Heterogeneous Machine Learning Ensembles for Predicting Train Delays

Mostafa Al Ghamdi

A thesis presented for the degree of
Doctor of Philosophy

School of Computing Sciences
University of East Anglia
United Kingdom

December 2022

Abstract

Train delays are a serious problem in the UK and other countries. Much research has gone into developing methods for predicting train delays. Most of these methods use only single models or homogeneous ensembles and their performance in terms of accuracy and consistency in general is unsatisfactory. We have therefore developed heterogeneous ensembles that use different types of regression models with an aim of improving their prediction performance.

We first looked at a wide range of base-learner models, including the state-of-the-art methods, Random Forest and XGBoost. Overall, our ensembles were more accurate than any of these single models.

We developed two methods for model selection when building the ensemble, the first uses accuracy and the second uses accuracy and diversity. We found that using accuracy resulted in the most accurate ensembles. We adapted the Coincident Failure Diversity measure for regression and compared its effectiveness with other diversity measures. While it proved the best, overall, we found no relationship between ensemble accuracy and diversity in the regression context. We also investigated the effect of ensemble size.

We compared the performance of our ensembles with the deep learning methods CNN and Tabnet and found that our ensembles were more accurate. However, ensembles of deep learning models proved to be more accurate than those of single machine learning models.

We tested our ensembles using a different set of train delay data and found that they produced more accurate and consistent results, indicating that our methods generalise well to new data.

Dedication

I wish to dedicate this thesis to the memory of my parents, Saeed and Fatimah Al Ghamdi, whose wisdom and love guided and inspired me throughout my early years, and enabled me to reach this stage in my life. I will always be grateful for all that they have done for me.

Acknowledgments

First I wish to thank God for enabling me to complete this thesis.

I would like to thank my supervisor Dr. Wenjia Wang who has guided and helped me during the time of my studies. I am particularly grateful for all the support he gave me during the time of the Covid-19 pandemic. His patience and knowledge were instrumental in enabling me to complete my studies.

I would also like to thank my secondary supervisor Professor Gerard Parr for all his support, comments and feedback throughout the course of my PhD.

I would also like to thank my friend Dr. Saleh Alyahyan, who helped me greatly in the early stages of my work and has supported me all the way to the completion of my thesis.

I would like to thank my family. My brothers Mohammed, Hatem, Ali and sisters Azzah, Sarah, Souad, Safa and Najla. I greatly appreciate all the encouragement and help that they have given me. Their love, support and prayers have been inestimable. I am especially grateful to my oldest brother Mohammed for all the help and guidance he has given me; throughout my life he has always been there to give support and encouragement.

Finally I wish to make special mention of my Wife Shaima, who has been so appreciative and tolerant of all the pressures and difficulties I have faced during the time of my PhD. She has willingly put up with long periods of separation and has always encouraged and supported me. She has always looked after me and been there for me and put up with all the extra pressures she has faced

during this period. I would also like to thank my son Taym, who has had to put up with me not spending the time with him that I wanted to and that he needed. His arrival during the time of my PhD here in the UK brought so much joy and happiness, and brightened my life during times of difficulty and pressure.

Acronyms

AE Averaging Ensemble.

BN Batch Normalization Layer.

BSM Best Single Model.

CFD Coincident Failure Diversity.

CM Collection Of Models.

COR Correlation.

COV Covariance.

DIS Disagreement.

DLHE Deep Learning Heterogeneous Ensemble.

FC Fully Connected Layer.

GLU Gated Linear Unit.

HAM Highest Accurate Model

HDM Highest Diversity Model.

HE Heterogeneous Ensemble .

HMLE Heterogeneous Machine Learning Ensemble .

MSM1 Model Selection Method 1.

MSM2 Model Selection Method 2.

MSM2a Model Selection Method 2a.

MSM2b Model Selection Method 2b.

SD Standard Deviation.

WE Weighted Ensemble.

Φ Selected models

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

Efficient train services are a vital part of transport networks worldwide, with many train services available every day with a broad choice of departure times and routes, linking city-centres, seven days per week, 24 hours per day, and 365 days of the year. Passenger travel by rail is overall less environmentally damaging than travel by road or air (European Environment Agency, 2020).

However, train delays are a major problem for both train companies and passengers in the UK (Network Rail, 2017). This has been the case for a number of years, a report by the UK National Audit Office stated that in 2006-7 delays on the UK rail network totalled over 14 million minutes, equivalent to over £1 billion in terms of lost time (National Audit office, 2008). Delays can occur for a variety of reasons but the inconvenience to the passengers affected is the same regardless of the nature of the delay. For the train operating companies delays can have a significant impact on the success of franchises (Murray, 2001).

In the UK, railway performance was officially measured using the Public Performance Measure (PPM), a measure that was devised to combine punctuality and reliability into a single value. PPM 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. For the purposes of PPM, "A train is defined as on time if it arrives at the destination within five minutes (i.e. 4 minutes 59 seconds or less) of the planned arrival time for London and South East or regional services, or 10 minutes (i.e. 9 minutes 59 seconds or less) for long distance services" (Network Rail, 2022a). The PPM itself is defined as "the percentage of trains which ran their entire planned journey calling at all scheduled stations and arriving at their terminating station within 5 minutes (for London and South East and regional services) or 10 minutes (for long distance services)" (Network Rail, 2022a).

There has been a consistent decline in the PPM over recent years, prior to the Covid-19 pandemic—from over 91% in 2013–14 to 85.6% in Q3 of 2018–19 (Office of Rail and Road, 2019). This was in spite of the fact that the entire rail industry has been working intensively to improve its performance.

In the UK, official Department for Transport figures show that the number of rail passenger numbers steadily increased by more than 150% between 1985 and 2019 (Department for Transport, 2020). While there was an initial drop in passenger numbers that occurred as a result of the Covid-19 pandemic, this has since begun to reverse and there is no reason to expect the passenger numbers to not continue to increase in the future as they have in the past, and for the PPM to continue to decline. This can be seen in that the recent PPM figure of 83.9% for 26 June to 24 July 2022 was lower than the figure of 90.3% for the equivalent period in 2021 (Network Rail, 2022b).

Initial delays (termed primary delays) can be the result of many causes, including trespassers on the line, signalling problems, accidents, fallen trees, equipment failure and construction work. (Oneto et al., 2018). An overview of the Delay Attribution Guide is provided by the Open Rail Data, which has a list of codes for different types of delay, Network Rail feeds and Train Movements (OpenRailData, 2019). The initial delays can then cause a chain of consequential delays on other trains, (which are termed reactionary delays) (Lee et al., 2016). As a result of the consistently increasing number of passengers, the rail networks have needed to run more train services to meet the demand. This has meant that the trains are closely scheduled to run with a minimum distance between them. The consequence of this is that there is little buffer time and space for any disruption in the rail system, so even a small initial delay can cause many secondary delays which disrupt a large number of train services, resulting in considerable inconvenience to the passengers.

Recognising that train delays are highly inconvenient and disruptive for train passengers, the railway networks and train companies try to do everything in their capacity to avoid train delays. When incidents occur, they provide passengers with information about the length and nature of the delay. In order to achieve this, train controllers must have some means of predicting train delays as early as possible. As well as enabling them to provide information for passengers this can enable them to take steps to reduce or prevent further delays. If the delays are expected, then the lead operation controller can be appointed at the regional control center, and a holding message can be issued to the affected lines and location (Network Rail, 2017).

In this research we intend to develop a machine learning ensemble that will combine multiple models generated from different types of standard learning algorithm, in order to enhance the accuracy and reliability of the prediction of train delays. At present, machine learning is an area undergoing very active development and this analytical approach has led to tremendous achievements. Since the amounts of data available are increasing rapidly, the role of machine learning is becoming increasingly important in offering big data solutions in predictive analytics. It thus has the potential to provide useful tools to enable train companies to more accurately predict delays and to ameliorate frustrations for passengers.

A key part of the project will be to construct a framework for producing models which can be incorporated into an ensemble for modelling and predicting delays. This research will give a review of the literature which deals with the problem of train delays. The methodology used will be explained in detail and justified.

1.2 Motivation

In recent years, there have been numerous instances of trains being delayed due to increasing numbers of passengers and the limitations of the rail network. Thus, being able to predict delays accurately is crucial when train controllers are trying to devise plans to prevent or reduce some of these delays. As a first consideration, ensembles provide better accuracy than a single classification or regression model. Many studies have shown that ensemble learning is more effective than using a single model due to the fact that combining the outputs of multiple models will often result in better results than using only one model (Dietterich, 2000; Breiman, 1996; González et al., 2020; Thompson, 2018). The success of the ensemble approach, which gives robust and consistent results, led us to use it in our study.

Such machine learning ensembles can be used to help improve the train service in the UK. This research will explore the use of various prediction models to build an ensemble to predict the extent of delays to train journey times, using historical data made available by Network Rail and their open source feeds. We will focus on the Norwich to London Liverpool Street service of the Greater Anglia area. However, the outcomes of this research and of the systems and models will be generalizable. Thus, the methods developed will be useful in dealing with data from any service within the railway system of the UK.

1.3 Research Aim and Objectives

The aim of this research is to develop heterogeneous machine learning ensemble techniques for predicting train delays.

The objectives of this research are as follows:

1. To develop methods for generating ensembles that contain models generated by more than one type of algorithm.
2. To develop methods for selecting which models to include in the ensemble.
3. To evaluate the performance of the methods developed, and determine which of them are best for predicting train delays.
4. To evaluate how well the methods developed generalise to new data.

It should be noted that majority of machine learning ensemble methods generate homogeneous ensembles, i.e., the ensemble is composed of one type of model. This is particularly so with train delay prediction where only one publication reporting a heterogeneous ensemble method has been published to date (Nair et al., 2019). The authors did not focus on the methodology, and therefore did not investigate how to build an effective heterogeneous ensemble, and their ensemble was sensitive to parameter values and not generalisable. Heterogeneous ensembles have proven effective in other problem areas (Alyahyan et al., 2016; Aytuğ, 2018; Gashler et al., 2008; Smętek and Trawiński, 2011), and therefore they should be able to work effectively on train delay data. Because they take advantage of the strengths of a variety of prediction methods, they should be particularly suited to a complex problem such as train delay prediction.

1.4 Research Questions

Based on the above objectives, this research seeks to answer the following questions:

1. How should a heterogeneous ensemble be built so that it performs better than single models?
2. What factors should be taken into consideration when selecting models for building into an ensemble? A number of factors will be examined, including accuracy, diversity and ensemble size.
3. Is there any relationship between accuracy and diversity in the context of building an ensemble for performing regression?
4. How should the models be combined to produce better results?
5. Do heterogeneous ensembles perform better than state-of-the-art methods such as deep learning?

1.5 Contributions of the Research

An effective solution is provided by this research that can predict train delays. This research also benefits passengers as it enables train operating companies to provide them with information about the nature and duration of delays very soon after they occur.

1. A study we conducted and published (Al Ghamdi et al., 2020) examined how heterogeneous models performed better than single models and how powerful the ensemble method can be in terms of accuracy and consistency when compared with single models. A subset of the initial experiment is used, which covers a period of 7 months, along with weather data. This work was applied and tested for each pair of stations. As we have observed, each trip behaves differently, which means that certain parts of the journey are subjected to more delays than others, which could indicate some uncertainty that a model cannot predict. Since no single model would work for all stations, this suggested the need for different algorithms, not just one. Based on literature reviews, we found that our ensemble performed better than the most well-known algorithms. This work is described in Chapter 4.

2. As an extension of our work, we built a heterogeneous ensemble which includes state of the art algorithms such as XGboost within the collection of models, along with established methods such as Random Forest. This work is described in Chapter 5.

3. We investigated the effect of the accuracy of the individual models on overall ensemble accuracy. We also investigated the effect of model diversity on overall ensemble accuracy. We found that accuracy of the individual models was important, but diversity had no effect. This work is described in Chapter 5.

4. We investigated the use of different metrics for measuring diversity. For this we redefined some of the existing metrics developed for regression, for example correlation. Then we redefined some of the existing metrics developed for classification, for example CFD, in order to apply them to

a regression problem. We also applied both pairwise and non-pairwise metrics. This work is described in Chapter 5, and has also been accepted as a paper for *IEEE Transactions on Intelligent Transportation Systems.*

5. We investigated the effect of the number of models on ensemble accuracy and found that the highest accuracy was achieved with a small (3-4) number of models. This work is described in Chapter 5.

6. We investigated the decision making function and compared the use of two different methods for combining the model outputs, namely *averaging* and *weighted averaging.* We found that weighted averaging gave more accurate ensembles. These methods are described in Chapter 3, and their use is investigated in Chapters 4 and 5.

7. We tested two Deep Learning benchmark methods, CNN and Tabnet, to see how well they would work in predicting the by using the DL models. One of these methods, Tabnet, has not previously been applied to train delay prediction. We found that there was no single model that performed better than the other and that when we applied our ensemble framework to deep learning, we found that ensembles of DL models performed better than single DL models. This work is described in Chapter 6.

8. In order to verify our methods, we evaluated them statistically and also tested them on new data collected from different train operating companies. As a result of our tests, we can confirm that our models are robust and generic. This is described in Chapter 7.

1.6 Outline of the Thesis

This section provides an overview of the structure of the thesis.

Chapter 1. Introduction. This chapter describes the background, motivation, and purposes of the study and the contributions of the research.

Chapter 2. Literature Review. In this chapter, we review existing literature relevant to this thesis. This chapter presents an overview of delay, specifically train delays and ensemble methods.

Chapter 3. Research Methodology. The methods and design of the study are presented in this chapter. It also explains the datasets that were used in all of our research.

Chapter 4. Heterogeneous Ensemble for Predicting Train Delay. This chapter presents the empirical investigation of the effects of heterogeneous ensembles and two aggregation functions for combining the outputs of individual models.

Chapter 5. Model Selection Methods for Building Heterogeneous Ensembles. This chapter examines model selection methods. It investigates the effects on ensemble accuracy of individual model accuracy and diversity, and the number of models. Two model selection criteria are proposed and investigated.

Chapter 6. Deep Learning Heterogeneous Ensemble. This chapter examines different Deep Learning benchmark methods tested and applied to the ensemble framework.

Chapter 7. Evaluation and Discussion This chapter presents the results of the research and work conducted for this thesis are discussed and evaluated.

Chapter 8. Conclusions and Suggestions for Further Work This chapter presents the conclusion and future work are discussed.

2 Literature Review

2.1 Related Work

In this review of the literature we will discuss some of the different methods that have been applied to train delay prediction to date. Various methods have been used for predicting train delays, including regression and classification. Regression has been studied much more, with relatively little work having been produced that uses classification.

A recent publication by Spanninger et al. (2022) gives a reasonably comprehensive review of publications on train delay prediction methods. They divide the methods into *event-driven*, which model the dependencies of the train arrivals, departures and other events on the network, and *data-driven* where the train-event dependency structure is not explicitly modelled. They conclude 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 apply in real time.

In this review we will look initially at work using methods that generate individual models, then at those that use methods that generate ensembles of models.

2.1.1 Train delay prediction using single models and hybrid approaches

Stochastic approaches have been much applied in this field. As early as 1994, Carey and Kwieciński (1994) used stochastic approaches to simulate interactions between trains to help avoid the effects of knock-on delays. Later, in 2008, Yuan and Hansen (2007) recommended the use of stochastic methods for estimating arrival and departure delays in Holland. Yuan and Hansen (2007) 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 explore delays resulting from conflicts of routes and the transfer of trains between connections. However, as noted in a recent paper by Li et al. (2022), approaches using probability distribution models have failed to provide accurate predictions of train delay durations when they occurred.

More recently, the use of Bayesian networks has been explored by a number of researchers. Lessan et al. (2019) made use of Bayesian network models for predicting the time 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 a level of accuracy of over 80% was achieved 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 system do not cover such huge distances, neither are they high-speed, and therefore it is not certain how well their approach would generalise to the UK network.

Bayesian networks were also used by Corman and Kecman (2018). These were applied to historical data from Sweden. Their method had the advantage that it was not restricted to static data but was also able to include dynamic characteristics of delays which were constantly fluctuating.

Support Vector Machines are another method which has been used in the field of transport to predict arrival times. In China, Yu et al. (2010) applied this hybrid method to the prediction of bus arrival times. Later, Marković et al. (2015) used SVM as a method for identifying any connections between train delays and railway network qualities. The work by Marković et al. (2015) 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 Marković et al. (2015) were hybrid simulation and machine learning, and multiple regression.

Artificial Neural Networks (ANN) are a class of algorithm which can be used to make predictions and produce train delay information. Yaghini et al. (2013) used ANNs to predict arrivals and departures on the Iranian train network. This entire investigation was completely different from all other research reviewed, because it permitted the implementation of classification in contrast to the more widely applied regression approach. (In a later study, Choi et al. (2016) used the classification method to predict airline delays due to bad weather.)

Yaghini et al. (2013) only used five input features. These were: the starting place, the destination, the route, as well as the month and year in which the journey is taking place. Their method worked very successfully, but we would note that dataset they were using was for a relatively simple railway network.

Extreme Learning Machines (ELMs), Shallow and Deep, were used by (Oneto et al., 2018) for creating a system to predict train delays because ELMs can learn faster than those which use traditional learning algorithms, which may not be fast enough, and they generalise well (Huang et al., 2006). They are also more appropriate for use with Big Data than methods which use univariate statistics because the model adapts and improves when fed external data (Oneto et al., 2018).

Deep learning networks have been used to model train delays as a time-series problem. Huang et al. (2020) combined three types of neural networks, 3D CNN, LSTM, and FCNN to produce a hybrid called CLF-Net which performed better than conventional machine learning models, and state-of-the-art deep learning models in terms of root mean squared error and mean absolute error. Zhang et al. (2021a) used a graph convolutional network model to capture the spatial-temporal characteristics of train operation data. They compared their model with ANNs, SVRs, RFs, and LSTM, and reported that it was better at predicting the cumulative effect of train delays.

While train delay prediction can be treated as a time-series problem, most reaearchers have treated it as a regression problem, which does not have the dependencies that the time series approch has. We will adopt the regression approach, using ensembles, which we will discuss in the next section.

2.1.2 Ensemble approaches

The methods employed in the research we have discussed above all produce single models. Even hybrid approaches, while they may incorporate more than one model, effectively function as a single model.

Ensemble techniques differ from both single models and hybrid approaches in that they employ a committee consisting of several models. They work in a way analagous to a committee of human experts whose combined decision is more reliable than that of one individual. The ensemble approach is based on the fact that while no one individual model may be perfect for solving a problem, a committee of several models can be. Ensemble methods almost invariably outperform single models (Wang, 2008).

Many types of ensemble methods have been developed and they have been applied in many fields, including the prediction of train delays.

Oneto et al. (2016) published a paper that focuses on highlighting the process of how a machine learning algorithm is used. Their approach proved to be a very efficient one and is of particular relevance to our research. It employs kernels, ensembles and neural networks, and works on the method of comparison and contrast between the performances of each. The problem is treated as a time series forecasting one in this model. Weather data is utilized in this model in order to extend the delay prediction model to enhance the performance, and the method uses both the historical and forecasted weather measures for assisting the prediction model. By including forecasted and historical weather information, a rise of 10% in the accuracy was observed. This makes it evident that incorporation of such information is very important, since those research projects which did not make use of such features had lower levels of accuracy. Oneto et al. (2016) uses the Random Forest approach, of which they state, “RF combine bagging to random subset feature selection”. They set the number of trees to 500 and use the ensemble technique. We will employ the ensemble approach in our research because, in comparison to using single classification or regression models, it is expected to perform better.

Wen et al. (2017) also compared the Random-Forest Model to a simple Multiple Linear Regression Model and found the Random Forest Model to have better levels of prediction accuracy. They determined the optimal number of trees to use by examining the error between different tree sizes, because they required an accurate but not overly complex model. Both Wen et al. (2017) and Oneto et al. (2016) state that Random Forest outperforms other approaches, and they suggest that it is the best algorithm for predicting train delays.

Several other authors have also used Random Forest for train delay predictions. Kecman and Goverde (2015) concluded that in their application it outperformed linear regression and decision trees. Gao et al. (2020) used a two stage RF model and found that it increased the accuracy of delay predictions. Nabian et al. (2019) used a bi-level RF approach, while Nair et al. (2019) used a large scale application of RF.

Shi et al. (2021) used the well-known XGBoost algorithm of Chen and Guestrin (2016), with hyperparameter tuning by Bayesian optimization, for predicting train delays. When tested using data for two high speed railway lines in China, it performed better than six other well-known algorithms including random forest, which gave the next best performance.

These 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 heterogenous 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 heterogenous ensemble has models produced by different algorithms and so are therefore methodologically heterogeneous. This means that they are less likely to make similar errors, resulting in a more accurate ensemble.

The only published example of a heterogenous ensemble in the context of predicting train delays that we are aware of is that of Nair et al. (2019). They developed a heterogenous ensemble consisting of three models: random forest, kernel regression and mesoscopic simulation of the network. They only generated one model of each type and performed no model selection. They found that the ensemble performed better than the individual models. However, they also found that their approach was sensitive to hyperparameters and tuning was required. This means that their method would not generalise well.

We would note that the work of Nair et al. (2019) also shows another advantage of the heterogeneous ensemble approach. This is, that *any* modelling algorithm can be used to generate the models including ensemble methods. Thus, it has the potential to bring together models generated by different ensemble methods.

2.1.3 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, such as airports, runways, airspace and boundaries. Problems can potentially arise due to weather conditions, air traffic control issues, mechanical issues, and capacity problems. The origin-destination matrix is complex and delays can be costly for passengers, airlines and other stakeholders. Carvalho et al. (2020) conducted a systematic mapping study of flight prediction research and produced a detailed taxonomy classifying prediction models according to their components. The components covered included air transportation system datasets, concerning airlines, airports or ensemble and carriers, plus information from service providers, regulatory bodies and government agencies. The review considered statistical analysis, most frequently using such techniques as regression models, multivariate analysis, correlation analysis, econometric models and parametric and non-parametric tests. It also reviewed the work of Tu et al. (2008) which used Probabilistic Models encompassing analysis tools which use historical data as the basis for estimating an event's probability of occurrence. Network representation, as used by Abdelghany et al. (2004), was also reviewed, in which graph theory is used to study flight systems with acyclic graphs employed to model airline schedules, detect possible delays and their effect elsewhere in the network. Operational Research can facilitate improved decision making and uses sophisticated methods of analysis such as simulations, optimization and queue theory.

Operational Research was used by Schaefer and Millner (2001) and by Hunter et al. (2007) to consider delays in arrivals and departures under varying weather conditions, and by Soomer and Franx (2008) to calculate the cost of delays.

Most pertinent to the current research, examples of machine learning, exploring the development of algorithms capable of learning from data and making predictions based upon it, are covered in the review by Sternberg et al. (2017). They noted that machine learning methodology is becoming increasingly important in flight systems analysis, particularly for prediction and classification, and the most frequently used methods are neural networks, k-Nearest Neighbor, SVM, random forests and fuzzy logic. For example, Rebollo and Balakrishnan (2014) predicted root delay at US airports over periods of 2, 4, 6 and 24 hours, using random forest and provided comparisons between their methods and regression. They found that as they expanded their forecast horizon, more test errors occurred. Khanmohammadi et al. (2014) 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.

Balakrishna et al. (2008) predicted taxi-out delays at JFK and Tampa Bay international airports by means of a reinforcement learning algorithm: a Markov decision process was used to model the problem, which was then solved using a machine learning algorithm. Their model performed well 15 minutes prior to the scheduled departure time. To predict propagation effect delays at airports, Zonglei et al. (2008) built a recommendation system prediction using the k-Nearest Neighbor algorithm and historical data to identify earlier, similar situations. They found various advantages to their approach, including fast response time and simple, logical comprehension.

Gui et al. (2019) considered a wide range of scope of factors which have the potential to impact on flight delay, and conducted a comparison of a number of machine learning-based models used for general flight delay prediction tasks. The dataset they created for their scheme comprised automatic dependent surveillance-broadcast (ADS-B) messages. These were received, pre-processed, and amalgamated with further data including airport information, weather information and flight schedules. Their design for prediction included various classification tasks plus a regression task. Their experiments indicated that that long short-term memory (LSTM) could cope with the aviation sequence data it received but that a problem arose with overfitting in their dataset, which was limited. However, 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.

Truong (2021) predicted the likelihood of flight delays using data mining and causal machine learning algorithms, in a process, known as USELEI process (Understanding, Sampling, Exploring, Learning, Evaluating, and Inferring). 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 an entire structural causal network from sizeable data sets. Data from various sources were used and the results suggested that predictors had significant effects on the probability of flight delays. These included capacity, reported arrivals and departures, demands on arrivals and departures, efficiency, volume of traffic at both origin and destination locations. Significantly, causal interrelationships among variables in a fully structural network were highlighted along with the way in which these predictors interacted the occurrences of delays resulting from such interactions. The predictive power and precision of the final network was high, with a 91.97% predictive accuracy and the validity of the model was demonstrated in the positive and negative predictive values of 91.56% and 95.45%, respectively, confirming the efficacy of USELEI for the prediction of non-delay as well as delay occurrences. Truong (2021) found that possible scenarios could be assessed by using sensitivity analysis and causal inference, which can then help lessen the chance of delays.

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. Models used for predicting delays for buses have tended to rely on limited data sources and simplistic model architectures, resulting in poor performance when predicting across an entire network. Models have therefore tended to focus on individual routes. However, rather than just looking at individual routes, Shoman et al. (2020) proposed prediction of bus delays across the entire bus network using a deep learning-based approach. Large quantities of heterogenous bus transit and vehicle probe data were used. The researchers used entity embeddings which made it possible for their model to fit functions and, at the same time, learn patterns from different types of data streams (categorical and continuous). One model was produced which was able to classify factors which impact on delays covering numerous routes and different stations simultaneously, at different times of the year and different times of day, in Saint Louis, Missouri. Their modelling framework performed well for delay prediction covering multiple stops. The mean absolute percentage error was just 6%. The high performance of the model of Shoman et al. (2020) over multiple routes was based on its use of heterogenous data and its ability to simultaneously model continuous and categorical data using deep learning.

Zhang et al. (2021b) analysed 2 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 the related transport areas of aviation and roads it is clear that similar approaches are being used in several transport areas. Therefore it is highly likely 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 for this thesis have strong potential to be used in other transport areas. However, the existing publications adopted the single model approach, and so would have the same limitations as existing single model solutions to train delay prediction. The disadvantage of the single model approach is that it is less accurate and less reliable than the ensemble approach. The ensemble approach is able to utilise the advantages of multiple methods and avoid their disadvantages.

2.2 Ensemble Learning

Predictive machine learning is a field where algorithms fit a model such as an artificial neural network to existing data, which can then be used to predict the values of new data. Common uses are classification, where the class of new samples is predicted and regression where the values of a continuous target variable are predicted.

An ensemble of models is a set of learning models whose predictions are combined in some way in order to obtain a more generalisable prediction by combining their predictions together (Dietterich, 2000). It was noted by Schapire (1990), that it is possible to create a strong learner by combining several weak learners into an ensemble. It has also been demonstrated by Hansen and Salamon (1990) empirically, that a prediction error made by an ensemble of multiple models can be less than the error of the best single model in the ensemble. In addition, other researchers (Alyahyan et al., 2016) have shown that an ensemble will outperform individual models (Brown et al., 2005; Wang et al., 2003). Furthermore, an ensemble offers a high level of reliability (Wang, 2008).

Two conditions must be met in order for an ensemble to outperform the individual members in it. First, the base learners need to be accurate (so that they outperform random guessing), and secondly, they need to be diverse (in other words, they should make different errors when making new predictions). (Dietterich, 2000; Hansen and Salamon, 1990)

An ensemble can be categorized into two types. A homogeneous ensemble is built with models generated by one type of base learner only, e.g. decision trees. In contrast, a heterogeneous ensemble is built using models generated by several different kinds of base learners.

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 comparing the effectiveness of heterogenous and homogenous ensembles (Alyahyan et al., 2016; Aytuğ, 2018; Gashler et al., 2008; Smętek and Trawiński, 2011; Nanglia et al., 2022) have shown that heterogeneous ensembles do perform better than homogenous ensembles in terms of both accuracy and reliability.

These studies have demonstrated that heterogeneous ensembles are generally more accurate and more reliable, which is particularly important in a critical industrial application such as train delay prediction, where consistent and robust predictions are more important than the absolute prediction accuracy, as long as the accuracy is within a tolerable limit, e.g one minute.

Because of the potential benefits of heterogeneous ensembles in terms of diversity, and therefore accuracy, we chose to investigate their use for the prediction of train delays in this study.

2.2.1 Ensemble Construction Methods

Much effort has been applied into developing effective ensemble learning methods and assessing their performance. In this section we will discuss ensemble construction methods and factors that affect ensemble performance.

Ensembles can be constructed using a variety of methods. We will now discuss the most widely used manipulation methods that are applied to a wide variety of ensemble learning algorithms. The review by Dietterich (2000) discusses reasons why ensembles may perform better than single models and methods for ensemble construction. We will now discuss three of these methods for ensemble construction: Data Level Manipulation, Algorithm Level Manipulation and Features Level Manipulation.

Data Level Manipulation

With data level manipulation, a number of training datasets are created by resampling the original data according to some sort of sampling distribution. By using a specific algorithm, a classifier or regression model is then constructed for each training set. Dietterich (2000) found that this approach is particularly effective for unstable learning algorithms such as decision trees, neural networks, and rule-based learning algorithms. (In contrast, linear regression, nearest neighbor, and linear threshold algorithms are very stable and so data level manipulation would be less effective with them.) Breiman (1996) introduced *bagging*, which is one of the most popular ensemble methods using this approach.

Algorithm Level Manipulation

There are two ways of applying algorithm level manipulation. The first applies the same method several times to the same training data in order to produce multiple models. This approach produces what is known as a *homogenous ensemble* where the models are all of the same type, for example an ensemble of decision trees, or of neural networks (Tan et al., 2016). There are various ways of ensuring that the models are not all identical, for example changes to the network topology or the weights of the links between neurons may result in the production of different models, in the case of an ensemble of artificial neural networks

The second way to apply algorithm level manipulation generates an ensemble by combining models produced by multiple learning algorithms, that is, different algorithms are applied to the data and the models combined into an ensemble. The ensembles generated this way are referred to as *heterogeneous ensembles.*

Features Level Manipulation

This approach involves repeatedly selecting a subset of feature inputs from the training dataset. Then this subset us used by the base classifier to generate a model. The models are then combined to produce the ensemble. The selection of the subset can be random or be based on certain criteria. When the dataset contains many repeating features, it can be particularly useful. An example of this approach is the Random Forests algorithm of Breiman (2001), where input features are manipulated and decision trees serve as the base classifiers (PN, 2006). Another example is the XGBoost algorithm of Chen and Guestrin (2016).

2.2.2 Popular Existing Homogeneous Ensemble Methods

The majority of ensemble methods generate homogeneous ensembles. The most commonly used approaches are Bagging, Boosting and Random Forest. XGBoost is another popular ensemble method that gives good performance. These are all described below.

Bagging

Bagging (from **b**ootstrap **agg**regat**ing**) was proposed by Breiman (1996).This method can be applied to regression and classification problems. It works by reducing the variance by sampling multiple subsets of the training data with replacement. As a result of this process, each subset has a $(1 - 1/e)$ probability ($\approx$ 63%) of containing any individual sample in the original training dataset Skurichina and Duin (2002). After generating individual models from each bagged subset, using the base learner, (e.g. decision tree,) the final output of the ensemble is obtained by averaging or voting on the outputs of the multiple models obtained.

Boosting

Boosting was proposed by Freund et al. (1999). The aim of boosting is to build a strong classifier by generating multiple models using a weak learner, and then combining their outputs in order to get a better performance. In boosting the models are generated sequentially by successive iterations of the algorithm, whereas with bagging they can be generated in parallel. One of the most well know boosting algorithms for classification is AdaBoost (from **Ada**ptive **Boost**ing). After each boosting iteration the weights applied to individual samples are increased for samples misclassified at the previous iteration. Thus, the algorithm focuses on the more difficult samples. The final ensemble output is made by weighted voting.

Random Forest

Random Forest is an ensemble algorithm that was proposed by Ho (1995) generates a variety of decision trees that can be used for classification and regression. It employs bagging of the features during tree construction and the trees are constructed by sampling with replacement samples from the original training dataset. Each tree node sample from the training dataset contains an attribute that is selected according to a random process. The ensemble prediction is achieved by averaging for a regression model and by majority voting for classification.

XGBoost

XGBoost (From e**X**treme **G**radient **Boost**ing) was proposed by Chen and Guestrin (2016) and is a boosting algorithm that has proven to be one of the most powerful ensemble algorithms. It was designed to be very efficient and employs Newton gradient descent. It has been used in many domains (Asselman et al., 2021; Li et al., 2020b; Liu et al., 2022), demonstrating that it has the ability to perform fast and accurate results. It can use parallel and distributed computing to speed up the learning process, resulting in a faster modelling process.

2.2.3 The factors that influence ensemble performance

The study by Wang (2008) discussed how an ensemble's accuracy can be influenced, by what factors and to what degree, and listed the following factors:

1. Accuracy of individual models.
2. Decision Making Function.
3. Diversity of individual regressors.
4. Number of models within an ensemble.
5. Methods for selecting models.

We now discuss these factors.

Accuracy of individual models

Ensemble accuracy depends on the accuracy of each individual model in the ensemble. In general terms, members whose accuracy is higher than a random guess should to be used, this would be expected to result in an ensemble having a more accurate result than a single model (Wang, 2008).

Decision Making Function

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 the ensemble (Wang, 2008).

(In our research we developed two functions for this, one employing *averaging* and the other employing *weighted averaging*.)

