The Aeronautical Journal (2023), **127**, pp. 2082–2104
doi:10.1017/aer.2023.103

REGULAR PAPER

Ensemble approaches for leveraging machine learning models in load estimation

C. Cheung[1], E. Seabrook[1], J.J. Valdés[2], Z.A. Hamaimou[1] and C. Biondic[1]

[1]Aerospace Research Centre, National Research Council Canada, 1200 Montreal Rd, Ottawa, ON K1A 0R6, Canada and
[2]Digital Technologies Research Centre, National Research Council Canada, 1200 Montreal Rd, Ottawa, ON K1A 0R6, Canada
Corresponding author: C. Cheung; Email: catherine.cheung@nrc-cnrc.gc.ca

Received: 21 May 2023; **Revised:** 21 September 2023; **Accepted:** 6 October 2023

Keywords: load estimation; health and usage monitoring; integrated vehicle health management; ensembles; machine learning

Abstract

Helicopter component load estimation can be achieved through a variety of machine learning techniques and algorithms. A range of ensemble integration techniques were investigated in order to leverage multiple machine learning models to estimate main rotor yoke loads from flight state and control system parameters. The techniques included simple averaging, weighted averaging and forward selection. Performance of the models was evaluated using four metrics: root mean squared error, correlation coefficient and the interquartile ranges of these two metrics. When compared, every ensemble outperformed the best individual model. The ensembles using forward selection achieved the best performance. The resulting output is more robust, more highly correlated and achieves lower error values as compared to the top individual models. While individual model outputs can vary significantly, confidence in their results can be greatly increased through the use of a diverse set of models and ensemble techniques.

Nomenclature

1D-CNN	one-dimensional convolutional neural network
AI	artificial intelligence
BPM	baseline prediction method
corr	correlation
extSAM	extended signal approximation method
FSCS	flight state and control system
HUMS	health and usage monitoring system
IQR	interquartile range
LSTM	long-short term memory recurrent network
MARS	multi-variate adaptive regression spline
ML	machine learning
NN-MLP	multi-layer perceptron neural network
NRC	National Research Council
RF	random forest
RMSE	root mean squared error
SAM	signal approximation method

A version of this paper first appeared at The Australian International Aerospace Congress 2021 (AIAC19).

1.0 Introduction

The popularity of leveraging machine learning (ML) and artificial intelligence (AI) solutions has dramatically increased in all applications, and the domain of aircraft loads and usage monitoring is no exception. Machine learning approaches have been used in various health and usage monitoring systems (HUMS) applications, such as regime recognition, load estimation and fault detection in bearings and gears. Certainly, there is tremendous potential for ML methods to be accurate and useful for these applications; however, in the rush to incorporate these solutions, the limitations of these methods are not always clearly expressed nor well understood. Furthermore, there is a growing number of machine learning model types and their countless variations that could be implemented in each application. As these solutions mature, there are a number of questions raised around certification, acceptance, building confidence and trust in ML solutions. Increasing understanding of the models and their limitations will help address the uncertainties around these challenging questions.

The topic of load estimation for helicopter component loads has been an active area of research for several decades. As aircraft platforms are being flown in expanded roles and pressures mount to postpone aircraft retirement or replacement, operators must closely monitor the usage of the aircraft and track component loads to support improved aircraft availability, reduced operating costs and improved capability. Load monitoring capability has many purposes, beyond potentially re-lifing parts or modifying maintenance intervals. Other important uses for load monitoring include verifying the aircraft usage spectrum and load spectrum, improved fatigue life estimation for more efficient structural life management, improved management of aircraft and/or fleet capability, and integration with other tools and analysis to support digital twin applications.

Direct measurement of helicopter component loads is uncommon due to the unwanted weight and cost of the sensor systems. Accurate load monitoring of aircraft component loads during flight is a challenge that has inspired the implementation of computational intelligence and machine learning techniques to replace the need for costly sensor systems. Using available flight data from existing aircraft instrumentation, approaches to better estimate aircraft component loads experienced in flight and improve component fatigue life tracking have been in development. The idea that using the sensors that are already in place for other purposes implies the implicit assumption that the information required for load prediction is somehow captured by them, which is not necessarily the case. While there are limitations to equip the aircraft with load specific sensors, as explained above, the expectation of a highly accurate prediction of the loads must be tempered. This situation emphasises the need of exploiting the available information as much as possible and justifies the efforts in that direction.

