the same mistakes, and as a result, a homogeneous ensemble may not perform better than individual models. On the other hand, a heterogeneous ensemble is built with the models that are generated by different types of learning algorithms to utilize the strengths of each type to therefore achieve more accurate results.

In this study, we built heterogeneous ensembles with multiple models generated by using a variety of machine learning algorithms. This should give a great advantage over both individual methods and homogeneous ensembles. Moreover, the aggregation function is also important because it determines how the outputs of individual models are combined to produce the final prediction. In this study, we devised some aggregation functions using averaging and weighted averaging and compared their performance on the test data. A framework has been built to implement these ensembles and the weighted aggregation function, and a case study on an intercity train service was conducted. In addition to this, we have investigated two criteria for selecting the models for building effective ensembles: accuracy and diversity.

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

II. Related Work

Various methods have been used for predicting train delays, including regression and classification. A recent paper [6] reviewed the methods for train delay prediction and divided them into two categories: *event-driven*, which models the dependencies of train arrivals, departures and other events in the rail network, and *data-driven* where the train-event dependency structure is not explicitly modelled. They concluded that while event-driven approaches are easily interpretable, the best data-driven methods give the most accurate predictions overall and have the additional advantage of being easier to be applied in real time.

This section reviews the related work in predicting train delay, by initially overviewing the basic methods that generate individual models, then focusing on ensemble methods. As some methods developed in one transportation type may be applicable to other types of transportation, so we will also briefly cover some publications in the areas of air and road transportation.

So, our reviews are organised by single models, ensemble methods and the work in other types of transportation.

A. Single Models

As the target of train delay prediction problem is a real value—usually the time variation from the scheduled time, so regression is the commonly used approach in this field. With various regression algorithms from simple linear to polynomial regression, to some stochastic and kernel-based machine learning methods, such as Support Vector Regression, Artificial Neural Networks, each of them can be used to generate single models from the data for predicting train delay.

In 1994, [7] used stochastic approaches to simulate interactions between trains to help avoid the effects of knock-on delays. Later, [8] recommended the use of stochastic methods for estimating arrival and departure delays in Holland. Reference [8] used a stochastic based modeling approach to highlight techniques that estimate reactionary delays, which are systematic in nature and are the cause of delays in railway stations. Their approach used probability distributions to deal with data fluctuations. They explored delays resulting from conflicts of routes and the transfer of trains between connections. However, as noted in a recent paper by [9], the approaches using probability distribution models have failed to provide accurate predictions of train delay durations when they occurred.

More recently, [10] used Bayesian network models for predicting delays. They stressed that traditional techniques require frequent updating, pointing out that if real-time train movement data is to be used, it will be extremely resource-intensive. They therefore used different structures, including the so-called hybrid structure, primitive-linear and heuristic hill-climbing. Their method aims at using the technique of data related to high-speed routes in China, which cover distances of over 1000 km. When applied to these routes, they achieved an accuracy of over 80% and so it is evident that modelling such routes can be done effectively. Their method also differentiates between the primary and reactionary delays. However, it should be noted that most lines in the UK railway networks do not cover such long distances, neither are they high-speed, but more interconnected, therefore it is not clear how well their approach would generalise to the UK rail networks.

Bayesian networks [11] were applied to this problem with historical data from Sweden. Their method was not restricted to static data and was also able to include dynamic characteristics of delays that were constantly fluctuating.

Support Vector Machines have also been used. For example, [12] applied a hybrid method to the prediction of bus arrival times; and [13] used SVM for identifying any connections between train delays and railway network qualities. This work focuses on anticipating and avoiding delays, particularly as finding any connections between the two could enable railway staff make use of learned choices to decrease delays.

Two further potential methods discussed by [13] were hybrid simulation and machine learning, and multiple regression.

Artificial Neural Networks (ANN) are also commonly used for predicting train delay [14], [15], [16]. Reference [14] used ANN to predict arrivals and departures on the Iranian train network. Their study modeled the train delay prediction as a classification problem, rather than a regression problem naturally considered by most other studies. Reference [15] used ANN, together with some other methods in their study for predicting train delays in the Italian rail network. The different approaches of [14] and [15] highlight the importance of tailoring the method employed and the features selected to suit the specific problems of the network. The Iranian rail network has a quite lower level of complexity than that of the Italian network. Since the UK rail network is extremely complex with a huge number of crossovers and the train services are run by many different privatised companies, the methods of [15] would be better suited for the UK rail network than that of [14].

Another type of neural net, Extreme Learning Machines (ELMs), Shallow and Deep, were used by [16] for predicting train delays because ELMs can learn faster than those which use traditional learning algorithms, which may not be fast enough, and they generalise well [17]. They were claimed to be more appropriate with big data than the methods that use univariate statistics because the model adapts and improves when fed external data [16].

In general, although these algorithms and models may produce some good results, they have a common weakness, that is, their performance varies quite considerably from model to model when used on different data, which means that they are not very consistent or are unreliable when working alone. However, this drawback can be compensated by using ensemble approaches, which will be reviewed below.

B. Ensemble Methods

An ensemble in machine learning can be simply defined as a committee of several models working together with a decision aggregation function with an aim of producing a better final output [5]. The basic idea is based on the fact that while no individual model may be perfect for solving a non-trivial problem, a committee constructed with several suitable models can work better than individuals working alone.

Reference [15] used standard the ensemble method Random Forest (RF) in their work and demonstrated that the ensembles are able to improve predictions. They compared their RF ensembles with kernels and neural networks. They noted that because they derived more detailed features in their dataset, the RF was able to use more subsets of features selected at random to generate as many as 500 decision trees and then combine them to produce a more accurate prediction, although it was more time consuming and used more resources.

Reference [18] also compared Random-Forest ensemble models to multiple linear regression models and found the RF models to be more accurate in prediction. They determined the number of trees to use by examining the error between different tree sizes, because they required an accurate but not overly complex model.

Several other authors have also used Random Forest for train delay predictions. Reference [19] concluded that in their application it outperformed linear regression and decision trees. Reference [20] used a two stage RF model and found that it increased the accuracy of delay predictions. Reference [21] used a bi-level RF approach, while [22] used a large scale application of RF.

All these studies demonstrated that the Random Forest ensembles outperform individual models generated by other approaches, and suggested that it was the best algorithm at the time for predicting train delays. But as their studies did not use a relatively new type of boosting algorithm—XGBoost [23], their suggestion should be taken with caution. A recent study [24] used XGBoost with hyperparameter tuning by Bayesian optimization for predicting train delays. When tested with the data for two high speed railway lines in China, it performed better than six other well-known algorithms including Random Forest, which gave the second best performance.

In summary, it is clear that ensemble methods are generally more accurate than individual models and, to date, Random Forest and XGBoost ensembles have been considered as the best methods for train delay prediction problem.

C. Work in Related Fields

Commercial aviation, like rail transport, is a complex system in which many stages of the process may be subject to delays caused by factors such as bad weather, mechanical issues and availability of aircraft, crew, embarking/disembarking ports, runways and airspace. Reference [25] reviewed studies of using machine learning algorithms for predicting flight delays. They noted that machine learning is becoming increasingly important in flight system analysis, and that the most frequently used methods are neural networks, k-Nearest Neighbor, SVM, Random Forest and fuzzy logic. For example, [26] predicted root delay at US airports using Random Forest and compared their methods with regression. Reference [27] predicted root delay using an adaptive network they had created using fuzzy inference systems. These predications were then input into fuzzy decision-making method for sequencing flight arrivals at JFK International Airport.

Reference [28] considered a wide range of factors with potential impact on flight delay, and conducted a comparison of several machine learning-based models for general flight delay prediction tasks. They created a dataset for this purpose containing automatic dependent surveillance-broadcast (ADS-B) messages combined with airport information, weather information and flight schedules. Their design for prediction included various classification tasks plus a regression task. Their experiments indicated that long short-term memory (LSTM) could cope with the aviation sequence data it received but had a problem with overfitting. In comparison with earlier models, their Random Forest-based model performed well in terms of prediction accuracy (90.2% for the binary classification) and it was able to surmount the overfitting problem.

Reference [29] predicted the likelihood of flight delays using data mining and causal machine learning algorithms, in a process, known as USELEI (Understanding, Sampling, Exploring, Learning, Evaluating, and Inferring) process. The process was used because CRISP-DM (Cross Industry Standard Process for Data Mining) and SEMMA (Sample, Explore, Modify, Model, Assess), which are commonly used for research in data mining, do not take into account important features of causal data mining which requires the identification of causal relationships between variables and the creation of a causal network from sizeable data sets. Data from various sources were used and the results indicated that predictors including capacity, efficiency and traffic volume, had significant effects on the probability of flight delays. The predictive power and precision of the final network was high, with a 91.97% predictive accuracy.

In the area of passenger road transport, passenger satisfaction and increasing use of bus services may also be affected by the accuracy of delay predictions. Reference [30] analysed two important factors in real time bus dispatching which can be used to deal with fluctuations in travel time due to traffic conditions and passenger numbers. They predicted bus arrival times by combining Support Vector Regression and Kalman Filters. They also proposed automatic timetable redesign using a circle search algorithm, and their results were verified in a case study in Shenzhen, China.

From our review of the published work on predicting train delays and the work in related transport areas of aviation and roads it is clear that similar approaches are being used across different areas. Therefore it is possible that methods developed in one area could be used in other areas, and thus the methods we have developed for rail transport in our research have strong potential to be used in other transport areas.

The existing studies that we have discussed showed that machine learning models are of great benefit in the prediction and avoidance of train delays. The ensemble approach has proven effective and in comparison to using single classification or regression models, it would be expected to perform better, as has proven to be the case. To date there has been very little work on using heterogeneous ensemble methods. Homogeneous ensembles have the advantage of using a committee of models. However, the models are all of the same kind. In contrast, a heterogeneous ensemble has models produced by different algorithms which are methodologically heterogeneous. These models are therefore less likely to make similar errors, resulting in a more accurate ensemble.