Two types of decision making function can be employed: *fusion* and *selection*.

Using the fusion approach, the individual predictions are merged to produce the final ensemble output. The term *fusion* refers to procedures that combine the decisions of all ensemble members. In contrast, ensemble *selection* procedures combine only a portion of the available decisions to create the ensemble output.

Diversity of individual regressors

The ensemble learning approach depends on the individual models being able to collectively give a better prediction than any one singe model. However, in order to do this they need to be diverse, that is, although they may make errors, the individual models must not make the same errors in terms of their predictions. Diversity measures have been devised with the aim of quantifying the overall amount of difference between the individual models making up an ensemble. Thus, a diversity measure is a measure of difference between models within an ensemble in terms of their predictions.

While there is no generally agreed absolute definition of diversity, there are several diversity definitions and methods for assessing diversity for classification problems (Wang, 2008). Ensemble diversity measures are typically recognized as representative approaches for reflecting failure independence among ensemble team member models, and are predicted to have a consistent association with ensemble accuracy (Wu et al., 2021). Diversity measures are divided into pairwise and non-pairwise. Pairwise measures only consider two models and non-pairwise can work with all the models. At present, no specific and widely accepted standard diversity measurement technique exists for classificaton or for regression problems.

For regression problems, it is more challenging in measuring diversity among the models, which is probably the reason that most diversity measures are defined for classification problems but not applicable to regression ensembles as they require the categorical outputs and cannot handle the real value output in regression problems. One review (Dutta, 2009) evaluated several measures for evaluating diversity in regression ensembles by using correlation coefficient, covariance, dissimilarity measure, Chi-square, and mutual information, etc.

In regression problems, diversity is explained based on ambiguity decomposition proposed by Krogh and Vedelsby (1994) who prove that the ensemble predictor guarantees a lower squared error than individual predictors. A typical study by Liu et al. (2000) 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 (Dutta, 2009).

In the literature, several ways for calculating diversity have been described. However, the majority of published measures of diversity are not applicable to regression ensembles because they were developed for classification ensembles and cannot be directly transferred to regression. One study (Dutta, 2009) evaluated several measures for evaluating diversity in regression ensembles (correlation coefficient, covariance, dissimilarity measure, chi-square, and mutual information). For this study we redefined some of the metrics evaluated by Dutta (2009) and also modified probably the most effective non-pairwise diversity measure for classification—the Coincident Failure Diversity(CFD) of Partridge and Krzanowski (1997), for regression problems. To the best of our knowledge this is the first time that CFD has been applied to a regression problem.

It is worth noting that the estimation of diversity is challenging and the existing diversity definitions may not necessarily have a linear relationship with the actual useful diversity, however, we would expect them to broadly reflect some level of the diversity, i.e. an ensemble with a high diversity measure value would be more diverse than one with a lower value.

It is important to note that different diversity measures will give values across different ranges, therefore when using diversity as a component of the fitness function, we normalised each diversity metric to range from 0 to 1, with 1 being the highest level of diversity. The metrics that we used in our study are described below:

Correlation In statistics, the correlation (COR) 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. As a result of the 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 models.

For a given pair of models x and y, where r = correlation between them, i is an index over the N data samples x_i = output value of x, y_i= output value of y, $\bar{x}$ = mean of x, and $\bar{y}$ = mean of y,

$$r = \frac{\sum(x_i - \bar{x}) \times \sum(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \times \sum(y_i - \bar{y})^2}}. \tag{2.1}$$

Covariance The covariance (COV) indicates whether the two variables x and y vary together in a correlated manner. Unlike the correlation, whose values are limited to -1 and +1, the values of covariance are unbounded, and can be anything between -∞ to +∞,

(The symbols are the same as in Equation 2.1)

$$cov = \frac{\sum(x_i - \bar{x}) \times \sum(y_i - \bar{y})}{N - 1}. \tag{2.2}$$

Disagreement The disagreement (DIS) 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 firstly converted into binary values with a threshold value θ, as per Equation 2.3,

$$f(x) = \begin{cases} 0, \ x < \theta \\ 1, \ x >= \theta \end{cases} . \tag{2.3}$$

Disagreement is then calculated as per Equation 2.4,

$$Disagreement = \frac{N^{01} + N^{10}}{N^{00} + N^{01} + N^{10} + N^{11}}, \tag{2.4}$$

where N^{11} represents the samples correctly predicted, and N^{00} represents the samples incorrectly predicted by a pair of predictors M_1 and M_2, respectively. Samples that are correctly predicted by M_1 and incorrectly by M_2 are represented by N^{10}, and samples which are correctly predicted by M_2 but incorrectly byM_1 are represented by N^{01}.

CFD The *Coincident Failure Diversity* score was defined by Partridge and Krzanowski (1997) to measure the probability that two or more models fail on a test data simultaneously and was also used for binary variables. But in a similar manner as mentioned above, it can be modified to handle continuous variables for regression problems, as shown below. CFD is calculated as:

$$CFD \ = \ \sum_{m=1}^{M} \frac{(M-m)}{(M-1)} \times f_m, \tag{2.5}$$

where, M = total number of models in ensemble, m = a number of models (between 1 and M), and:

$$f_m = \frac{\text{number of samples incorrectly predicted by } m \text{ models}}{\text{number of samples incorrectly predicted by at least one model}}. \tag{2.6}$$

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 other members. So an ensemble with a maximum diversity should produce a perfect answer, although its members may make some mistakes.

CFD has previously been reported to be the best non-pairwise measure of diversity (Bian, 2006). In a study involving ten non-pairwise diversity measures, Bian and Wang (2007) noted that CFD values are relatively independent from the number and accuracy of models in the ensemble.

Number of models within an ensemble

As with a committee of human experts, an ensemble with a large number of members can be expected to be more accurate than one with a small number. However, having more models in the ensemble will only be beneficial if they are diverse from one another, so that they do not make the same mistakes and the optimum size of the ensemble will vary according to the dataset characteristics, model accuracy and model diversity. Thus, a small ensemble with high diversity could give better performance than a large one with low diversity.

Methods for selecting models

Many ensemble selection approaches have been published in the literature in order to reduce computation time, as well as categorizing them as follows: Cluster-based, Optimization-based and Ordering-based selection (Zhou, 2012). Mohammed et al. (2022) demonstrate how to build low-complexity, small-size, and highly accurate ensembles using ensemble selection. These results indicate that the proposed method of ensemble selection is a suitable alternative to large-size ensembles.

Cluster-based. As a general rule, clustering is used to partition individual learners into groups of learners, where individuals within the same group behave similarly, while different groups behave differently with great diversity (Zhou, 2012).

Optimization-based. In order to find the most efficient solution, ensemble selection is formulated as an optimization problem. Learners are included in the final ensemble with the aim of improving generalization (Zhou, 2012).

Ordering-based. In this method, the regressor or classifiers in the pool of models are sorted according to some criteria such as accuracy or diversity.

Using ordered aggregation, an empirical study by Martínez-Muñoz et al. (2008) showed that selection ensembles generated through ordered aggregation can be both competitive and robust compared with computationally more expensive approaches that select optimum or near-optimal subensembles directly. A study of 26 UCI data sets was conducted by Lu et al. (2010) to examine how ensemble pruning or selection can be used to construct subensembles based on the order in which the ensemble members outperformed the original ensemble. They showed that the subensembles generated outperformed the original ensemble.

In our study we will use this approach, as overall it is the most appropriate for our purposes.

2.3 Summary

In this chapter we have discussed previous work done in the field of train delay prediction and also looked at work in the related fields of air and road passenger transport delay prediction. Random Forest has been the most effective algorithm for train delay prediction overall and has been used by several researchers.

The work of Oneto et al. (2016) who employed the use of many detailed features, such as weather conditions and position of other trains on the network contrasts with that of of Yaghini et al. (2013), discussed above, who used a relatively simple dataset. The different approaches of Yaghini et al. (2013), who used data for the Iranian network, and Oneto et al. (2016), who focused on the Italian rail network, highlight the importance of tailoring the method employed and the features selected to suit the specific problems of the network. The Iranian network has a much lower level of complexity than the Italian network. Since the UK rail network is extremely complex with a considerable number of crossovers, it might therefore be thought that Oneto's method would be better suited to it than that of Yaghini et al. (2013).

Several other authors have also used Random Forest for train delay predictions. Kecman and Goverde (2015) concluded that in their application it outperformed linear regression and decision trees. Gao et al. (2020) used a two stage RF model and found that it increased the accuracy of delay predictions. Nabian et al. (2019) used a bi-level RF approach, while Nair et al. (2019) used a large scale application of RF. These applications demonstrate how powerful the Random Forest ensemble method is and for this reason we will include it as one of the base learners to be used in building our heterogeneous ensembles, and also as a benchmark method for comparison purposes.

We have discussed ensemble learning methods and how they might be applied to the problem of train delay prediction. We noted that heterogeneous ensembles have been shown to give better performance than homogeneous ensembles. But we also noted that almost all existing applications of ensembles to train delay prediction were homogenous ensembles, and the only application of heterogeneous ensembles that we are aware of would not generalise well.

The factors affecting ensemble performance have been considered, and we especially noted the importance of diversity and discussed measures of ensemble diversity. A number of diversity metrics were discussed and it was noted that the CFD score of Partridge and Krzanowski (1997) has been reported by previous researchers to be the best diversity measure. We discussed existing work on the use of ensemble selection to improve ensemble performance, and noted that ordering-based selection is, overall, the best approach.

3 Research Methodology

3.1 Introduction

This chapter provides an overview of the design and methods used in this research in order to accomplish its aim. We will present the ensemble framework that was used for all our experiments and describe its components. We also present the datasets that we used to test our methods. We also present the methods used for evaluating the significance of the results.

The rest of the chapter is organized as follows:

Section 3.2 will present the research research design and methods.

Section 3.3 describes the dataset and preprocessing.

Section 3.4 will provide details of the methods for evaluating and presenting the results.

Section 3.4.4 describes the statistical tests for comparison.

Section 3.5 will give a summary of the chapter.

3.2 Research Design and Methods

The aim of our research is to develop heterogeneous ensemble machine learning methods that can be used for predicting train delays. The research will be carried out as a case study using train data on the Greater Anglia line between Norwich and London Liverpool Street stations.

The ensemble framework is introduced in Subsection 3.2.1 and described in detail in Subsections 3.2.2–3.2.6.

The data used covers a period of 2 years and 5 months between 2017 and 2019. It is described in Section 3.3.

3.2.1 Heterogeneous Ensemble Framework for Predicting Train Delay

The overall framework of our Heterogeneous Machine Learning Ensemble (HMLE) is shown in Fig. 3.1. It consists of five phases: (1) data preprocessing and feature extraction, (2) data partitioning, (3) modelling, (4) Model Selection, and (5) building the ensemble. These phases are described in Subsections 3.2.2–3.2.6. The framework will be adapted for each experiment. A detailed description of the framework used for each experiment will be given in the relevant chapter.

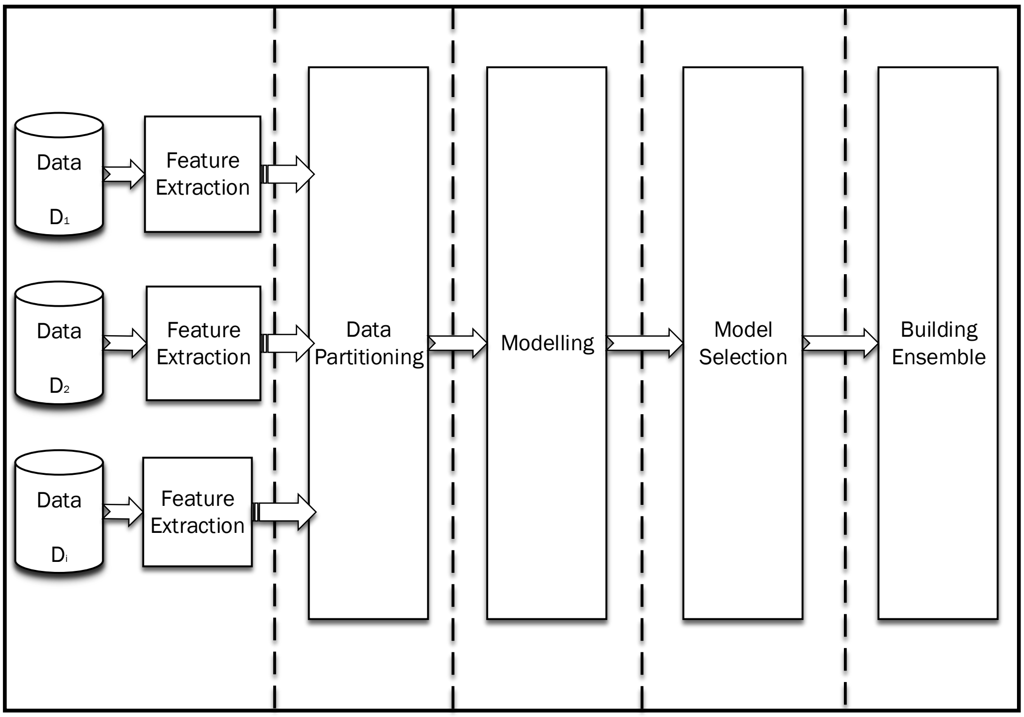


Figure 3.1: Ensemble Framework, showing the stages in ensemble construction.

3.2.2 Data Preprocessing and Feature extraction

The dataset we have used consists of a flat table where rows represent samples, i.e., timepoints where a train passes a point on the railway line and columns represent the attributes of the data. The database and its preprocessing are described in detail in Section 3.3.

In our system, the input data X will consist of features derived from two sources, one from the train company, which was used in all our experiments, and one from the weather company, which was used in our initial experiments $X = \{x_1, x_2, .., x_j, .., x_J\}$, where J is the total number of features. For example, x_1 could represent the departure delay for the last station and x_2 the departure delay for the current station. Different features were used in different experiments.

It should be noted that while several datasets can be merged into one and then used to generate multiple models that are then combined into an ensemble, it is not possible to modify or "update" that ensemble to take into account new data, e.g. by adding more models generated using the new data. If new data is added, then a completely new ensemble must be generated using the new, combined dataset.

3.2.3 Data Partitioning

In this phase, the dataset will be split into training, validation and testing sets. For all partitioning random seeds will be used in order to ensure reproducibility of results and shuffling the dataset. To conduct this research, the initial dataset was partitioned into training, validation and testing subsets.

The *Train_test_split* function provided by Scikit-Learn (Buitinck et al., 2013) was used to achieve this splitting of the data. When performing experiments with multiple runs, we used a different random seed for each run, in order to achieve different partitions and to ensure reproducibilty.

3.2.4 Modelling

In the modelling phase, a collection of models, CM, will be produced. $CM = \{m_1, m_2, \ldots, m_i, \ldots, m_I\}$ where each m_i will come from a regression model. The models produced at this stage will be the candidates for being selected to build the ensemble. A wide range of algorithms could be used for generating the models, including deep learning, existing ensemble methods such as Random Forest and conventional methods e.g. regression. By using different algorithms to generate the models a heterogeneous ensemble can be generated. All models will be learned from the training data. In this study, we are mainly concerned with ensemble learning, therefore these learning algorithms are regarded as black boxes in our research. For this study, we used a range of different types of standard base learner which are available in Scikit-Learn (Buitinck et al., 2013) and other libraries, and are well-known methods that have previously been used in train delay prediction. In each chapter describing experiments, the base learners used are listed. This phase of modelling entails two important considerations: (1) What is a suitable base learner? (2) What is the optimum number of models?

In our experiments we used the default parameter settings for the base learner algorithms that we used. This was because our focus was on the development of ensembles and to investigate whether they had ability to do better than individual models working alone. To attempt to optimise parameters for every algorithm would have entailed a large amount of work which would have given little benefit, as the default values generally give good results.

3.2.5 Model Selection

The model selection is the fourth step of the proposed framework. In connection with this step, it is important to note that the models in the ensemble for predicting train delays must have two important characteristics—one is *accuracy* and the other is *diversity*.

Model selection involves choosing from among many different models stored in the collection of models. Different aspects were taken into account during the development of this stage. We began by investigating the size of the ensemble and how this may affect its performance. We then explored the effect of the accuracy of the models and finally, we examined the effects of their diversity.

In order to carry out the investigation of model selection two different selection methods were devised. The first (MSM1) only considers the accuracy, the second (MSM2) considers both accuracy and diversity, which was computed using different metrics. These selection methods will be fully described in Sections 5.2.1 and 5.2.1, respectively, in Chapter 5.

3.2.6 Building Ensemble

Building the ensemble involves combining together some or all of the models selected in order to build an ensemble that will produce the best prediction for the delay at a station by combining the results from the M models chosen. For this process, two techniques will be used, *Averaging* and *Weighted Averaging*, which are described below.

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 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. (Equation 3.1.)

$$y_a = \frac{\sum_j^M y_j}{M}, \tag{3.1}$$

where, y_j is the output from member model j, M is the number of models in the ensemble, $1 \leq j \leq M$.

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 then chosen metrics, e.g. R^2 or *MAE*. In this study, for our initial experiments we used the metrics described in Section 4.4.1, for our later experiments, described in Chapters 5 to 7, we used R^2. The procedure is as follows (using R^2 as an example): for model j, its weight w_j is computed with equation 3.2, 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 $a < 1$, 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, as per Equation 3.2,

$$w_j = a_j R_j^2. \tag{3.2}$$

The weighted output of the ensemble y_w can then be computed by Equation 3.3,

$$y_w = \frac{\sum_j^M y_j \times w_j}{\sum_j^M w_j}, \tag{3.3}$$

where, y_j is the output from member model j, w_j is the weight for model j, M is the number of models in the ensemble, $1 \leq j \leq M$.

3.2.7 Prediction Modelling Scheme

Delay Prediction Representation

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, of train T_i at an intended station S_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 calculates the target variable y,

$$y = \Delta t(T_i, S_j) = t_{aa}(T_i, S_j) - t_{pa}(T_i, S_j). \tag{3.4}$$

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 and when it is negative, it means that a train arrives early. 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.

These predictions are based on the running time. With these features, different models are trained as candidate models for building heterogeneous ensembles.

3.3 Datasets and Pre-processing

3.3.1 Datasets

A dataset covering a period of 2 years and 5 months between 2017 and 2019, train running data was collected from the Historic Service Performance data repository (HSP) (NRE, 2018). Weather data was also collected from the weather stations nearest to the railway stations in question, which were used in the initial experiment Chapter 4 . This dataset was originally collected by Mr. Douglas Fraser of the School of Computing Sciences, University of East Anglia. For our initial investigations, discussed in Chapter 4, a subset of this dataset covering a period of 7 months was used. It is described in Section 4.1. For the remaining work the entire dataset was used, excluding the weather data. Table 3.1 lists the features from the raw data that were selected from the dataset of train running data.

Table 3.1: Description of the key features of the raw data.

Key	Description
RID	A rid is a unique identifier for a journey.
tbl	Code for train location
wta	Planned arrival times
wtd	Planned departure times
arr_at	Actual arrival times
dep_at	Actual departure times
wtp	Planned passing point
pass_at	Actual passing point
canc_reason	A train delay code that represents the reason for the cancellation

3.3.2 Pre-processing

The data used need to be as complete, accurate, and consistent as possible, in order achieve the best results. Therefore detailed data cleansing was performed. We did not perform any imputation of missing data. This is because we had enough data to carry out our research, even when all records with missing data were removed, and data imputation can introduce errors. We preferred to ensure that the dataset was as free from errors as possible.

1. Firstly all the csv files were concatanated into a single file.

2. Then journey level data were processed to distinguish between stopping stations and passing points/non-stopping stations. (At a stopping station passengers can get in and out, whereas the train just passes a passing point/non-stopping station).

3. We identified the stations where the train stopped in each complete journey. We then made a list of all the various combinations of stations stopped at. This enabled us to identify journeys where the trains had stopped at the same combination of stations.

4. We converted columns wta, wtp , arr_at and dep_at to datetime format, which makes calculations easier.
5. Duplicate records were removed.
6. The column indicating whether a journey was cancelled was removed.
7. Records with missing values were removed.
8. We ran numerous logical checks to identify records containing errors. (E.g. some trains were listed as arriving earlier than they left, this was one example of inconsistency.) These records were deleted.
9. Features were then extracted. (Different combinations of features were extracted for different experiments.)
10. Columns of numeric data were adjusted to zero mean and unit standard deviation using scikitlearn StandardScaler function.
11. Categorical data, such as day of week and day of month, were all converted to binary using one hot encoding.

3.4 Evaluation Methods and Metrics

The evaluation of results is a critical part of all research and different researchers into train delay prediction have adopted different methods and metrics to calculate performance. The choice of metric in train delay prediction is important because while predicting the delay is a regression problem, the criterion for a train being officially delayed is normally based on an absolute cut off, e.g. 1 minute, and delay prediction is therefore also a classification problem. Furthermore, data relating to timetables and recorded delay values are rounded to e.g. the nearest minute. (See Oneto et al. (2018).)

It should be noted that while the current PPM standard on the UK network declares a train to be late only if it is more than 5 minutes behind schedule, this can be the result of a series of smaller delays at several stations, so for our evaluation we will use a smaller cutoff. Different measures for reporting train delays have been used by different researchers, the measures we will use are described below.

3.4.1 Percentage correct prediction after rounding (%CP)

The continuous output produced by regression models is unlikely to be whole integers. Output values are therefore rounded to the nearest integer. Comparison is then made with the actual outputs in the test dataset to assess the number of journeys the model accurately predicted to the minute following rounding.

3.4.2 Percentage within 1 minute prediction after rounding (%|P| <1)

This is the same as the above evaluation, using comparison to assess whether the model predicted exactly or within 1 minute either way.

3.4.3 Regression Metrics

We employed four common metrics to evaluate the prediction accuracy of the proposed technique for predicting next arrival delay in this study. These were, mean absolute error (MAE), mean squared error (MSE), root mean squared error ($RMSE$) and R-squared (R^2):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \tag{3.5}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{3.6}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \tag{3.7}$$

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y}_i)^2}, \tag{3.8}$$

where N is the number of samples and i is the index of the sample, $1 \leq i \leq N$.

3.4.4 Statistical Tests for Comparison

The performance of the ensemble methods was evaluated using the Friedman test to compare all the methods used. Critical difference diagrams were used to view the results.

Critical Difference Diagrams

Critical difference (CD) diagrams (Demšar, 2006), enabled statistical comparisons between our results to be made. In CD diagrams the results are presented on a numerical range and results that do not differ significantly are grouped using solid black lines. Two results are significantly different if they are not found in any common group.

The comparison is a two step process. The null hypothesis is that the average ranks of k methods do not differ significantly, the alternative hypothesisis is that the mean rank of least one regressor is different.

In step one, the k methods are ranked according to their performance; then, the average ranking of each algorithm is calculated. The null hypothesis, can be tested using the Friedman test (Equation 3.9),

$$Q = \frac{12l}{k(k+1)} \cdot \left[\sum_{j=1}^{k} \bar{r}_j^2 - \frac{k(k+1)^2}{4} \right], \tag{3.9}$$

where Q is the Friedman test statistic, l is the number of runs, $\bar{r}_j^2$ is the rank of the jth of k algorithms and the statistic is estimated using a chi-squared distribution with $k - 1$ degrees of freedom. The rejection criterion used was $p < 0.05$.

This calculation is often conservative, so Demšar (2006) used an alternative statistic, devised by Iman and Davenport (1980):

$$F_F = \frac{(l-1)Q}{l(k-1) - Q}, \tag{3.10}$$

where Q is the Friedman test statistic.

Under the null hypothesis this equation follows an F distribution with $(k - 1)$ and $(k - 1)(l - 1)$ degrees of freedom.

Step two is applied if the null hypothesis is rejected. Here, *post-hoc* pair-wise Nemenyi tests are used. These indicate significant differences between the average ranks of two regressors if their difference is equal to or more than the *critical difference*, CD, as per Equation 3.11,

$$CD = q_a \sqrt{\frac{k(k+1)}{6l}}. \tag{3.11}$$

Here q_a is calculated from the difference in the range of standard deviations between the sample with the smallest value and the sample with the largest value. The CD diagram presents the results.

The CD diagram (see for example Figure 4.9) presents the algorithms in rank order according to accuracy from highest to lowest, where the algorithm with highest accuracy is in position 1, the next in position 2, and so on.

The thick lines on the CD diagram link the algorithms that do not differ significantly. Hence, it is easy to identify which algorithms differ significantly from each other, because they are not linked by the same thick line.

3.5 Summary

In this chapter we have given an overview of our research methodology. We have presented the framework for generating heterogeneous ensembles and given details of its phases. The five phases are: (1) data preprocessing and feature extraction, (2) data partitioning, (3) modelling, (4) selecting models, and (5) building the ensemble.

(1) Data preprocessing involves cleaning the data to remove incomplete records, standardising numerical data and converting categorical data to binary. No imputation of missing data was performed in order to avoid introducing errors into the data we used.

(2) Data partitioning was used to split the data into training, validation and testing datasets.

(3) Modelling employs different algorithms to generate machine learning models for inclusion in the ensemble.

(4) Selection of models was performed to decide which models to include in the ensemble. Different criteria (accuracy and diversity) were used to to do this.

(5) The models were combined together in order to build the ensemble. Two different criteria, averaging and weighted averaging, were used for performing this.

We have described the data preprocessing and feature extraction, the data partitioning, the model selection and the methods for combining the models. The datasets have been described. Finally, we have described the statistical tests and critical difference diagrams we will use to help in the evaluation of our results.

4 Heterogeneous Ensemble for Predicting Train Delay

4.1 Introduction

Over recent years ensemble learning methods have gained a lot of attention and have become one of the most popular ways for solving real data analysis problems in a wide range of research and competitions. Despite the fact that many algorithms have been applied in the past for the prediction of train delays, there has been only one application of a heterogeneous ensemble (HE) in this area.

A heterogeneous ensemble contains models that are different because they have been generated by different base learner algorithms. Therefore, for our initial experiments we devised an ensemble that was able to generate models from a wide variety of base learner algorithms and combine them together.

The rest of the chapter is organized as follows:

Section 4.2 Describes the proposed heterogeneous ensemble.

Section 4.3 Describes the dataset and feature extraction.

Section 4.4 Presents the experimental design.

Section 4.5 Presents and discusses the experimental results.

Section 4.6 Summarises the chapter.

4.2 Proposed heterogeneous ensemble

For these initial experiments we devised a framework (Figure 4.3) adapted from our overall framework described in Section 3.2.1. The main difference was that for these initial experiments we did not apply or investigate any model selection in this experiment. (Chapter 5 will cover the process of model selection.)

The framework consists of five phases: (1) data preprocessing and feature extraction, (2) data partitioning, (3) modelling, (4) collection of models, and (5) building the ensemble.

For phase (1) data preprocessing was performed as described in Section 3.3.2, then feature extraction was performed as described below in Section 4.3.2.

For phase (2) the dataset was partitioned into training, validation and testing subsets. This was performed using a random seed to enable reproducibility, and also to enable different random partitions to be made by using different random seeds.

Phase (3) was the modelling phase. Here different regression algorithms were used to generate different regression models from the training data. (The fact that each model was generated from a different algorithm is what makes the ensemble heterogeneous.)

In phase (4) the models generated in phase (3) were all put into a collection of models.

Phase (5) was the final stage where the models were combined together into the heterogeneous ensemble. This was performed using either averaging or weighted averaging.

4.3 Dataset and features

Because this was an initial experiment we used a subset of the dataset described in Section 3.3. It contained data for a seven month period from 01-01-2017 to 01-08-2017. In total, it contained data relating to 5499 valid journeys that could be used in the modelling process. (The total number of station-to-station journey instances was 33293.)

Figure 4.1 shows the proportions of journeys that were early, on time and delayed. By on time we mean that there is no difference between the scheduled time and the actual time. It can be seen that 27.4% were not delayed, 21.3% were on time and 51.4% were delayed.

In addition, for this dataset, we also included weather data from a dataset collected from the weather stations nearest to the railway stations in question, provided free of charge by Weatherquest Ltd. This data was preprocessed by Mr Bradley Lewis Thompson of the University of East Anglia for his MSc dissertation (Thompson, 2018) who extracted data for the nearest hour for the weather station nearest to the rail station from which the train departed for each record in the train delay dataset. The data were only complete for temperature and precipitation, so these were the only features that we used.

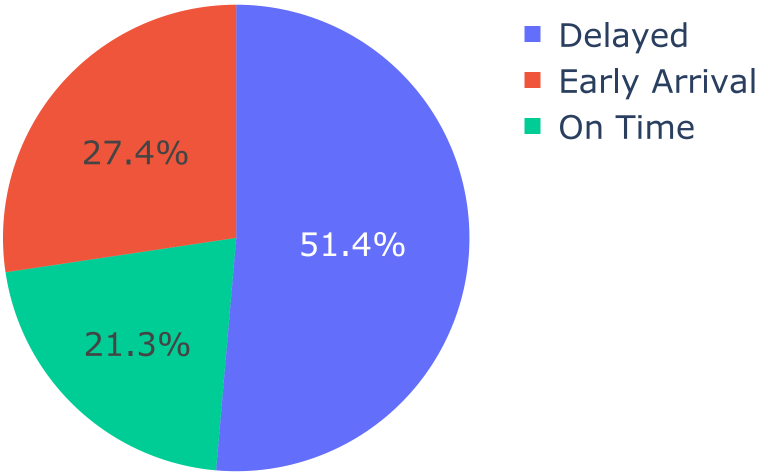


Figure 4.1: Proportions of all NRW – LST journey delays. The pie chart shows the percentage of journeys that were early, on time or delayed.

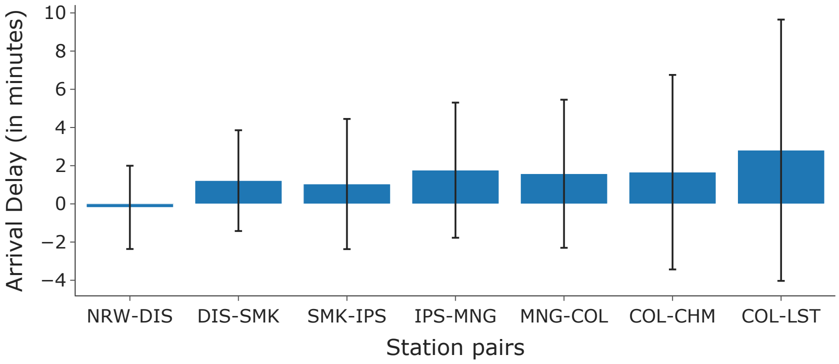


Figure 4.2: Bar chart of arrival delays between the station pairs, NRW–DIS, DIS–SMK, SMK–IPS, IPS–MNG, MNG–COL, COL–CHM, COL–LST. The average arrival delays are shown, with the SDs indicated by error bars.

4.3.1 Station pairs

The full list of stations for the Norwich to London Liverpool Street journey is: Norwich (NRW), Diss (DIS), Stowmarket (SMK), Ipswich (IPS), Manningtree (MNG), Colchester (COL), Chelmsford (CHE) and London Liverpool Street (LST). However, not every train stops at every station. Therefore data are available for journeys between the following pairs of stations: NRW–DIS, DIS–SMK, DIS–IPS, SMK–IPS, IPS–MNG, MNG–COL, COL–LST, COL–CHM.

The arrival delays are shown for NRW–DIS, DIS–SMK, SMK–IPS, IPS–MNG, MNG–COL, COL–CHM, COL–LST in Figure 4.2, with the SDs indicated by error bars. The smallest average delay is for NRW–DIS, which is the first stage of the journey, and the largest average delay is for COL–LST, LST being the terminal station on the journey. This indicates that the delays accumulate through the course of the journey. Similarly, the SDs are larger for stations that are later in the journey, again this would be the result of the accumulating delays. This pattern is to be expected and the difficulty of the prediction task can be seen from this figure.

4.3.2 Features derived for Model Input

Normally some sort of feature selection and engineering would be applied to any prediction problem. This dataset contains a large number of records and a relatively low number of features. Train delays are caused by many factors and each one may have only a small impact. We found that only one feature was in itself a strong predictor of the arrival delay, which was the departure delay.

For this study we were able to derive the following features from the raw data which had an impact on the delay:

- The departure delay for the last station.
- The departure delay for the current station.
- Day of the Week: Monday = 1, Sunday = 7.
- Day of the Month: The day of the month = 1–31.
- Weekday/Weekend: True = Weekday; False = Weekend.
- On-Peak/Off-Peak: True = on-peak; False = off peak.
- Hour of the Day: the hour period of the day the journey 0–23.
- Temperature: for the relevant hour, reading in degrees Celsius taken from the nearest weather station for the current station.
- Rainfall: Levels of precipitation, according to the nearest weather station.

While there are many factors that could impact on the train delays, such as number of passengers, type of train, distance between stations and driver experience, these were not available in the data we had.

The effect of including the weather data was investigated in Section 4.5.2.

4.4 Experimental Design

In these initial experiments we were investigating the feasibility of building ensembles containing models generated by different base learner algorithms.

In order to do this we used the framework described above and performed the following steps.

1. First, feature extraction from the dataset was performed as described in Section 4.3

2. We then divided the data into training, validation, and testing sub-datasets. This is a commonly used practice, where the validation set is used to check the accuracy of the generated models during training, and the test set is used to test the accuracy of the final ensemble. There is no standard ratio for dividing up the dataset, but normally the validation and test sets would each contain between 10 and 20% of the samples. A balance has to be struck between having as many samples as possible for training, and sufficient to perform accurate validation and testing. We decided that using the split 70%:15%:15% did this. We split the dataset using a random seed. Each experiment was repeated five times, using different random seeds, and the results were averaged.