As described in Section 2.0, there are a wide range of approaches that could be applicable for this problem. Within each approach, there could be numerous solutions or models developed. How do we choose one model as the solution? Should we choose just one model? The ability to have a diverse set of models from which to draw should be quite valuable, particularly in the landscape of the numerous machine learning techniques that are currently available and that will continue to be developed in the future. Having a framework that accommodates more than one solution provides additional robustness and flexibility, not to mention achieving improved performance.

The authors have been investigating the use of a variety of machine learning models for estimating helicopter loads based on existing aircraft sensor data [1, 2]. The estimates of load and fatigue life have shown tremendous potential for accurate and consistent estimates for several helicopter platforms. Efforts have now shifted to examining approaches which leverage a wide range of machine learning models through the construction of appropriate ensemble models [3, 4]. It is well-known in the machine learning community that there is no "silver bullet" perfect solution for a given problem, which justifies the consideration and use of a diverse repertoire of individual models, since many models could be weak. Ensembles compensate the error distribution of the individual models and reduce the variance of the estimates.

The work described in this paper occurred in two phases. The first phase was aimed at investigating robustness and stability of model configurations across hyperparameter values and random seed, while implementing simple approaches to ensembles. In the second phase of this work, the goal was to expand the variety of machine learning algorithms and prediction methods used to generate a widely

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

encompassing, heterogeneous set of load prediction models and implement a more advanced ensemble approach.

In this paper, a range of ensemble techniques are described and implemented, from simple ensembles based on rank sum and rank product, to a more complex ensemble approach using a forward selection process. The development of the ensembles helps to highlight some of the challenges and limitations of machine learning models, and demonstrate the influence of certain aspects of implementation, such as metrics and optimisation considerations.

This paper is structured as follows: Section 2.0 reviews related work in load estimation, Section 3.0 details the methodology for load estimation, Section 4.0 describes the ensemble integration approaches, Section 5.0 covers the experimental settings and model configurations, Section 6.0 presents the results from individual models and the ensemble approaches, and Section 7.0 summarises the work and provides recommendations.

2.0 Related work

Polanco [5] carried out an early survey of load estimation-related work in order to calculate helicopter component fatigue damage. Three approaches for deducing the loads on dynamic components were identified, with the difference being the information used to estimate the loads. These three categories were: information from loads measured on fixed components, from flight parameter measurements, and flight parameter and fixed system load measurements. At that point, neural network models were already used broadly in load estimation efforts.

More recently, with the surge of development and interest in ML tools, two main categories of approaches have emerged for aircraft load estimation or aircraft load monitoring:

- machine learning models based on sensor data for load prediction; and
- machine learning models trained on simulated flight data for aircraft wear and maintenance.

The first and largest category of papers relates to systems that use sensor data to build machine learning models that then predict the loads on the aircraft over time. A few such as Oldersma [6], Qing [7], Mucha [8] and Gallimard [9], use physical monitoring sensors such as strain gauges and temperature sensors to determine the state of the loads and train the machine learning model. Others, such as Isom [10] and Qing [7], use external validation sensors like vibration, Piezoelectric sensors, or accelerometers to infer the state of the aircraft at a point in time. Sikorsky has implemented onboard their rotary-wing aircraft their Virtual Monitoring of Loads technology, a ML-based technology for load estimation relying on aircraft sensor measurements and their onboard real-time HUMS data [10, 11].

Another category of techniques involves using simulated flight data to determine wear and maintenance needs of an aircraft. For example, Owen [12] uses simulated helicopter landing data to determine the interactions between the aircraft and ship loads during takeoff and landing. Ling [13] used a Bayesian probabilistic model to analyse structural health monitoring data in conjunction with fatigue damage prognosis models.

Although many of these works were successfully able to build highly accurate machine learning models, most of them are not currently in use beyond the proof-of-concept stage. This is in part due to a lack of guidance on integrated systems using machine learning as described by Gallimard [9] and in part due to difficulties in the explainability of machine learning models. Clearly, the topic of load estimation is very active with a range of pursued approaches and possible solutions.