The only published example of a heterogeneous ensemble in the context of predicting train delays that we are aware of is that of [22]. They developed a heterogeneous ensemble consisting of three models: Random Forest, kernel regression and mesoscopic simulation of the network. They only generated one model of each type and used them as they are without any assessment or selection. They found that the ensemble performed better than the individual models. However, they also found that their approach was sensitive to hyperparameters and fine tuning was required. This means that their method would not generalise well. In a previous study [31] we showed that heterogeneous ensembles outperformed random forest for predicting train delays. The motivation of this research is to help improve the UK train network by developing an accurate and reliable machine learning ensemble by efficiently combining multiple models generated from different standard learning algorithms into a heterogeneous ensemble.

III. Delay Prediction Modelling Scheme

We can represent the train delay prediction as follows. A given train service journey, J, will contain several stations, i.e., $J = \{S_1, S_2, \ldots, S_i, \ldots, S_{N-1}, S_N\}$ starting with the initial station S_1, then intermediate stations S_2 to S_{N-1}, (where the train stops for passengers embarking and disembarking,) and ending with the terminal station S_N. There will be a series, T, of n trains: $T = \{T_1, T_2, \ldots, T_j, \ldots, T_{n-1}, T_n\}$, moving (or stationary) at various times and positions according to the timetable, assuming that they are running on time.

We can then define the delay prediction problem for a train T_i that has just departed from a Station S_i, then we want to predict its arrival time at the next station S_j, and at all the subsequent stations of the journey. The modelling scheme for this is described as follows.

Rather than predicting the actual arrival time of a given train T_i at its next station, we transform the value to predict the difference between the scheduled and actual arrival times. Let t_{pa} and t_{aa} represent the scheduled and actual arrival times, respectively, of train T_i at an intended station S_j. The difference, Δt, between these values is calculated by the following equation,

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

A positive value for Δt indicates a delay and a negative value indicates an early arrival. This predicted delay will be taken, together with other variables, as the inputs to the next model for predicting the arrival time at a subsequent station.

IV. Heterogeneous Ensemble Methods

Ensemble methods are techniques for combining several base learning models together in order to achieve a better accuracy than that single models can have. However, an ensemble may not necessarily do better if it is not properly constructed and a very important condition for an ensemble to be better is that its member models must be diverse enough from each other when making their decisions [5].

As noted earlier, ensembles can be classified into two types: homogeneous and heterogeneous. a homogeneous ensemble is built with models generated by just one type of base learner only, e.g. decision trees. In contrast, a heterogeneous ensemble is built using models generated by several different types of base learners. A homogeneous ensemble method, although using just one learning algorithm, attempts to generate diverse models by either manipulating training data with different sampling strategies, or using different parameters of the base learner. But as these models are of methodologically the same type, they are generally similar to each other and hence more likely to make the same errors [32]. As a result, the improvement from a homogeneous ensemble can sometimes be very small, if any. But on the other hand, the models that are generated by using different learning algorithms should be more diverse from each other to reduce the probability of making the same errors simultaneously, that is why a heterogeneous ensemble is more likely to perform better in terms of accuracy and consistency.

In our study, we proposed a framework for building heterogeneous ensembles and compared our heterogeneous ensembles with some popular homogeneous ensembles, which are briefly described as follows for convenience.

A. Existing Popular Homogeneous Ensemble Methods

The most common ensemble or arguably state of the art approaches are Random Forest, Boosting and XGBoost.

1) Random Forest: is an ensemble algorithm that was proposed by [33] that generates a variety of decision trees that can be used for classification and regression. The basic idea is that it selects some features at random to induce a decision tree and repeats this process for many times by sampling features t at random with replacement to generate a forest of many trees, i.e. a homogeneous ensemble of decision. The ensemble prediction is achieved by averaging for regression and by majority voting for classification. It has been applied to many areas and usually achieved very good results.

2) Boosting: [34] is to generate a series of models with a boosting mechanism that attempts to make the next model improves the errors of previous models. One of the most well know boosting algorithms is AdaBoost (from **Ada**ptive **Boost**ing). After each boosting iteration the weights applied to individual data samples that have not been correctly learned by the model at previous iteration are increased, i.e. boosted, so that the current model will pay more attention to learn those samples correctly. The boosting process usually stops at a given number of iterations. These models essentially form a homogeneous ensemble and the final output is determined by combining the outputs of all the models with different weights, computed based on their accuracy.

3) XGBoost: e**X**treme **G**radient **Boost**ing [23] is an extension of the Boosting framework. It employs distributed Newton gradient descent and parallel tree boosting to improve efficiency. It has been used in many domains [35], [36], [37], demonstrating that it has the ability to perform fast and achieve accurate results. It can use parallel and distributed computing to speed up the learning process, resulting in a faster and scalable modelling process. XGBoost has been popularly used in various applications and has often produced better results than other methods, even including some deep learning methods.

Due to their excellent performance, XGBoost and Random Forest are considered as state of the art methods in machine learning for many real-world applications. That is why they were chosen for comparison with our methods.

B. Heterogeneous Ensembles

Heterogeneous ensembles are expected to perform better than homogenous ensembles because they are built with methodologically different models, which may have learned different aspects of a problem from the training data and could be more diverse from each other to avoid making the same mistakes. Previous studies [38], [39], [40], [41] have shown that heterogeneous ensembles can perform better than homogenous ensembles, not only being more accurate but more reliable as well. Train delay prediction has proven to be a difficult problem to solve and many attempts have been made to apply machine learning methods to it. Because of the advantages of heterogeneous ensemble techniques, and the fact that almost no work has been published describing their application to train delay prediction, we therefore chose to develop heterogeneous ensembles in this study.

1) Building Heterogeneous Ensembles: The process we proposed for building heterogeneous ensembles consists of the following steps.

- **Dataset:** Some data of train operations in the UK were collected and used in this study. The basic form of the train running data is provided in a format similar to a train timetable. For each train, it lists the planned departure time, actual departure time, planned arrival time and actual arrival time for each stop station along the journey of a train service.
- **Feature Extraction:** The raw data was transformed into a structured representation by extracting the features listed in Section VII-A.
- **Data Partitioning:** The data was then partitioned into training, validation and testing datasets, the proportions being 70%, 15% and 15%, respectively, using a random seed for reproducibility.
- **Base Learning Algorithms:** In the next stage of the process several different learning algorithms are employed to generate predictive models as the candidates to be selected for building the ensemble.
- **Collection of Trained Models:** All the trained models are put into a Collection of Models (CM) as candidates for further processing.
- **Model Selection:** This step selects some models as the member models of a heterogeneous ensemble. We devised two selection methods: MSM1 and MSM2, based on two different criteria: *accuracy* and *diversity*, respectively, calculated on the validation dataset. We will discuss these criteria in detail later.
- **Decision Making Function:** The final stage is to combine the results of the chosen member models in an ensemble in order to produce the final prediction. For this process, two combination techniques: *Averaging* (AE) and *Weighted Averaging* (WE) were employed and their details will be described in the next subsection.
- **Heterogeneous Ensemble:** Once this is done, the ensemble is complete and can be used for prediction or further tested on the test data.

This process was implemented in Python, using Scikit-learn and other libraries.

2) Decision Fusion Strategies: Any ensemble requires a decision making function to combine the outputs of the individual models to produce the final output. This function plays a very critical role in determining the performance of an ensemble [5]. We devised two functions for this: *Averaging (AE)* and *Weighted Averaging (WE).*

a) Averaging (AE): This is a technique that computes the mean of the outputs from all the individual models in an ensemble as the final output of the ensemble. It is the simplest decision fusion function and is often used for regression problems. In this technique, all the models in an ensemble are used to make their prediction independently for each data point and their predictions in real value are then averaged to produce a final prediction, y_a. (Equation 2.)

$$y_a = (\sum_{j}^{M} y_j)/M \tag{2}$$

where, y_j is the output from member model j, M is the number of models in the ensemble.

b) Weighted averaging (WE): This is a variation of Averaging, or a generalised averaging fusion function. The important difference is that outputs of individual models are assigned with different weights based on their performance when computing the final value. There can be different ways to calculate the weights, based on the chosen metrics, e.g. R^2 or *MAE*. In this study, for model j, its weight w_j is computed with equation 3, based on its R_j^2 on the validation data, where a is the multiplying factor to further adjust influence of a model. When $a > 1$, the weight is boosted to increase the influence of a good model in making the final decision; whilst when $a < 1$, the weight is further reduced for the models with poor performance. When $a = 1$, the value of R^2 of a model is just taken as its weight, which was used in this study for simplicity, without loss of generality.

$$w_j = a_j R_j^2 \tag{3}$$

The output of a weighted ensemble, y_w, can then be computed by Equation 4.

$$y_w = (\sum_{j}^{M} y_j \times w_j) \Big/ \sum_{j}^{M} w_j \tag{4}$$

where, y_j is the output from member model j, w_j is the weight for model j.

In order to construct a heterogeneous ensemble with different models that are as diverse as possible, we chose 12 different learning algorithms to generate candidate models, which will be listed later.

For a given training dataset each algorithm is used to generate a model. Thus 12 models are generated as candidates for forming an ensemble. For comparison, we built various ensembles by using two model selection methods, based on accuracy (MSM1) and diversity (MSM2), which will be described in Section VI.