3. We applied a series of regressor algorithms from the scikit-learn library to generate models. Various algorithms were selected to build the models in these ensembles. We deliberately chose a wide variety of algorithms in order to make the ensemble as heterogeneous as possible. Of the fifteen chosen, twelve generate individual models; Linear Regression (LR), Multi-layer Perceptron (MLP) (Rosenblatt, 1961), ElasticNet (EN), k-nearest neighbours Regressor (KNN). Support Vector Regression (SVR), Kernel Ridge Regression (KR), Gaussian process regression (GPR), Bayesian Ridge (BR) (Box and Tiao, 2011), Stochastic Gradient Descent (SGD) (Jain et al., 2018), Lasso (Santosa and Symes, 1986), Ridge (Hoerl and Kennard, 1970) and LassoLars (LL); and three produce

essentially homogeneous ensembles; Extra Trees Regressor (ET), Gradient Boosting Regression (GBR) (Friedman, 2001) and Random Forest (RF) (Ho, 1995).

As noted in Section 3.2.4, 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.

4. All the models generated were put into a collection of models.

5. These generated models were then used to form an ensemble. Here two different decision making functions were applied to combine the outputs of the models in a heterogeneous ensemble to produce a final output. Because of the different decision making functions the ensembles are named as Averaging Ensemble (AE) and Weighted Ensemble (WE). The weighted averaging was performed using the performance of the models on the validation data. The performance was calculated using the metrics described in Section 4.4.1. Finally the completed ensemble was tested using the testing data. The performance of the individual models on the validation data and the performance of the ensemble on the testing data were calculated using the metrics described below in Section 4.4.1.

In a further experiment we used the same procedure and investigated how beneficial it was to include weather data in the dataset. For this we only used the data for the journey from Norwich to Diss.

Overall we conducted a total of 1220 experimental runs for the work describe in this chapter.

Figure 4.3 illustrates the framework of our HE.

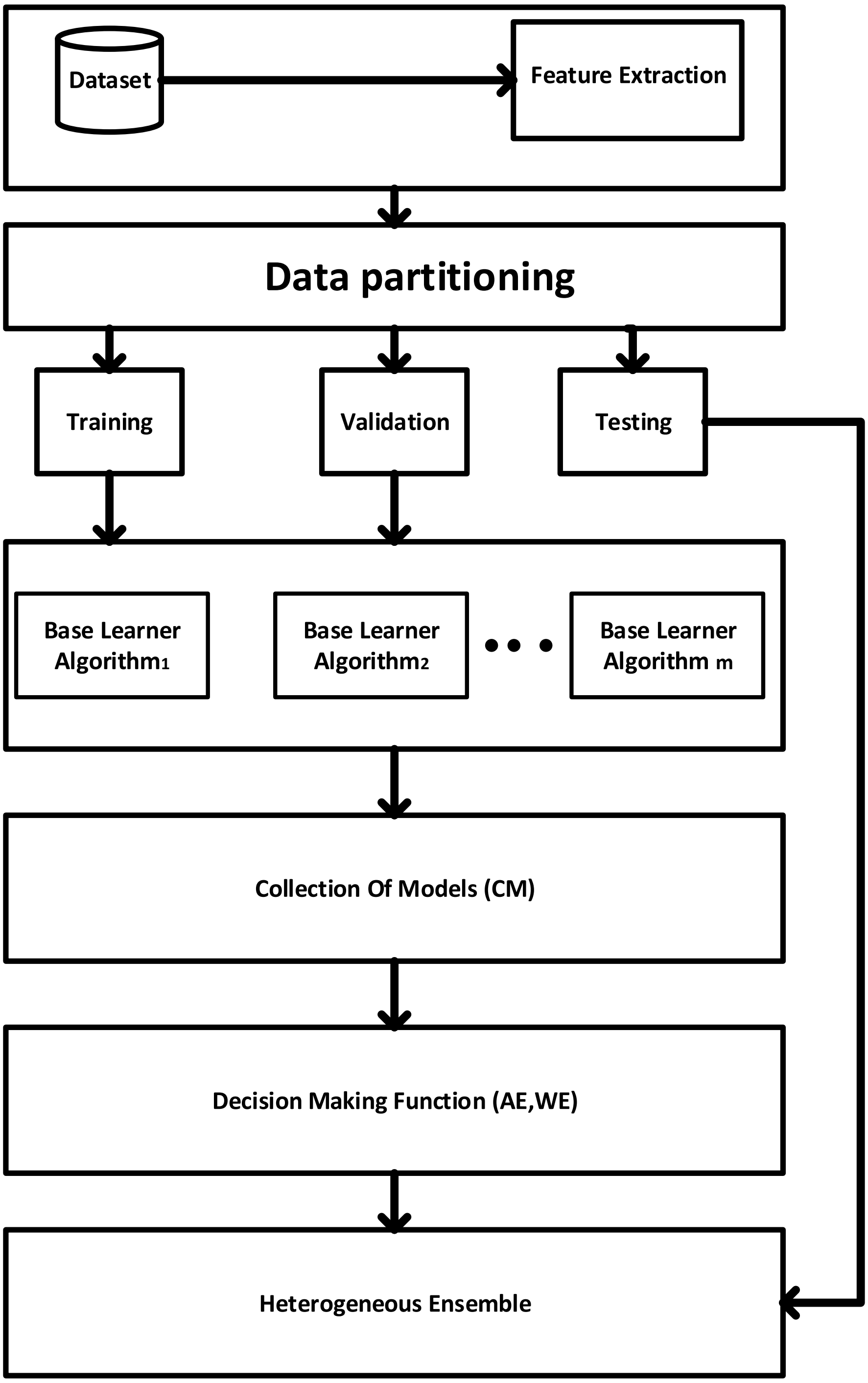


Figure 4.3: Framework of Heterogeneous Ensemble showing the process of generating the ensemble.

4.4.1 Evaluation Methods

In this initial experiment and investigation, we evaluated the methods using the following metrics.

Percentage correct prediction after rounding (%CP). The continuous output produced by regression models is unlikely to be whole integers. Output values are therefore rounded to the nearest integer. Comparison is then made with the actual outputs in the test dataset to assess the number of journeys the model accurately predicted to the minute following rounding.

Percentage within 1 minute prediction after rounding (%|P| <1). This is the same as the above evaluation, using comparison to assess whether the model predicted exactly or within 1 minute either way.

4.5 Results and discussion

A number of experiments were carried out to investigate the performance of our ensemble models.

4.5.1 The general results for all available station pairs

Table 4.1 presents the results for the delays between all available pairs of stations, measured by both percentage of correct prediction and by percentage of prediction accurate to within one minute. Figures 4.4 and 4.5 present this data in graphical form.

Figure 4.4 shows the percentage of correct prediction between pairs of stations.

In these results we present the average values of accuracy measures of single models, the averaging ensembles and the weighted ensembles. The average of single models result is the value obtained from averaging the accuracy obtained from generating models using each algorithm separately. The averaging ensemble result is the value obtained when all the algorithms are used to generate models and those models are put into an ensemble, and assigned equal weights. The weighted ensemble result is the value obtained when all the algorithms are used to generate models and those models are put into an ensemble, and the models are weighted according to accuracy.

The worst performance was for the COL–LST journey, and we will discuss possible reasons for this later. For the remaining pairs of stations, the average accuracy of single models never produced more than 50% correct predictions and falls as low as 25%. With the averaging ensembles, the prediction accuracy showed a marked improvement, the lowest average percentage correct prediction being 30% and the highest 71%. The best results were produced by the weighted averaging ensembles. Here the standard deviation was constant. Its lowest correct prediction, between Diss and Ipswich was 47% and the highest, between Ipswich and Manningtree was 82%.

The results for COL–LST are noticeably worse than any other pair of stations. This could be due to a number of reasons. For example, LST is the terminal station on the line and there could be delays as the trains are waiting to enter. However, we note that in this situation the weighted ensemble achieved a much better performance than either the average of single models or the averaging ensemble. Thus, the weighting mechanism can be seen to work effectively, even when the performance of the individual models was poor.

It is noteworthy that the pattern of correct predictions was remarkably consistent for each of the three methods. 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.

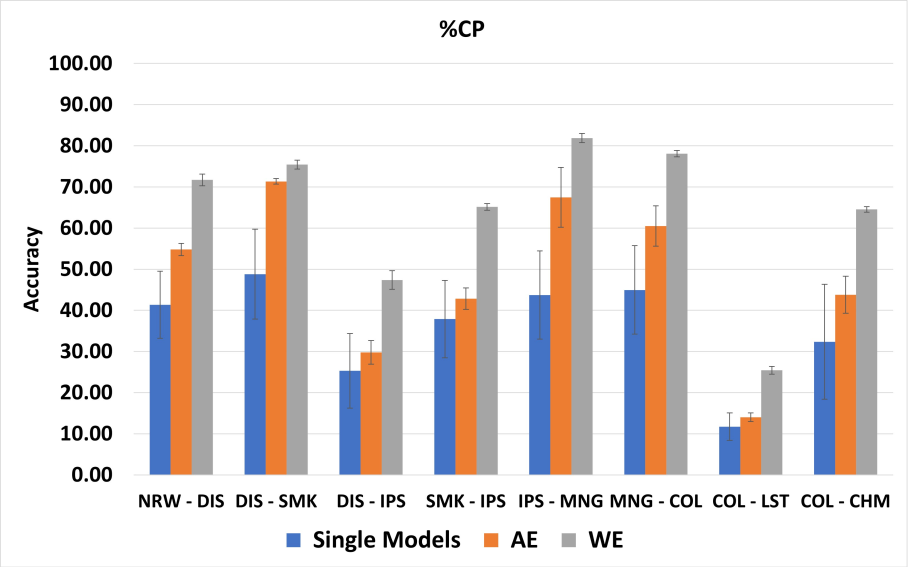


Figure 4.4: The results obtained for predicting the delays between pairs of stations using average of single models, averaging ensembles and weighted averaging ensembles. The SDs of the values are indicated by bars.

A considerable improvement in accuracy was demonstrated when the average prediction was accurate to within one minute, as shown in Fig 4.5. Once again, the pattern was similar to that of correct prediction but the lowest mean was 65%, for the average of single models. For weighted averaging, the accuracy levels reached as high as 98%

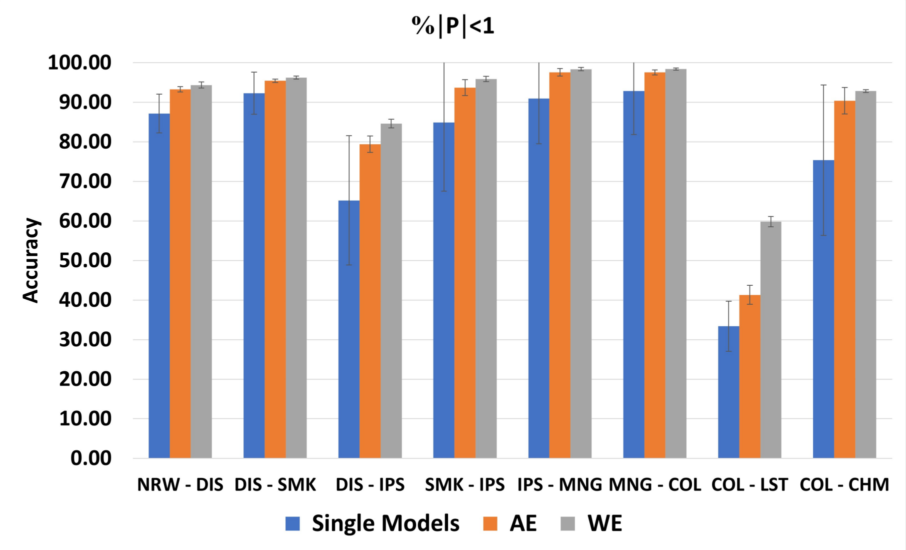


Figure 4.5: The results for correct prediction within one minute for the delay between pairs of stations obtained using average of single models, averaging ensembles and weighted averaging ensembles. The SDs of the values are indicated by bars.

The results presented in Table 4.1, Figure 4.4 and Figure 4.5 are the means and the standard deviations of the predictions over five runs. Table 4.2 shows the results obtained for the individual fifteen base learner models with the five partitions of the data. From this table it can be seen that no single model gives the best performance in each run, although some models perform better than others more often.

Table 4.1: Comparison of the results for the delays between all available pairs of stations. Showing % CP and %$\lvert P\rvert$<1 for average of single models, averaging ensembles and weighted averaging ensembles.

Section	Average of Single Models				Averaging Ensemble				Weighted Averaging Ensemble			
	% CP		%$\lvert P\rvert$<1		% CP		%$\lvert P\rvert$<1		% CP		%$\lvert P\rvert$<1	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
NRW - DIS	41.34	8.17	87.12	4.88	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
DIS - IPS	25.30	9.05	65.16	16.34	29.77	2.88	79.37	2.09	47.37	2.26	84.61	1.12
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 - LST	11.69	3.35	33.36	6.36	14.00	1.06	41.29	2.40	25.42	0.94	59.82	1.31
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

4.5.2 Investigating the usefulness of the Weather Data

In order to investigate the usefulness of the weather data we conducted two sets of experiments, using the journey from Norwich to Diss. The first used all the features, the second omitted the weather features (temperature and precipitation). Each set was run five times with different data partitions to test the consistency of prediction.

Using All Features

Fig 4.6 shows results for correct delay prediction and for correct delay prediction within one minute, for average of single models, averaging ensembles, and weighted ensembles, for Norwich-Diss, using all the features for prediction. It also shows the standard deviation for each result, indicated by the error bar. Average of single models performed least well, with a mean accuracy of only 41%. The averaging ensembles produced a mean percentage accuracy of 55%, while the best accuracy levels were produced the weighted average, at 72%. It was interesting to note that when a margin of error of one minute was allowed, there was a lower difference between the three methods. Nevertheless, the single models were still worst, while the weighted ensembles performed best. However, the difference between the averaged and the weighted ensembles is only about 1%.

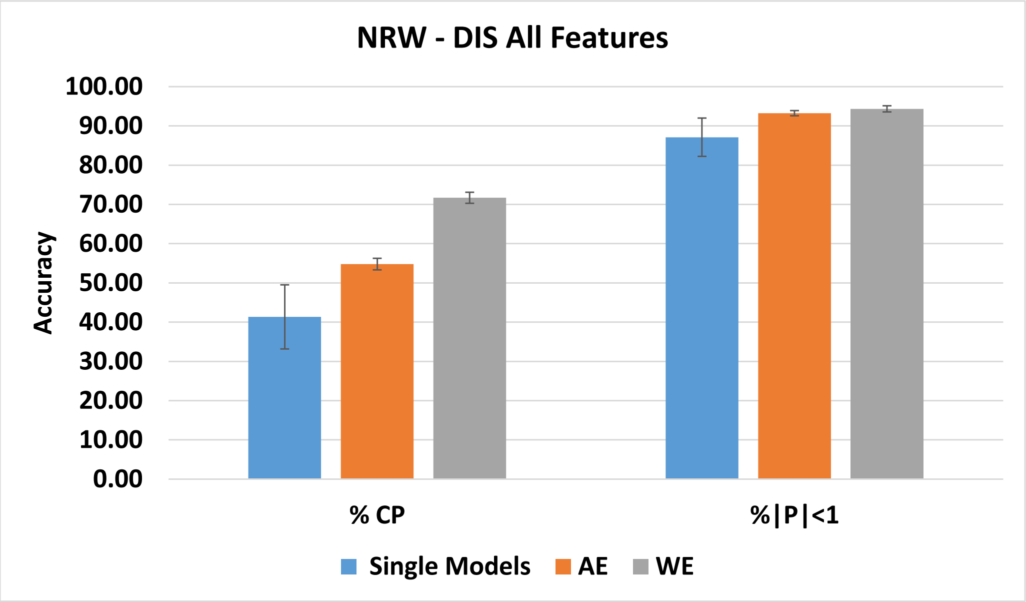


Figure 4.6: The results for correct prediction and correct prediction within one minute for the delays between Norwich and Diss obtained using average of single models, averaging ensembles and weighted averaging ensembles, where all the features were used in training. The SDs of the values are indicated by bars.

Removing Weather Data

In our inputs, some were defined as weather features as described in Section 4.5.2. In order to investigate whether they were useful or not in our modelling, this experiment removed these features from the dataset, then trained the models and build two types of ensembles. In Figures 4.7 and 4.8 it can be seen that removing these weather features decreased the accuracy of the ensemble. These results indicate that weather features do capture some useful information that is learned by ensembles.

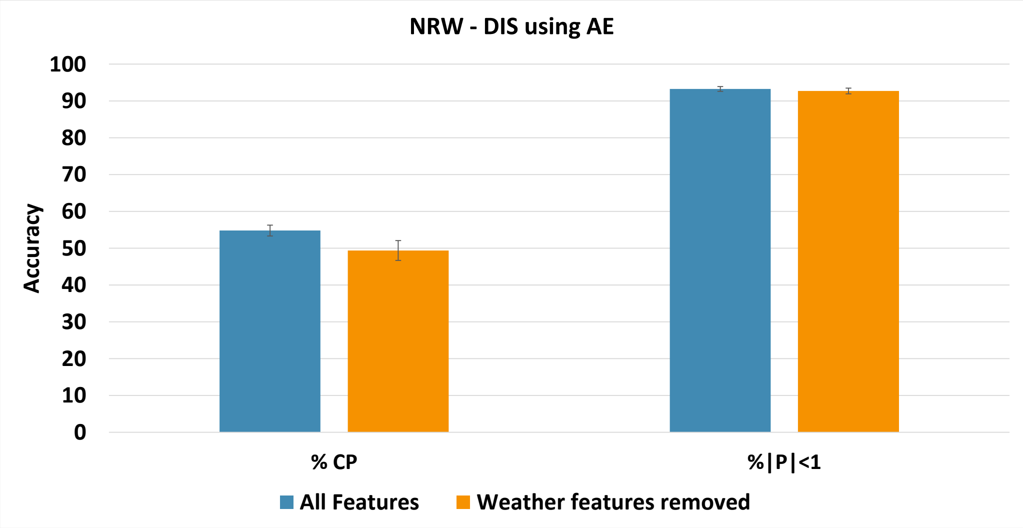


Figure 4.7: Results produced for prediction of arrival time between Norwich-Diss, using averaging ensembles, when using all features, and when weather features were removed. The SDs of the values are indicated by bars.

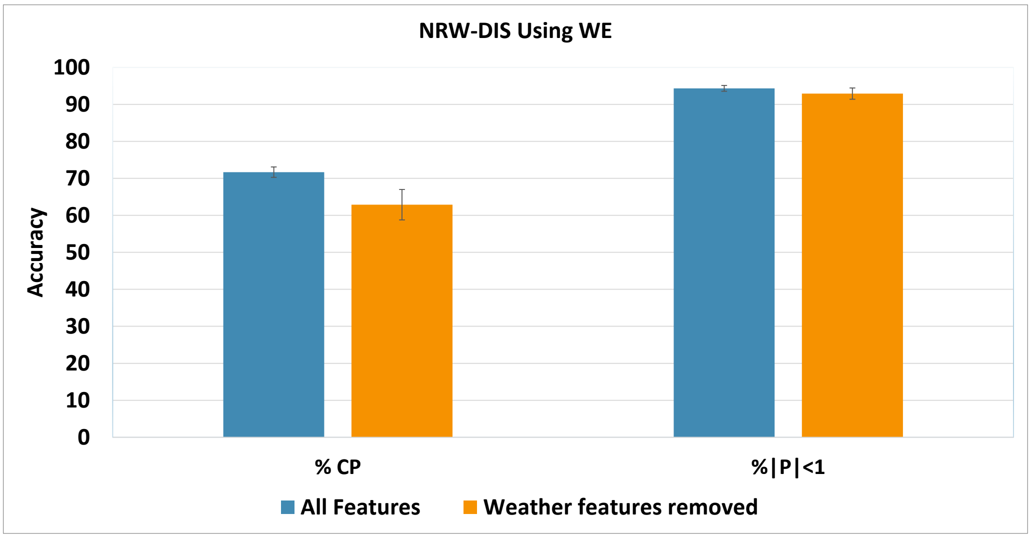


Figure 4.8: Results produced for prediction of arrival time between Norwich-Diss, using weighted averaging ensembles, when using all features, and when weather features were removed. The SDs of the values are indicated by bars.

Table 4.2: Results obtained for NRW–DIS for the individual fifteen base learner models with five different partitions of the data. The best performing model for each run, as indicated by both the correct percentage and percentage correct within one minute predictions is indicated by **bold** type.

	Run 1			Run 2			Run 3			Run 4			Run 5	
Model	% CP	%$\|P\|$<1	Model	% CP	%$\|P\|$<1	Model	% CP	%$\|P\|$<1	Model	% CP	%$\|P\|$<1	Model	% CP	%$\|P\|$<1
RF	50.64	**92.88**	RF	47.18	**91.09**	RF	49.09	88.36	RF	51.09	88.82	RF	49.91	88.36
MLP	29.55	70.18	MLP	31.18	71.91	MLP	27.55	70.36	MLP	31.55	73.45	MLP	28.64	68.55
BR	40.18	89.82	BR	42.64	88.82	BR	40.55	89.27	BR	42.18	89.27	BR	42.18	89.27
LR	39.73	84.36	LR	43.27	86.18	LR	39.18	84.45	LR	41.27	85.73	LR	40.36	84.36
SVR	**53.36**	91.55	SVR	51.18	90.27	SVR	**53.64**	**90.55**	SVR	52.55	**90.82**	SVR	54.09	91.36
KR	40.36	87.27	KR	43.64	87.73	KR	40.73	86.36	KR	41.36	87.64	KR	51.42	**94.30**
GPR	33.55	88.45	GPR	35.09	88.27	GPR	32.45	88.18	GPR	33.91	88.36	GPR	31.64	88.91
SGD	29.82	82.82	SGD	37.09	86.64	SGD	37.82	88.00	SGD	34.36	83.00	SGD	46.09	89.82
Lasso	33.36	88.45	Lasso	34.64	88.27	Lasso	32.36	88.18	Lasso	33.73	88.27	Lasso	31.55	89.09
ET	50.36	87.91	ET	48.18	87.18	ET	47.55	86.64	ET	48.00	87.09	ET	48.36	85.27
GBR	49.55	88.91	GBR	47.27	89.45	GBR	48.27	88.09	GBR	51.73	89.36	GBR	48.91	87.82
Ridge	40.45	87.27	Ridge	43.55	87.73	Ridge	40.73	86.27	Ridge	41.36	87.64	Ridge	40.73	86.55
KNN	53.27	90.91	KNN	**52.27**	90.00	KNN	**53.64**	90.09	KNN	**53.36**	90.09	KNN	**54.64**	90.55
LL	33.36	88.45	LL	34.64	88.27	LL	32.36	88.18	LL	33.73	88.27	LL	31.55	89.09
EN	33.36	88.45	EN	34.64	88.27	EN	32.36	88.18	EN	33.73	88.27	EN	31.55	89.09

4.5.3 Critical Comparisons

The statistical tests described in Section 3.4.4 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 (Oneto et al., 2016).

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 in second place, Random Forest third and individual models fourth. The critical distances among these four methods are represented by the critical distance diagrams Fig. 4.9 and Fig. 4.10. 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 averaging ensembles), AE (averaging ensembles) and RF, and the second group includes, AE, RF and average of single models.

Fig. 4.9 presents the results obtained with %CP. With this metric our WE ensembles gave the highest accuracy and they were statistically different from the average of single models. The AE ensembles and RF were also more accurate than the average of single models, but not statistically significantly different. Ensembles are used because, in principle, they can give more accurate performance than single models, although not all ensembles actually achieve this. From these results it can be seen that our WE ensembles have achieved better accuracy than single models. Both RF and our AE ensembles were also more accurate than the single models, but overall only our WE ensembles were statistically better than the single models.

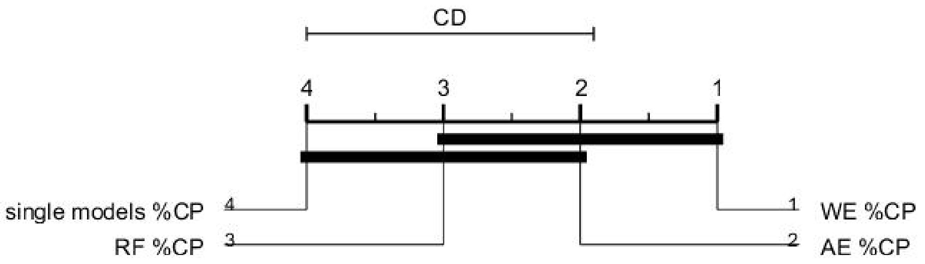


Figure 4.9: Critical difference comparison between our ensembles, average of single models and Random Forest ensembles, using correct prediction measure %CP.

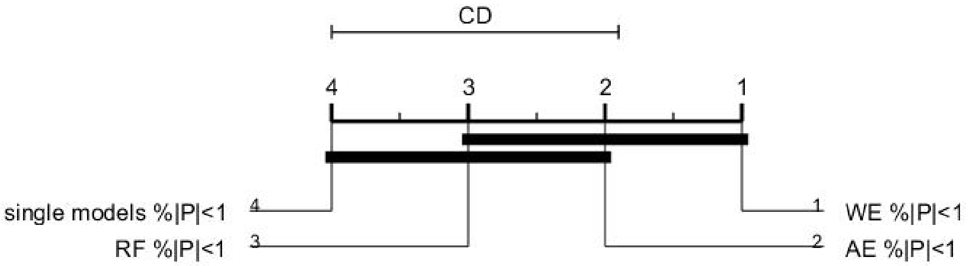


Figure 4.10: Critical difference comparison between our ensembles, average of single models and Random Forest ensembles, using the measure of correct prediction within one minute %|P| <1.

Fig. 4.10 presents the results with %|P| <1. It can be seen that the rank order of the four methods is the same as with %CP, the WE ensembles being the most accurate, then the AE ensembles, followed by RF and with the average of single models being the least accurate. As with %CP, the only statistically significant difference was that the WE ensembles were more accurate than the single models. Thus with both performance metrics, our WE ensembles achieved significantly more accurate performance than the single models.

4.6 Summary

We built heterogeneous ensembles by using 15 models in combination, using an averaging decision making function. We also used a weighted averaging decision making function to see whether this produced any improvement in the results. The experiments were run, in five sets, for eight pairs of stations between Norwich and London Liverpool Street. The average for correct prediction from single models never rose above 50%. But averaging ensembles showed improved accuracy over average of single models, and the weighted ensembles produced the best overall results. A similar pattern was noted when the mean prediction metric was accurate to within one minute.

Two sets of experiments were then conducted, to investigate the benefit of the weather data, with each run five times for the Norwich to Diss section of the journey. The first experiment used all features for single models, averaging and weighted averaging ensembles, the second did the same but with the weather data removed. Overall, removing weather data led to a decrease in accuracy for both averaging and weighted ensembles. The patterns were similar for the correct prediction and for predictions to within one minute.

In conclusion, it can be stated that averaging ensembles produce better results than both average of single models and random forest, and that weighted averaging ensembles show improved accuracy over averaging ensembles. Our ensembles appear to learn better and produce more accurate results when all features are included.

These initial investigations and experiments thus produced encouraging results. These results demonstrated that our proposed ensemble methods are technically feasible and effective.

For these initial investigations we have only used accuracy to assess the performance of our ensembles. In the next chapter, where we will investigate the effect of ensemble size, and the use of accuracy and diversity as model selection criteria when building heterogeneous ensembles, we will employ additional standard criteria, such as R^2 to assess their performance.

5 Model Selection Methods for Building Heterogeneous Ensembles

5.1 Introduction

In the previous chapter we reported our initial experiments on using heterogeneous ensembles for analysing train delay data. Our initial results were encouraging and showed that heterogeneous ensembles performed better than Random Forest. In this chapter we will describe experiments investigating methods for model selection when building heterogeneous ensembles. We have called these ensembles Model Selecting Heterogeneous Ensembles (MSHE).

The rest of the chapter is organized as follows:

Section 5.2 Describes the proposed model selection methods.

Section 5.3 Describes the dataset and feature extraction.

Section 5.4 Presents the experimental design.

Section 5.5 Presents and discusses the experimental results.

Section 5.6 Summarises the chapter.

5.2 Proposed model selection methods

For our work in this chapter we extended the framework described in Chapter 4 by adding a model selection step. The new framework for our MSHE is illustrated in Figure 5.1. In this figure the additional model selection step can be seen between the collection of models and the decision making function.

For this model selection step we propose two model selection methods (MSM) based on two different criteria to build the heterogeneous ensemble. The first—

MSM1—only considers the accuracy, and the second—MSM2—considers both accuracy and diversity.

The purpose of employing model selection is ensure that only the best models are used in the ensemble. A large variety of base learners can be used to construct models, but it is not known how well they will perform on a given dataset. By constructing models on the training data and then measuring their performance on the validation data an assessment can be made on their effectiveness and a decision made as to which models to include in the ensemble. Our model selection methods are described in detail below in Section 5.2.1.

We investigated the use of two different model selection criteria: accuracy and diversity. For the measurement of diversity we investigated pairwise and non-pairwise measures. Because several of these measures were developed for classification ensembles they cannot be directly applied to regression ensembles, so we adapted them to work in the regression context. These diversity measures are explained in detail below in Section 5.2.2.

5.2.1 Model selection methods

Ensemble model selection attempts to select a subset of the suitable models from a collection of the trained models to build an ensemble with the aim of achieving the maximum accuracy with as fewer models as possible (Mohammed et al., 2022).

This is because more models require more resources (time and space), so when building an ensemble, it is thus beneficial to prune models while preserving accuracy and diversity (Tsoumakas et al., 2008).

In our research we proposed two model selection methods (MSM) based on two different criteria to build heterogeneous ensembles. The first, Model Selection Method 1 (MSM1), only considers the accuracy, and the second Model Selection Method 2 (MSM2), considers both accuracy and diversity.

MSM1

This selection method only considers the accuracy of individual models. 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 the ensemble Φ. Then the performance 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 on-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 models by producing an accuracy plot over the growth of an ensemble from empty to the full size, as shown by the figures in Results Section.

This selection method can be generalised by setting up a selection batch size, say V. That is, in every iteration, V 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 sizes 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 $V = 1$, with an intention of evaluating the contribution of each individual model.