3.0 Methodolgy for load estimation

Research has been carried out to develop a methodology for load signal and fatigue life estimation using only data from standard flight state and control system parameters. The National Research Council's

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

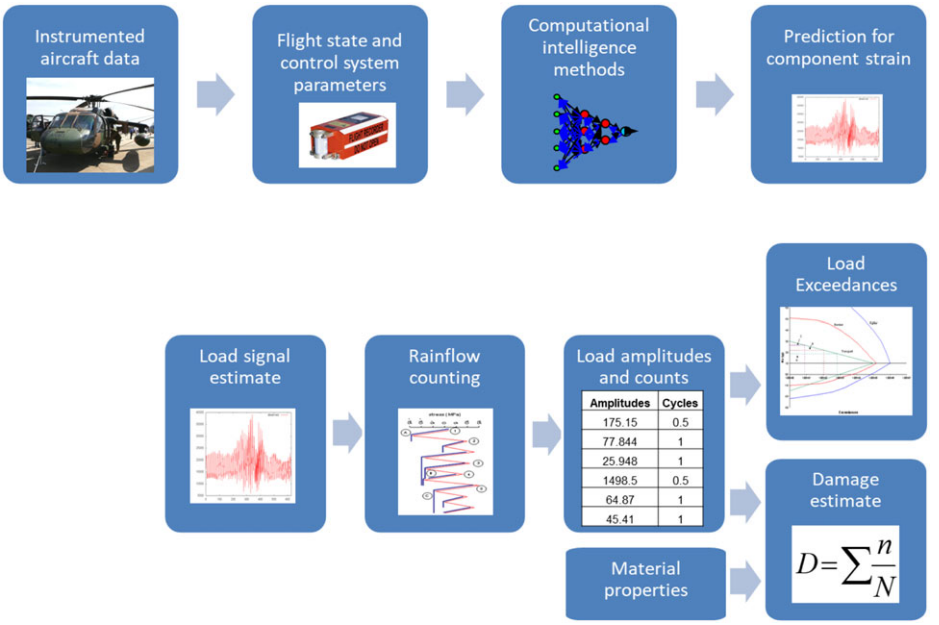

Figure 1. *Load estimation methodology.*

(NRC) approach to load and usage monitoring is centred on leveraging the data recorded by existing instrumentation using machine learning models and data mining techniques. In this work, a variety of machine learning models have been used to estimate main rotor yoke loads from 28 flight state and control system (FSCS) input parameters. The general methodology for load estimation is shown in Fig. 1. From the instrumented aircraft data, the FSCS parameters and sensor measurements were extracted as the input parameters and the target parameter, respectively. Machine learning techniques were then applied to train models which predict the component load signal from the FSCS parameters. Since both the inputs and more importantly the target output were continuous real-valued variables, the natural and most straight-forward problem formulation was as a regression problem.

From the load-time signal prediction, other structural parameters can be determined, such as load exceedances, fatigue damage, and fatigue life following a traditional fatigue damage calculation using cycle counting, such as Rainflow counting, and fatigue damage accumulation theory. Further details of this process are provided in References [2, 14].

The machine learning algorithm to train the model for load signal prediction could be any supervised regression algorithm, depicted in Fig. 2. The error function driving the training was the root mean squared error (RMSE), while the models were evaluated based on RMSE and the correlation between the observed target signal and the predicted signal. The formulae for the RMSE and correlation (*corr*) are presented below:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(y_{pred} - y_{true}\right)^2}{N}} \tag{1}$$

$$corr = \frac{\sum\left(y_{pred,i} - \overline{y_{pred}}\right)\left(y_{true,i} - \overline{y_{true}}\right)}{\sqrt{\sum\left(y_{pred,i} - \overline{y_{pred}}\right)^2 \sum\left(y_{true,i} - \overline{y_{true}}\right)^2}} \tag{2}$$

where y_{pred} is the predicted value, y_{true} is the true value, and N is the number of data points.