V. Diversity Measures

A fundamental philosophy that makes an ensemble better is that the individual models in an ensemble must be diverse enough from each other to avoid making the same mistakes simultaneously. So, having a certain level of appropriate diversity among the member models is essential [5]. However, measuring the diversity among the models is tricky as there are different definitions of diversity in the literature and most of them were defined for classification problems. Diversity measures can be generally divided into two categories: pairwise and non-pairwise. Pairwise measures only consider the difference between two models at a time and non-pairwise measures try to estimate the diversity among all the models. Although there are dozens of definitions for diversity, almost all of them are not really effective as they measure a specific diversity that is not directly related to the decision marking function [5]. On the other hand, when a diversity is defined to represent the failure independence among the member models, which is used in the final decision fusion function, it is demonstrated to have a consistent association with ensemble accuracy [42].

For regression problems, it is more challenging to measure diversity among the models because the diversity measures defined for classification problems require categorical outputs and cannot handle the real value in a regression problem output hence they are not directly applicable to regression ensemble. One review [43] evaluated several diversity measures in regression ensembles by using correlation coefficient, covariance, dissimilarity measure, Chi-square, mutual information, etc. A typical study [44] demonstrated that negative correlation can be useful to push the models apart if it is integrated in the training of neural networks. But again, this mechanism is not related to the final decision making and hence its effectiveness is limited. In addition, these studies do not consider how the diversity in an ensemble can be affected and measured when a model is added to or removed from an ensemble, which may affect its overall prediction [43].

For this study we redefined some of the metrics evaluated by [43] and also modified probably the most effective non-pairwise diversity measure for classification, the Coincident Failure Diversity (CFD) [9], [32], to fit regression problems. To the best of our knowledge this is the first time that CFD has been applied to a regression problem.

It is important to note that different diversity measures will give values across different ranges, therefore when using diversity as a component in the fitness function, we normalise values of diversity metrics. The metrics that we derived or used in our study are described below.

A. Correlation

In statistics, the correlation coefficient is a measure of how strongly two variables are related to one another based on their relative movements, with a range of values between -1.0 and 1.0. A correlation of -1.0 indicates that there is a perfect negative correlation, while a correlation of 1.0 indicates that there is a perfect positive correlation. With a correlation value of 0.0, no linear relationship can be found between the two variables. Therefore, the diversity is inversely proportional to correlation, i.e., when the correlation is high, the diversity will be low and vice versa.

It can be used as a diversity measure between a pair of two models, which we call as the correlation diversity D_r as defined below.

For a given pair of models with their outputs: y_i and y_j, with k as an index over the intended N data samples.

$$\begin{aligned} &D_r = \tfrac{1-r}{2}, \\ &\text{Where, } r = \frac{\sum(y_{ik}-\bar{y_i})\times\sum(y_{jk}-\bar{y_j})}{\sqrt{\sum(y_{ik}-\bar{y_i})^2\times\sum(y_{jk}-\bar{y_j})^2}} \end{aligned} \tag{5}$$

Where r is the standard Pearson's correlation coefficient. When, $r = -1$, D_r is 1, meaning that the maximum diversity is achieved; $r = 0 => D_r = 0.5$, indicating that two models have a random diversity between them; and when

$r = 1 => D_r = 0$, there is no diversity between two models. So, for any pair of models, they must meet the condition: $D_r > 0.5$, to be considered diverse enough.

B. Covariance

The covariance indicates whether the two variables vary together in a correlated manner. Unlike the correlation, whose values are limited to -1 and $+1$, the values of covariance are unbounded, could be anything between -∞ and $+\infty$, which can be difficult to interpret or to use as a diversity measure, so we need to convert it to a limited and meaningful representation. We employ the sigmoid function to convert the range of possible covariance values to (0, 1). So we define the covariance diversity D_v as follows.

$$D_v = \frac{2e^{-|cov|}}{1 + e^{-|cov|}} \tag{6}$$

$$cov = \frac{\sum(y_{ik} - \bar{y_i}) \times \sum(y_{jk} - \bar{y_j})}{N-1} \tag{7}$$

where, cov is a normal covariance coefficient between two outputs, y_i and y_j. With this definition, the bigger $|cov|$ is, the smaller diversity is, and vice versa. When $|cov|$ is small or close to zero, which means that two variables do not show correlation in their trends, so D_v is close to 1, i.e. maximum diversity.

C. Disagreement

This score measures how dissimilar the predictions from two different models are. It is mainly used for binary variables, but may be indirectly applied to the continuous output after it is first converted into a binary value with a threshold value θ, as per Equation 8.

$$f(x) = \begin{cases} 0, & x < \theta \\ 1, & x >= \theta \end{cases} \tag{8}$$

The disagreement between outputs of two models can be used as a diversity metric, which is defined as follows.

$$Disagreement = \frac{N^{01} + N^{10}}{N^{00} + N^{01} + N^{10} + N^{11}} \tag{9}$$

where N^{11} represents the number of samples that are correctly predicted by a pair of models M_1 and M_2, and N^{00} represents the number of samples incorrectly predicted by two models. N^{10} - the number of samples that are correctly predicted by M_1 and incorrectly by M_2 and N^{01} - the number of samples that are incorrectly predicted byM_1, but correctly by M_2.

When *Disagreement* is 0, it means that two models have no disagreement between them, i.e. they always produce the same outputs, either correct or wrong on the same date at the same time, so they are identical. In this case, there is no need to have another model in an ensemble. When *Disagreement* is 1, it means that two models always give the opposite answers on every data point, hence have the maximum diversity, which is not necessarily a good thing either as they will always cancel each other out. Where it may be useful is when some values are below the middle point (0.5) as it means that the two models have some common knowledge for dealing with majority of data, whilst each has some unique knowledge to cover unusual data.

D. CFD

The *Coincident Failure Diversity* score was defined by [32] to measure the probability that two or more models fail on test data simultaneously and was also used for binary variables. But in a similar manner as mentioned above, we modified it to handle continuous variables for regression problems as follows:

$$CFD = \sum_{m=1}^{M} \frac{(M-m)}{(M-1)} \times f_m \tag{10}$$

where, $m(= 1, 2, \ldots M)$ is the number of models that produce a wrong prediction on the data between 1 and M), and f_m is the failure frequency of m models and is defined as:

$$f_m = E_m / E_{any} \tag{11}$$

where E_m is the number of samples incorrectly predicted by m models and E_{any} is the number of samples incorrectly predicted by at least one model.

When $CFD = 0$, this means that all the members of an ensemble are the same, hence there is no diversity. When $CFD = 1$, the ensemble members have a maximum diversity, indicating that all members make distinct errors that are compensated by the other members. So an ensemble with a maximum diversity should produce a perfect answer, although its members may make some mistakes.

VI. Ensemble Construction Algorithms

In order to construct an effective and efficient ensemble, a very important step is to decide what models and how many models should be used [45]. This is because that the characteristics of individual models in terms of accuracy, efficiency and diversity, as well as the number of models are two essential factors that affect the performance of an ensemble [5]. More models used means more resources (time and space) required, so when building an ensemble it is thus more economical and efficient to use as few models as possible while preserving accuracy and diversity [46].

In our research we proposed two model selection methods: MSM1 and MSM2, based on two different criteria, to build various heterogeneous ensembles. MSM1 only considers accuracy of individual models, and MSM2 takes both accuracy and diversity into consideration.

A. MSM1

This selection method only considers the accuracy of individual models on the validation data. Firstly, it starts with a collection of the models (CM) that have been generated with various learning algorithms or provided with a collection of some existing pre-trained models, and their accuracies are evaluated on a given validation dataset with a chosen metric, such as R^2 or any other suitable one. All the models are then ranked in a descending order according to their validation accuracy. The ensemble Φ starts empty. Then, starting from the top of the ranking, we choose the highest ranked model in the Collection and add it to ensemble Φ. Then the accuracy of the in-building ensemble will be evaluated in the next component.

This selection process repeats until as long as that the accuracy of the in-building ensemble keeps improving and stops when the accuracy starts to drop. However, in our experiments, we let it continue until there is no model left in the Collection just to examine the effect of a permutation of the entire Collection of the modules by producing an accuracy plot over the growth of an ensemble from empty to the full size, as shown by the figures in Section VII.

This selection method can be generalised by setting up a selection batch size, say Q. That is, in every iteration, Q models are selected together as a batch and then added to a growing ensemble, rather than just one model at a time. This can speed up the process of ensemble construction. The batch size can be determined or varied by a number of factors, such as the difference of accuracy among the models, or the size of the model Collection and the size of an intended ensemble, etc. In our experiments, as the size of the model collection is relatively small, we set $Q = 1$, with an intention of evaluating the contribution of each individual model.

Algorithm 1 for **MSM1**

```
Input: Collection of models, CM, validation data Val
Output: The best ensemble $\Phi_{best}$
N=count(CM)
for $i = 1$ to $N$ do
    calculate Accuracy $R^2$ on Val
end for
sort CM in descending order according to their accuracy $R^2$
for $i = 1$ to $N$ do
    select the $i$th model and add to $\Phi_i$
    evaluate $\Phi_i$, and record the best fo far, $\Phi_{best}$
    if $\Phi_{best} > \Phi_i$ then
        Stop
    else
        Continue
    end if
end for
```

B. MSM2

This selection method takes account of both accuracy and diversity when selecting models. First, the highest accuracy model (HAM) is selected from the collection of models (CM). Then the second model is chosen with the highest diversity model (HDM) to the HAM. As we applied two different diversity measures: Pairwised and non-pairwised (CFD), this selection method has two variants, MSM2a and MSM2b. For MSM2a, the diversity between models in CM is calculated with a pairwise diversity measure such as correlation, covariance and disagreement. For MSM2b, we use the CFD that considers the combinations of the models in the CM and each combination consists of HAM, HDM and the remaining model, then they are added to the Ensemble.