Algorithm 1 for **MSM1**

```
Input: Collection of trained models, $\boldsymbol{CM}$, validation data $\boldsymbol{Val}$
Output: The best ensemble $\Phi_{best}$
$B$=count($\boldsymbol{CM}$)
for $i = 1$ to $B$ do
    calculate Accuracy $R^2$ on $\boldsymbol{Val}$
end for
sort $\boldsymbol{CM}$ in descending order according to their accuracy $R^2$
for $i = 1$ to $B$ 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
```

MSM2

This selection method takes account of both accuracy and diversity when selecting models. First, the highest accurate model (HAM) is selected from the collection of models (CM). Then the second model is chosen with the highest diversity model (HDM) to 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 trained models, CM, Diversity-metric, validation data Val
Output: The selected models $\Phi$
$B$=count(CM)
for $i = 1$ to $B$ 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 $B$ do
    calculate diversity (HAM, $CM_i$))
end for
sort CM in descending order according to their diversity
for $i = 1$ to $B$-1 do
    select first $i$ models and add them to a new set called $NCM$
    add model combination(HAM,$NCM$) to $\Phi$
end for
```

5.2.2 Diversity Measures used with MSM2

For the diversity measures to use with MSM2 we chose Disagreement, Covariance, Correlation and CFD. Of these CFD is a non-pairwise measure, the others are pairwise.These measures were described in detail in Chapter 2. As noted there, it is important to note that different diversity measures will give values across different ranges, therefore when using diversity as a component of the fitness function, we need to normalise each diversity metric to range from 0 to 1, with 1 being the highest level of diversity. In order to do this we have redefined Correlation and Covariance, as described below.

Algorithm 3 for **MSM2b** non-pairwise