In this work, we have implemented a variety of ML algorithms including multi-variate adaptive regression splines (MARS) [15], random forest (RF) [16], long-short term memory (LSTM) recurrent

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

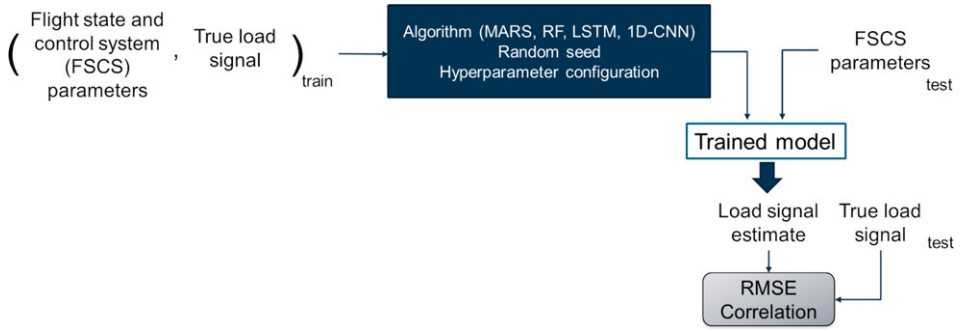

Figure 2. *Machine learning model training.*

networks [17], multi-layer perceptron neural networks (NN-MLP) [18] and 1-dimensional convolutional neural networks (1D-CNN) [19].

Several load estimation methodologies were presented in previous work [2] and include the NRC's baseline prediction method (BPM), NRC's signal approximation method (SAM), and NRC's extended signal approximation method (extSAM). The baseline prediction method is the most direct method of modelling the target load signal and estimating the component fatigue life from the flight state and control system parameters using one of the machine learning algorithms. The signal approximation method drastically improves the peak magnitude predictions and the resulting fatigue damage accumulation estimates, as compared to BPM predictions. A variation of SAM (extSAM, as defined earlier) combines the accurate prediction of signal phase and secondary peaks from the BPM with the signal characteristics from the SAM. These methods are described in more detail in [2]. These three load estimation methods are used with any of the machine learning algorithms (e.g. MARS, RF, NN-MLP, LSTM or 1D-CNN) to construct an individual load estimation model.

4.0 Ensembles

There are many models that can be developed for a particular problem. It may be logical that a single model with a unique spatial configuration for a given set of load data may not be the best model to estimate loads in other locations. However, it is not clear whether customising a singular type of model to the different loads will improve load estimation performance. It is commonly seen in other HUMS-related research that a single machine learning algorithm is selected and is the only one considered for solving the problem [20]. We were interested in keeping a wide variety of model types in our portfolio, especially given that there may be new models developed in the future that are worth exploring. One way to address this issue is to retain multiple high performing models in an ensemble and leverage all of these individual models [21].

The concept of merging predictions instead of selecting the single best is commonly accepted in statistics and has exhaustive theoretical background [22]. A significant reduction in error can emerge when combining multiple predictions through ensemble integration.

4.1 Rank sum and rank product for subset selection

In the first phase of this work, we looked at simple methods for creating ensembles of models. In order to select the subset of highest performing models to retain for the ensemble, rank sum and rank product algorithms were used.

Rank sum is a form of additive scoring, as it sums the score of the multiple different rankings considered. Rank product is a form of multiplicative scoring [23]. Rank sum can be calculated using Equation

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

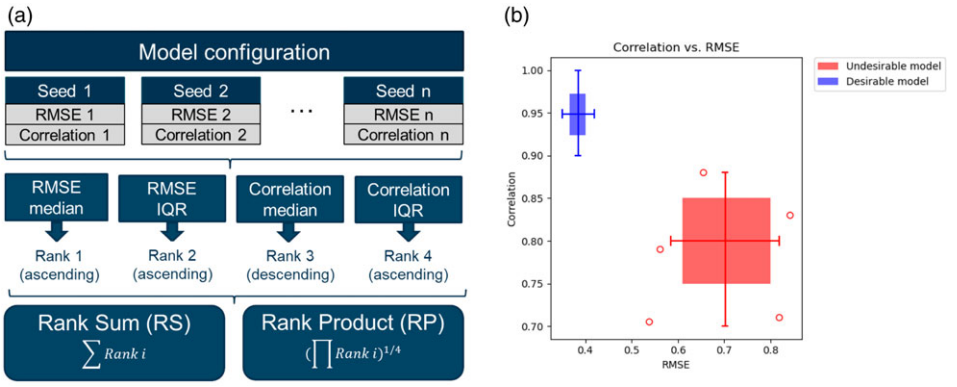

Figure 3. *Rank sum and rank product to select top performing models.*

(3), while rank product is expressed in Equation (4):