Algorithm 2 for **MSM2a** Pairwise

```
Input: Collection of models, CM, Diversity-metric, validation data Val
Output: The selected models Φ
N=count(CM)
for $i = 1$ to $N$ do
    calculate Accuracy $R^2$ on Val
end for
sort CM in descending order according to their accuracy $R^2$
HAM=the highest accuracy model in CM
remove HAM from CM
for $i = 1$ to $N$ do
    calculate diversity (HAM, $CM_i$))
end for
sort CM in descending order according to their diversity
for $i = 1$ to N-1 do
    select first $i$ models and add them to a new set called NCM
    add model combination(HAM, $NCM$) to Φ
end for
```

C. Evaluation Metrics

We employed two standard metrics to evaluate the accuracy of models and ensembles for predicting train arrival delay. These are the Mean Absolute Error (MAE) and R-squared (R^2):

1) MAE:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{12}$$

2) R Squared:

$$R^2 = 1 - (\sum(y_i - \hat{y}_i)^2 / \sum(y_i - \bar{y}_i)^2) \tag{13}$$

D. Statistical Tests for Comparing the Results

Statistical significance tests, which compare one method against others, were used to evaluate the significance of performance differences between the proposed ensemble methods and the compared models and existing ensembles. These tests are chosen based on the experimental design. The Friedman test was used in our study to compare all the methods used and the results presented by the critical difference diagram. A Friedman test compares multiple learning algorithms through nonparametric procedures. The results of the test show if there is a statistical difference between the accuracies of the algorithms. We have used critical difference diagrams, as introduced by [47], to provide a visual representation of the overall performance. In this type of diagram, ensembles that do not differ significantly are grouped into “cliques” identified by lines. When two ensembles are not members of any common clique, their performances are significantly different.

Algorithm 3 for **MSM2b** Non-Pairwise

```
Input: Collection of models, CM, Diversity-metric, validation data Val
Output: The selected models $\Phi$
N=count(CM)
for $i = 1$ to $N$ do
    calculate Accuracy $R^2$ on Val
end for
sort CM in descending order according to their accuracy $R^2$
HAM=the highest accurate model in CM
remove HAM from CM
for $i = 1$ to N-1 do
    Find all possible combinations of $i$ models and add them to a new set called NCM
    M= count (NCM)
    for $j = 1$ to $M$ do
        Compute Diversity (HAM, $NCM_i$)
    end for
    sort NCM in descending order according to their diversity
    select model combination(HAM, $NCM_0$)
    add selected models to $\Phi$
end for
```

VII. Experiments and Results

A. Data and Features

We have been collecting train running data from the Network Rail Darwin data feed site [48] and also the Open Rail Data Historic Service Performance Data Repository (HSP) [49]. Besides containing vast amounts of historical data, the HSP site allows users to filter data by period.

The data used in the experiment should be complete, accurate, and consistent in representing the normal operations of train services in the UK. Therefore, we chose a dataset, coded as NRW, over a period of about 2 years before the COVID-19 pandemic started simply because that when the pandemic started, many people were allowed to work from home, hence the number of train services was significantly reduced and the remaining trains with a very low number of passengers ran mostly on-time. The chosen data was pre-processed by following the common practice in machine learning research. Specifically, the records with missing data and duplicate records were deleted. We ran numerous logical checks on our datasets to identify and correct errors. Some trains arrived earlier than they left, which was one example of inconsistency. Records with inconsistent data were deleted. Numeric data were scaled. Categorical data were converted to numeric.

In order to convert the train running timetable-like data to a structured representation, we derived the following features:

- The planned travel time from the current station to the next.
- The actual travel time from the current station to the last.
- The planned travel time from the current station to the last.
- The planned dwell time at the current station.
- The actual dwell time at the current station.
- The arrival delay for the current station.
- The departure delay for the last station.
- The departure delay for the current station.
- The number of passing points. (Places where the train's passing is recorded.)
- Day of the Month
- Day of the Week
- Hour of the Day

B. Experiment Design

The experiments were all coded in Python, using scikit-learn and the XGBoost library for generating algorithms. They were run on a personal computer with an Intel Core i5-7500 CPU running at 3.4 GHz, with 32 GB RAM, and using the Windows-10 64-Bit operating system.

We designed our experiments to examine the performance of our heterogeneous ensembles and the chosen comparative targets: two existing ensembles—Random Forest (RF) and XGBoost. In order to investigate the effectiveness of our model selection methods MSM1 and MSM2 and the influence of ensemble size, we varied the number of models in a heterogeneous ensemble from 2 to 12. In addition, we examined the impact of diversity on accuracy using four diversity measures.

Each experiment was repeated five times with variations of different data samplings. Overall, we conducted over 150 experiments. The results presented are the average and standard deviation (SD) of the repetitions.

C. Base Learning Algorithms

In this study, we chose 12 different Regressors with an aim of representing a wider spectrum of machine learning themes from the baseline method to "state-of-the-art" methods, to generate candidate models for building heterogeneous ensembles. Of them, eight generate individual models: Linear Regression, Bayesian Ridge [50], Stochastic Gradient Descent [51], Lasso [52], Ridge [53], K-nearest neighbours Regressor, Decision tree [54] and Multi-layer Perceptron [55]. Four algorithms produce essentially homogeneous ensembles with decision trees: Random Forest [33], ElasticNet, Gradient Boosting [56], and XGBoost [23].

The default parameter settings were used for all these algorithms as our focus was to investigate whether our ensembles have ability to do better than individual models that work separately, no matter how well an individual model does.

D. Experimental Results

Figures 1 and 2 present the results for single models, averaging ensembles (AE) and weighted averaging ensembles (WE). These heterogeneous ensembles of variable sizes (from 2 to 12) were built with the algorithm MSM1, and the AE and the WE fusion functions. In Figure 1, R^2 values of the predictions are given, with the standard deviations (SD) over five runs. These results are also presented in Table I which shows the means and standard deviations (SD) of R^2 of

TABLE I
THE RESULTS FOR AE, WE, SINGLE MODELS USING MSM1

M	Average Ensemble		W. Avg. Ensemble		Avg. single models	
	R^2	SD	R^2	SD	R^2	SD
2	0.7909	0.0057	0.7909	0.0057	0.7888	0.0010
3	0.7917	0.0052	0.7917	0.0052	0.7885	0.0012
4	0.7911	0.0053	0.7911	0.0053	0.7840	0.0091
5	0.7892	0.0055	0.7892	0.0055	0.7803	0.0114
6	0.7867	0.0055	0.7869	0.0055	0.7765	0.0138
7	0.7849	0.0053	0.7850	0.0053	0.7735	0.0149
8	0.7829	0.0052	0.7831	0.0052	0.7713	0.0152
9	0.7810	0.0051	0.7813	0.0051	0.7693	0.0151
10	0.7796	0.0052	0.7799	0.0052	0.7664	0.0149
11	0.7770	0.0052	0.7776	0.0052	0.7590	0.0335
12	0.7734	0.0052	0.7747	0.0052	0.7499	0.0441

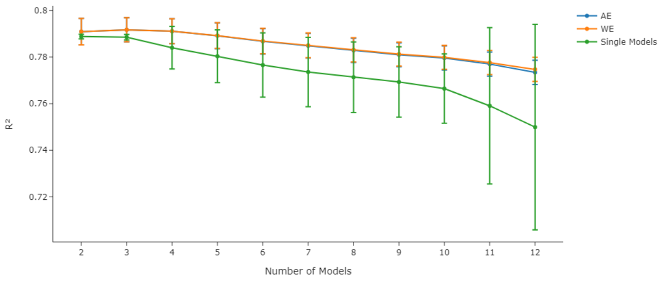

Fig. 1. R^2 values of AE and WE using MSM1 and single models.

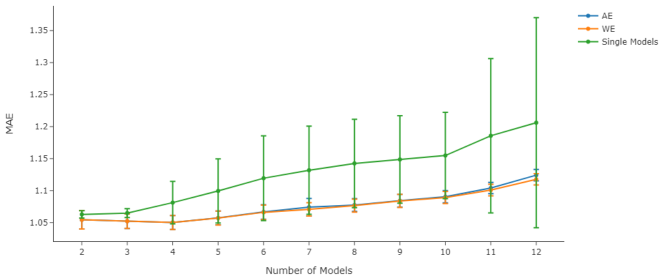

Fig. 2. MAE of AE and WE ensembles built with MSM1 and single models.

TABLE II
THE RESULTS FOR AE, WE, SINGLE MODELS USING MSM2 (CFD)

M	Average Ensemble		W. Avg. Ensemble		Avg. single models	
	R^2	SD	R^2	SD	R^2	SD
2	0.7568	0.0086	0.7612	0.0098	0.7209	0.0950
3	0.7689	0.0069	0.7732	0.0066	0.7384	0.0729
4	0.7776	0.0071	0.7795	0.0071	0.7482	0.0628
5	0.7787	0.0053	0.7798	0.0057	0.7508	0.0561
6	0.7810	0.0041	0.7820	0.0047	0.7560	0.0520
7	0.7812	0.0052	0.7819	0.0058	0.7571	0.0479
8	0.7798	0.0051	0.7804	0.0056	0.7570	0.0443
9	0.7785	0.0048	0.7790	0.0054	0.7570	0.0415
10	0.7767	0.0048	0.7776	0.0054	0.7568	0.0391
11	0.7768	0.0052	0.7771	0.0057	0.7573	0.0372
12	0.7734	0.0052	0.7747	0.0057	0.7497	0.0440

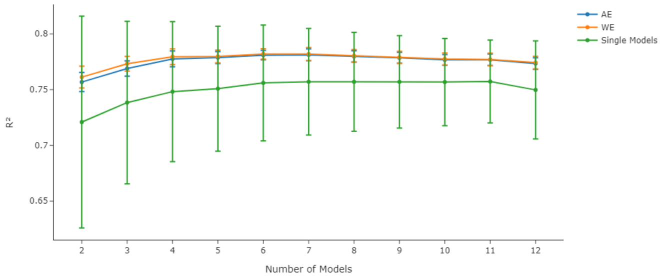

Fig. 3. R^2 of AE and WE ensembles with MSM2b(CFD) and single models.