```
Input: Collection of trained models, CM, Diversity-metric, validation data Val
Output: The selected models $\Phi$
$B$=count(CM)
for $i = 1$ to $B$ 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 $B$-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
```

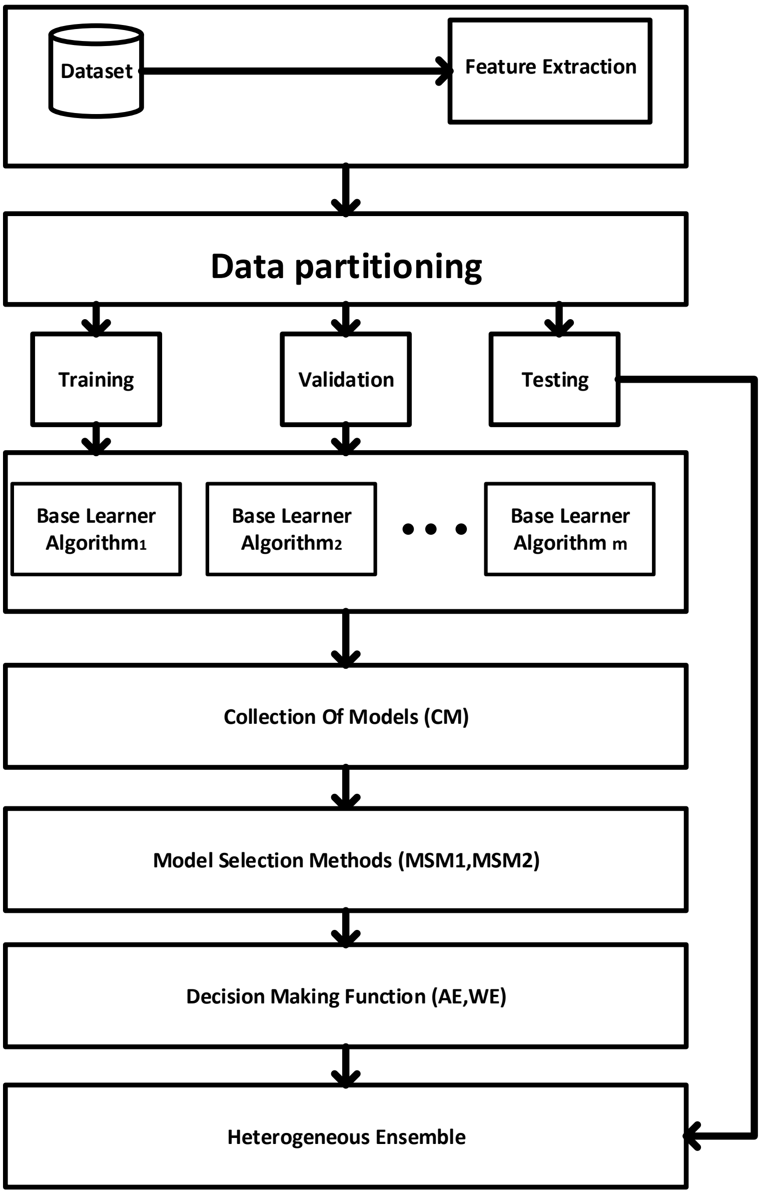


Figure 5.1: Framework of Model Selecting Heterogeneous Ensemble showing the process of generating the ensemble.

Correlation

In order to use this as a diversity measure between a pair of models, which we call as the correlation diversity D_r we have redefined it as 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,

$$\begin{aligned} &D_r = \tfrac{1-r}{2}, \\ &\text{where, } r = \tfrac{\sum(y_{ik}-\bar{y_i})\times\sum(y_{jk}-\bar{y_j})}{\sqrt{\sum(y_{ik}-\bar{y_i})^2\times\sum(y_{jk}-\bar{y_j})^2}}. \end{aligned} \tag{5.1}$$

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.

Covariance

Similarly we have redefined Covariance. The covariance indicates whether 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 $-\infty$ and $+\infty$, which can be difficult to interpret or be used as a diversity measure, we need to convert it to a limited and meaningful representation. We have employed the sigmoid function to convert the range of possible covariance values to (0, 1) and we define the covariance diversity D_v as follows:

$$D_v = \frac{2e^{-|cov|}}{1 + e^{-|cov|}}, \tag{5.2}$$

$$\text{where, } cov = \frac{\sum(y_{ik} - \bar{y}_i) \times \sum(y_{jk} - \bar{y}_j)}{N - 1}. \tag{5.3}$$

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.

5.3 Dataset and features

For this experiment we used the full dataset described in Section 3.3.1. This contained data for a period of two years and five months from 01-01-2017 to 05-05-2019. In total, it contained data relating to 16371 valid journeys that could be used in the modelling process. (The total number of station-to-station journey instances was 107431.)

Figure 5.2 shows the proportions of journeys that were early, on time and delayed. By on time we mean that there is no difference between the scheduled time and the actual time. It can be seen that 29.2% were not delayed, 8.54% were on time and 52.3% were delayed.

No weather data were included in this dataset because we were unable to obtain any. We purchased weather data covering the period of the dataset, but it was found to be incomplete and was not as accurate as the weather data provided by Weatherquest for the seven month period of the initial study described in Chapter 4. We noted in Section 4.5.2 that there was some benefit from including the weather data, however for this study we were focusing on the

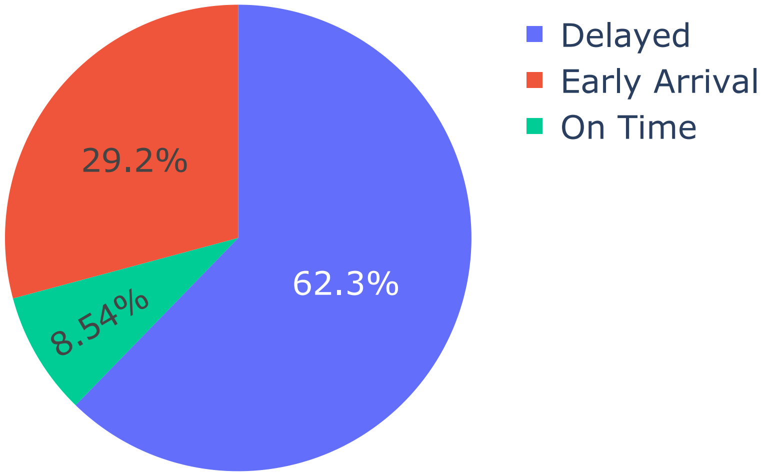


Figure 5.2: Proportions of all NRW – LST journey delays. The pie chart shows the percentage of journeys that were not delayed, on time and delayed.

ensemble development and it was not judged necessary to include the weather data.

In the published literature a number of factors have been reported to have an impact on train delay. For this experiment we did not use the same features as we used in our initial experiments in Chapter 4, because we were trying to build more precise models. Therefore we derived the following features from the raw data, which we found to have an effect on the delay:

- 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.

5.4 Experimental design

Our experimental design was based on that used in our initial experiments, described in Section 4.3, but with three differences. Firstly the base learner models were not exactly the same. Secondly there was the additional step of model selection in the framework. Thirdly, in these experiments we have also investigated the effect of the size of the ensemble on its performance.

We performed the following steps.

1. First, feature extraction from the dataset was performed, as described in Section 5.3
2. We then divided the data into training, validation, and testing datasets using a 70% : 15% : 15% split. We split the dataset using a random seed. Each experiment was repeated five times, using different random seeds, and the results were averaged.

3. Because our purpose in this work was to build a heterogeneous ensemble we employed a wide range of base learner algorithms. The regressors we used were chosen with the aim of representing a wide spectrum of machine learning themes from the baseline method to "state-of-the-art" methods. Of them, eight generate individual models; Linear Regression, Bayesian Ridge (Box and Tiao, 2011), Stochastic Gradient Descent (Jain et al., 2018), Lasso (Santosa and Symes, 1986), Ridge (Hoerl and Kennard, 1970), K-nearest neighbours Regressor, ElasticNet and Multilayer Perceptron (Rosenblatt, 1961); and four produce essentially homogeneous ensembles; Random Forest (Ho, 1995), Decision tree (DT), Gradient Boosting (Friedman, 2001), and XGBoost (XGB) (Chen and Guestrin, 2016).

   We have not used all the base learners we used in Chapter 4, because some gave very poor performance and we did not consider that it was worthwhile to continue to use them. However for these experiments we also included XGBoost, because is a very well known state-of-the-art algorithm which is known to give good performance.

   As noted in Section 3.2.4, 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.

4. All the models generated were put into a collection of models.

5. A model selection method, either MSM1 or MSM2 was applied to select models for inclusion in the ensemble, using the validation data. MSM1 selects on the basis of accuracy, while MSM2 selects on the basis of accuracy and diversity. MSM2 is further divided into MSM2a, which uses

a pairwise diversity measure, either covariance, correlation or disagreement; and MSM2b which uses the non-pairwise diversity measure, CFD.

6. These generated models were then used to form an ensemble. Here two different decision making functions were applied to combine the outputs of the models in a heterogeneous ensemble to produce a final output. Because of the different decision making functions the ensembles are named as Averaging Ensemble (AE) and Weighted Ensemble (WE). The weighted averaging was performed using the performance of the models on the validation data. The performance was calculated using the metrics described in Section 5.4.1. Finally the completed ensemble was tested using the testing data. The performance of the individual models on the validation data and the performance of the ensemble on the testing data were calculated using the metrics described below in Section 5.4.1.

In order to test our results we have employed a number of statistical methods and used CD diagrams, as we did with our results in Chapter 4.

5.4.1 Evaluation methods

In contrast to Chapter 4, we evaluated the results using the four metrics listed in Section 3.4.3. These are, mean absolute error (MAE), mean squared error (MSE), root mean squared error ($RMSE$) and R-squared (R^2). These are standard metrics used for assessing the results obtained from regression experiments. For most of the experiments we have only presented the R^2 and MAE values because RMSE and MSE showed comparable patterns. The values of all four measures are presented for the examples of single runs.

Using R^2 with different numbers of features can cause problems, because the value of R^2 increases with larger numbers of features. For this reason, the adjusted R^2 measure (see Equation 5.4) is often used, which takes into account the number of features. However, we found that with our results, because the number of samples was much larger than the number of features, the difference between the two measures was negligible. The equation for adjusted R^2 is shown in Equation 5.4:

$$adjusted\ R^2 = 1 - \frac{(1 - R^2)(N - 1)}{N - J - 1}, \tag{5.4}$$

where N is the number of samples and J is the number of features.

It is obvious that where N is much larger than J the difference between R^2 and adjusted R^2 will be very low. For example, in our experiments in this chapter, the number of samples was 107431 and the number of features was 12, so an unadjusted R^2 value of 0.79090 (the first R^2 value in Table 5.1) becomes 0.79088 after adjustment. Therefore we used R^2 throughout our studies.

5.5 Results and discussion

In our initial experiments in Chapter 4 presented results for each pair of stations separately. To do this for the work in the current chapter was considered impractical, and so the results presented here are for all the pairs of stations. This also more generic, as the methods would have to be generalisable to give consistent results across multiple pairs of stations.

In the experiments described in this chapter and subsequent chapters we applied model selection, and ensembles were generated with different numbers of models. When presenting the results, the results for the ensembles are for ensembles of different sizes, from 2 to 12. The corresponding results for accuracy of single models are the average of the individual models that were used in that ensemble. So, for example, where an ensemble contained two models, the corresponding result for accuracy of single models was the average accuracy of those two models. (This will differ from the accuracy of the ensemble, of course, because in the ensemble the outputs of the models are merged by the decision making function before the accuracy is measured.)

In these results we present the average values of accuracy of single models, the averaging ensembles and the weighted ensembles. The average of single models result is the value obtained from averaging the accuracy obtained from generating models using each algorithm separately. The averaging ensembles result is the value obtained when all the algorithms are used to generate models and those models are put into an ensemble, and assigned equal weights. The weighted ensemble result is the value obtained when all the algorithms are used to generate models and those models are put into an ensemble, and the models are weighted according to accuracy.

Figures 5.3 and 5.4 present the results for average of 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 5.3 R^2 values of the predictions are given, with the standard deviations (SD) over five repeat runs. These results are also presented in Tables 5.1 and 5.2 which show the means and standard deviations (SD) of R^2 and MAE, respectively, of predictions made by the single models, and the heterogeneous ensembles. In Figure 5.4 the MAE values of these predictions are presented. (Note: In Figure 5.3 and some subsequent figures the plots are moved slightly along the x-axis in order to show them clearly by avoiding overlapping of the error bars.)

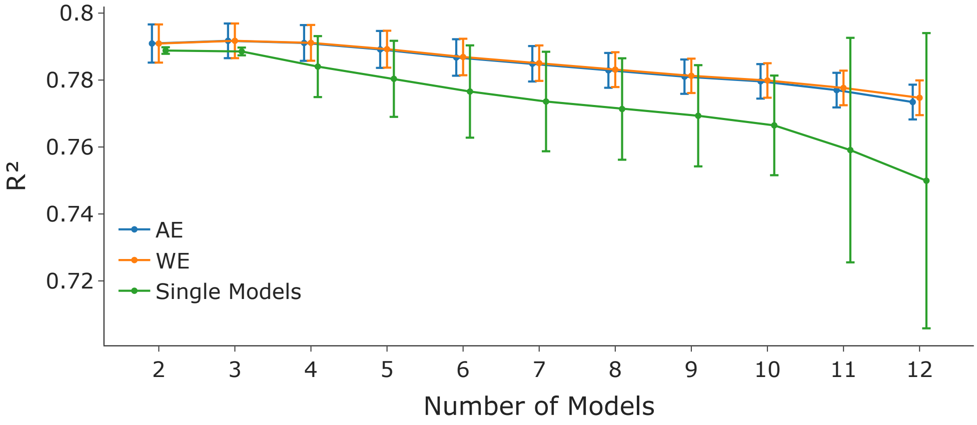


Figure 5.3: R^2 values for AE, WE and average of single models using MSM1, with SD values indicated by error bars. The ensembles showed higher accuracy than the average of single models.

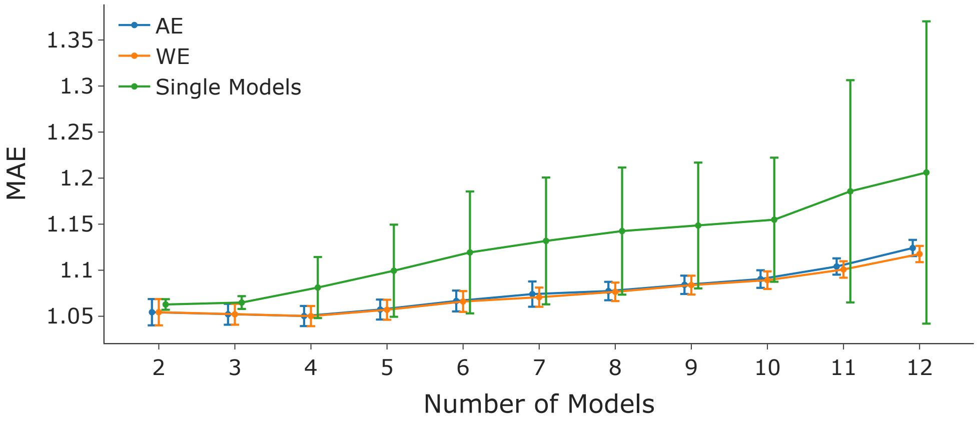


Figure 5.4: MAE values for AE, WE and average of single models using MSM1, with SD values indicated by error bars. The ensembles showed lower error than the average of single models.

From these results it can be seen that both types of ensemble (AE and WE) had consistently higher R^2 values that 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.

The SD values for single models were sometimes quite large. For example, in Figure 5.3 the error bars (showing the SD) become larger as the number of models increases. The reason for the high values with the larger numbers of models is that the most accurate models were being added to the ensemble in its initial stages of construction, and the least accurate when it was at its largest, the least accurate models also had the greatest inconsistency, resulting in the largest SD. Where MSM2 was being used with pairwise diversity measures the SD values were large even with a small number of models, for example in Figure 5.11. This was because when the non-pairwise diversity measures were used the least accurate models were being selected in the early stages of the ensemble construction. As explained below, this was because they were the most diverse from the most accurate model, which was always used as the first model in the ensemble. Hence, even with only two models the SD value was high. As more models were added, the more accurate ones were included in the ensemble, and the SD values reduced, but never became very low.

Thus we can say that the ensembles outperform the single models not only terms of 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. Thus there are advantages of the ensembles in terms of both accuracy and consistency, with the WE being best overall. The fact that the ensembles are more consistent in very important because it indicates that they are more robust. Therefore if used for predictions in a real time system it will have a greater level of reliability than a system based on single models.

The ensembles generated with 3 to 4 models achieved the best performance, as shown by Figures 5.3 and 5.4. This is important because it indicates that the ensembles were achieving the best performance with a small number of models. As the size of the ensembles increased beyond 4 models the performance deteriorated. Thus ensembles generated using our strategy achieve their best performance using a small number of iterations. A consequence of this is that it should be possible to identify the models that are used most often when constructing ensembles, and those that are not. Thus the poorly performing base learners could be excluded from future ensemble building in order to reduce the size of the collection of models and the build time.

It is important to note that the ensembles outperformed the best single model (BSM), as can be seen in Figures 5.5 and 5.6, which show the performance of the best single model and that of the ensembles for separate single runs of the methods, i.e., they are not averages over multiple runs. For the runs presented in Figure 5.5 the ensembles sized 3 and 4 outperformed the best single model, for those presented in Figure 5.6 the ensembles sized 2, 3 and 4 outperformed the best single model. Thus the best performing ensemble outperformed the best single model. Thus the heterogeneous ensembles sized 2 and 3 were able to perform consistently, which gives them a definite advantage over single models, or a homogeneous ensemble composed of one type of model. Thus, even if the best single model was a state-of-the-art method such as XGBoost or Random Forest, the ensembles performed better.

Figures 5.7 and 5.8 show the R^2 and MAE values obtained, respectively, for ensembles of different sizes, with AE, WE and average of single models using MSM2 with CFD for diversity measurement.

In Figure 5.7 R^2 values of the predictions are given, with the standard deviations (SD) over five repeat runs. These results are also presented in Tables 5.3 and 5.4 which show the means and standard deviations (SD) of R^2 and MAE, respectively, of predictions made by the single models, and the heterogeneous ensembles. In Figure 5.8 the MAE values of these predictions are presented. It can be seen that AE and WE consistently perform better than single models.

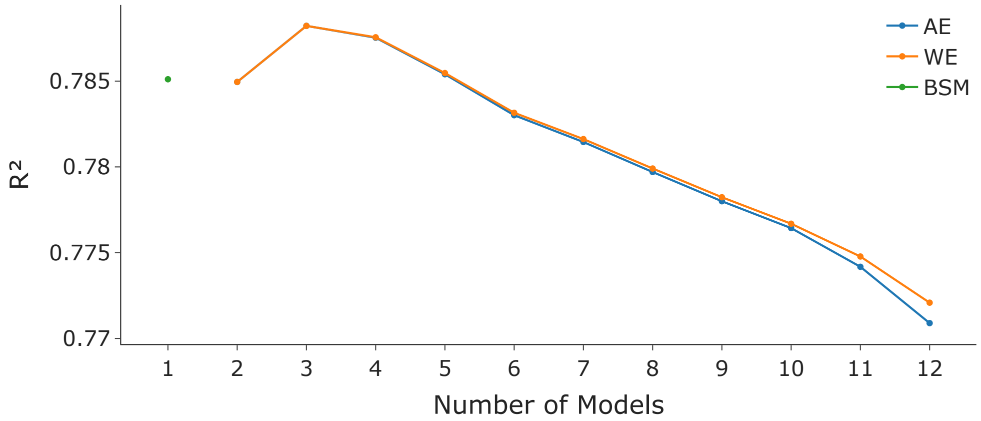


Figure 5.5: R^2 values obtained for one individual run for AE, and WE using MSM1, with the value for the most accurate single model indicated. It can be seen that ensembles sized 2 and 3 were more accurate than the most accurate single model.

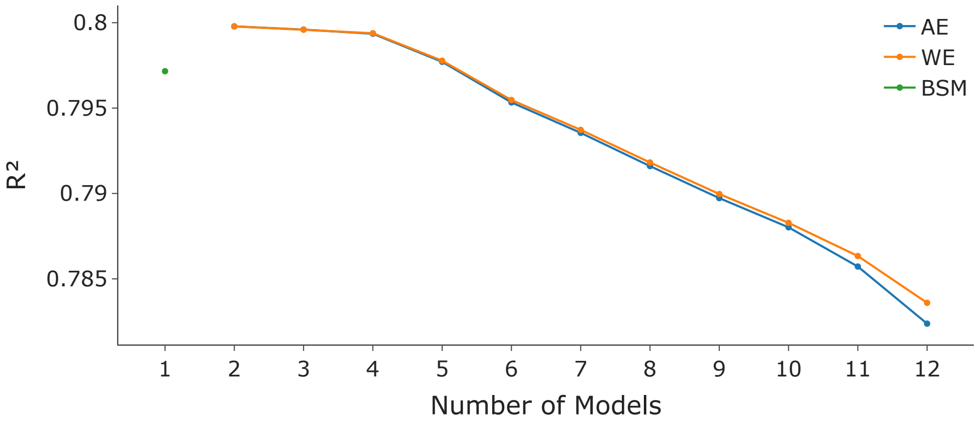


Figure 5.6: R^2 values obtained for one individual run for AE, and WE using MSM1, with the value for the most accurate single model indicated. This is a different run from that presented in Figure 5.5. In this run ensembles of sizes 2, 3 and 4 were more accurate than the most accurate single model.

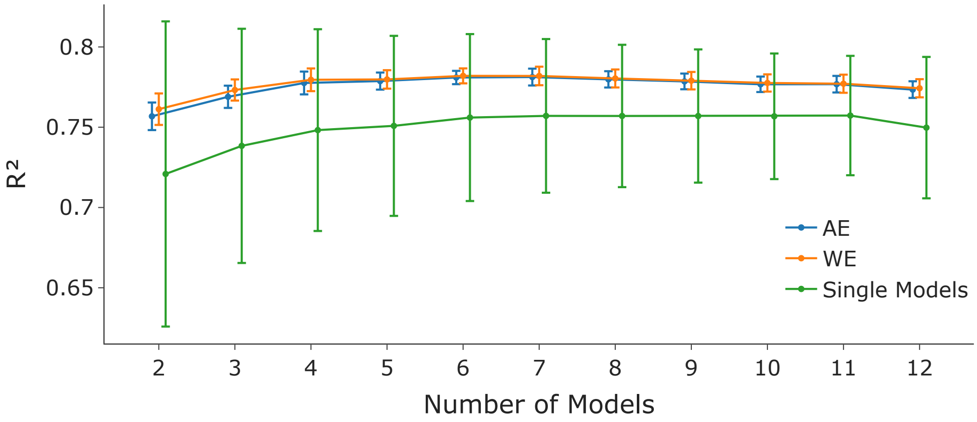


Figure 5.7: R^2 values for AE, WE and average of single models using MSM2 and Coincidence Failure Diversity as the diversity measure. SD values are indicated by error bars.

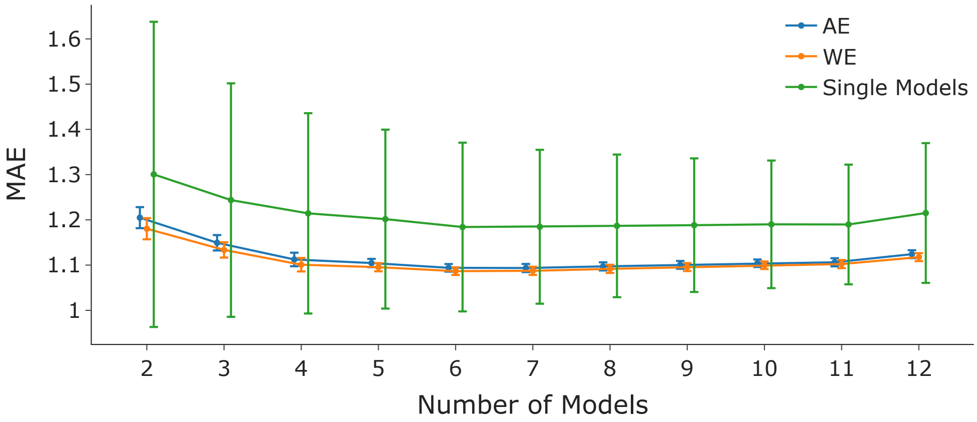


Figure 5.8: MAE values for AE, WE and average of single models using MSM2 and Coincidence Failure Diversity as the diversity measure. SD values are indicated by error bars.

Tables 5.5–5.9 show the means and standard deviations (SD) of R^2 and MAE of predictions made using MSM2 with the other diversity measures.

Figures 5.9 and 5.10 present comparisons of the R^2 values obtained for different size ensembles using MSM1 and MSM2 for AE and WE ensembles, respectively. These results are also presented in Tables 5.11 and 5.12, respectively.

These results demonstrate that with MSM1 for both AE and WE, performance remains constant when varying the size of ensembles from 2 to 12 models and is always the best.

With CFD as the criterion for model selection as models were added to the ensemble beyond the initial 2, the performance, (as measured by an increase in R^2 and a decrease in MAE,) improved, until the ensemble contained 5 models. Thus using MSM2 with CFD for building ensembles was effective, although it was not as effective as using MSM1 which gave a better performance, and did so with fewer models.

Table 5.13 presents the results for a single run of MSM2 with CFD and shows the scores obtained as models were added to the ensemble. It can be seen that the CFD score decreased as each model was added, but that the accuracy increased initially, then decreased. The reason for the decrease in diversity was that the most diverse models were added in the early stages of ensemble construction. We have not presented comparable results for the other diversity measures since only CFD was effective and the other measures did not give any benefit.

The other three diversity measures, all gave equivalent patterns to each other. Figures 5.11 to 5.16 and Tables 5.5 to 5.10 present the results. (It should be noted that MAE gives a graph that has a pattern that is the reverse of Correlation and R^2 because it is measuring *error*, i.e. a lower value indicates higher accuracy.) As with CFD, for a given ensemble size WE ensembles always give the best performance, closely followed by AE then by single models.

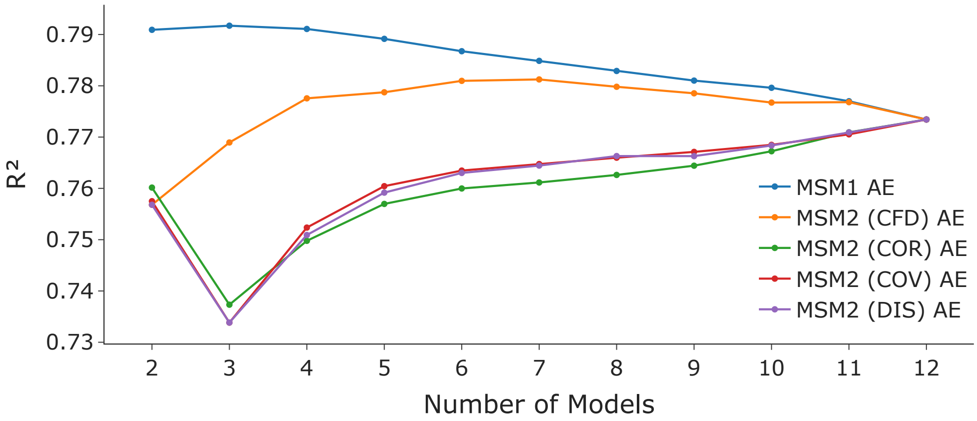


Figure 5.9: R^2 values obtained with AE using the two selection methods MSM1 and MSM2, different diversity measures with MSM2, and different ensemble sizes.

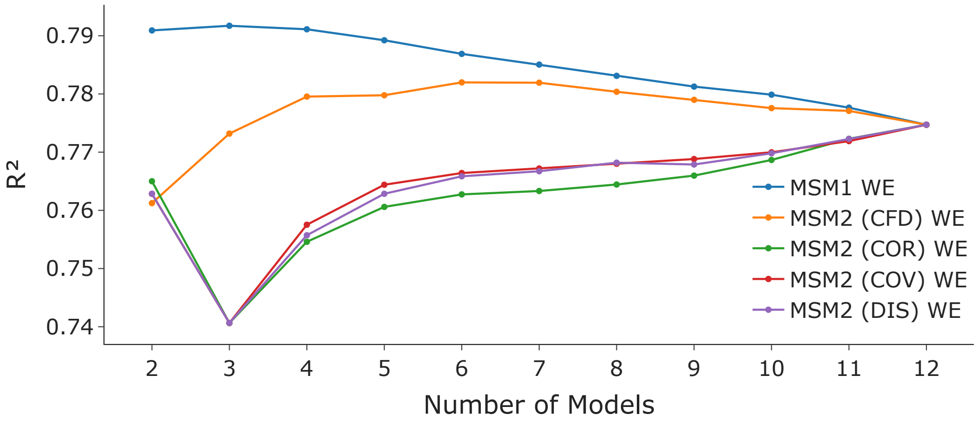


Figure 5.10: R^2 values obtained with WE using the two selection methods MSM1 and MSM2, different diversity measures with MSM2, and different ensemble sizes.

With regard to the ensemble performance as models are selected and added to it, when the third model is added to the ensemble the R^2 value decreases and the MAE value increases. (Figures 5.9 and 5.10, Tables 5.1–5.10, and

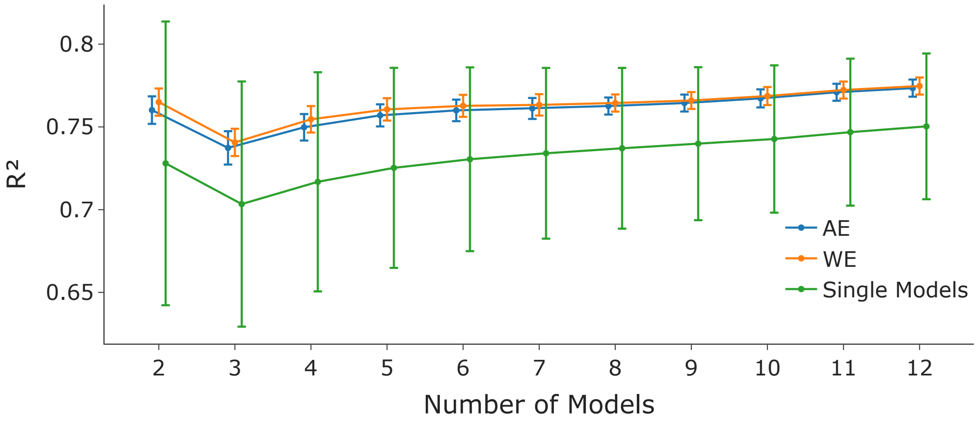


Figure 5.11: R^2 values for AE, WE and average of single models using MSM2 and correlation as the diversity measure. SD values are indicated by error bars.

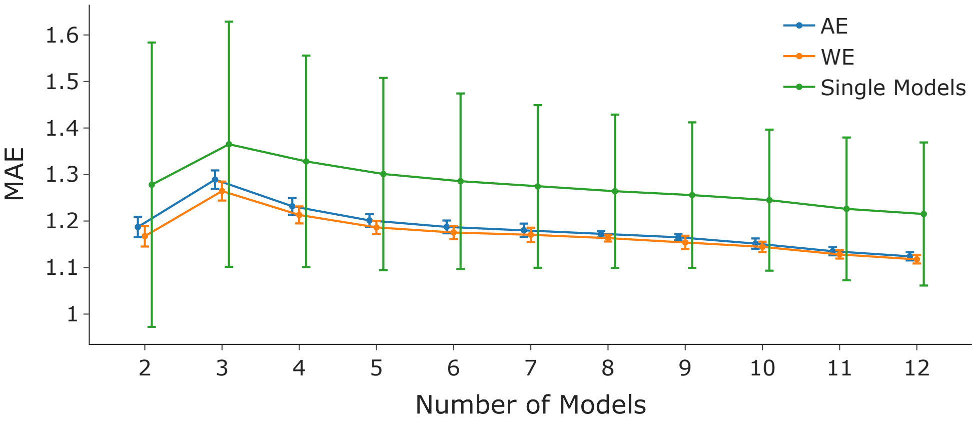


Figure 5.12: MAE values for AE, WE and average of single models using MSM2 and correlation as the diversity measure. SD values are indicated by error bars.

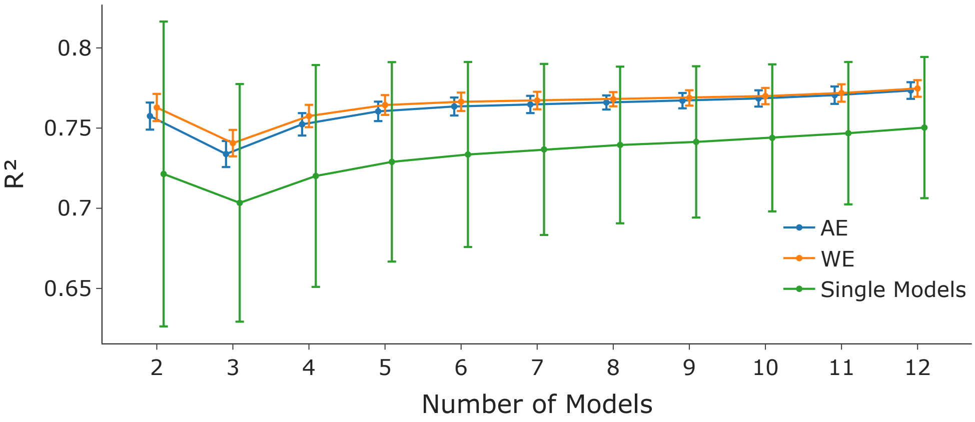


Figure 5.13: R^2 values for AE, WE and average of single models using MSM2 and covariance as the diversity measure. SD values are indicated by error bars.

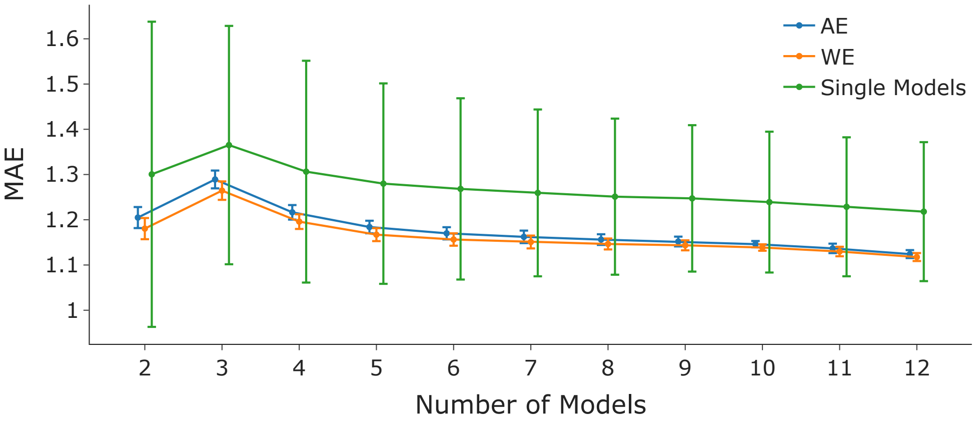


Figure 5.14: MAE values for AE, WE and average of single models using MSM2 and covariance as the diversity measure. SD values are indicated by error bars.

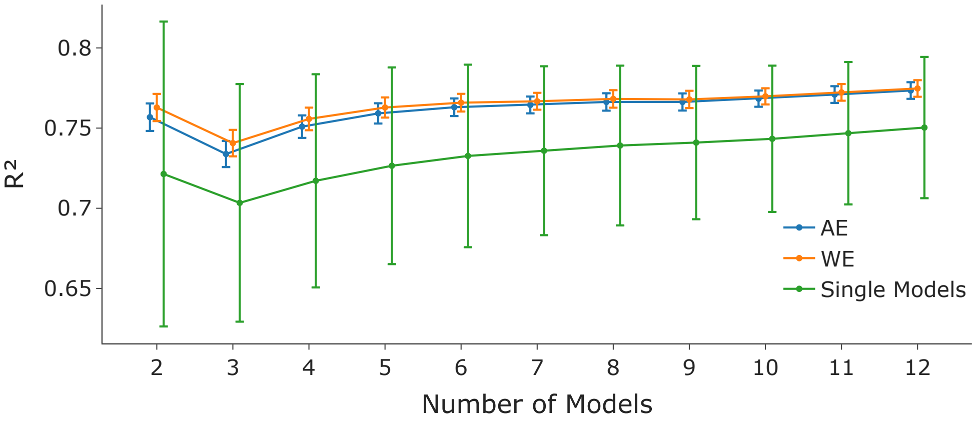


Figure 5.15: R^2 values for AE, WE and average of single models using MSM2 and disagreement as the diversity measure. SD values are indicated by error bars.

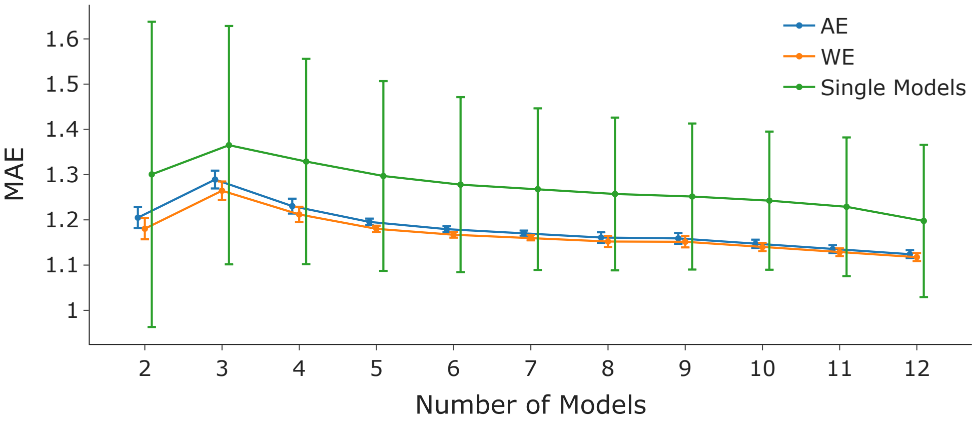


Figure 5.16: MAE values for AE, WE and average of single models using MSM2 and disagreement as the diversity measure. SD values are indicated by error bars.

Figures 5.11 to 5.16.) Then as further models are added the R^2 value increases a small amount as each model is added. Similarly the MAE value decreases by a small amount as each model is added. Thus the performance of the ensemble decreases to its lowest when the ensemble contains 3 models and then as more models are added the performance improves. Compared with the drop in performance when the third model is added, the improvement with each successive model is relatively small and it takes several models to be added to the ensemble before the performance is better than the initial value with 2 models. Overall the worst performance is given with three models and the best is only achieved when all the models have been added.

This is, of course, completely different from the pattern for MSM2 with CFD, and when using MSM1. It indicates that the selection mechanism is failing in its aim of selecting the best models early on in the building of the ensemble and in fact it appears to be selecting the worst performing model as the third model. The improvement in performance observed as more models are added is simply due to the better performing models then being added. Overall we can say that using these diversity measures to select the best models has totally failed. This is confirmed by the fact that the best performance is not obtained until all the models have been added.

In contrast, while MSM2 with CFD was not as effective as MSM1 at building heterogeneous ensembles, it did give an initial improvement in performance as models were added, and performance reached a peak before it began to drop as the poorer performing models were added to the ensemble. Thus, to a certain extent, using CFD as a criterion for selecting models was effective.

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 of there are almost no diversity measures designed for regression problems we adapted some existing classification diversity measures for this purpose, in particular

Table 5.1: R^2 values and SDs for AE, WE and average of single models using MSM1.

M	MSM1 AE		MSM1 WE		Average of 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

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 of models have not only the highest accuracies but also the most consistent results. The accuracy of single models often varies considerably over multiple runs. A current most accurate model may perform considerably worse in another run and it is very difficult to predict 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.

Table 5.2: MAE values and SDs for AE, WE and average of single models using MSM1.

M	MSM1 AE		MSM1 WE		Average of single models	
	MAE	SD	MAE	SD	MAE	SD
2	1.0544	0.0143	1.0544	0.0143	1.0628	0.0057
3	1.0521	0.0113	1.0521	0.0113	1.0649	0.0069
4	1.0503	0.0109	1.0502	0.0109	1.0812	0.0332
5	1.0573	0.0109	1.0571	0.0109	1.0995	0.0500
6	1.0666	0.0113	1.0660	0.0113	1.1193	0.0662
7	1.0741	0.0137	1.0707	0.0104	1.1318	0.0689
8	1.0774	0.0100	1.0765	0.0100	1.1425	0.0690
9	1.0842	0.0100	1.0838	0.0102	1.1486	0.0683
10	1.0904	0.0096	1.0892	0.0096	1.1548	0.0674
11	1.1040	0.0089	1.1008	0.0089	1.1857	0.1207
12	1.1241	0.0088	1.1176	0.0088	1.2061	0.1641

Table 5.3: R^2 values and SDs for AE, WE and average of single models using MSM2 with Coincident Failure Diversity as the diversity measure.

M	MSM1 AE		MSM1 WE		Average of 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

Table 5.4: MAE values and SDs for AE, WE and average of single models using MSM2 with Coincident Failure Diversity as the diversity measure.

M	MSM1 AE		MSM1 WE		Average of single models	
	MAE	SD	MAE	SD	MAE	SD
2	1.2049	0.0233	1.1804	0.0234	1.3005	0.3373
3	1.1493	0.0170	1.1336	0.0170	1.2437	0.2581
4	1.1124	0.0150	1.1009	0.0149	1.2144	0.2214
5	1.1046	0.0092	1.0954	0.0091	1.2017	0.1978
6	1.0940	0.0084	1.0866	0.0085	1.1841	0.1865
7	1.0935	0.0090	1.0874	0.0090	1.1847	0.1701
8	1.0971	0.0090	1.0917	0.0091	1.1867	0.1577
9	1.1004	0.0087	1.0956	0.0088	1.1883	0.1477
10	1.1040	0.0083	1.0997	0.0084	1.1901	0.1410
11	1.1062	0.0090	1.1023	0.0091	1.1898	0.1323
12	1.1241	0.0088	1.1176	0.0088	1.2151	0.1545

Table 5.5: R^2 values and SDs for AE, WE and average of single models using MSM2 with correlation as the diversity measure.

M	MSM1 AE		MSM1 WE		Average of 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 5.6: MAE values and SDs for AE, WE and average of single models using MSM2 with correlation as the diversity measure.

M	MSM1 AE		MSM1 WE		Average of single models	
	MAE	SD	MAE	SD	MAE	SD
2	1.1872	0.0220	1.1674	0.0222	1.2781	0.3056
3	1.2891	0.0197	1.2644	0.0204	1.3651	0.2634
4	1.2318	0.0182	1.2131	0.0183	1.3282	0.2275
5	1.2013	0.0135	1.1863	0.0139	1.3011	0.2065
6	1.1874	0.0137	1.1753	0.0144	1.2856	0.1886
7	1.1802	0.0143	1.1704	0.0153	1.2743	0.1748
8	1.1717	0.0070	1.1633	0.0073	1.2641	0.1647
9	1.1651	0.0067	1.1540	0.0144	1.2557	0.1564
10	1.1516	0.0110	1.1445	0.0111	1.2449	0.1516
11	1.1352	0.0089	1.1282	0.0089	1.2261	0.1535
12	1.1241	0.0088	1.1176	0.0088	1.2151	0.1538

Table 5.7: R^2 values and SDs for AE, WE and average of single models using MSM2 with covariance as the diversity measure.

M	MSM1 AE		MSM1 WE		Average of 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 5.8: MAE values and SDs for AE, WE and average of single models using MSM2 with covariance as the diversity measure.

M	MSM1 AE		MSM1 WE		Average of single models	
	MAE	SD	MAE	SD	MAE	SD
2	1.2049	0.0233	1.1804	0.0234	1.3005	0.3373
3	1.2891	0.0197	1.2644	0.0204	1.3651	0.2634
4	1.2164	0.0161	1.1959	0.0162	1.3063	0.2452
5	1.1840	0.0139	1.1671	0.0143	1.2799	0.2215
6	1.1701	0.0135	1.1564	0.0137	1.2682	0.2003
7	1.1622	0.0138	1.1508	0.0142	1.2594	0.1844
8	1.1562	0.0118	1.1465	0.0121	1.2511	0.1725
9	1.1519	0.0108	1.1435	0.0111	1.2473	0.1618
10	1.1461	0.0070	1.1386	0.0071	1.2391	0.1557
11	1.1368	0.0104	1.1298	0.0105	1.2286	0.1536
12	1.1241	0.0088	1.1176	0.0088	1.2180	0.1536

Table 5.9: R^2 values and SDs for AE, WE and average of single models using MSM2 with disagreement as the diversity measure.

M	MSM1 AE		MSM1 WE		Average of 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 5.10: MAE values and SDs for AE, WE and average of single models using MSM2 with disagreement as the diversity measure.

M	MSM1 AE		MSM1 WE		Average of single models	
	MAE	SD	MAE	SD	MAE	SD
2	1.2049	0.0233	1.1804	0.0234	1.3005	0.3373
3	1.2891	0.0197	1.2644	0.0204	1.3651	0.2634
4	1.2304	0.0165	1.2120	0.0170	1.3289	0.2270
5	1.1954	0.0073	1.1801	0.0070	1.2970	0.2097
6	1.1794	0.0064	1.1671	0.0064	1.2778	0.1935
7	1.1707	0.0057	1.1604	0.0057	1.2679	0.1786
8	1.1609	0.0116	1.1521	0.0121	1.2572	0.1689
9	1.1590	0.0118	1.1515	0.0124	1.2516	0.1614
10	1.1471	0.0092	1.1398	0.0092	1.2425	0.1527
11	1.1352	0.0087	1.1283	0.0087	1.2289	0.1533
12	1.1241	0.0088	1.1176	0.0088	1.1976	0.1683

Table 5.11: R^2 values obtained with AE using the two selection methods MSM1 and MSM2, different diversity measures with MSM2, and different ensemble sizes.

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 5.12: R^2 values obtained with WE using the two selection methods MSM1 and MSM2, different diversity measures with MSM2, and different ensemble sizes.

M	MSM1	MSM2	MSM2	MSM2	MSM2
		(CFD)	(COR)	(COV)	(DIS)
	WE	WE	WE	WE	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

Table 5.13: Example single run of MSM2 using Coincidence Failure Diversity as criterion for model selection when building the ensemble, showing the models selected for the ensemble, the R^2, MAE and RMSE values, and the CFD score.

M	**Selected Models in Ensemble**	R^2	**MAE**	**MSE**	**RMSE**	**CFD Score**
2	GBR,EN	0.7490	1.2341	4.1836	2.0454	0.6191
3	GBR,EN,KNN	0.7630	1.1716	3.9507	1.9876	0.6136
4	GBR,EN,KNN,MLP	0.7768	1.1158	3.7211	1.9290	0.6112
5	GBR,EN,KNN,MLP,SGD	0.7764	1.1098	3.7273	1.9306	0.6083
6	GBR,EN,KNN,MLP,SGD,XGB	0.7791	1.0981	3.6815	1.9187	0.6041
7	GBR,EN,KNN,MLP,SGD,XGB,DT	0.7786	1.1016	3.6902	1.9210	0.5997
8	GBR,EN,KNN,MLP,SGD,XGB,DT,BR	0.7771	1.1047	3.7156	1.9276	0.5922
9	GBR,EN,KNN,MLP,SGD,XGB,DT,BR,RF	0.7764	1.1080	3.7265	1.9304	0.5862
10	GBR,EN,KNN,MLP,SGD,XGB,DT,BR,RF,Ridge	0.7752	1.1112	3.7473	1.9358	0.5810
11	GBR,EN,KNN,MLP,SGD,XGB,DT,BR,RF,Ridge,LR	0.7740	1.1147	3.7672	1.9409	0.5768
12	GBR,EN,KNN,MLP,SGD,XGB,DT,BR,RF,Ridge,LR,Lasso	0.7709	1.1314	3.8188	1.9542	0.5554

5.5.1 Critical Comparisons

Figures 5.17–5.22 present our results using Critical Difference diagrams.

Figure 5.17 is the CD diagram of the results with two selection methods for AE ensembles of different sizes. (These were presented in Figures 5.9.) It can be seen that MSM1 AE has the highest accuracy. It is more accurate than MSM2(CFD) AE, but the difference is not statistically significant. However, it is more accurate than MSM2(COV) and the difference is statistically significant. The other MSM2 ensembles, MSM2(COR) AE and MSM2(DIS) AE, are less accurate than MSM2(COV) AE. Thus MSM1 AE achieves an accuracy that is statistically significantly better than the ensembles using pairwise diversity measures (COV, COR and DIS) for selecting models, and it is also more accurate than the ensemble using the non-pairwise diversity measure (CFD) to select the models. The rank order from highest to lowest accuracy is MSM1 AE, MSM2(CFD) AE, MSM2(COV) AE, MSM2(COR) AE and MSM2(DIS) AE.

Figure 5.18 presents the equivalent comparison for the ensembles using weighed averaging. (These were presented in Figures 5.10.) The rank order is the same, and MSM1 WE is statistically significantly more accurate than MSM2(COV) WE, MSM2(COR) WE and MSM2(DIS) WE. MSM1 WE is also more accurate than MSM2(CFD) but the difference is not statistically significant.

Figures 5.19–5.22 compare the results for ensembles ranging in size from 2–5, generated with MSM1 with the results for Random Forest and XGBoost.

For the ensembles of size 2 (Figure 5.19) the AE ensemble is more accurate than than WE, but the difference is not statistically significant. Both the AE and WE ensembles are statistically significantly more accurate than RF. They are also more accurate than XGBoost, but the differences are not statistically significant. XGBoost is more accurate than RF, but not significantly. The rank order from highest to lowest accuracy is AE, WE, XGBoost, RF.

For ensembles of size 3 (Figure 5.20) WE performs better than AE, but there is no significant difference. As with the ensembles of size 2, both AE and WE size 3 ensembles are statistically significantly more accurate than RF. WE and AE ensembles are more accurate than XGBoost, but the difference is not significant.

The interesting difference between the ensembles of size 3 and those of size 2 is that the size 3 WE were more accurate than the AE, while the size 2 AE were more accurate than the WE. This is probably because when there are three models, it is always possible for two accurate models to compensate for one inaccurate model to give an overall result that is accurate. Whereas when there are only two models, one accurate model cannot compensate for one inaccurate one. The ensemble principle depends on having an overall majority of good models, which is not possible when there are only two.