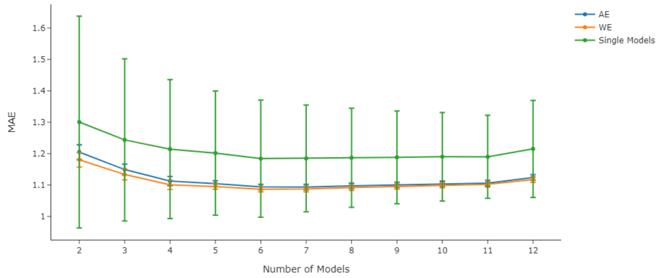

Fig. 4. MAE of AE and WE with MSM2b(CFD) and single models.

predictions made by the single models, and the heterogeneous ensembles. In Figure 2 the MAE values of these predictions are presented.

From these results it can be seen that both types of ensemble (AE and WE) had consistently higher R^2 values than the single models. In addition their SD values were lower, except for the ensemble sizes 2 and 3, where the SD values were relatively small for both ensembles and single models. Also, the SD values for both AE and WE are much more consistent across the number of models than the SD for SM. The R^2 values are slightly better for WE than AE.

Thus we can say that the ensembles outperform the single models not only in accuracy but also in consistency (with much smaller SDs). The AE and WE ensembles were nearly the same to begin with, but as the number of models increased, the WE performed slightly better. This is because there are significant variations in the weights, which aid the ensemble by giving more weight to the most accurate models.

Figures 3 and 4 show the R^2 and MAE values obtained, respectively, for ensembles of different sizes, with AE, WE and single models using MSM2 with CFD for diversity measurement.

In Figure 3, R^2 values of the predictions are given, with the standard deviations (SD) over five repeat runs. These results are also presented in Table II which shows the means and standard deviations (SD) of R^2 of predictions made by the single models, and the heterogeneous ensembles. In Figure 4 the MAE values of these predictions are presented. It can be seen that AE and WE consistently perform better than single models. Tables III–V show the mean and standard deviations (SD) of R^2 of predictions made using MSM2 with other diversity measures. Similar patterns to these found using CFD were found using these other diversity measures, but the best results were obtained by using CFD. However, overall, the best results were obtained with MSM1 rather than MSM2, thus using diversity when selecting models did not improve the accuracy of the ensemble.

Figures 5 and 6 present comparisons of the R^2 values obtained for different sized AE and WE ensembles using

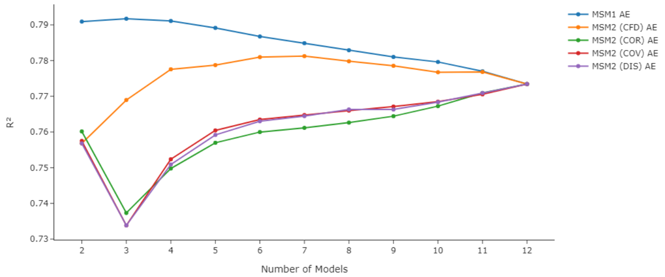

Fig. 5. Comparison of AEs ensembles built with two selection methods, different diversity measures and sizes.

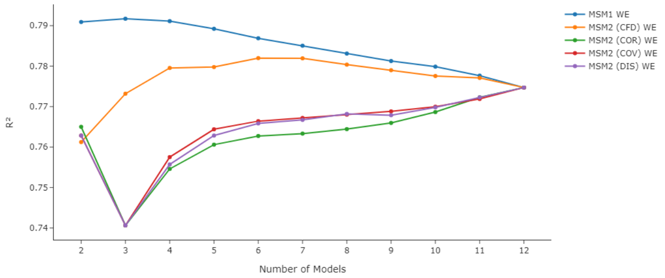

Fig. 6. Comparison of WEs ensembles built with two selection methods, different diversity measures and sizes.

TABLE III

THE RESULTS FOR AE, WE, SINGLE MODELS USING MSM2 (COR)

M	Average Ensemble		W. Avg. Ensemble		Avg. single models	
	R^2	SD	R^2	SD	R^2	SD
2	0.7602	0.0083	0.7650	0.0082	0.7280	0.0857
3	0.7373	0.0101	0.7406	0.0080	0.7034	0.0741
4	0.7498	0.0080	0.7546	0.0080	0.7168	0.0662
5	0.7570	0.0067	0.7606	0.0068	0.7253	0.0604
6	0.7600	0.0065	0.7627	0.0066	0.7305	0.0555
7	0.7612	0.0063	0.7633	0.0065	0.7341	0.0516
8	0.7626	0.0052	0.7644	0.0052	0.7371	0.0486
9	0.7644	0.0051	0.7660	0.0051	0.7399	0.0462
10	0.7672	0.0055	0.7687	0.0055	0.7427	0.0445
11	0.7709	0.0051	0.7723	0.0051	0.7468	0.0440
12	0.7734	0.0052	0.7747	0.0052	0.7503	0.0440

TABLE IV

THE RESULTS FOR AE, WE, SINGLE MODELS USING MSM2 (COV)

M	Average Ensemble		W. Avg. Ensemble		Avg. single models	
	R^2	SD	R^2	SD	R^2	SD
2	0.7575	0.0084	0.7628	0.0085	0.7214	0.0950
3	0.7338	0.0081	0.7406	0.0082	0.7034	0.0741
4	0.7524	0.0070	0.7575	0.0070	0.7202	0.0692
5	0.7604	0.0061	0.7644	0.0062	0.7290	0.0622
6	0.7635	0.0056	0.7664	0.0058	0.7335	0.0577
7	0.7647	0.0054	0.7672	0.0055	0.7367	0.0533
8	0.7660	0.0044	0.7680	0.0044	0.7395	0.0489
9	0.7671	0.0048	0.7688	0.0048	0.7414	0.0472
10	0.7685	0.0051	0.7700	0.0051	0.7439	0.0458
11	0.7705	0.0054	0.7719	0.0054	0.7468	0.0444
12	0.7734	0.0052	0.7747	0.0052	0.7503	0.0440

TABLE V

THE RESULTS FOR AE, WE, SINGLE MODELS USING MSM2 (DIS)

M	Average Ensemble		W. Avg. Ensemble		Avg. single models	
	R^2	SD	R^2	SD	R^2	SD
2	0.7568	0.0086	0.7628	0.0085	0.7214	0.0950
3	0.7338	0.0081	0.7406	0.0082	0.7034	0.0741
4	0.7509	0.0070	0.7557	0.0070	0.7171	0.0665
5	0.7592	0.0063	0.7628	0.0063	0.7265	0.0613
6	0.7630	0.0055	0.7658	0.0055	0.7326	0.0569
7	0.7644	0.0053	0.7667	0.0053	0.7359	0.0527
8	0.7663	0.0054	0.7682	0.0055	0.7391	0.0498
9	0.7663	0.0053	0.7679	0.0054	0.7410	0.0478
10	0.7684	0.0051	0.7698	0.0050	0.7433	0.0456
11	0.7709	0.0052	0.7723	0.0052	0.7468	0.0444
12	0.7734	0.0052	0.7747	0.0052	0.7503	0.0440

TABLE VI

COMPARISON OF TWO SELECTION METHODS IN DIFFERENT SIZES OF AE ENSEMBLES

M	MSM1 AE	MSM2 (CFD) AE	MSM2 (COR) AE	MSM2 (COV) AE	MSM2 (DIS) AE
2	0.7909	0.7568	0.7602	0.7575	0.7568
3	0.7917	0.7689	0.7373	0.7338	0.7338
4	0.7911	0.7776	0.7498	0.7524	0.7509
5	0.7892	0.7787	0.7570	0.7604	0.7592
6	0.7867	0.7810	0.7600	0.7635	0.7630
7	0.7849	0.7812	0.7612	0.7647	0.7644
8	0.7829	0.7798	0.7626	0.7660	0.7663
9	0.7810	0.7785	0.7644	0.7671	0.7663
10	0.7796	0.7767	0.7672	0.7685	0.7684
11	0.7770	0.7768	0.7709	0.7705	0.7709
12	0.7734	0.7734	0.7734	0.7734	0.7734

TABLE VII

COMPARISON OF TWO SELECTION METHODS IN DIFFERENT SIZES OF WE ENSEMBLES

M	MSM1 WE	MSM2 (CFD) WE	MSM2 (COR) WE	MSM2 (COV) WE	MSM2 (DIS) WE
2	0.7909	0.7612	0.7650	0.7628	0.7628
3	0.7917	0.7732	0.7406	0.7406	0.7406
4	0.7911	0.7795	0.7546	0.7575	0.7557
5	0.7892	0.7798	0.7606	0.7644	0.7628
6	0.7869	0.7820	0.7627	0.7664	0.7658
7	0.7850	0.7819	0.7633	0.7672	0.7667
8	0.7831	0.7804	0.7644	0.7680	0.7682
9	0.7813	0.7790	0.7660	0.7688	0.7679
10	0.7799	0.7776	0.7687	0.7700	0.7698
11	0.7776	0.7771	0.7723	0.7719	0.7723
12	0.7747	0.7743	0.7747	0.7747	0.7747

MSM1 and MSM2, respectively. These results are also presented in Tables VI and VII, respectively.

These results demonstrate that with MSM1 for both AE and WE, their performance remains constantly best when varying the size of ensembles from 2 to 12 models. When diversity is used in selecting models, using CFD with MSM2 is better than using pairwise measures. However, using any diversity measure is not as effective as MSM1.

In this work we have examined whether diversity can be useful for improving the accuracy of an ensemble when used for selecting models to build it. Our results show that there is no observable relationship between diversity and accuracy in regression problems. Compared to other diversity metrics, CFD stands out as the best.

Clearly, the ensembles built with selection method MSM1, i.e. using accuracy measure as its selection criterion, produced the most accurate results. The fact that MSM2 did not give as accurate ensembles as MSM1 is surprising in view of the fact that for classification ensembles diversity among models has been shown to be an important factor affecting the overall accuracy of the ensemble, and an ensemble of weak learners can still give very high accuracy providing they are sufficiently diverse. However, the measurement of diversity is not trivial and several measures of diversity have been developed for classification ensembles. These diversity measures do not all perform equally well, i.e. some are more effective at measuring diversity than others. Because there are almost no diversity measures designed for regression problems, we adapted some existing classification diversity measures for this purpose, in particular CFD since it is recognised to be the best diversity measure for classification ensembles. However it is possible that these adapted measures were not able to capture the diversity in the regression ensembles and for this reason MSM2 was not as effective for model selection as MSM1.

When compared with the single models, our ensembles have not only the highest accuracies but also the most consistent results as indicated by smaller SD. Accuracy of single models often varies considerably over multiple runs and data. A current most accurate model may perform considerably worse in another run with a different partition of the data and it is very difficult to predict in which run a single model can produce the best result. In contrast, an ensemble can perform consistently well in any run, and this high reliability, as represented by their smaller standard deviations, is more important in real-world applications.

E. Critical Comparison and Discussion

Figures 7–12 present our results using Critical Difference diagrams. Figure 7 is the CD diagram of the results with two selection methods for AE ensembles of different sizes. (These were presented in Figures 5 and 6.) It can be seen that while there is no significant difference between MSM1 AE and MSM2(CFD) AE, there is a significant difference with MSM2(COV) AE. Figure 8 presents the equivalent comparison for the WE ensembles of different sizes, which are very similar.

Figures 9–12 compare the results for ensembles ranging in size from 2–5, generated with MSM1 with the results for Random Forest and XGBoost. For an ensemble size of 2 (Figure 9) the AE ensemble performs better than WE; for size 3, WE performs better than AE, but there is no significant difference. Overall our ensembles perform better than both Random Forest and XGBoost. We did not present the results beyond ensemble size 5. The performance deteriorates for the larger ensembles and this is most likely due to the poorer performing models being included, that were excluded by the selection process in the smaller ensembles.

Overall the key results of this study are that: (1) By incorporating models produced by the most well known and state-of-the-art methods, Random Forest and XGBoost, into our heterogeneous ensembles we have been able to take advantage of both methods and improve upon their performance. (2) The most accurate results were obtained using ensembles generated with model selection according to accuracy and weighed averaging of the model outputs.

These results demonstrate the benefit of the heterogeneous ensemble approach in general, and also show that our specific implementation approach using model selection according to accuracy and weighed averaging of the model outputs is effective.

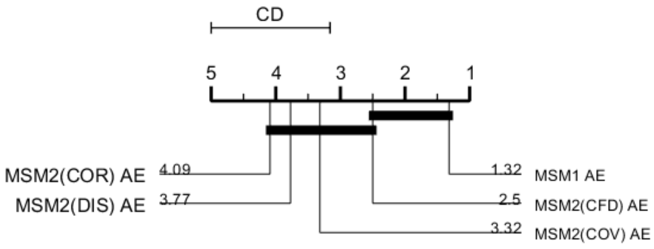

Fig. 7. CD diagram for results with two selection methods for AE ensembles of different sizes.

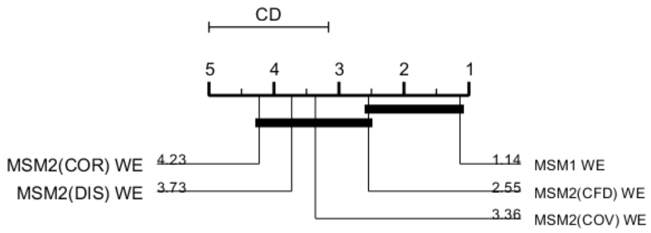

Fig. 8. CD diagram for two selection methods for WE ensembles of different sizes.

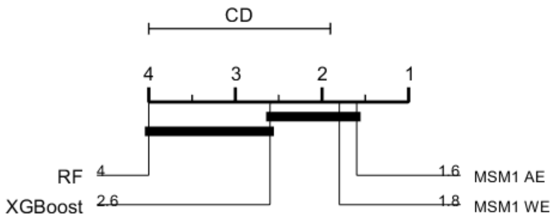

Fig. 9. CD diagram for AE and WE ensembles of size 2 with MSM1, Random Forest model, and XGBoost model.

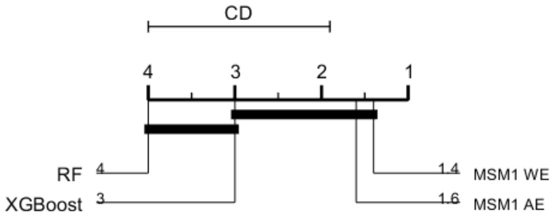

Fig. 10. CD diagram for AE and WE ensembles of size 3 with MSM1, Random Forest model, and XGBoost model.

F. Applying Our Heterogeneous Ensembles to a New Dataset

In order to investigate the generalisation ability of our heterogeneous ensemble methods we tested them on a new and different dataset of train delays. It was collected from another major intercity train service from a southwestern coastal city to London (longer than the journey of the first dataset) from a different train operating company (TOC) in the UK. This is because we believe that using the train service data from a different region, journey and TOC should provide a good indication of the generalisation and robustness of

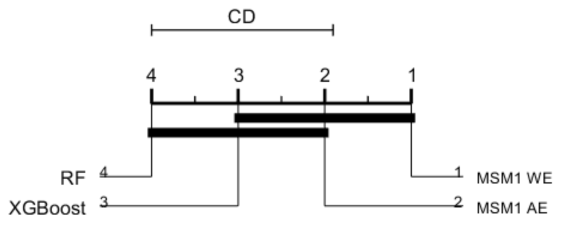

Fig. 11. CD diagram for AE and WE ensembles of size 4 with MSM1, Random Forest and XGBoost models.

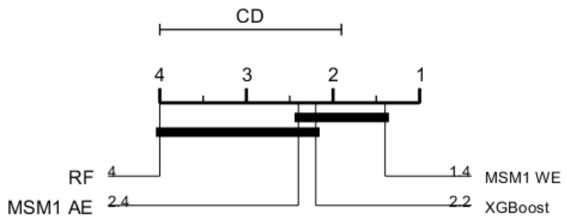

Fig. 12. CD diagram for AE and WE ensembles of size 5 with MSM1, Random Forest and XGBoost models.

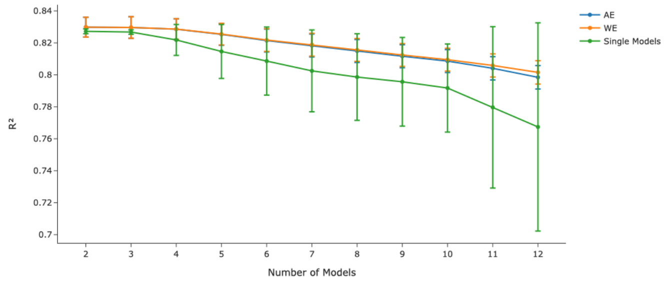

Fig. 13. Plots of the R^2 values obtained for AE, WE ensembles and single models on a new dataset—WEY. It can be seen clearly that the ensembles are not only more accurate but also more consistent (with very small standard deviations) than those of single models.

our heterogeneous ensemble methods. This dataset, coded as WEY, covered a two year period from 2017 to 2018 and contained more train services with more stations than those in the first dataset. To save time, we applied this new dataset only to the methods AE and WE using MSM1, which achieved the best results on the first dataset, as our aim was to verify if the good or best performance of our heterogeneous ensembles can be reproduced in a new dataset.

The data were pre-processed and partitioned in the same manner as for the first dataset. The experiments were run with the same settings, except using the new dataset. The results on the test data of the new dataset are visualised in Figure 13. It can be seen that the results achieved from AE and WE ensembles were very similar or identical when the ensembles were smaller. But when more models were added the WE ensembles produced slightly higher accuracies than the AE ones. It should be noted that the accuracies of single models were always lower than those of the ensembles. On consistency, it is clear that the standard deviations (SD) of the ensembles were significantly smaller than those of individual models, which is particularly true when the ensembles got bigger. It demonstrated again that our ensembles are more consistent and reliable than individual models.

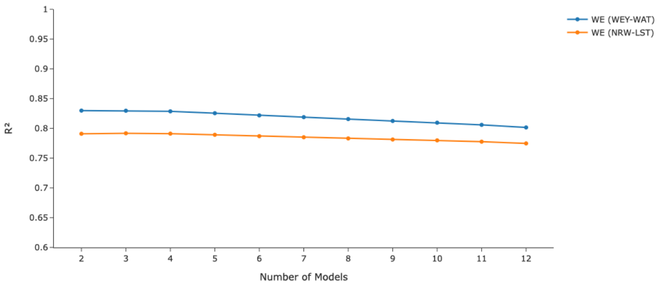

Fig. 14. Comparison of R^2 values obtained from Weighted Ensembles (WE) on the two datasets: the first one—NRW and the new one—WEY. The results show that the WE ensembles have reproduced the good performance on a new dataset and actually did better consistently.

These results showed the same, or similar, patterns that were observed from the results on the first dataset, NRW. Actually, the results on the second dataset, WEY, are consistently better than those from the first dataset, as shown clearly by Figure 14, which could be because WEY is larger. So, in summary, this test with a new dataset has not only verified the strong generalisation ability of our methods but has also demonstrated that they can do even better when they were provided with more data.