For ensembles of size 4 (Figure 5.21) the rank order is the same as for ensembles size 2 and 3. The WE and AE ensembles are both more accurate than RF, but the difference in accuracy is only statistically significant for the WE. This is probably because the fourth model is not very accurate, and so the weighting mechanism of the WE ensembles causes the more accurate models to have a greater input to the overall output of the ensemble, whereas in the AE ensembles all the models contribute equally. This shows the benefit of having the weighting mechanism in the ensemble.

For ensembles of size 5 (Figure 5.22) the rank order is different from that with ensembles sized 2–4. The WE ensembles are still the most accurate, and are statistically more significantly accurate than RF. However, XGBoost is now more accurate than the AE ensemble. This is probably because the fifth model added to the WE and AE ensembles is of much lower accuracy than the other models, and while the weighting mechanism in the WE ensembles is able to compensate for this, in the case of the AE ensembles the presence of a model with very low accuracy cannot be compensated by the more accurate models.

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.

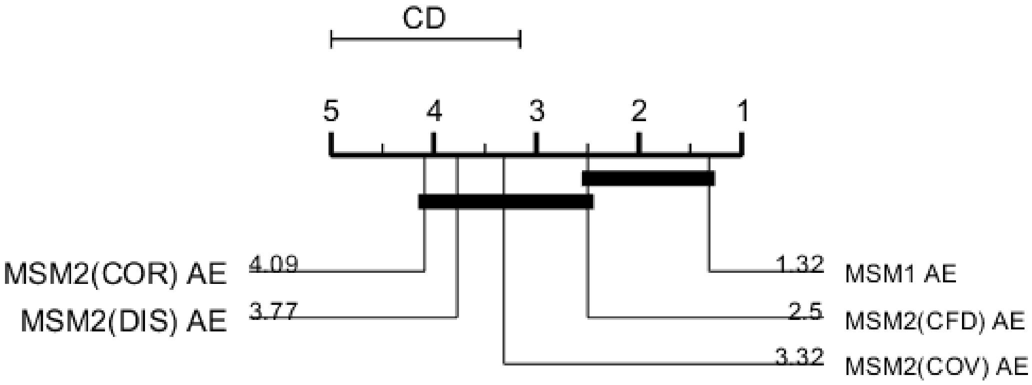


Figure 5.17: Critical difference comparison for results with the two selection methods, MSM1 and MSM2, with different diversity measures for MSM2, for AE ensembles of different sizes.

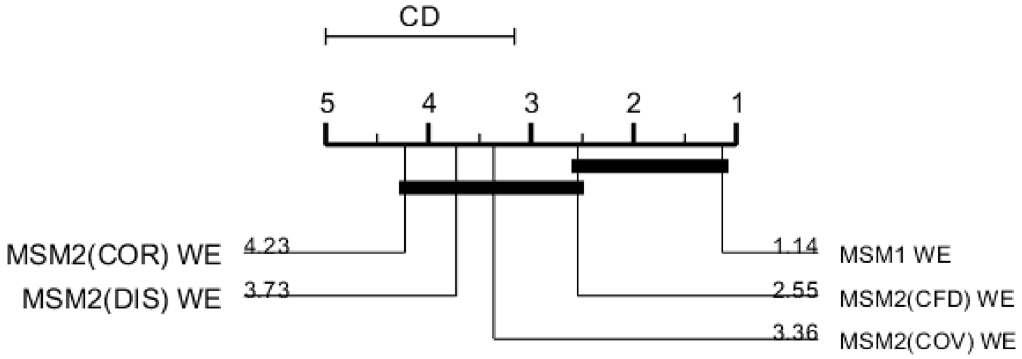


Figure 5.18: Critical difference comparison for results with the two selection methods, MSM1 and MSM2, with different diversity measures for MSM2, for WE ensembles of different sizes.

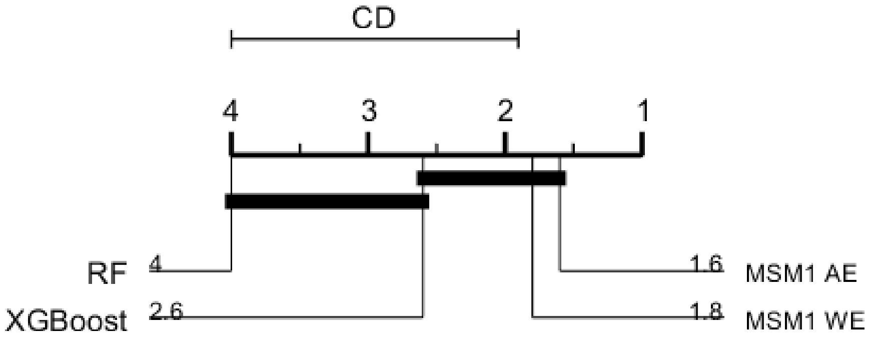


Figure 5.19: Critical difference comparison for ensembles size 2 with MSM1 AE and MSM1 WE, Random Forest model, and XGBoost model.

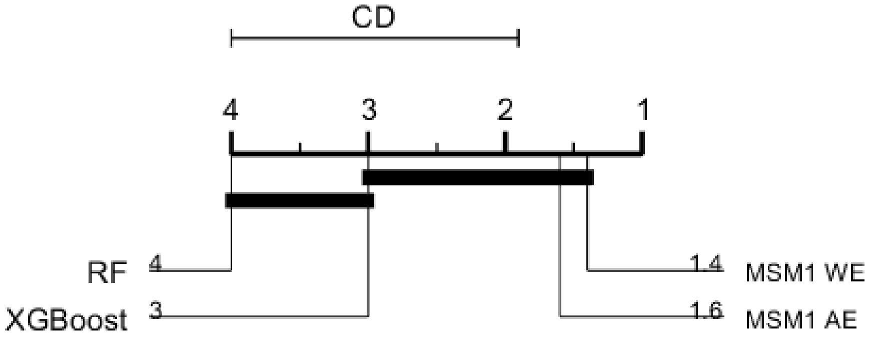


Figure 5.20: Critical difference comparison for ensembles size 3 with MSM1 AE and MSM1 WE, Random Forest model, and XGBoost model.

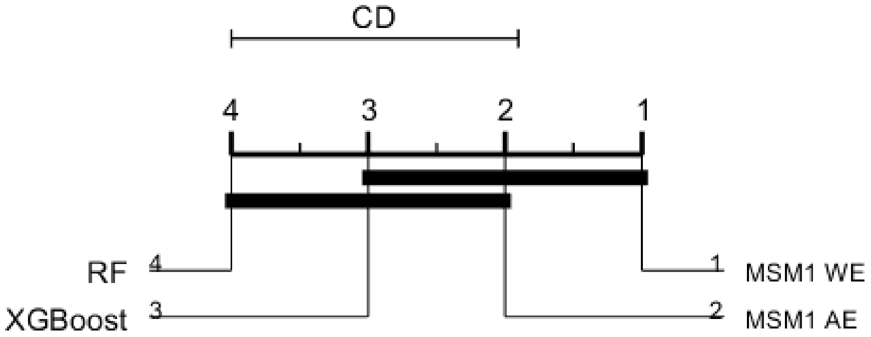


Figure 5.21: Critical difference comparison for ensembles size 4 with MSM1 AE and MSM1 WE, Random Forest model, and XGBoost model.

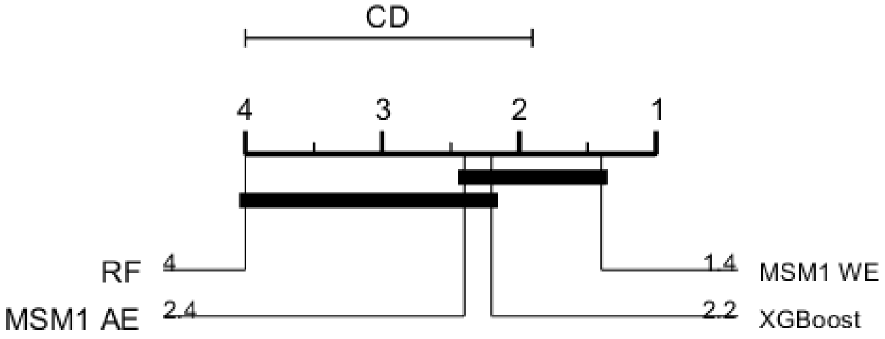


Figure 5.22: Critical difference comparison for ensembles size 5 with MSM1 AE and MSM1 WE, Random Forest model, and XGBoost model.

5.6 Summary

The experiments described in this chapter were performed in order to investigate the effect of ensemble size, and the use of accuracy and diversity as model selection criteria when building heterogeneous ensembles. Our results have helped to answer the research questions:

- What factors should be taken into consideration when selecting models? (For which we have examined a number of factors, including accuracy, diversity and ensemble size.)
- Is there any relationship between accuracy and diversity in the context of building an ensemble for performing regression?

We have developed two methods, MSM1 and MSM2, to investigate the use of Accuracy, and Accuracy and Diversity, respectively, in ensemble construction. For MSM2 we used both pairwise and non-pairwise diversity measures. These were all used with the Average and Weighted Average decision making functions that were developed in Chapter 4.

MSM1 considers only accuracy when selecting models to build the ensemble. Models are selected based on their performance on the validation data and sorted in descending order based on R^2 accuracy. They are then added to the ensemble, one by one.

MSM2 considers accuracy and diversity when selecting models to build the ensemble. Initially the highest accuracy model is selected and put into the ensemble, from then on models are selected based on diversity. We tested several diversity measures, both pairwise and non-pairwise. Two of these (CFD and Disagreement) were designed for classification problems, so we redefined them in order that they could be applied to regression problems. The other measures (Covariance and Correlation) were redefined so that their outputs were comparable with those of CFD and Disagreement, a range from 0 to 1, where 0=no diversity and 1=maximum diversity.

We assessed our ensembles using both R^2 and MAE. Other measures (RMSE and MSE) were also employed but were not presented (except for the examples of single runs) as they showed comparable patterns. We also employed the Friedman test to compare our ensembles with the state-of-the-art methods, XGBoost and Random Forest. Critical distance diagrams to visualise the results.

The results showed that MSM1 generated more accurate ensembles than MSM2. Thus, model selection by accuracy was more effective than model selection by accuracy and diversity. Of the diversity measures tested, only CFD gave any benefit. This was the only non-pairwise measure tested, the rest were all pairwise. There is no generally agreed definition of diversity, however diversity has been shown to be an important factor affecting classification ensemble performance. Our results suggest that there is no direct relation between ensemble performance and diversity for regression problems. This may be because the measures being used were not able to capture the diversity. We note that in the context of classification ensembles, non-pairwise measures are recognised to be better than pairwise, and the fact that it was only the non-pairwise measure that gave any benefit when selecting models for our regression ensembles does suggest that diversity can be captured and used in the regression context. However, there needs to be more work done on developing effective non-pairwise diversity measures for regression ensembles.

Of the two decision making methods, WE gave ensembles with slightly better accuracy than AE, and both types of ensemble gave better performance than single models.

We found that the ensembles with 3 to 4 models had the highest accuracy. This shows that our methods for building heterogeneous ensembles work effectively and are able to identify the best models to include in the ensembles.

Our ensembles were more consistent than the single models. The SD values of the ensemble accuracies for ensembles with 4 and more models were consistently lower than the SD values of the single models. With ensembles size 2 and 3 their SD values were relatively small for both ensembles and single models. The results we have presented are the averages of five runs, however, we have also shown the results of two individual runs where it can be seen that the most accurate ensembles (i.e. sized 3 and 4) are more accurate than the best single model.

The results of the Friedman test, shown in the critical difference diagrams, indicate that our ensembles were better than the state-of-the-art methods, XGBoost and Random Forest

In the next chapter we will investigate the use of deep learning, which is now a popular approach a number of fields, in order to compare it with our ensembles.

6 Deep Learning Heterogeneous Ensemble

6.1 Introduction

In the work described in the previous chapters we were focusing on heterogeneous ensemble construction and used machine learning (ML) algorithms for the base learners.

We tried to cover the most well-known algorithms that have been used previously as base learners in regression ensembles. We also used a sufficient number of different algorithms, in order to be able to examine the impact of the number of models the overall accuracy of the ensemble, as well as using a wide variety of base learner algorithms.

In this chapter we will explore the use of deep learning as a new approach with our heterogeneous ensembles for predicting train delay.

Deep learning (DL) has become increasingly popular in recent years and some studies (Huang et al., 2020; Zhang et al., 2021a) have used it for predicting train delays.

Huang et al. (2020) used a time-series approach in which the non-time series data, time-series data and spacio-temporal data were fed separately into a fully connected neural network (FCNN), long short term memory recurrent neural network and a 3-dimensional CNN respectively. These then all fed into a FCNN. Zhang et al. (2021a) also used a time series approach but split the data into weekly, daily and recent subsets, since there can be delay patterns according to the time of day or the day of the week. Each subset was processed using a graph convolution network feeding into a convolutional layer which fed into a FCNN. The outputs of the FCNNs were weighted and fused. But their description is not clear on how the weighting is actually determined.

It should be noted that our data are tabular in nature, and the use of DL architectures with tabular data has only recently been explored. (Arik and Pfister, 2021) In general, DL has mainly been used for computer vision and natural language processing.

Recently there has been a growing interest in finding a DL architecture for tabular data (TD). The purpose of this part of our study was to search for a benchmark DL method that could handle the data that we have. That is, we wanted to try and use DL and test it on our train delay data, and then compare it with machine learning in the heterogeneous ensemble context.

We used two different DL architectures, namely the Tabnet network and the CNN network. These are described in Section 6.2. For these algorithms hyperparameter tuning was necessary, this is described in Section 6.3.1.

The rest of this chapter is organised as follows:

Section 6.2 Presents the deep learning methods and the proposed deep learning heterogeneous ensemble.

Section 6.3 Presents the experimental design.

Section 6.4 Presents and discusses the experimental results.

Section 6.5 Summarises the chapter.

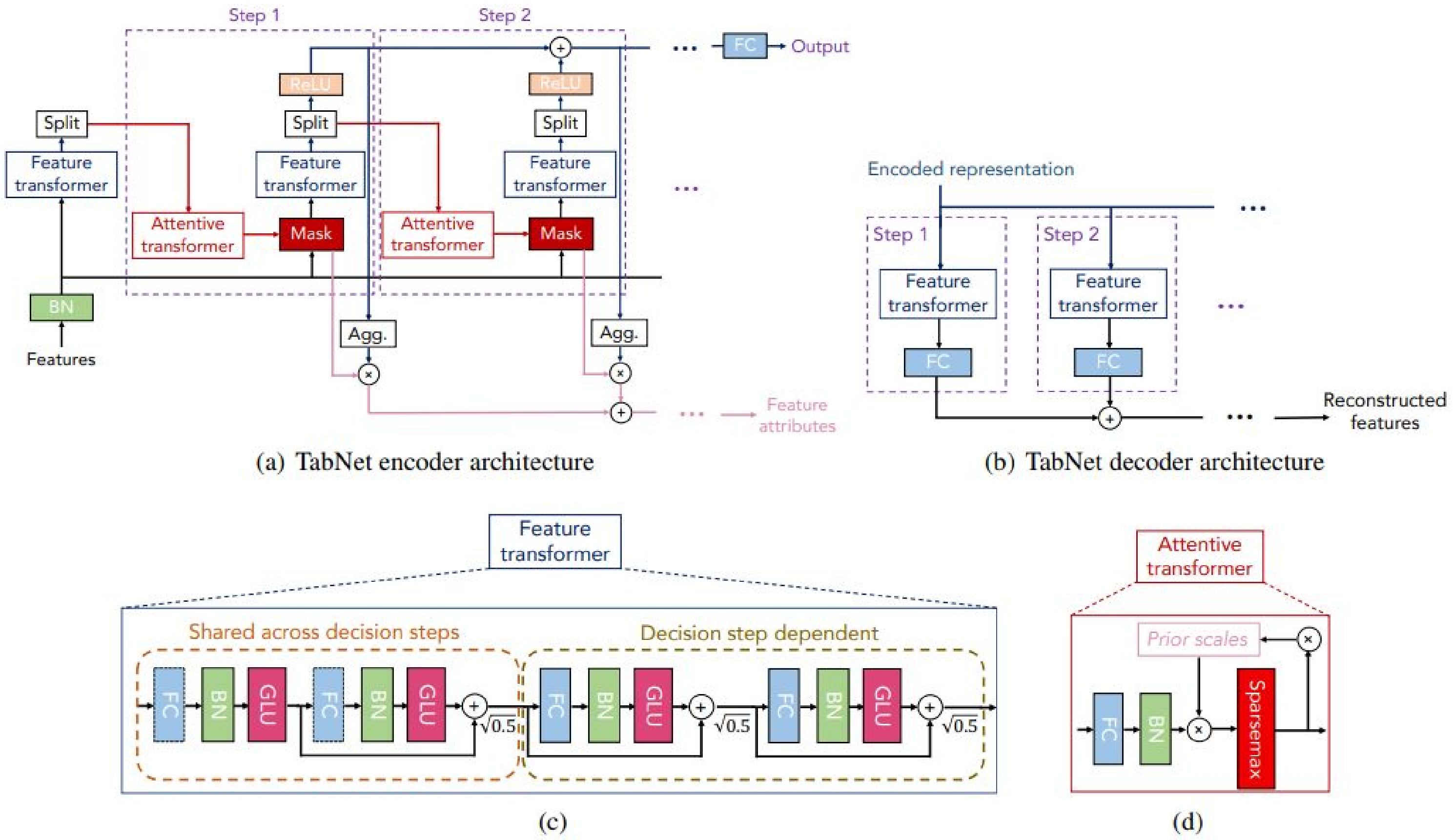
6.2 Deep learning methods

For our work in this chapter we devised a new framework based on that in Chapter 4. This employs deep learning methods instead of machine learning methods and uses the same dataset as that used in Chapter 5. This framework is shown in Figure 6.2.3.

The deep learning algorithms that we used were Tabnet and CNN. These are described below.

6.2.1 Tabnet

Tabnet is a deep neural network designed for tabular data. Decision trees were incorporated into deep learning in this architectural approach, which was first proposed by Google (Arik and Pfister, 2021). The tabnet was composed of an encoder and a decoder, as seen in Figure 6.1. Because our dataset is labelled, we will focus on the encoder. (The decoder is used for unsupervised learning with unlabelled data and is therefore not applicable.)



(a) TabNet encoder architecture

(b) TabNet decoder architecture

(c)

(d)

Figure 6.1: TabNet architecture. Figure taken from Arik and Pfister (2021) showing: (a) the encoder architecture, (b) the decoder architecture (this is not used in our application), (c) example of 4 layer network, with 2 shared across all decision steps and 2 are decision step dependent, and (d) the attentive transformer.

Encoder: the encoder, (see Figure 6.1(a)), consists of multiple layers. Each layer has 3 components—feature transformer, attentive transformer and mask. The features are input as raw features into a batch normalization, then the raw features are passed through four layers of the feature transformer, which are shown in Figure 6.1(c). This is followed by the attentive transformer and then important features are selected by feature masking.

Feature transformer: Figure 6.1(c)), shows the example from Arik and Pfister (2021) containing 2 shared layers and 2 decision step dependent layers. The fully connected (FC) layer feeds into a batch normalization (BN) layer which feeds into a gated linear unit (GLU) which is used to prevent exploding or vanishing gradients. The residual connection is normalised and is multiplied by 0.5, to ensure that the variance throughout the network does not vary widely.

Attentive Transformer: as can be seen in Figure 6.1(d) this consists of a fully connected (FC) layer, a batch normalization (BN) layer, prior scales layer and sparsemax layer. Each step of the process is controlled by the attentive transformer. During the first step, the input is passed to the FC layer, which is then followed by batch normalization, which is then multiplied by the prior scale. In other words, the prior scale reveals how much information is known about the features already from the previous steps, and how many features have already been used in the steps that came before. This that means that the output all integrates into overall decision making.

Feature masking: the outputs from the attentive transformer step are then fed to a mask. This simply ensures that only the selected features will be input to the model.

6.2.2 CNN

The Convolutional Neural Network (CNN) is another well known deep learning architecture. The most common applications of CNNs are in computer vision (Ding et al., 2018), speech recognition (Palaz et al., 2019) and face recognition (Li et al., 2020a), but they can be applied in many other areas.

Artificial neural networks function in an analogous way to neurons in the human brain, and CNNs are based on the organization of neurons in the visual cortex of the human brain.

A CNN consists of a convolutional layer, a pooling layer and a fully connected layer. The convolutional layer is responsible for extracting the features from the data using a kernel function. The only difference between our application of CNN and the standard CNN is the way the filter is applied, because our data are structured in 1D, instead of 2D as in the case of images. The pooling layer is designed to reduce the number of parameters while retaining the important information in the data, we used standard max-pooling in this layer. The fully-connected layer is a traditional multi layer perceptron where every neuron is connected to another.

The CNN structure we adopted for our experiments is shown in Figure 6.2. It contains an input layer, a convolutional layer, a pool layer where we used max-pooling, a flattening layer and three dense layers.

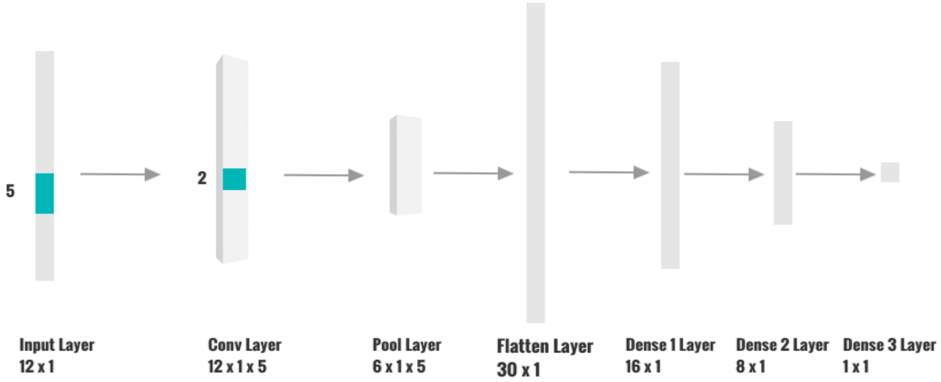


Figure 6.2: Our CNN architecture, as used in our application, consisting of an input layer, a convolutional layer, a pool layer where we used max-pooling, a flattening layer and three dense layers.

6.2.3 Deep Learning Heterogeneous Ensemble (DLHE)

For these experiments we devised a framework (Figure 6.3) based on the framework described in Chapter 4 (Figure 4.3).

This employs deep learning methods instead of machine learning methods and uses the same dataset as that used in Chapter 5.

The framework consists of five phases: (1) data preprocessing and feature extraction, (2) data partitioning, (3) modelling, (4) collection of models, and (5) building the ensemble.

For phase (1) data preprocessing and feature extraction were performed as previously described in Section 5.3, in order to give a fair comparison with our previous experiments.

For phase (2) the dataset was partitioned into training, validation and testing subsets. This was performed using a random seed to enable reproducibility,

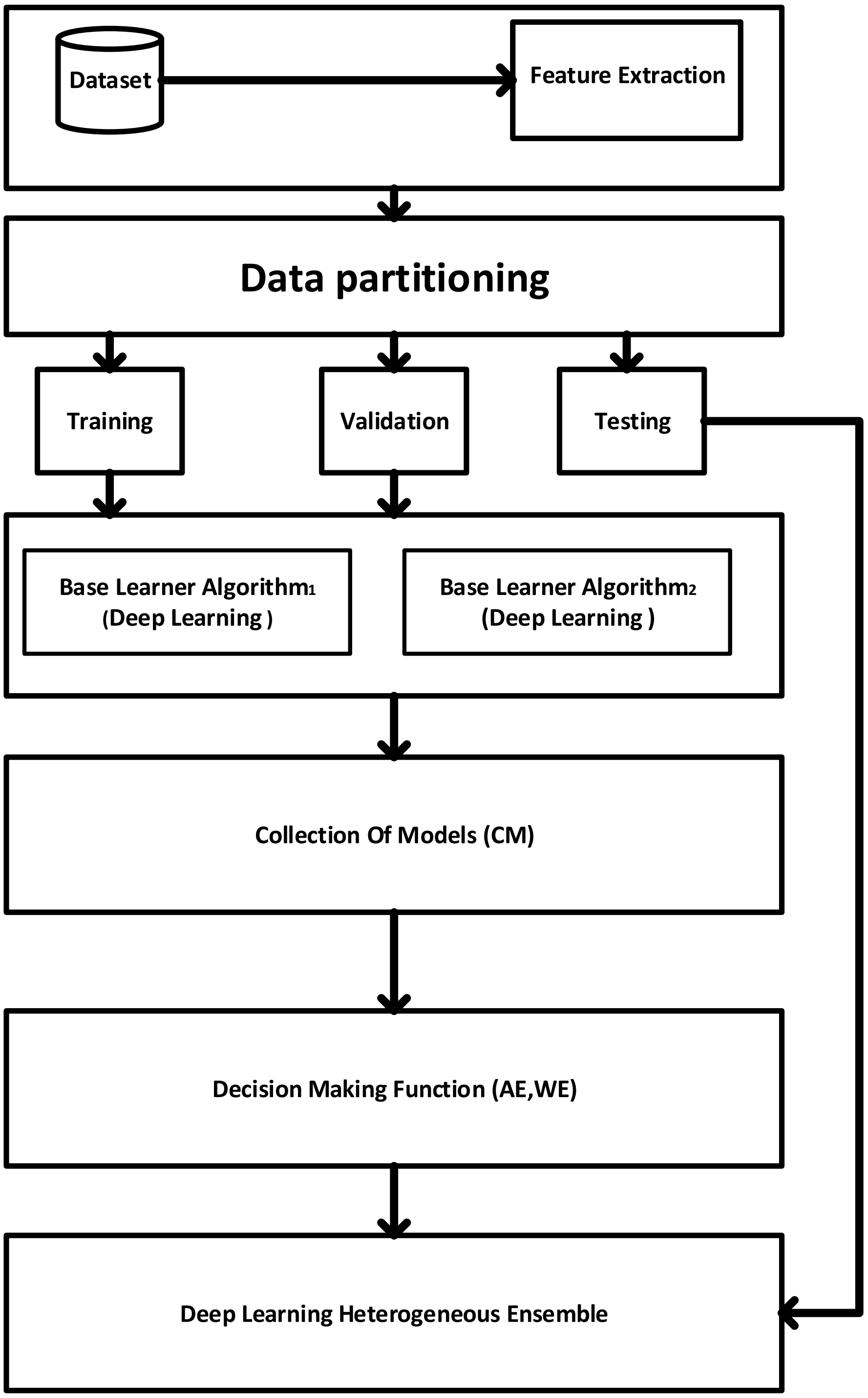

Dataset
Feature Extraction
Data partitioning
Training
Validation
Testing
Base Learner Algorithm1 (Deep Learning)
Base Learner Algorithm2 (Deep Learning)
Collection Of Models (CM)
Decision Making Function (AE,WE)
Deep Learning Heterogeneous Ensemble

Figure 6.3: Framework of the Deep Learning Heterogeneous Ensemble showing the process of generating the ensemble.

and also to enable different random partitions to be made by using different random seeds.

Phase (3) was the modelling phase. Here we employed the deep learning algorithms instead of machine learning algorithms. It is important to note that hyperparameter tuning must be employed with deep learning algorithms. Therefore we performed initial experiments to determine appropriate values for the hyperparameters of the algorithms used.

In phase (4) the models generated in phase (3) were put into the collection of models.

Phase (5) was the final stage where the models were combined together into the deep learning heterogeneous ensemble. This was performed using either averaging or weighted averaging.

6.3 Experimental design

We performed extensive initial experiments to fine tune the hyperparameters of the DL algorithms that we were going to use in our heterogeneous ensembles. (The DL algorithms are described in Section 6.2 and the tuning of their hyperparameters is described below in Section 6.3.1.)

Then, we tested these hyperparameter-optimised DL algorithms with the heterogeneous ensembles. For this we employed the framework shown in Figure 6.3, using the hyperparameter-optimised DL algorithms for the base learners.

The dataset was divided (70% : 15% : 15%) into *training*, *validation* and *testing* datasets, using a random seed to enable reproducibility and also to ensure that different partitions could be generated (by using different seeds) when performing an experiment multiple times.

The hyperparameter-optimised DL models (as described in Section 6.3.1) are then generated and put into the collection of models, the decision making function is applied and the heterogeneous ensemble generated, for this we used averaging and weighted averaging as in previous experiments. We call this a Deep Learning Heterogeneous Ensemble (DLHE).

Python, TensorFlow, and Pytorch-tabnet 2.0.0 were used in the implementation of these experiments which were performed on a standard PC with an Intel i5 processor and 16GB RAM.

6.3.1 Parameter Setting

The Hyperparameters of Tabnet

The hyperparameter settings are listed in Table 6.1. The parameters N_d, N_a and N_steps are the most important parameters. There are recommended default values for them suggested by Arik and Pfister (2021), but because this is a relatively new algorithm we tested a range of values for them. In order to prevent overfitting we used the early stopping technique. We performed training using the Adam optimiser and used mean squared error for the loss function. We used 200 epochs of training and a batch size of 128.

Table 6.1: Hyperparameter values used with Tabnet.

Hyperparameter	Description	Value
N_d	Width of the decision prediction layer	8
N_a	Width of the attention embedding for each mask	8
N_steps	Number of steps in the architecture	5
Lr	Learning rate	0.01
optimizer_fn	optimizer	Adam
gamma	the coefficient for feature reusage in the masks	1.3
N_shared	Number of shared Gated Linear Units at each step	2

The Hyperparameters of our CNN

Our CNN architecture is shown in Figure 6.2. Table 6.2 lists the hyperparameter values we adopted after testing a wide range of values for them. We employed early stopping to prevent overfitting. Training was performed using the Adam optimiser and the loss function was mean squared error. We used 200 epochs of training and a batch size of 128.

Table 6.2: Hyperparameter values used with our CNN.

Layer Number	Layer	Output Size	Kernel Size	Stride	Activation
1	Input	12 x 1	-	-	-
2	Convolution	12 x 1 x 5	5 x 1	1	Relu
3	Pooling	6 x 1 x5	2 x 1	2	-
4	Flatten	30 x 1	-	-	-
5	Dense	16 x 1	-	-	Relu
6	Dense	8 x 1	-	-	Relu
7	Dense	1 x 1	-	-	Linear

6.3.2 Evaluation metrics

We evaluated the results using the four metrics listed in Section 3.4.3. These are, mean absolute error (MAE), mean squared error (MSE), root mean squared error ($RMSE$) and R-squared (R^2). These are standard metrics used for assessing the results obtained from regression experiments. The values of all four measures are presented for all the single runs and the averages of five runs.

6.4 Experimental Results and Discussion

6.4.1 Experiment Results

Figures 6.4 to 6.6 present the results obtained with the single DL models (Tabnet and CNN) and DLHEs. It can be seen that the R^2 values for the AE and WE DLHEs are higher than those of the single DL models, indicating that they are more accurate. The values of MAE, RMSE and MSE also show that the DLHEs (AE and WE) are more accurate than the single DL models.

Table 6.7 gives the SD values from five runs for each algorithm. It can be seen that the SD values of the AE and WE DLHEs are consistently lower than the SD values of the single DL models. Thus the ensembles containing the DL models are more consistent than the single DL models. Thus the ensembles of the DL models display a combination of increased accuracy and increased consistency, just as the ensembles of ML models displayed compared with the single DL models.

We used the same type of framework for these experiments with DL models that we used in Chapters 4 and 5 for our experiments with ML models. The fact that we have obtained comparable results shows that our framework is able to work with a wide range of base learner algorithms for building heterogeneous ensembles.

Our results also show that CNN achieved a greater accuracy than Tabnet, even though our data are tabular and Tabnet was developed for tabular data while CNN was not. This does suggest that there still needs to be more work done in developing DL methods that perform well on tabular data. Shwartz-Ziv and Armon (2022) concluded that in their experiments ML methods outperformed DL methods on tabular data, and our results are consistent with their conclusions.

The WE DLHEs achieved a slightly higher R^2 score than the AE, but the values of the other metrics were the same. The fact that there is such a similar performance from both WE and AE strongly suggests that their accuracies are extremely similar. This is also confirmed in that the values for the comparable individual runs for AE and WE (i.e. using the same random seed) also match (see Tables 6.5 and 6.6). We only had two DL models in these ensembles. With a larger number of different models, as with our experiments using ML models, it is unlikely that they would all give the same performance, so we would expect that the WE ensembles would be more accurate.

Tables 6.3 to 6.6 present the results of all five runs of each experiment we performed using DL algorithms, giving the values of all four metrics (R^2, MAE, RMSE and MSE) with the averages. It was not possible to present all the results of all our experiments in Chapter 5 in this way, but we have been able to do so in this chapter. It can be seen that the four metrics all show comparable patterns.

Figure 6.7 presents the results (average R^2 of five runs) obtained for single DL models, DLHEs (AE and WE), and also for the heterogeneous ensembles (AE and WE, using MSM1) containing two ML models. Figures 6.8 and 6.9 present the equivalent results for the heterogeneous ensembles (HEs) generated using

MSM1 with three and four ML models, respectively. (Note that the DLHE ensemble results presented in Figures 6.8 and 6.9 are for two model ensembles, because we only investigated two different DL algorithms.)

It can be seen that the heterogeneous ensembles with MLs were more accurate than the single DL models. They also had lower SD values. Thus they were more accurate and more consistent. We conclude from these results that our heterogeneous ML ensembles are better than the DL algorithms we have tested. We would note as well that the hyperparameters of the DL algorithms needed to be tuned in order to achieve the performance they did, while the ML algorithms we employed were used with the default parameters.

The DLHEs outperformed the single DL models and the HEs with MLs. However, their SD values were higher.

Table 6.3: R^2, MAE, RMSE and MSE values obtained with five runs of Tabnet using different random seeds when partitioning the data, and the average values.

Tabnet	R^2	MAE	RMSE	MSE
Run1	0.79288	1.05847	1.87057	3.49904
Run2	0.77431	1.08104	1.93952	3.76175
Run3	0.78511	1.04094	1.84547	3.40575
Run4	0.78271	1.07811	1.88220	3.54266
Run5	0.79404	1.03728	1.83332	3.36106
Avg	0.78581	1.05917	1.87422	3.51405

Table 6.4: R^2, MAE, RMSE and MSE values obtained with five runs of our CNN using different random seeds when partitioning the data, and the average values.

CNN	R^2	MAE	RMSE	MSE
Run1	0.79189	1.05642	1.84940	3.42027
Run2	0.78616	1.07217	1.88794	3.56431
Run3	0.78277	1.04595	1.85550	3.44287
Run4	0.79374	1.05871	1.83381	3.36286
Run5	0.79159	1.05487	1.84416	3.40093
Avg	0.78923	1.05762	1.85416	3.43825

Table 6.5: R^2, MAE, RMSE and MSE values obtained with five runs of DLHE using AE, and using different random seeds when partitioning the data, with the average values.

AE	R^2	MAE	RMSE	MSE
Run1	0.80591	1.02805	1.81080	3.27899
Run2	0.80849	1.02479	1.79872	3.23538
Run3	0.78735	1.03293	1.83585	3.37033
Run4	0.79344	1.04882	1.83514	3.36775
Run5	0.79633	1.03298	1.82310	3.32369
Avg	0.79830	1.03351	1.82072	3.31523

Table 6.6: R^2, MAE, RMSE and MSE values obtained with five runs of DLHE using WE, and using different random seeds when partitioning the data, with the average values.

WE	R^2	MAE	RMSE	MSE
Run1	0.80601	1.02805	1.81080	3.27899
Run2	0.80861	1.02479	1.79872	3.23538
Run3	0.78735	1.03293	1.83585	3.37033
Run4	0.79346	1.04882	1.83514	3.36775
Run5	0.79633	1.03298	1.82310	3.32369
Avg	0.79835	1.03351	1.82072	3.31523

Table 6.7: Summary of results with deep learning single models and deep learning heterogeneous ensembles using AE and WE, listing R^2, MAE, RMSE and MSE values, with the SDs.

M	R^2	SD	MAE	SD	RMSE	SD	MSE	SD
Tabnet	0.78581	0.00996	1.05917	0.02030	1.87422	0.04136	3.51405	0.15615
CNN	0.78923	0.00899	1.05762	0.00946	1.85416	0.02049	3.43825	0.07634
AE	0.79830	0.00879	1.03351	0.00923	1.82072	0.01601	3.31523	0.05821
WE	0.79835	0.00885	1.03351	0.00923	1.82072	0.01601	3.31523	0.05821

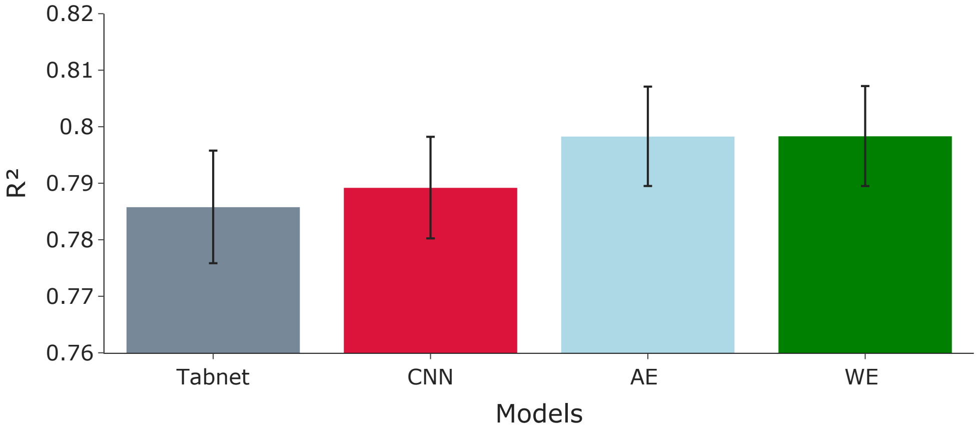


Figure 6.4: R^2 values with deep learning single models and deep learning heterogeneous ensembles using AE and WE. The SDs of the values are indicated by bars.

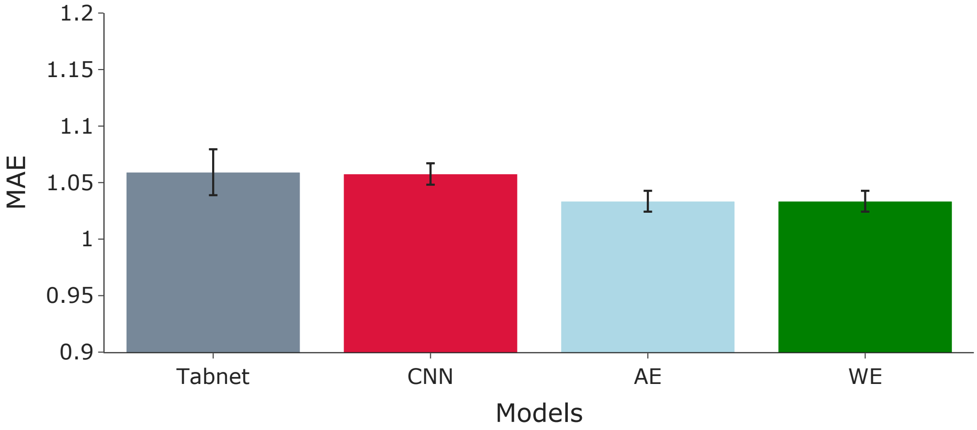


Figure 6.5: MAE values with deep learning single models and deep learning heterogeneous ensembles using AE and WE. The SDs of the values are indicated by bars.

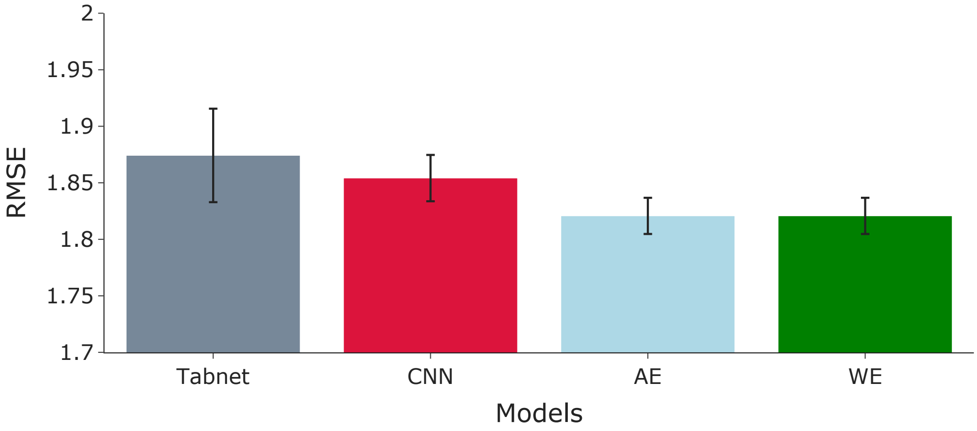


Figure 6.6: RMSE values with deep learning single models and deep learning heterogeneous ensembles using AE and WE. The SDs of the values are indicated by bars.

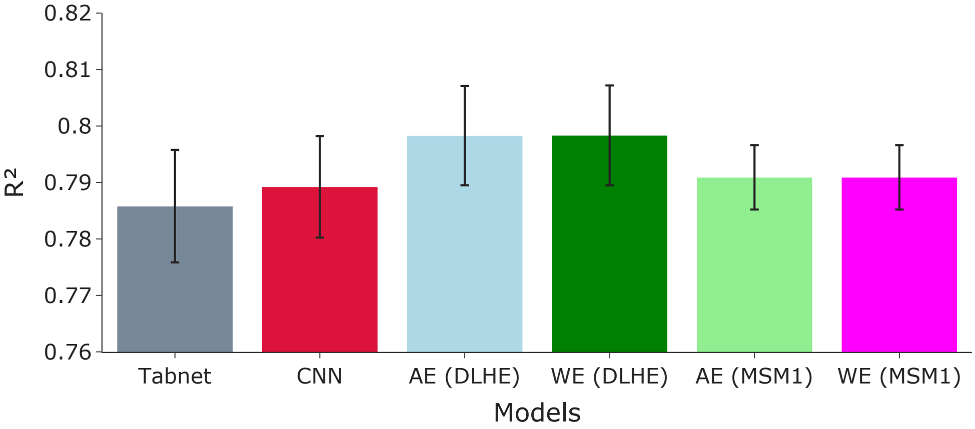


Figure 6.7: R^2 values with deep learning single models, deep learning heterogeneous ensembles, and machine learning heterogeneous ensembles using MSM1 size 2. The SDs of the values are indicated by bars.

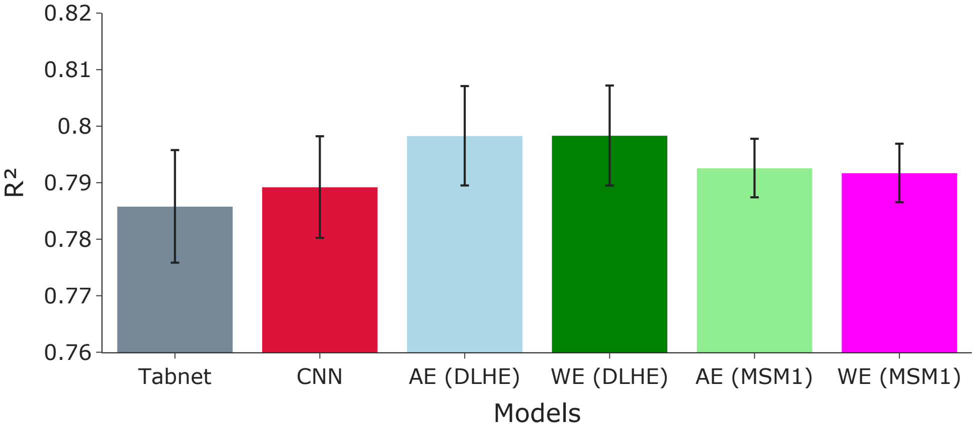


Figure 6.8: R^2 values with deep learning single models, deep learning heterogeneous ensembles, and machine learning heterogeneous ensembles using MSM1 size 3. The SDs of the values are indicated by bars.

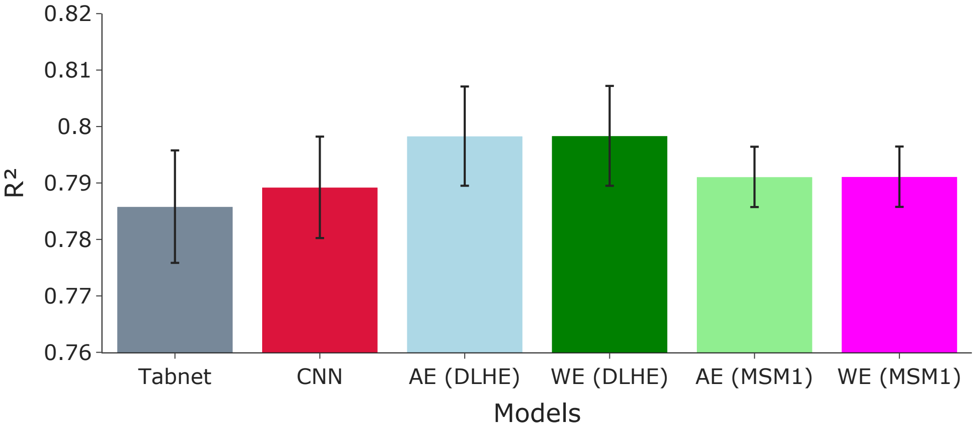


Figure 6.9: R^2 values with deep learning single models, deep learning heterogeneous ensembles, and machine learning heterogeneous ensembles using MSM1 size 4. The SDs of the values are indicated by bars.

6.4.2 Critical Comparisons

Figures 6.10 to 6.13 are CD diagrams comparing the values of metrics for the single DL models and the AE and WE DLHEs.

Figure 6.10 is the CD diagram of R^2 values. It can be seen that the ensembles (WE and AE) were both more accurate than the DL single models. However, the difference was not statistically significant, as indicated by the fact that they are all linked under one bar. The rank order, from highest to lowest accuracy, was WE, AE, CNN, Tabnet.

Figure 6.11 is the CD diagram of MAE values. This shows some differences with the R^2 diagram. The ensembles were more accurate than the DL single models, but the WE and AE ensembles were of the same accuracy. The ensembles were also statistically significantly more accurate than Tabnet, but the difference between the ensembles and CNN was not significant. The rank order, from highest to lowest accuracy, was AE and WE (equal), CNN, Tabnet.

Figure 6.12 is the CD diagram of RMSE values. With this measure the ensembles were more accurate than the DL single models, but the differences were not statistically significant. The AE and WE ensembles were of the same accuracy and the rank order, from highest to lowest accuracy, was AE and WE (equal), CNN, Tabnet.

Figure 6.13 is the CD diagram of MSE values. This measure gave the same overall pattern as RMSE, although the actual values were different. The ensembles were more accurate than the DL single models, but the differences were not statistically significant. The AE and WE ensembles were of the same accuracy and the rank order, from highest to lowest accuracy, was AE and

WE (equal), CNN, Tabnet. MSE and RMSE are similar measures, so it is not surprising that they gave similar patterns.

From these CD diagrams it can be seen that while the differences between the ensembles and the single DL models were only significantly different with the MAE measure, they were more accurate, which shows that the ensemble approach did improve the accuracy. These ensembles only contained two models, if more models could be included then it is possible that more of the measures would indicate that the ensembles were statistically significantly more accurate than the single models.

Figures 6.14 to 6.16 are CD diagrams of R^2 values obtained with DL models, DLHEs, and HEs generated using MSM1 containing from 2 to 4 models. (Note that in all three figures the DLHE ensemble results presented are for two model ensembles, because we only investigated two different DL algorithms.)

Figure 6.14 is the CD diagram for the 2 model HEs. It can be seen that the DLHE ensembles were the most accurate, then the HEs, then the DL single models. The rank order was WE(DLHE), AE(DLHE), AE(MSM1), WE(MSM1), Tabnet, CNN. Thus, the ensembles of DL models were more accurate than the ML ensembles of two models. None of the differences in accuracy was statistically significant.

Figure 6.15 is the CD diagram for the 3 model HEs. Here the DLHE ensembles were again the most accurate, then the ML ensembles. However, the WE(MSM1) ensemble was more accurate than the AE(MSM1) ensemble. This was probably because since the ML ensembles contained 3 models, instead of 2, the weighting mechanism was more effective. The DL single models had the lowest accuracy. The rank order was, WE(DLHE), AE(DLHE), WE(MSM1), AE(MSM1), CNN, Tabnet. None of the differences in accuracy was statistically significant.

Figure 6.16 is the CD diagram for the 4 model HEs. The rank order of the ensembles was the same: WE(DLHE), AE(DLHE), WE(MSM1), AE(MSM1). However, there was a large difference in accuracy between WE(MSM1) and AE(MSM1). This was probably because they contained 4 models and the fourth model was not very accurate, so the AE(MSM1) ensemble's accuracy was affected, but the weighting mechanism of the WE(MSM1) ensemble was able to compensate for the poor accuracy of the fourth model.