VIII. Conclusion

In general, there are two types of machine learning ensembles: homogeneous and heterogeneous. A homogeneous ensemble is defined as being built with models of only one type. For example, some arguably state of the art ensemble methods—Random Forest, AdaBoosting and XGBoost, are each constructed with just one kind of model, i.e. decision trees only. Although they are powerful, they have a common problem, that is, their member models are methodologically the same so they are likely to be highly correlated and make the same mistakes at the same time when making their decisions [5]. A heterogeneous ensemble, on the other hand, is built with models of different types, which are methodologically heterogeneous and should be more diverse in terms of the knowledge learned from the data to compensate the weaknesses of each other. Hence, heterogeneous ensembles should be more accurate and robust in general.

In this paper, we have developed a procedure to investigate the best approach and strategies for building more accurate and reliable heterogeneous ensembles. We looked at three important aspects: firstly, the generation of more diverse models by using different machine learning algorithms; then how to determine and select the models that are suitable to be the members of a heterogeneous ensemble; and lastly how to aggregate the outputs of the chosen models to produce an improved output from the ensemble.

For the first aspect, we searched and used as many as 12 different machine learning algorithms that are suitable for our task—predicting train delays—which is basically a regression problem. It should be pointed out that as our procedure for building a heterogeneous ensemble is flexible, which means it is able to use any types of regressors as its base learners, it is not limited to the 12 algorithms we used.

So if one wishes to use other algorithms, such as some new deep learning methods, it would be a straightforward matter to include them in the collection. So, we consider our approaches "future-proof" as they can take advantage of future advances in machine learning methods for regression.

For the second, we devised two new methods for selecting models from the collection of trained models by using two criteria—accuracy and diversity, independently or jointly, to build a heterogeneous ensemble. For the last aspect, we devised a new weighted decision fusion function to combine the outputs of models in an ensemble with weights computed based on the accuracy of individual models. This weighted averaging function was compared with the standard averaging function in our experiments.

We have tested our approaches to build various heterogeneous ensembles for predicting train delays on a real-world dataset. The testing results have clearly demonstrated that the heterogeneous ensembles built with our first selection method and the weighted decision fusion function produced the best results, which are not only more accurate, but also more robust, compared with other state-of-the-art methods, including the homogeneous ensembles Random Forest and XGBoost.

To verify the generalisation and reproduction ability of our ensembles, we tested them on a new dataset, collected from a different train service provided by a different train operating company. The results are in a high degree of agreement with the patterns observed in the results of the same types of ensembles on the first dataset and are actually consistently better in all the experiments. We believe that is primarily because the second dataset contained more data, although it is more complex, so our ensembles are able to learn more from the data and hence perform better.

There are of course some limitations in our methods. A relatively obvious weakness of our methods is their efficiency in exploring the ways for building the most accurate and reliable ensembles as they need to investigate various combinations of model selection methods, decision fusion functions and ensemble size. All these take considerable time to complete. Nevertheless, once these experiments are completed and the best ensembles are found, they can be implemented and run much faster in real time for real world applications.

It would not be possible to continuously update the models using live data, since the build time would be too long, however once a model has been made for a given railway route, it is not expected that it will need to be updated frequently.

Another weakness is that our methods were developed using data for a UK mainline railway, and validated using a different UK mainline railway. It is not certain how well it would work in extreme situations, such as a complex urban railway system with many stations, for example the London Underground, or very long cross-continental routes as are found in China or North America. However, in principle, we can see no reason why it should not be applied in such cases.

The method also requires a number of years data for building an accurate model. This is no problem for an established line or route where data is collected and stored. However, for a new line or a new route on existing lines, it would be necessary to collect enough data relating to the route in order to build an accurate model. Similarly in some countries the data might not normally be collected or stored, and so to apply the method data would need to be collected first. Also, to use the method predictively in real time, the current data needs to be available. However, this would apply to any method that is being used predictively in real time.

Overall our developed method is not only accurate, but more robust and it has been shown to work on a different dataset and these merits mean that it is suitable for a real world application, where consistently, accurate results are needed.

For further work, we suggest that different weighting mechanisms could be investigated in order to identify the optimal weighting strategy, for example, adjusting and adapting the weights based on multiple criteria. In addition, there is no suitable diversity measure for building regression ensembles, and although we attempted to use and modify some existing diversity measures used for classification ensembles, our experiment results showed that these diversity measures have not helped much in selecting more diverse models that can improve performance of ensembles. Given that, in principle, the basic assumption for an ensemble to work better is that its member models must be different to compensate each other's shorting comings, and the fact that our heterogeneous ensembles improved accuracy, their models must be diverse in some ways. One reason that these diversity measures did not show a clear correlation between their values and the ensemble accuracy is that they do really not measure the useful diversity among the models. So, we suggest that another area for further work is to explore new diversity measures.

Overall, our methods have built some very successful heterogeneous ensembles that can predict train delays more accurately and reliably than not only individual models, but also some well-known homogeneous ensembles.

Acknowledgment

The authors would like to thank the support from following organisations and people in their research: the Rail Safety and Standards Board (RSSB) for awarding a grant for conducting a feasibility study on developing AI techniques for predicting train delays, and through this grant, they built a data stack at UEA to collect the data and laid the foundation of their research; the Cultural Bureau of Saudi Arabia (SACB) for sponsoring the author Mostafa Al Ghamdi to do his Ph.D. at UEA, Doug Fraser for collecting data from various sources, Bradley Thompson for doing the initial work in extracting features, and finally Dr. Geoffrey Guile for reviewing their manuscript and providing critical comments.

References

[1] Network Rail. *Railway Performance*. accessed: Oct. 1, 2022. [Online]. Available: www.networkrail.co.uk/who-we-are/how-we-work/performance/railway-performance/

[2] Office Of Rail and Road (ORR). *Passenger Rail Performance Q3 2022*. Accessed: May 15, 2023. [Online]. Available: https://dataportal.orr.gov.uk/media/2189/passenger-performance-oct-dec-2022.pdf

[3] Network Rail. *Railway Delays*. Accessed: Oct. 30, 2022. [Online]. Available: www.https://www.networkrail.co.uk/running-the-railway/looking-after-the-railway/delays-explained/

[4] W.-H. Lee, L.-H. Yen, and C.-M. Chou, "A delay root cause discovery and timetable adjustment model for enhancing the punctuality of railway services," *Transp. Res. C, Emerg. Technol.*, vol. 73, pp. 49–64, Dec. 2016.

[5] W. Wang, "Some fundamental issues in ensemble methods," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intelligence)*, Jun. 2008, pp. 2243–2250.

[6] T. Spanninger, A. Trivella, B. Buchel, and F. Corman, "A review of train delay prediction approaches," *J. Rail Transp. Planning Manag.*, vol. 22, Jun. 2022, Art. no. 100312.

[7] M. Carey and A. Kwiecinski, "Stochastic approximation to the effects of headways on knock-on delays of trains," *Transp. Res. B, Methodol.*, vol. 28, no. 4, pp. 251–267, Aug. 1994.

[8] J. Yuan and I. A. Hansen, "Optimizing capacity utilization of stations by estimating knock-on train delays," *Transp. Res. B, Methodol.*, vol. 41, no. 2, pp. 202–217, Feb. 2007.

[9] Z. Li, P. Huang, C. Wen, X. Jiang, and F. Rodrigues, "Prediction of train arrival delays considering route conflicts at multi-line stations," *Transp. Res. C, Emerg. Technol.*, vol. 138, May 2022, Art. no. 103606.

[10] J. Lessan, L. Fu, and C. Wen, "A hybrid Bayesian network model for predicting delays in train operations," *Comput. Ind. Eng.*, vol. 127, pp. 1214–1222, Jan. 2019.

[11] F. Corman and P. Kecman, "Stochastic prediction of train delays in real-time using Bayesian networks," *Transp. Res. C, Emerg. Technol.*, vol. 95, pp. 599–615, Oct. 2018.

[12] B. Yu, Z. Yang, K. Chen, and B. Yu, "Hybrid model for prediction of bus arrival times at next station," *J. Adv. Transp.*, vol. 44, no. 3, pp. 193–204, Jul. 2010.

[13] N. Markovic, S. Milinkovic, K. S. Tikhonov, and P. Schonfeld, "Analyzing passenger train arrival delays with support vector regression," *Transp. Res. C, Emerg. Technol.*, vol. 56, pp. 251–262, Jul. 2015.

[14] M. Yaghini, M. M. Khoshraftar, and M. Seyedabadi, "Railway passenger train delay prediction via neural network model," *J. Adv. Transp.*, vol. 47, no. 3, pp. 355–368, Apr. 2013.

[15] L. Oneto et al., "Advanced analytics for train delay prediction systems by including exogenous weather data," in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2016, pp. 458–467.

[16] L. Oneto et al., "Train delay prediction systems: A big data analytics perspective," *Big Data Res.*, vol. 11, pp. 54–64, Mar. 2018.

[17] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231206000385

[18] C. Wen, J. Lessan, L. Fu, P. Huang, and C. Jiang, "Data-driven models for predicting delay recovery in high-speed rail," in *Proc. 4th Int. Conf. Transp. Inf. Saf. (ICTIS)*, Aug. 2017, pp. 144–151.

[19] P. Kecman and R. M. P. Goverde, "Predictive modelling of running and dwell times in railway traffic," *Public Transp.*, vol. 7, no. 3, pp. 295–319, Dec. 2015.

[20] B. Gao, D. Ou, D. Dong, and Y. Wu, "Predictive modelling of running and dwell times in railway traffic," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 30, no. 7, pp. 921–940, 2019.