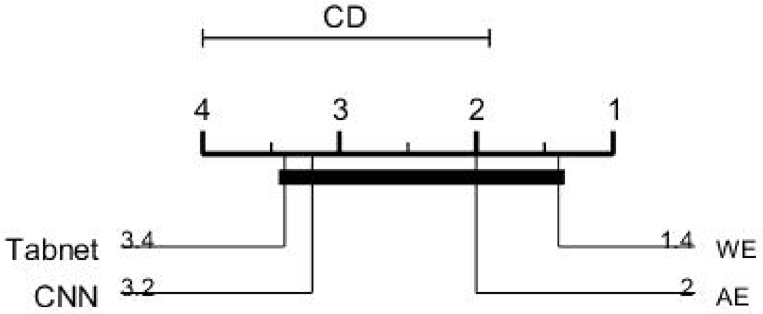
Overall, the results from the work presented in this chapter clearly show the benefit of our heterogeneous ensemble approach, whatever type of base learner is being used, and benefit of the weighting mechanism can also seen from the results presented.

6.5 Summary

In this chapter we have investigated the use of DL algorithms as base learners with our heterogeneous ensemble approach.

We investigated two different DL algorithms, Tabnet and CNN. Tabnet was particularly developed for use with tabular data, which our train delay data is.

We carried out hyperparameter tuning to these DL algorithms by applying a wide range of values, in order to get the best performance from them. We then employed the same framework as used in our previous experiments, but used



,

Figure 6.10: Critical difference comparison for R^2 values obtained with deep learning single models Tabnet and CNN, and deep learning heterogeneous ensembles using AE and WE.

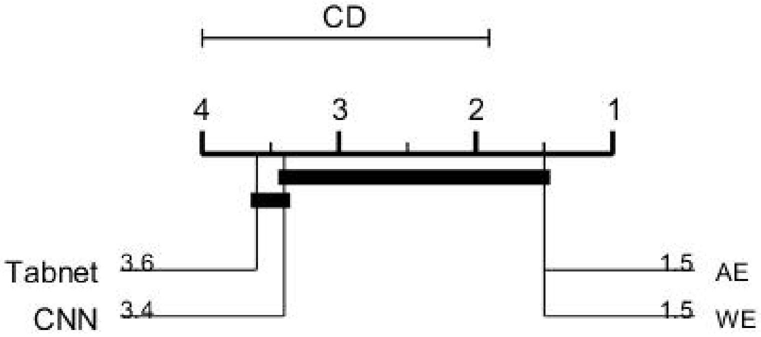


Figure 6.11: Critical difference comparison for MAE values obtained with deep learning single models Tabnet and CNN, and deep learning heterogeneous ensembles using AE and WE.

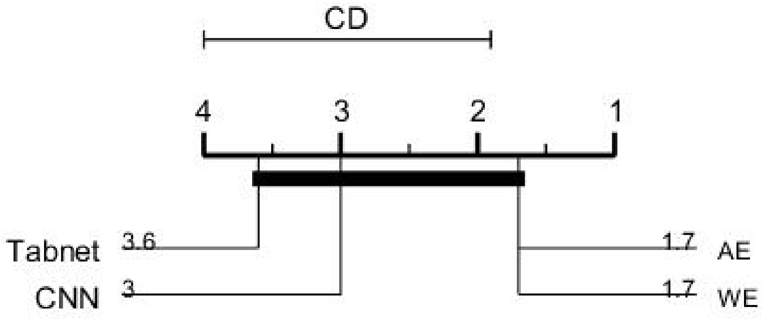


Figure 6.12: Critical difference comparison for RMSE values obtained with deep learning single models Tabnet and CNN, and deep learning heterogeneous ensembles using AE and WE.

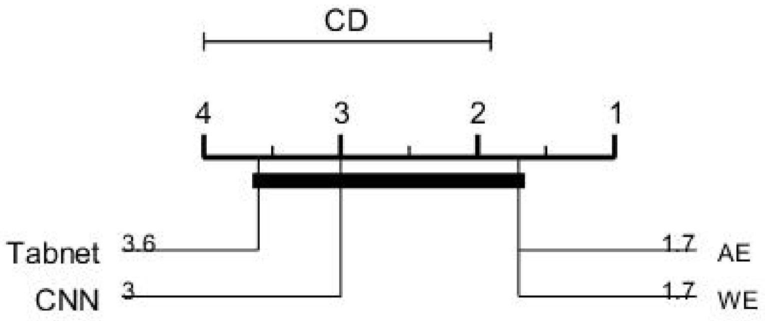


Figure 6.13: Critical difference comparison for MSE values obtained with deep learning single models Tabnet and CNN, and deep learning heterogeneous ensembles using AE and WE.

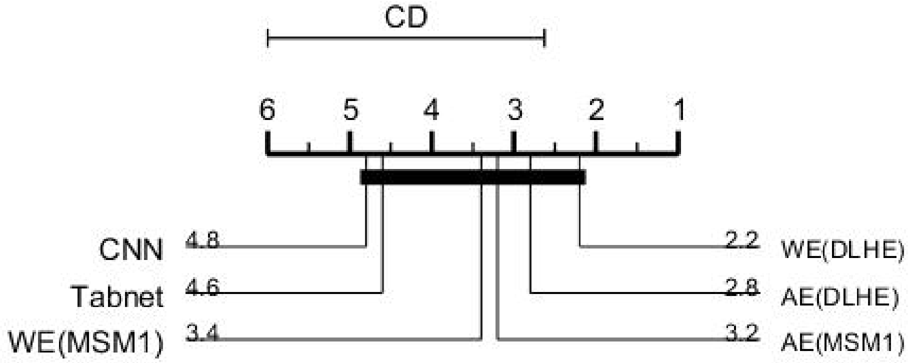


Figure 6.14: Critical difference comparison for R^2 values obtained with deep learning single models Tabnet and CNN; deep learning heterogeneous ensembles using AE and WE; and machine learning heterogeneous ensembles using MSM1, using AE and WE, containing 2 models.

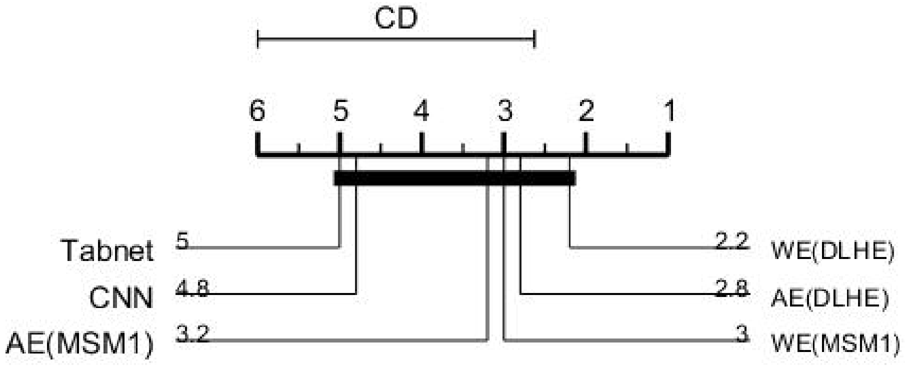


Figure 6.15: Critical difference comparison for R^2 values obtained with deep learning single models Tabnet and CNN; deep learning heterogeneous ensembles using AE and WE; and machine learning heterogeneous ensembles using MSM1, using AE and WE, containing 3 models.

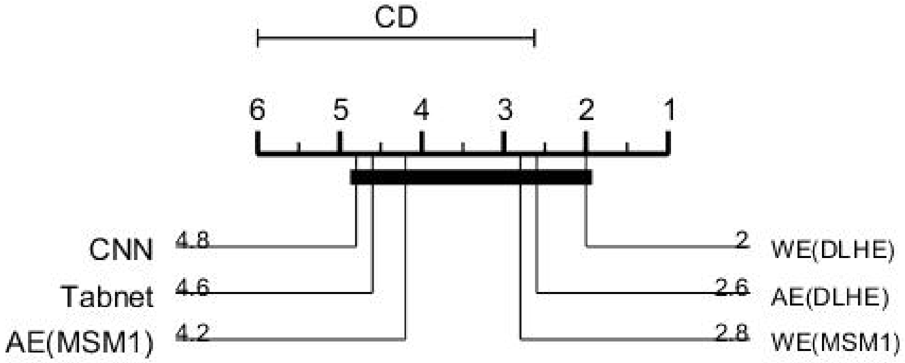


Figure 6.16: Critical difference comparison for R^2 values obtained with deep learning single models Tabnet and CNN; deep learning heterogeneous ensembles using AE and WE; and machine learning heterogeneous ensembles using MSM1, using AE and WE, containing 4 models.

the hyperparameter-optimised DL algorithms for the base learners instead of ML algorithms.

As in previous experiments we generated ensembles using Averaging and Weighted Averaging for the decision making function.

We assessed our results using four metrics: R^2, MAE, RMSE and MSE. We applied the Friedman test and viewed the results using critical difference diagrams. These enabled us to compare the DL algorithms and ensembles with our ML ensembles.

We found that heterogeneous ensembles of DL algorithms gave more accurate results than single DL models. We also found that heterogeneous ensembles of ML models gave better results than the single DL models, and were also more consistent. The DL ensembles were more accurate than the ML ensembles, but were not as consistent.

In our list of research questions we asked the question "Do heterogeneous ensembles perform better than state-of-the-art methods such as deep learning?" and we believe that the answer to this question is "Yes." Our results clearly show that our HEs perform better than the two DL methods tested. In addition we would note that our HEs achieve their performance using the default hyperparameters, while the DL algorithms needed hyperparameter tuning to achieve their results.

In the next chapter we will evaluate our work and discuss it in detail.

7 Evaluation and Discussion

7.1 Introduction

The aim of this research was to develop heterogeneous machine learning ensemble techniques for predicting train delays, that are more accurate and more reliable than single models. Having presented our results in Chapters 4 to 6, in this chapter we will give an overall evaluation of our work and discuss it further.

In addition we will present results from testing our methods on a new set of train delay data. This dataset is from a different UK railway region from the dataset we used to develop our methods, and testing our methods with it will enable us to assess how well our methods generalise to new data.

The rest of the chapter is organized as follows:

Section 7.2 We give an overview of our work.

Section 7.3 Presents and discusses the experimental results obtained with the new dataset.

Section 7.4 Summarises the chapter.

7.2 Overview

The ensemble approach seeks to combine the outputs of several models with the aim of giving more accurate and reliable outputs than any one single model, and it has been applied in many areas, including train delay prediction. Ensembles function like a human committee and they often do outperform single models. However, most ensemble methods produce *homogeneous* ensembles, that is, all the models are of the same type, e.g. decision trees. A *heterogeneous* ensemble, in contrast, consists of models of more than one type, e.g. decision trees and artificial neural networks. The reasoning behind this approach is that they are able to take advantage of the strengths of different types of model, and therefore can potentially give better performance than ensembles composed of only one type of model. Heterogeneous ensembles have been used in a number of problem areas, but, to date, there has only been one study applying them in train delay prediction. However, that particular study had a number of disadvantages: the ensembles only used three types of base learner and performed no model selection. In addition, the generated ensembles were sensitive to hyperparameter values and tuning was required, this means that they would not generalise well.

Because of the potential benefits of heterogeneous ensembles, we decided to undertake this research, in order to produce heterogeneous ensembles that perform well and would generalise well. To achieve our aim we set the following objectives:

1. To develop methods for generating ensembles that contain models generated by more than one type of algorithm.
2. To develop methods for selecting which models to include in the ensemble.
3. To evaluate the performance of the methods developed, and determine which of them are best for predicting train delays.
4. To evaluate how well the methods developed generalise to new data.

In order to achieve our first objective, we developed ensemble methods that utilised a wide range of machine learning algorithms and evaluated their performance. We included both well established methods such as decision trees, and also included state-of-the-art methods such as Random Forest and XGboost. This work was reported in Chapters 3 to 5. The ensemble framework was described in Chapter 3, the initial experiments were described in Chapter 4 and further experiments in Chapter 5. We also investigated the use of the Deep Learning methods Tabnet and CNN, and this work was reported in Chapter 6.

To achieve our second objective we developed methods for selecting models for inclusion in the ensemble. We investigated the use of two different criteria for model selection, *accuracy* and *diversity.* This work was reported in Chapter 5. For this we developed two model selection methods, MSM1, which only considers accuracy, and MSM2 which considers both accuracy and diversity.

MSM1 works by starting with a collection of models generated by different base learners that have been evaluated by a chosen metric such as R^2. These models are ranked according to accuracy and added iteratively to the ensemble starting with the most accurate; the accuracy of the ensemble is then evaluated and once the ensemble accuracy has reached its highest value the construction is terminated.

MSM2 starts by taking the most accurate model, then it adds further models to the ensemble, based on diversity. Thus the second model chosen is the one that results in the highest diversity in the ensemble, and further models are added based on diversity. For the diversity measure we tested both pairwise and non-pairwise measures. The majority of diversity measures were developed for classification ensembles and CFD has been reported to be the best non-pairwise measure of diversity. Therefore, we adapted it for use in the regression context. For pairwise diversity measures we used correlation, covariance and disagreement. The outputs of correlation and covariance were adapted to be on the same 0 to 1 scale as CFD and disagreement, to ensure compatibility.

To achieve our third objective we evaluated our ensembles in two different ways. For our initial experiments, described in Chapter 4, we used two measures, *percentage correct prediction after rounding* and *percentage within one minute after rounding*. These were based on the standard railway industry practice for measuring train delays. For our later experiments, described in Chapters 5 and 6, we used four standard statistical metrics: R^2, MAE, RMSE and MSE, in order to perform a rigorous assessment of our ensembles' performance. We also employed the Friedman test and viewed the results using critical difference diagrams.

To achieve our fourth objective we tested our methods on a new set of data that relates to a different railway region in the UK from the dataset used during the development of all our methods. This work will be described in Section 7.3, later in this chapter.

Based on the above objectives, we sought to answer the following research questions:

1. How should a heterogeneous ensemble be built so that it performs better than single models?
2. What factors should be taken into consideration when selecting models for including in an ensemble? A number of factors will be examined, including accuracy, diversity and ensemble size.
3. Is there any relationship between accuracy and diversity in the context of building an ensemble for performing regression?
4. How should the models be combined to produce better results?

5. Do heterogeneous ensembles perform better than state-of-the-art methods such as deep learning?

The first of these questions was addressed in Chapters 3 and 4. In Chapter 3 we described our framework for building heterogeneous ensembles. In Chapter 4 we described our initial experiments and presented the results from them.

The framework of our Heterogeneous Ensemble consists of the following five phases: (1) data preprocessing and feature extraction, (2) data partitioning, (3) modelling, (4) model selection, and (5) building the ensemble.

For phase (1) we used data collected by a colleague for a single rail route (Norwich to London Liverpool Street) in the Greater Anglia region of the UK rail network. This covered a period of 2 years 5 months. For our initial experiments described in Chapter 4 we used the first 7 months of the dataset and for our later experiments we used the entire dataset. We performed cleansing and preprocessing of the raw data so that it was in a suitable format to use for modelling. It is essential that appropriate cleansing and preprocessing is performed on any dataset before it is used for machine learning.

We also had weather data for the first 7 months of our train delay dataset. We did not have weather data for the remainder of the period, as explained in Section 5.3.

In phase 2 we split the dataset into training, validation and testing subsets. This is a very important practice to follow. The use of the separate testing dataset ensures that the algorithms building the models have not “seen” the data on which they are tested and so the test data can be used to provide an independent check on the model accuracy. In the same way use of a validation dataset provides an independent means of assessing the performance of the models during the ensemble building process. This is important, for example, when weights are assigned to models within the final ensemble, based on their performance. When performing the splitting we used a random seed to ensure reproducibility, the use of different random seeds also allows different splits of the dataset to be made so that multiple repetitions of a given experiment can be performed, and the results compared to check for reproducibility.

Phase 3 is the modelling phase. Here we used a wide variety of machine learning algorithms as base learners. This was important because these algorithms work in different ways, and so an ensemble built from them is heterogeneous. In principle, any algorithm that can fit a regression model to data can be used, and so we were able to include a wide spectrum of algorithms including basic linear regression, K-nearest neighbours and state-of-the-art methods such as Random Forest and XGboost. We were also able to use the Deep Learning methods Tabnet and CNN. This ability to use a wide variety of algorithms is the key to the success of heterogeneous ensemble methods.

In phase 4 model selection is performed. This phase was not implemented in our initial experiments described in Chapter 4, but was developed in Chapter 5. The use of model selection is a key part of our completed ensemble approach. Some algorithms will perform better than others on different datasets. Selecting the best models is essential in order to generate the best ensemble. Therefore we devised a model selection process that selected the best performing models one by one, until the ensemble reached its highest accuracy. For our research we continued the building of the ensemble in order to see how the performance changed as the more poorly performing models were added, but in a real world application model selection would stop when peak performance was reached.

In order to select models we devised two methods, which we called MSM1 and MSM2. MSM1 selects models based on their accuracy, while MSM2 selects models on the basis of accuracy and diversity. We devised these two methods because it is generally accepted that the two important factors affecting the overall accuracy of an ensemble are the accuracy and diversity of its component models, and we wished to investigate how these factors could be used to build our heterogeneous ensembles. In principle, any diversity measure that can be used with regression ensembles could be used in this context, and we tested both pairwise and non-pairwise methods. In practice it is necessary for them to measure diversity on the same scale, and we modified some of the methods we used so that they measured diversity on a scale from 0=no diversity to 1=maximum diversity. Also, most diversity measures were developed for classification ensembles and cannot work directly with regression ensembles. We therefore modified CFD so that it could work in the regression context.

The final phase is phase 5 where the decision making function combined the outputs of the models in the ensemble. This is a very important stage in the process. We devised two different decision making functions. The first, which we called *AE*, combined the outputs by averaging. The second, which we called *WE*, combined them by weighted averaging, based on their accuracy on the validation data. It would be expected that different base learners would perform differently on a given dataset, and therefore weighting the model outputs according to their performance might be expected to result in a more accurate ensemble. We devised these two methods in order to test this.

The experiments described in Chapter 4 used the framework presented in Chapter 3, apart from the model selection step which was investigated in Chapter 5.

In these experiments we compared the performance of single models and the performance of our heterogeneous ensembles. Table 4.1 lists the key results from these experiments.

It can be seen that for every pair of stations the AE and WE ensembles always had higher accuracy than the average accuracy of single models. In addition, the SD values were always lower for the ensembles than for the average accuracy of single models. Thus, the ensembles were more accurate and more consistent. It is important to note that in these initial experiments no model selection was being performed, and all the models were put into the ensemble. The better performance of the ensemble compared with the average of single models was entirely due to the ensemble principle working effectively.

We can therefore conclude that the first research question has been answered, in that we have demonstrated the building of ensembles that are consistently more accurate than the average of single models.

Question 2 was "What factors should be taken into consideration when selecting models for including in an ensemble?" and question 3 was "Is there any relationship between accuracy and diversity in the context of building an ensemble for performing regression?"

In our experiments in Chapter 5 we investigated the use of accuracy and diversity when building ensembles. Our model selection methods MSM1 and MSM2 used accuracy, and accuracy and diversity, respectively, when building ensembles. We found that ensembles produced using MSM1 were more accurate than those produced by MSM2. Figure 7.1 presents the R^2 values obtained for MSM1, with averaging, weighted averaging and for the average of single models. It shows that the ensembles had more accurate results than any of the single models. The AE and WE ensembles had similar results, how-

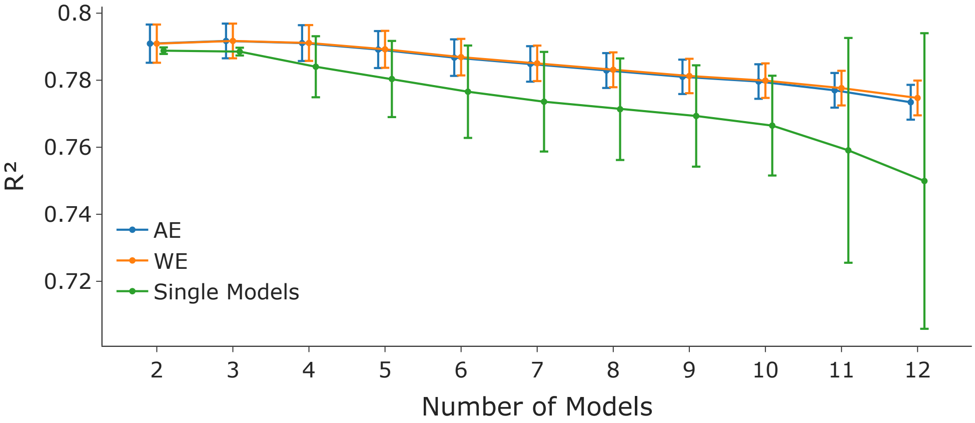


Figure 7.1: R^2 values for AE, WE and average of single models using MSM1, with SD values indicated by error bars. The ensembles showed higher accuracy than the single models. Repeat of Figure 5.3

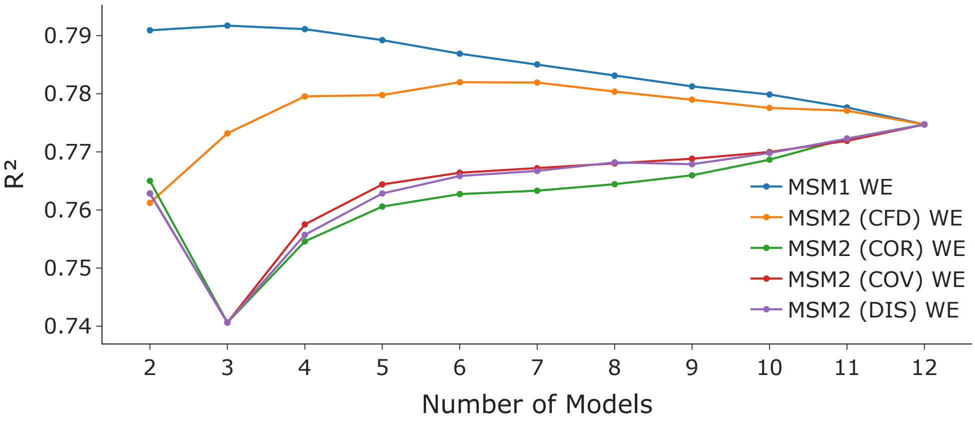


Figure 7.2: R^2 values obtained with WE using the two selection methods MSM1 and MSM2, different diversity measures with MSM2, and different ensemble sizes. Repeat of Figure 5.10.

ever, for when a larger number of models had been added the WE ensembles were slightly more accurate. This no doubt reflects the fact that the less accurate models were being given lower weighting by WE, and hence the overall ensemble accuracy was higher.

Figure 7.2 presents the results with MSM1 and MSM2 with WE ensembles. It can be clearly seen that MSM1, which uses accuracy for model selection, gives more accurate results than all the variants of MSM2 which uses accuracy and diversity for model selection. This suggests that model accuracy is the most important factor to consider when building a heterogeneous regression ensemble and that diversity is not as important. This is different from what has been observed with classification ensembles and is somewhat surprising. However it must be remembered that there is no general agreement on a definition of diversity for either classification and regression ensembles, and also most measures of diversity were developed for classification ensembles. We adapted CFD to work in the regression context, because it is recognised as being the best measure of diversity for classification ensembles. As we noted in Section 5.5 the performance drops with 3 models and then improves as more are added. This is due to selection mechanism failing in its aim of selecting the best models early on in the building of the ensemble and it appears to be selecting the worst performing model as the third model. This would be due to this model resulting the greatest diversity in the ensemble. The performance then improves as other, better performing models are added.

It is interesting to note that of the diversity measures we tested for MSM2, it resulted in the most accurate ensembles, even thought they were not as accurate as those obtained with MSM1. (See Figure 7.2.) The other measures tested were all pairwise measures, which in the classification context are recognised as not performing as well as non-pairwise measures. Therefore, when using diversity for model selection in the context of a regression ensemble, the performance of the measures are comparable with the classification context. This does suggest that the diversity measures are capturing the diversity of the ensemble to some degree. However, the fact that simply selecting models on the basis of accuracy (MSM1) gives more accurate ensembles may indicate that not all the diversity is being captured, i.e. they are not as effective at capturing diversity in the regression context.

Overall our results do not show any relationship between diversity and ensemble accuracy. Intuitively this should not be the case and we think it is likely that the measures of diversity used are not reflecting the actual diversity. An alternative explanation is that because the first model selected is the most accurate, then the model that is most diverse from this one is likely to be less accurate and therefore when the second model is selected it sometimes does not improve the accuracy of the ensemble or even causes a drop in accuracy when it is added. One possible way to mitigate this would be to require a minimum level of accuracy before a model could be added, even if it would result in higher diversity. Overall, our results suggest that more research is needed on the measurement of diversity in regression ensembles.

Our methods are able to determine the optimum ensemble size, and we found that a small number of models were able to give excellent performance. The optimum ensemble size was 3-4 models. (See Figure 7.2.)

This shows that excellent performance can be achieved with a small number of models if they are selected properly, and our methods are able to do this.

The fact that our methods are able to select the best models from the collection of models means that all the models do not need to be included in the ensemble. This shows the benefit of the model selection strategy.

It is possible that a different dataset may need a larger number of models. In such a situation our methods would identify this and put the appropriate models into the ensemble. This means that our methods would generalise to other types of data and produce the optimum size of ensemble.

We consider that questions 2 and 3 of our research questions have been answered by these results, in that accuracy has been found to be the most important factor to take into consideration when building ensembles, and that we have found no relationship between diversity and accuracy in the context of building a regression ensemble.

We have found throughout our experiments that weighted averaging of the models according to accuracy gave the most accurate ensembles. We employed weighting according to R^2, but other metrics could be used and this would be a suitable area for future research. Where the models have similar performance, the weights will, of course, be similar, but where there is a large variation in model performance there will be a greater difference in the weights. Thus poorly performing models will have less effect on the ensemble accuracy, and the ensembles will show a greater improvement over those that use just accuracy.

This is seen in the results presented in Figure 7.3. This shows the results obtained with MSM2 using the disagreement diversity measure. In this case the models have been selected according to disagreement, i.e. they are different in their predictions and as a result poorly performing models are put into the ensemble at an early stage of its construction. Here the WE ensembles show

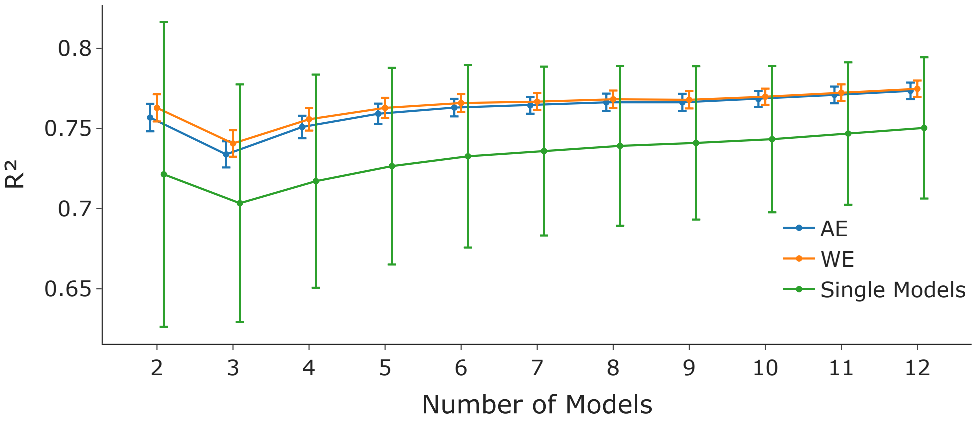


Figure 7.3: R^2 values for AE, WE and average of single models using MSM2 and disagreement as the diversity measure. SD values are indicated by error bars. Repeat of Figure 5.15

a higher accuracy than the AE ensembles, and as the ensemble construction continues, and better performing models are added, performance of both AE and WE improve, and the difference between them becomes less. We consider that the results with the AE and WE have answered our 4th research question.

We investigated the use of Deep Learning (DL) algorithms with heterogeneous ensembles. Table 7.1 presents the results obtained for two types of single DL models, Tabnet and CNN, and for AE and WE DLHEs constructed using these models. It can be seen that the DLHEs are more accurate than the single DL models. Figure 7.4 compares the R^2 values obtained with the DL single models, DLHEs and HEs. The HEs show higher performance than the DL single models, but the DLHEs show higher performance than the HEs. Thus, we can conclude that our heterogeneous ensembles do outperform state-of-the-art deep learning methods, and we therefore conclude that we have answered question 5 of our research questions. We would note as well that

Table 7.1: Summary of results with deep learning single models and deep learning heterogeneous ensembles using AE and WE, listing R^2, MAE, RMSE and MSE values, with the SDs. Repeat of Table 6.7

M	R^2	SD	MAE	SD	RMSE	SD	MSE	SD
Tabnet	0.78581	0.00996	1.05917	0.02030	1.87422	0.04136	3.51405	0.15615
CNN	0.78923	0.00899	1.05762	0.00946	1.85416	0.02049	3.43825	0.07634
AE	0.79830	0.00879	1.03351	0.00923	1.82072	0.01601	3.31523	0.05821
WE	0.79835	0.00885	1.03351	0.00923	1.82072	0.01601	3.31523	0.05821

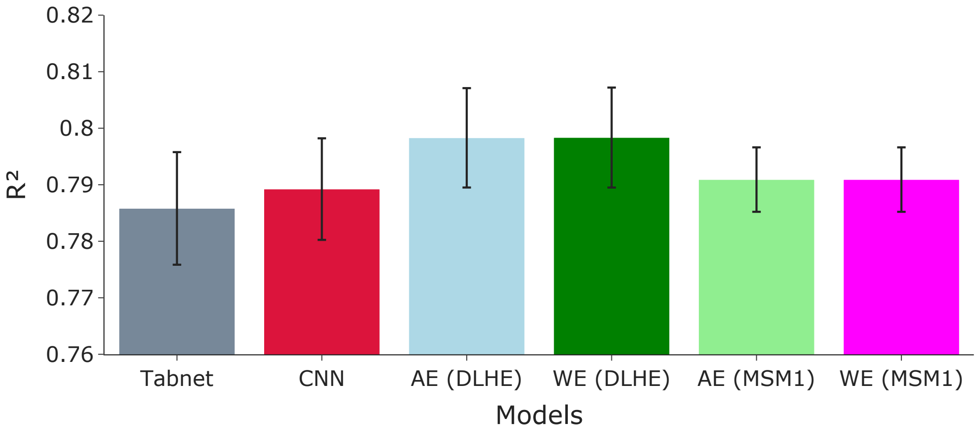


Figure 7.4: R^2 values with deep learning single models, deep learning heterogeneous ensembles, and machine learning heterogeneous ensembles using MSM1 size 2. The SDs of the values are indicated by bars. Repeat of Figure 6.7

our heterogeneous ensemble approach can be used with DL base learners to generate ensembles that outperform single DL models, which also shows the benefit of it. It would, of course, be perfectly possible to generate ensembles containing both ML and DL models, and this would be a suitable area for future research.

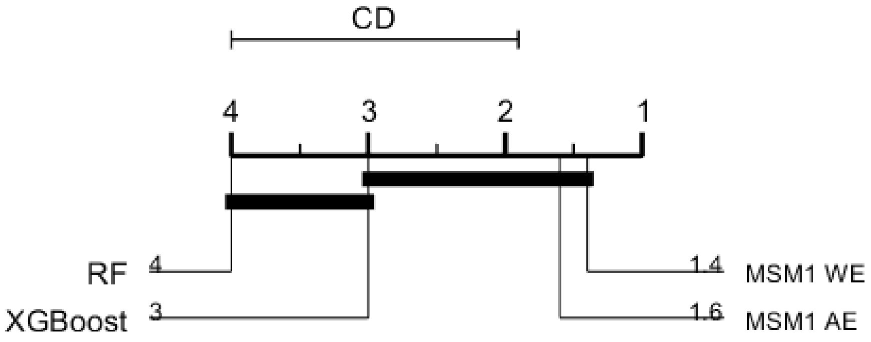


Figure 7.5: Critical difference comparison for ensembles size 3 with MSM1 AE and MSM1 WE, Random Forest model, and XGBoost model. (Repeat of Figure 5.20)

7.2.1 Comparison of Heterogeneous and Homogeneous ensembles

The work of this thesis was undertaken because heterogeneous ensembles have been reported to perform better than homogeneous. We have developed heterogeneous ensembles and shown that they gave more accurate results than single models. However, we can also compare our heterogeneous ensembles with existing state-of-the-art homogeneous ensembles, Random Forest and XGBoost, since these methods were used as base learners when developing our heterogeneous ensembles.

Figures 7.5 and 7.6 are CD diagrams comparing the results obtained using Random Forest and XGBoost with heterogeneous ensembles sized 3 and 4, respectively. It can be seen that the heterogeneous ensembles sized 3 and 4 gave better performance that the homogeneous ensembles Random Forest and XGBoost. The results obtained using a new dataset of train delay data, described below in Section 7.3 and presented in Figure 7.10, show comparable patterns.

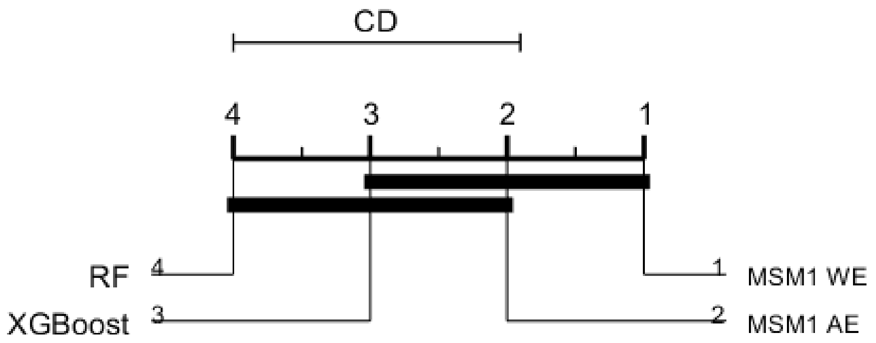


Figure 7.6: Critical difference comparison for ensembles size 4 with MSM1 AE and MSM1 WE, Random Forest model, and XGBoost model. Repeat of (Figure 5.21)

The rank order from highest to lowest accuracy was the same (WE(MSM1), AE(MSM1), XGBoost, RF), and WE(MSM1) was statistically significantly more accurate than RF. These results show that our heterogeneous ensembles do give more accurate results than homogeneous ensembles, and confirm that our hypothesis that heterogeneous ensembles would give more accurate results than homogeneous ensembles was correct.

7.3 Application of our Heterogeneous Ensemble methods to a new dataset

In order to investigate how our heterogeneous ensemble methods would generalise to new data we tested them on a second dataset of train delay data. This contained data for the Weymouth to London Waterloo line of the UK South Western Railway region. This dataset covered a two year period from 2017 to 2018. Using data from a different region should provide a good indication of how robust our methods are to handling different datasets.

This dataset contained 128120 instances, and our previous (Norwich to London Liverpool Street) dataset that we used to develop our methods contained 107431, making the new dataset slightly larger.

We tested this new dataset with our best performing methods, MSM1 using AE and WE. We did not test MSM2 because it was not as effective as MSM1 and here we were concerned with testing the generalisation ability of our heterogeneous ensemble methods, and because MSM1 gave the best performance.

Table 7.2 presents the results we obtained, and they are are plotted in Figure 7.7. It can be seen that the values for AE and WE were very similar, being identical with for the smaller ensembles, but as more models were added the WE ensembles had slightly higher accuracy than the AE ones. The single model performance was always lower than that of the ensembles. This is broadly the same pattern that was seen with the NRW–LST dataset. This is also seen in the SD values, where that of the single models increases much more than that of the ensembles as the size of the ensemble increases. Thus the overall patterns of results obtained with the two datasets are very similar.

Figures 7.8 and 7.9 show plots of the R^2 values obtained using MSM1 with the NRW–LST and WEY–WAT datasets. It can be seen that higher accuracies were obtained for the WEY–WAT dataset than for the NRW–LST dataset. This is very interesting, because the ensembles have performed better on the new dataset than on the dataset that was used to develop our methodology. There may be a number of reasons for this, but we would note that the fact that our ensembles performed better on the new dataset is very strong evidence that they generalise to new data.

It can be seen that there is no difference in accuracy between the AE and WE ensembles for the smaller sized ensembles. (See Table 5.1 for NRW–LST and Table 7.2 for WEY–WAT.) This is almost certainly due to there being no difference in the accuracy of the most accurate individual models, and so there is no difference in the weighing being applied to them. As the less accurate models are added to the ensemble, the WE ensembles then become slightly more accurate than the AE ensembles, because in them less weight is applied to the less accurate models.

Overall, we can conclude that whatever the nature of the data, our heterogeneous ensembles are very likely to perform better than any single models in terms of accuracy and consistency and so are generalisable.

Figures 7.10 to 7.12 are critical difference diagrams for the results obtained with the new dataset. Figure 7.10 presents the results for WE and AE ensembles size 2 with the state-of the-art methods random forest and XGboost. It can be clearly seen that the ensembles are better than these methods, as was the case when they were applied to the NRW–LST dataset.

Table 7.2: R^2 values and SDs obtained with the Weymouth to London Waterloo dataset for MSM1 using AE and WE, and for average of single models.

M	MSM1 AE		MSM1 WE		Average of single models	
	R^2	SD	R^2	SD	R^2	SD
2	0.8298	0.0062	0.8298	0.0062	0.8272	0.0015
3	0.8296	0.0067	0.8296	0.0067	0.8268	0.0014
4	0.8286	0.0065	0.8286	0.0065	0.8218	0.0097
5	0.8253	0.0068	0.8255	0.0068	0.8146	0.0169
6	0.8214	0.0072	0.8218	0.0071	0.8086	0.0213
7	0.8183	0.0072	0.8188	0.0071	0.8025	0.0256
8	0.8149	0.0072	0.8156	0.0071	0.7986	0.0271
9	0.8116	0.0072	0.8124	0.0071	0.7957	0.0277
10	0.8086	0.0072	0.8095	0.0071	0.7917	0.0276
11	0.8041	0.0073	0.8059	0.0072	0.7796	0.0505
12	0.7985	0.0073	0.8015	0.0073	0.7674	0.0651

Figure 7.11 also includes the single models and overall we can see that the ensembles were best. The ensembles were significantly better than RF. The rank order from highest to lowest accuracy was WE, AE, XGBoost, single models, RF. The ensembles were significantly more accurate than RF, the other differences were not statistically significant. Figure 7.12 presents the results for ensembles size 3. Here the ensembles again are seen to be more accurate than single models, XGBoost and RF. They are also consistently better, while while XGboost is now in 4th position, compared to 3rd in Figure 7.11. As in Figure 7.11 the ensembles are significantly more accurate than RF and the other differences are not statistically significant.

7.4 Summary

This study was undertaken in order to develop heterogeneous ensembles for predicting train delays. Previously there has only been one application of

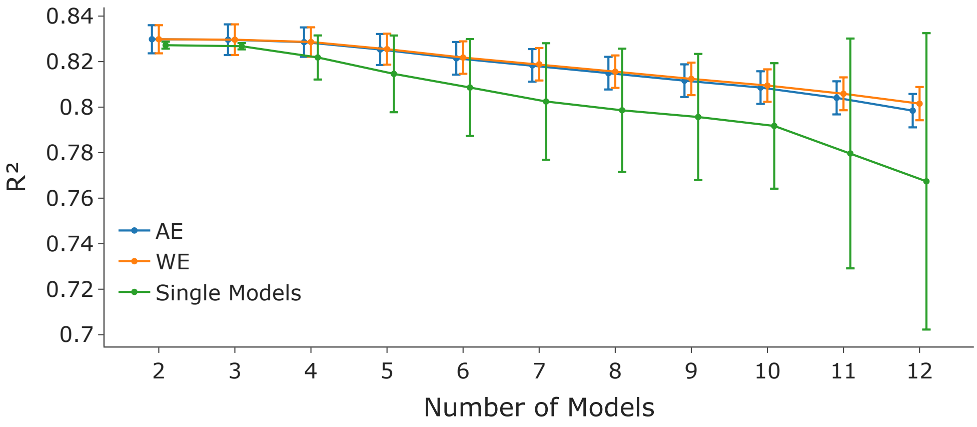


Figure 7.7: Plots of the R^2 values obtained using MSM1 AE, MSM1 WE and average of single models with the WEY–WAT dataset. SD values are indicated by error bars.

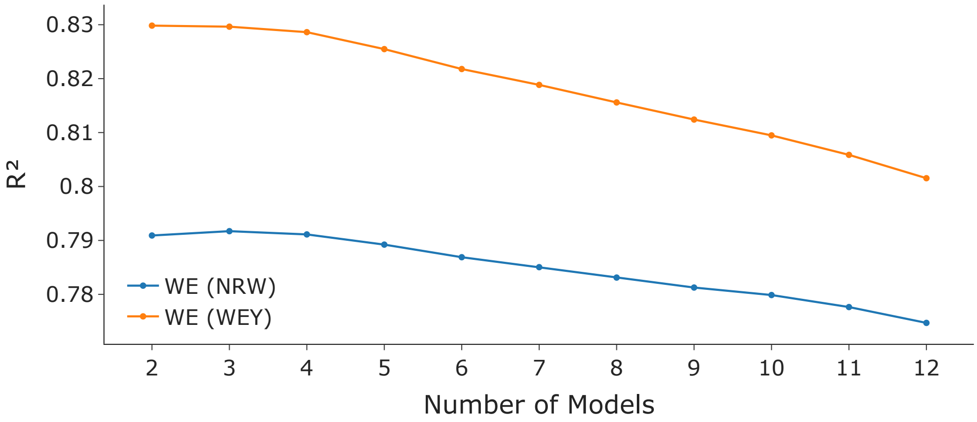


Figure 7.8: Comparison of R^2 values obtained using MSM1 (WE) on the two datasets: the first one (NRW) and the new one (WEY). The results show that the WE ensembles have reproduced the good performance on a new dataset.

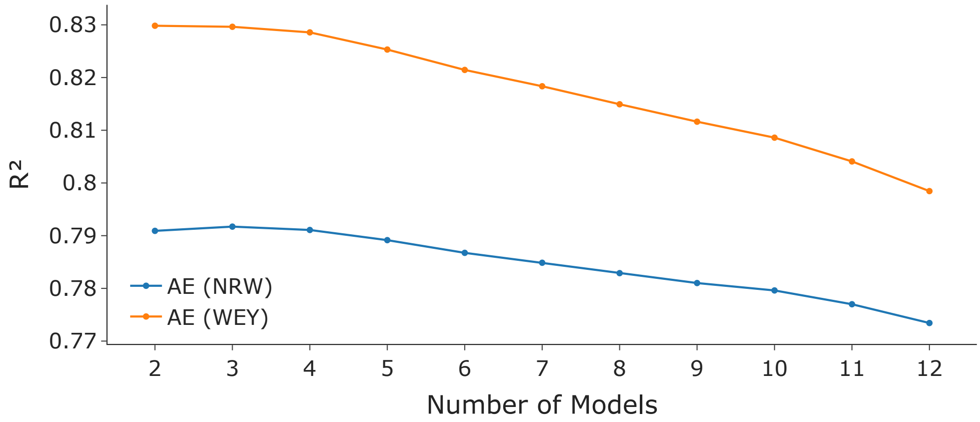


Figure 7.9: Comparison of R^2 values obtained using MSM1 (AE) on the two datasets: the first one (NRW) and the new one (WEY). The results show that the AE ensembles have reproduced the good performance on a new dataset.

heterogeneous ensembles to train delay prediction and that was a very limited study that did not investigate a wide variety of algorithms or mechanisms for combining models. It was sensitive to hyperparameter values and was therefore not generalisable. We undertook a much deeper study and investigated the use of a wide variety of models, model selection and methods for combining models. We also investigated the use of deep learning. We compared our methods with well-known algorithms and evaluated them using a number of metrics. We also applied the Friedman test and used CD diagrams to view our results. Finally we used a new dataset and applied our methods to it. The results showed that our methods generalise to new data.

At the start of this study an number of objectives were set, in this chapter we have shown that were all met. We also asked a number of research questions which we have shown were all answered.

Our objectives were:

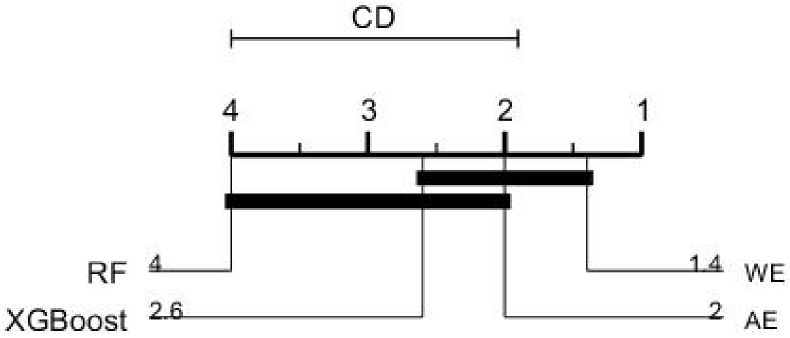


Figure 7.10: Critical difference comparison for RF; XGboost; and MSM1 AE and MSM1 WE, size 2; for WEY–WAT dataset.

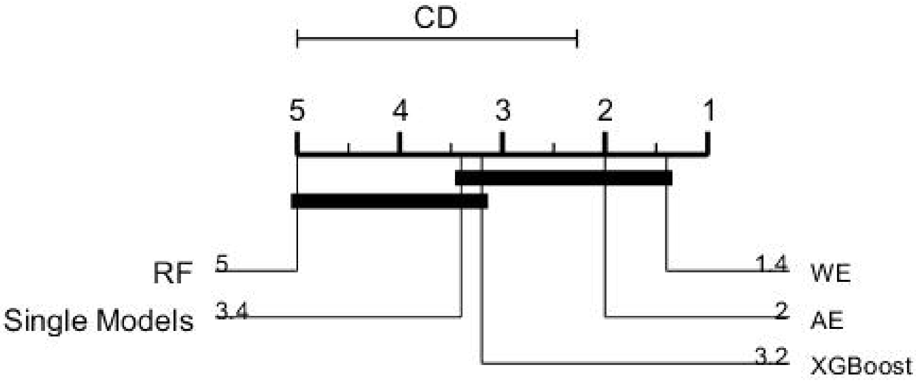


Figure 7.11: Critical difference comparison for RF; XGboost; single models; and MSM1 AE and MSM1 WE, size 2; for WEY—WAT dataset.

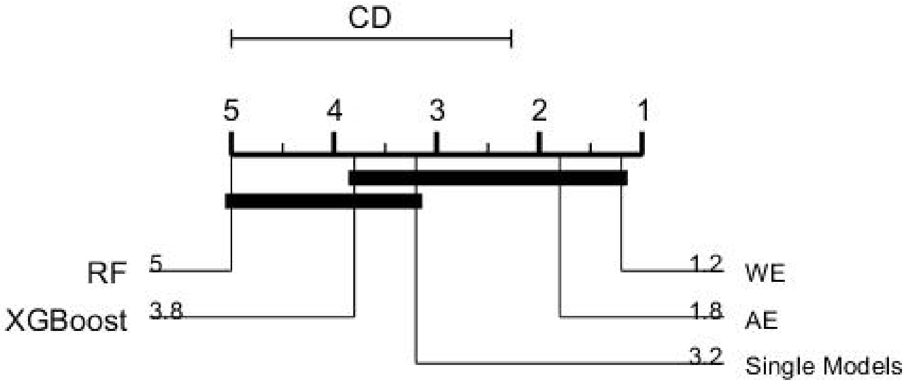


Figure 7.12: Critical difference comparison for RF; XGboost; single models; and MSM1 AE and MSM1 WE, size 3; for WEY—WAT dataset.

1. To develop methods for generating ensembles that contain models generated by more than one type of algorithm.
2. To develop methods for selecting which models to include in the ensemble.
3. To evaluate the performance of the methods developed, and determine which of them are best for predicting train delays.
4. To evaluate how well the methods developed generalise to new data.

Objective 1 was met in that we have developed methods for generating heterogeneous ensembles from a wide variety of algorithms. For the base learners we tested basic methods such as decision trees, state of the art methods random forest and XGboost; we also tested the use of the deep learning methods CNN and Tabnet. An important point to note about our heterogeneous ensembles is that they can use any regressor as the base learner, including homogeneous ensemble methods.

Objective 2 has been met in that we developed two model selection methods, MSM1 and MSM2, that select models based on accuracy, and accuracy and diversity, respectively. These methods work by adding models to the ensemble until maximum accuracy is reached. They enable an ensemble to be generated that achieves maximum accuracy and minimum size. (For development purposes we added models to the ensemble until all were used, but this was in order to see what effect they had on the ensemble.)

Objective 3 was met in that we employed various metrics and tested our results statistically. We found that the best method for predicting train delays was an ensemble generated by selecting models based on accuracy and employing a decision making function that used weighted averaging. For our initial experiments, described in Chapter 4, we used a measure based on the way train operators measure train performance according to the percentage that are on time or late. In our later experiments we employed a number of metrics to perform a more rigorous evaluation of the performance of our methods. We have statistically tested our results for significance.

Objective 4 was met by applying our methods to a new dataset of train delay data, for a rail route operated by a different company in a different part of the country. We found that our methods performed well on this data and we therefore concluded that they are generalisable. In fact, our methods gave greater accuracy with this new dataset than the one we used during their development. While this was almost certainly due to the characteristics of the dataset, it nevertheless indicates how well our methods generalise.

Our research questions were:

1. How should a heterogeneous ensemble be built so that it performs better than single models?
2. What factors should be taken into consideration when selecting models for including in an ensemble? A number of factors will be examined, including accuracy, diversity and ensemble size.
3. Is there any relationship between accuracy and diversity in the context of building an ensemble for performing regression?
4. How should the models be combined to produce better results?

5. Do heterogeneous ensembles perform better than state-of-the-art methods such as deep learning?

Question 1 we devised a framework for building heterogeneous ensembles, which we described in Chapter 3. This was used in our initial experiments in Chapter 4, where we built heterogeneous ensembles using a wide variety of base learner models. We devised two decision making functions, one combined the model outputs using averaging, the other using weighted accuracy. Both these types of heterogeneous ensemble achieved higher accuracy than any of the single models, including the state-of-the-art methods XGboost and random forest. In our further experiments in Chapters 5 and 6 the same decision making methods were used and our heterogeneous ensembles consistently outperformed both the machine learning methods and also the deep learning methods, CNN and Tabnet. Thus we consider that we have answered this question.

Question 2 and 3 we investigated the use of accuracy and diversity when building ensembles in our experiments in Chapter 5. We devised two methods for selecting models, MSM1 and MSM2. MSM1 only considers accuracy, while MSM2 considers both accuracy and diversity. We found that accuracy was the most important factor to take into consideration when building ensembles. We found no relation between diversity and ensemble accuracy. In regard to ensemble size, we found that maximum accuracy was usually achieved with between 3 and 4 models in the ensemble. While the optimum number of models may vary with individual datasets, the fact that our ensembles can achieve maximum accuracy with a small number of models means that the ensemble size can be kept small. Thus we consider that we have answered questions 2 and 3.

Question 4 we built ensembles using two different decision making functions, the first using averaging and the second using weighted averaging. We found that the weighted averaging gave consistently better accuracy than averaging. Thus we concluded that the best way to combine the models was using weighted averaging. We therefore consider that we have answered this question.

Question 5 we investigated the use of deep learning algorithms in Chapter 6. We tested the deep learning methods Tabnet, which was specifically designed for use with tabular data, (which our train delay data is) and CNN, which is a general deep learning algorithm. We found that our machine learning heterogeneous ensembles were more accurate than the deep learning methods. In addition we found that heterogeneous ensembles built with the deep learning algorithms were more accurate than both the individual deep learning models and the heterogeneous ensembles using machine learning methods. Therefore we can answer this question in the positive: yes, heterogeneous ensembles perform better than state-of-the-art methods such as deep learning; but we would also note that deep learning methods can be used as base learners in heterogeneous ensembles.

Overall, we consider that we have met all our objectives and answered all our research questions. In the next chapter we will draw our final conclusions and make our suggestions for further work.

8 Conclusions and Suggestions for Further Work

8.1 Conclusions

For this thesis we have developed heterogeneous ensemble techniques for predicting train delays using regression models for the base learners. Heterogeneous ensembles have proven effective in other application areas, but almost nothing has been done in the field of train delay prediction using heterogeneous ensembles.

We investigated how heterogeneous ensembles can be built in order to achieve accuracy and consistency. In order to do this we devised a framework for building heterogeneous ensembles, which we used as the basis for all our experiments.

We performed initial experiments which investigated how models should be combined in a heterogeneous ensemble, and tested the use of a wide variety of base learners.

We investigated the use of accuracy and diversity for model selection when building an ensemble.

We investigated how heterogeneous machine learning ensembles compare with state-of-the-art machine learning methods and deep learning methods.

As a result of our research we have achieved the following:

1. We devised two methods for combining the models in an ensemble, one using averaging, the other using weighted averaging.
2. We devised two methods for selecting which models to include when building an ensemble, one using accuracy, the other using accuracy and diversity.

3. Our model selection methods were able to achieve maximum ensemble accuracy using a small number of models.

4. We showed that selecting models on the basis of accuracy resulted in more accurate ensembles than using accuracy and diversity.

5. We found no relationship between ensemble diversity and ensemble accuracy.

6. We showed that our heterogeneous ensembles were more accurate than any of the single models tested, including the state-of-the-art methods random forest and XGboost.

7. We also showed that our heterogeneous ensembles were more accurate than the deep learning algorithms Tabnet and CNN.

8. We were able to use deep learning algorithms (Tabnet and CNN) as base learners in heterogeneous ensembles, and found that they were more accurate than heterogeneous ensembles using machine learning algorithms as base learners.

9. We showed that our methods generalise to new data by testing them on a new train delay dataset from a different UK railway region.

In summary, the work undertaken for this thesis has made the following contributions to knowledge:

1. We developed methods for building heterogeneous regression ensembles for predicting train delays.

2. We developed two methods, MSM1 and MSM2, for selecting models to include in the ensemble, (based on accuracy, and accuracy and diversity,

respectively) for selecting the subset of models from a collection of models to achieve maximum accuracy with minimum size.

3. We adapted the Coincident Failure (CFD) diversity measure to work in the regression context.
4. We found that there is no relationship between ensemble accuracy and diversity as measured by currently existing diversity measures.
5. We used deep learning methods as base learners in heterogeneous regression ensembles.

8.2 Suggestions for Future Work

We developed our methods using a dataset where we had removed any entries that were incomplete. For the future it would be beneficial to be able to handle incomplete data. Therefore methods for imputing missing data into a train delay dataset should be investigated.

We used two methods when combining models: averaging and weighted averaging. The weighted averaging was linear. We did perform some initial investigations using the softmax function and the beta function. (Results not shown.) However, we did not find that either was helpful. The softmax function does not differentiate between the weights and gave similar results to averaging. The beta function gives very high priority to the best performing models and severely penalises the lower performing models which meant that all their input was lost and hence negated the benefit of having an ensemble. The main focus of our work was on building the ensembles, so we did not carry out extensive investigations into methods for combining the model outputs. However, optimising the combining function would be beneficial and therefore for future work it would be worth investigating different ways of adjusting the weighting. For example weighting according to the square of the accuracy in order to bias it strongly toward the more accurate models.

We investigated the use of accuracy and diversity as measures when selecting, but other measures could be used, for example combining the accuracy and diversity into a new measure by multiplying them together. In addition, the methods we used were ordering-based, and other types of method could be investigated, such as cluster-based or optimization-based.

We found no relationship between ensemble diversity and accuracy. However, this may be due to the limitations of the existing diversity measures. It would be very beneficial to have a diversity measure that truly reflects the diversity of a regression ensemble. Therefore research should be conducted to develop diversity measures for the regression context.

We primarily investigated the use of machine learning (ML) algorithms for the base learners in our ensembles. However, our work using deep learning (DL) models showed that when an ensemble was built with them, it was more accurate than the ensembles built with the ML models. Since our ensembles can be built using any type of regression model as a base learner, it would be interesting to investigate the use of both ML and DL models in the same ensemble, to see if ensembles build with both types of model give even higher accuracies.

Having additional data relevant to train delays could aid in making more accurate delay predictions. Therefore it would be beneficial to have additional data such as passenger numbers and train type.

9 Bibliography

Abdelghany, K. F., Shah, S. S., Raina, S., and Abdelghany, A. F. (2004). A model for projecting flight delays during irregular operation conditions. *Journal of Air Transport Management*, 10(6):385–394.

Al Ghamdi, M., Parr, G., and Wang, W. (2020). Weighted ensemble methods for predicting train delays. In *International Conference on Computational Science and Its Applications*, pages 586–600. Springer.

Alyahyan, S., Farrash, M., and Wang, W. (2016). Heterogeneous ensemble for imaginary scene classification. In *KDIR*, pages 197–204.

Arik, S. Ö. and Pfister, T. (2021). Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6679–6687.

Asselman, A., Khaldi, M., and Aammou, S. (2021). Enhancing the prediction of student performance based on the machine learning xgboost algorithm. *Interactive Learning Environments*, pages 1–20.

Aytuğ, O. (2018). Particle swarm optimization based stacking method with an application to text classification. *Academic Platform-Journal of Engineering and Science*, 6(2):134–141.

Balakrishna, P., Ganesan, R., Sherry, L., and Levy, B. S. (2008). Estimating taxi-out times with a reinforcement learning algorithm. In *2008 IEEE/AIAA 27th Digital Avionics Systems Conference*, pages 3–D. IEEE.

Bian, S. (2006). *Data Mining Ensemble Hierarchy, Diversity and Accuracy.* PhD thesis, University of East Anglia.

Bian, S. and Wang, W. (2007). On diversity and accuracy of homogeneous and heterogeneous ensembles. *International Journal of Hybrid Intelligent Systems*, 4(2):103–128.

Box, G. E. and Tiao, G. C. (2011). *Bayesian inference in statistical analysis*, volume 40. John Wiley & Sons.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Brown, G., Wyatt, J., Harris, R., and Yao, X. (2005). Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

Carey, M. and Kwieciński, A. (1994). Stochastic approximation to the effects of headways on knock-on delays of trains. *Transportation Research Part B: Methodological*, 28(4):251–267.

Carvalho, L., Sternberg, A., Maia Gonçalves, L., Beatriz Cruz, A., Soares, J. A., Brandão, D., Carvalho, D., and Ogasawara, E. (2020). On the relevance of data science for flight delay research: a systematic review. *Transport Reviews*, pages 1–30.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Choi, S., Kim, Y. J., Briceno, S., and Mavris, D. (2016). Prediction of weather-induced airline delays based on machine learning algorithms. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–6. IEEE.

Corman, F. and Kecman, P. (2018). Stochastic prediction of train delays in real-time using bayesian networks. *Transportation Research Part C: Emerging Technologies*, 95:599–615.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30.

Department for Transport (2020). Transport statistics great britain 2020. `https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/945829/tsgb-2020.pdf`. Accessed: 2022-09-131.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Ding, L., Fang, W., Luo, H., Love, P. E., Zhong, B., and Ouyang, X. (2018). A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory. *Automation in construction*, 86:118–124.

Dutta, H. (2009). Measuring diversity in regression ensembles. In *IICAI*, volume 9, page 17p. Citeseer.

European Environment Agency (2020). Train or plane. https://www.eea.europa.eu/publications/transport-and-environment-report-2020. Accessed: 2022-09-23.

Freund, Y., Schapire, R., and Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Gao, B., Ou, D., Dong, D., and Wu, Y. (2020). Predictive modelling of running and dwell times in railway traffic. *International Journal of Software Engineering and Knowledge Engineering*, 30(7):921–940. Special Issue: Selected Papers from 4th Int. Conf. on Electrical Engineering and Information Technologies for Rail Transportation (EITRT2019).

Gashler, M., Giraud-Carrier, C., and Martinez, T. (2008). Decision tree ensemble: Small heterogeneous is better than large homogeneous. In *2008 Seventh International Conference on Machine Learning and Applications*, pages 900–905. IEEE.

González, S., García, S., Del Ser, J., Rokach, L., and Herrera, F. (2020). A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. *Information Fusion*, 64:205–237.

Gui, G., Liu, F., Sun, J., Yang, J., Zhou, Z., and Zhao, D. (2019). Flight delay prediction based on aviation big data and machine learning. *IEEE Transactions on Vehicular Technology*, 69(1):140–150.

Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Huang, G.-B., Zhu, Q.-Y., and Siew, C.-K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1):489–501. Neural Networks.

Huang, P., Wen, C., Fu, L., Peng, Q., and Tang, Y. (2020). A deep learning approach for multi-attribute data: A study of train delay prediction in railway systems. *Information Sciences*, 516:234–253.

Hunter, G., Boisvert, B., and Ramamoorthy, K. (2007). Advanced national airspace traffic flow management simulation experiments and vlidation. In *2007 Winter Simulation Conference*, pages 1261–1267. IEEE.

Iman, R. L. and Davenport, J. M. (1980). Approximations of the critical region of the fbietkan statistic. *Communications in Statistics-Theory and Methods*, 9(6):571–595.

Jain, P., Kakade, S. M., Kidambi, R., Netrapalli, P., and Sidford, A. (2018). Accelerating stochastic gradient descent for least squares regression. In *Conference On Learning Theory*, pages 545–604. PMLR.

Kecman, P. and Goverde, R. (2015). Predictive modelling of running and dwell times in railway traffic. *Public Transport*, 7(3):295–319. harvest.

Khanmohammadi, S., Chou, C.-A., Lewis, H. W., and Elias, D. (2014). A systems approach for scheduling aircraft landings in jfk airport. In *2014 IEEE*

International Conference on Fuzzy Systems (FUZZ-IEEE), pages 1578–1585. IEEE.

Krogh, A. and Vedelsby, J. (1994). Neural network ensembles, cross validation, and active learning. *Advances in neural information processing systems*, 7.

Lee, W.-H., Yen, L.-H., and Chou, C.-M. (2016). A delay root cause discovery and timetable adjustment model for enhancing the punctuality of railway services. *Transportation Research Part C: Emerging Technologies*, 73:49–64.

Lessan, J., Fu, L., and Wen, C. (2019). A hybrid bayesian network model for predicting delays in train operations. *Computers & Industrial Engineering*, 127:1214–1222.

Li, H.-C., Deng, Z.-Y., and Chiang, H.-H. (2020a). Lightweight and resource-constrained learning network for face recognition with performance optimization. *Sensors*, 20(21):6114.

Li, M., Fu, X., and Li, D. (2020b). Diabetes prediction based on xgboost algorithm. In *IOP conference series: materials science and engineering*, volume 768, page 072093. IOP Publishing.

Li, Z., Huang, P., Wen, C., Jiang, X., and Rodrigues, F. (2022). Prediction of train arrival delays considering route conflicts at multi-line stations. *Transportation Research Part C: Emerging Technologies*, 138:103606.

Liu, W., Chen, Z., and Hu, Y. (2022). Xgboost algorithm-based prediction of safety assessment for pipelines. *International Journal of Pressure Vessels and Piping*, 197:104655.

Liu, Y., Yao, X., and Higuchi, T. (2000). Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation*, 4(4):380–387.

Lu, Z., Wu, X., Zhu, X., and Bongard, J. (2010). Ensemble pruning via individual contribution ordering. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 871–880.

Marković, N., Milinković, S., Tikhonov, K. S., and Schonfeld, P. (2015). Analyzing passenger train arrival delays with support vector regression. *Transportation Research Part C: Emerging Technologies*, 56:251–262.

Martínez-Muñoz, G., Hernández-Lobato, D., and Suárez, A. (2008). An analysis of ensemble pruning techniques based on ordered aggregation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):245–259.

Mohammed, A. M., Onieva, E., and Woźniak, M. (2022). Selective ensemble of classifiers trained on selective samples. *Neurocomputing*, 482:197–211.

Murray, A. (2001). *Off the Rails: Britain's Great Rail Crisis: Cause, Consequences and Cure*. Verso.

Nabian, M. A., Alemazkoor, N., and Meidani, H. (2019). Predicting near-term train schedule performance and delay using bi-level random forests. *Transportation Research Record*, 2673(5):564–573.

Nair, R., Hoang, T. L., Laumanns, M., Chen, B., Cogill, R., Szabó, J., and Walter, T. (2019). An ensemble prediction model for train delays. *Transportation Research Part C: Emerging Technologies*, 104:196–209.

Nanglia, S., Ahmad, M., Khan, F. A., and Jhanjhi, N. (2022). An enhanced predictive heterogeneous ensemble model for breast cancer prediction. *Biomedical Signal Processing and Control*, 72:103279.

National Audit office (2008). Reducing passenger rail delays by better management of incidents.

Network Rail (2017). How we keep you updated when there's a delay.

Network Rail (2022a). How we keep you updated when there's a delay. Accessed: 2022-07-11.

Network Rail (2022b). Railway performance. `www.networkrail.co.uk/who-we-are/how-we-work/performance/railway-performance/`. Accessed: 2022-09-01.

NRE (2018). Darwin data feeds. Available at: http://www.nationalrail.co.uk/100296.aspx. Accessed 10/2019].

Office of Rail and Road (2019). 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 05/04/2019.

Oneto, L., Fumeo, E., Clerico, G., Canepa, R., Papa, F., Dambra, C., Mazzino, N., and Anguita, D. (2016). Advanced analytics for train delay prediction systems by including exogenous weather data. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 458–467. IEEE.

Oneto, L., Fumeo, E., Clerico, G., Canepa, R., Papa, F., Dambra, C., Mazzino, N., and Anguita, D. (2018). Train delay prediction systems: a big data analytics perspective. *Big data research*, 11:54–64.

OpenRailData (2019). Train movements.

Palaz, D., Magimai-Doss, M., and Collobert, R. (2019). End-to-end acoustic modeling using convolutional neural networks for hmm-based automatic speech recognition. *Speech Communication*, 108:15–32.

Partridge, D. and Krzanowski, W. (1997). Software diversity: practical statistics for its measurement and exploitation. *Information and software technology*, 39(10):707–717.

PN, T. (2006). M., steinbach, and v. kumar. introduction to data mining.

Rebollo, J. J. and Balakrishnan, H. (2014). Characterization and prediction of air traffic delays. *Transportation research part C: Emerging technologies*, 44:231–241.

Rosenblatt, F. (1961). Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Lab Inc Buffalo NY.

Santosa, F. and Symes, W. W. (1986). Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1307–1330.

Schaefer, L. and Millner, D. (2001). Flight delay propagation analysis with the detailed policy assessment tool. In *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat. No. 01CH37236)*, volume 2, pages 1299–1303. IEEE.

Schapire, R. E. (1990). The strength of weak learnability. *Machine learning*, 5(2):197–227.

Shi, R., Xu, X., Li, J., and Li, Y. (2021). Prediction and analysis of train arrival delay based on xgboost and bayesian optimization. *Applied Soft Computing*, 109:107538.

Shoman, M., Aboah, A., and Adu-Gyamfi, Y. (2020). Deep learning framework for predicting bus delays on multiple routes using heterogenous datasets. *Journal of Big Data Analytics in Transportation*, 2(3):275–290.

Shwartz-Ziv, R. and Armon, A. (2022). Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90.

Skurichina, M. and Duin, R. P. (2002). Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis & Applications*, 5(2):121–135.

Smętek, M. and Trawiński, B. (2011). Selection of heterogeneous fuzzy model ensembles using self-adaptive genetic algorithms. *New Generation Computing*, 29(3):309.

Soomer, M. J. and Franx, G. J. (2008). Scheduling aircraft landings using airlines' preferences. *European Journal of Operational Research*, 190(1):277–291.

Spanninger, T., Trivella, A., Büchel, B., and Corman, F. (2022). A review of train delay prediction approaches. *Journal of Rail Transport Planning & Management*, 22:100312.

Sternberg, A., de Abreu Soares, J., Carvalho, D., and Ogasawara, E. S. (2017). A review on flight delay prediction. *CoRR*, abs/1703.06118.

Tan, P.-N., Steinbach, M., and Kumar, V. (2016). *Introduction to data mining*. Pearson Education India.

Thompson, B. L. (2018). Predicting train delay.

Truong, D. (2021). Using causal machine learning for predicting the risk of flight delays in air transportation. *Journal of Air Transport Management*, 91:101993.

Tsoumakas, G., Partalas, I., and Vlahavas, I. (2008). A taxonomy and short review of ensemble selection. In *Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications*, pages 1–6.

Tu, Y., Ball, M. O., and Jank, W. S. (2008). Estimating flight departure delay distributions—a statistical approach with long-term trend and short-term pattern. *Journal of the American Statistical Association*, 103(481):112–125.

Wang, H., Fan, W., Yu, P. S., and Han, J. (2003). Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 226–235. AcM.

Wang, W. (2008). Some fundamental issues in ensemble methods. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 2243–2250. IEEE.

Wen, C., Lessan, J., Fu, L., Huang, P., and Jiang, C. (2017). Data-driven models for predicting delay recovery in high-speed rail. In *2017 4th International Conference on Transportation Information and Safety (ICTIS)*, pages 144–151.

Wu, Y., Liu, L., Xie, Z., Chow, K.-H., and Wei, W. (2021). Boosting ensemble accuracy by revisiting ensemble diversity metrics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16469–16477.

Yaghini, M., Khoshraftar, M. M., and Seyedabadi, M. (2013). Railway passenger train delay prediction via neural network model. *Journal of advanced transportation*, 47(3):355–368.

Yu, B., Yang, Z.-Z., Chen, K., and Yu, B. (2010). Hybrid model for prediction of bus arrival times at next station. *Journal of Advanced Transportation*, 44(3):193–204.

Yuan, J. and Hansen, I. A. (2007). Optimizing capacity utilization of stations by estimating knock-on train delays. *Transportation Research Part B: Methodological*, 41(2):202–217.

Zhang, D., Peng, Y., Zhang, Y., Wu, D., Wang, H., and Zhang, H. (2021a). Train time delay prediction for high-speed train dispatching based on spatio-temporal graph convolutional network. *IEEE Transactions on Intelligent Transportation Systems*, 23(3):2434–2444.

Zhang, X., Yan, M., Xie, B., Yang, H., and Ma, H. (2021b). An automatic real-time bus schedule redesign method based on bus arrival time prediction. *Advanced Engineering Informatics*, 48:101295.

Zhou, Z.-H. (2012). *Ensemble methods: foundations and algorithms*. CRC press.

Zonglei, L., Jiandong, W., and Guansheng, Z. (2008). A new method to alarm large scale of flights delay based on machine learning. In *2008 International Symposium on Knowledge Acquisition and Modeling*, pages 589–592. IEEE.