[21] M. A. Nabian, N. Alemazkoor, and H. Meidani, "Predicting near-term train schedule performance and delay using bi-level random forests," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2673, no. 5, pp. 564–573, May 2019, doi: 10.1177/0361198119840339.

[22] R. Nair et al., "An ensemble prediction model for train delays," *Transp. Res. C, Emerg. Technol.*, vol. 104, pp. 196–209, Jul. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X18317984

[23] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[24] R. Shi, X. Xu, J. Li, and Y. Li, "Prediction and analysis of train arrival delay based on XGBoost and Bayesian optimization," *Appl. Soft Comput.*, vol. 109, Sep. 2021, Art. no. 107538. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494621004610

[25] A. Sternberg, J. Soares, D. Carvalho, and E. Ogasawara, "A review on flight delay prediction," 2017, *arXiv:1703.06118*.

[26] J. J. Rebollo and H. Balakrishnan, "Characterization and prediction of air traffic delays," *Transp. Res. C, Emerg. Technol.*, vol. 44, pp. 231–241, Jul. 2014.

[27] S. Khanmohammadi, C.-A. Chou, H. W. Lewis, and D. Elias, "A systems approach for scheduling aircraft landings in JFK airport," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2014, pp. 1578–1585.

[28] G. Gui, F. Liu, J. Sun, J. Yang, Z. Zhou, and D. Zhao, "Flight delay prediction based on aviation big data and machine learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 140–150, Jan. 2020.

[29] D. Truong, "Using causal machine learning for predicting the risk of flight delays in air transportation," *J. Air Transp. Manag.*, vol. 91, Mar. 2021, Art. no. 101993.

[30] X. Zhang, M. Yan, B. Xie, H. Yang, and H. Ma, "An automatic real-time bus schedule redesign method based on bus arrival time prediction," *Adv. Eng. Informat.*, vol. 48, Apr. 2021, Art. no. 101295.

[31] M. Al Ghamdi, G. Parr, and W. Wang, "Weighted ensemble methods for predicting train delays," in *Computational Science and Its Applications—ICCSA*, O. Gervasi, B. Murgante, S. Misra, C. Garau, I. Blecic, D. Taniar, B. O. Apduhan, A. M. A. Rocha, E. Tarantino, C. M. Torre, and Y. Karaca, Eds. Cham, Switzerland: Springer, 2020.

[32] D. Partridge and W. Krzanowski, "Software diversity: Practical statistics for its measurement and exploitation," *Inf. Softw. Technol.*, vol. 39, no. 10, pp. 707–717, Jan. 1997.

[33] T. Kam Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, 1995, pp. 278–282.

[34] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *J. Jpn. Soc. Artif. Intell.*, vol. 14, no. 771, p. 1612, Sep. 1999.

[35] A. Asselman, M. Khaldi, and S. Aammou, "Enhancing the prediction of student performance based on the machine learning XGBoost algorithm," *Interact. Learn. Environ.*, vol. 31, no. 6, pp. 3360–3379, Aug. 2023.

[36] M. Li, X. Fu, and D. Li, "Diabetes prediction based on XGBoost algorithm," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 768, no. 7, Mar. 2020, Art. no. 072093.

[37] W. Liu, Z. Chen, and Y. Hu, "XGBoost algorithm-based prediction of safety assessment for pipelines," *Int. J. Pressure Vessels Piping*, vol. 197, Jun. 2022, Art. no. 104655.

[38] S. Alyahyan, M. Farrash, and W. Wang, "Heterogeneous ensemble for imaginary scene classification," in *Proc. 8th Int. Joint Conf. Knowl. Discovery, Knowl. Eng. Knowl. Manag.*, 2016, pp. 197–204.

[39] A. Onan, "Particle swarm optimization based stacking method with an application to text classification," *Academic Platform J. Eng. Sci.*, vol. 6, no. 2, pp. 134–141, Aug. 2018.

[40] M. Gashler, C. Giraud-Carrier, and T. Martinez, "Decision tree ensemble: Small heterogeneous is better than large homogeneous," in *Proc. 7th Int. Conf. Mach. Learn. Appl.*, 2008, pp. 900–905.

[41] M. Smętek and B. Trawinski, "Selection of heterogeneous fuzzy model ensembles using self-adaptive genetic algorithms," *New Gener. Comput.*, vol. 29, no. 3, pp. 309–327, Jul. 2011.

[42] Y. Wu, L. Liu, Z. Xie, K.-H. Chow, and W. Wei, "Boosting ensemble accuracy by revisiting ensemble diversity metrics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 16464–16472.

[43] H. Dutta, "Measuring diversity in regression ensembles," in *Proc. IICAI*, vol. 9, 2009, p. 17.

[44] T. Higuchi, X. Yao, and Y. Liu, "Evolutionary ensembles with negative correlation learning," *IEEE Trans. Evol. Comput.*, vol. 4, no. 4, pp. 380–387, Nov. 2000.

[45] A. M. Mohammed, E. Onieva, and M. Wozniak, "Selective ensemble of classifiers trained on selective samples," *Neurocomputing*, vol. 482, pp. 197–211, Apr. 2022.

[46] G. Tsoumakas, I. Partalas, and I. Vlahavas, "A taxonomy and short review of ensemble selection," in *Proc. Workshop Supervised Unsupervised Ensemble Methods Appl.*, 2008, pp. 1–6.

[47] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.

[48] Network Rail Open Rail Data. (2018). *Darwin Data Feeds*. Accessed: Oct. 2019. [Online]. Available: http://www.nationalrail.co.uk/100296.aspx

[49] Network Rail Open Rail Data. (2016). *Historical Service Performance (HSP)*. Accessed: Nov. 13, 2019. [Online]. Available: https://wiki.openraildata.com/index.php/HSP

[50] G. E. Box and G. C. Tiao, *Bayesian Inference in Statistical Analysis*, vol. 40. Hoboken, NJ, USA: Wiley, 2011.

[51] P. Jain, S. M. Kakade, R. Kidambi, P. Netrapalli, and A. Sidford, "Accelerating stochastic gradient descent for least squares regression," in *Proc. Conf. Learn. Theory*, 2018, pp. 545–604.

[52] F. Santosa and W. W. Symes, "Linear inversion of band-limited reflection seismograms," *SIAM J. Sci. Stat. Comput.*, vol. 7, no. 4, pp. 1307–1330, Oct. 1986.

[53] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 42, no. 1, pp. 80–86, Feb. 2000.

[54] X. Wu, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008, doi: 10.1007/s10115-007-0114-2.

[55] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Washington, DC, USA: Spartan, Jan. 1962.

[56] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

Mostafa Al Ghamdi received the B.Sc. degree from Al-Baha University, Saudi Arabia, in 2010, the M.Sc. degree in computer science from Bridgeport University, USA, in 2014, and the Ph.D. degree from the University of East Anglia, U.K., in 2023, with a focus on heterogeneous ensemble learning. He is currently an Assistant Professor with Al-Baha University. His current research interests include machine learning, deep learning and generative artificial intelligence, and their applications to various real-world problems.

Gerard Parr He was the Head of the School of Computing Sciences, University of East Anglia, from 2015 to 2023. He is currently the Full Chair of the Telecommunications Engineering. He is an Invited Member with the EPSRC Peer Review College. He has collaborated widely with many Institutes, including MIT, the Georgia Institute of Technology, UC Berkeley, UC San Diego, USC-ISI Los Angeles, University College London, University College Oxford, the University of St Andrews, the University of Cambridge, the Beijing University of Posts and Telecommunications (BUPT), Tsinghua University, Peking University, and the Indian Institutes of Technology Bombay, Mumbai. His research interests include wireless sensor clouds, disaster response communications, big data event management, ICT for the rural economy, delay-sensitive protocols, transportation networks, and the IoT-edge computing. He was a co-investigator to PI Wenjia Wang for the grant from the Rail Safety and Standards Board (RSSB), U.K., and the follow Innovation Grant from the University of East Anglia. He made a significant contribution in these research projects, and was a co-supervisor of Mostafa Al Ghamdi's Ph.D. study. He was previously appointed as a Senior Guest Editor of prestigious IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS for a Special Issue on Communications Challenges and Dynamics in UAVs.

Wenjia Wang received the Ph.D. degree from The University of Manchester. He is currently an Associate Professor of artificial intelligence with the University of East Anglia (UEA), Norwich, U.K. He leads a group of Ph.D. students and research associates to develop machine learning ensemble methods for various tasks, including classification, clustering, feature selection, prediction, pattern recognition, and anomaly detection and correction. As a PI, he has been awarded over 20 research grants from U.K. Research Councils, Innovate U.K., Industries and Charities. Particularly, as a PI, leading a consortium of UEA, Greater Anglia Trains-a Train Operating Company (TOC) and Network Rail, he was awarded a grant after winning the Big Data Sandbox Competition organized by the Rail Safety and Standards Board (RSSB) in 2017 to conduct a feasibility study developing AI ensemble methods for predicting and preventing train delays, which played a significant role in the early stages of this research. He was the primary supervisor of Mostafa Al Ghamdi's Ph.D. He has published about 100 peer-reviewed papers in journals and international conferences. His research has been applied to real-world problems, such as predicting train delays, identifying data file types, classifying undersea power cable failures, and correcting anomalies in time series water level data. In the last few years, supported by three grants, including a knowledge transfer partnership (KTP) grant from Innovate U.K., and collaborating with a regional company, he led a team successfully in developing an AI system for improving seabed mapping from sonar surveys. The AI system was integrated into an existing system with a single AI button that works in real-time and saves a huge amount of time and resources. The project was graded OUTSTANDING and nominated as a finalist for U.K.'s Best KTP Awards 2023. Recently, he was awarded another Grant from Innovate U.K. to develop an AI system for monitoring the conditions of baby during birth.