$$RS\,(op) = \sum_{i=1}^{k} r_{op,i} \tag{3}$$

$$RP\,(op) = \left(\prod_{i=1}^{k} r_{op,i} \right)^{1/k} \tag{4}$$

where op is the specific option/configuration, k is the number of rankings being considered ($k = 4$ in this work) and r indicates each of the different rankings for that specific option.

Rankings in four categories were used as shown in Fig. 3a: RMSE interquartile range (IQR), correlation IQR, RMSE median, and correlation median. The IQR corresponds to the range of the 25th to 75th percentile results from the distribution across the random seeds in the configuration for the Phase 1 work. While machine learning methods are non-deterministic methods that rely on some degree of randomness, achieving a certain level of consistency and reproducibility of results is important in a load estimation application. Therefore, models with a smaller IQR were sought in order to have consistent and robust model predictions. The more accurate and reliable models are those that have high correlation, low RMSE, and small IQRs in both metrics, located in the top-left corner of the plot in Fig. 3b. In this work, all four of the rankings, RMSE IQR, correlation IQR, RMSE median and correlation median, are important for a successful model. In order to simultaneously consider all of these requirements, two ranking methods, rank sum and rank product, were considered for comparing ranking in the various criteria.

4.2 Simple and weighted average ensemble

For a simple average ensemble, the predictions of all the top models are averaged for each data point in the test set. The RMSE and correlation are then calculated based on the new ensemble prediction values. As all the models in a simple ensemble are weighted equally, the load signal prediction can be described by Equation (5):

$$y_{ensemble} = \frac{\sum_1^n y_{op}}{n} \tag{5}$$

where $y_{ensemble}$ is the ensemble load signal prediction, y_{op} is the load signal prediction from the individual model, and n is the number of individual models in the subset.

A variation on the simple average ensemble is to use a weighted average. In this work, we follow on with the rank sum and rank product rankings to determine the weightings of each model in the ensemble.

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

For the weighted average ensembles, the weight of a configuration is the normalised inverse of the rank sum or rank product of the configuration. The weightings for the rank sum ensemble used Equation (6), while the rank product ensemble weightings followed Equation (7):

$$y_{ensemble} = \frac{\sum_1^n \frac{y_{op}}{RS_{op}}}{\sum_1^n \frac{1}{RS_{op}}} \tag{6}$$

$$y_{ensemble} = \frac{\sum_1^n \frac{y_{op}}{RP_{op}}}{\sum_1^n \frac{1}{RP_{op}}} \tag{7}$$

where RP_{op} is the rank product for that individual model, and RS_{op} is the rank sum for that individual model.

4.3 Forward selection

The second phase of this work looked to incorporate a more comprehensive ensemble approach. Several ensemble methodologies have been developed for supervised machine learning problems in various areas and applications involving classification and regression.

The forward selection algorithm is implemented in this work. Forward selection is a greedy search algorithm that attempts to find the globally optimal subset of models to include in the ensemble through incremental decisions that optimise an evaluation function locally at each step. Beginning from an empty ensemble, the algorithm selects the model that optimises the RMSE of the ensemble. Greedy search algorithms are prone to overfitting the dataset they are trained on because of the locality of the decision at each step. This motivates slight modifications to the forward selection algorithm to improve the ability of the ensemble to generalise [24].

Selection is performed with replacement allowing models to be added multiple times. This prevents models that reduce the ensemble's performance from being added and allows selection to fine-tune the ensemble weights: models added multiple times receive more weight. To further improve the generalisation capacity of the forward selection ensemble, the ensemble is initialised with the top five performing models on the validation set. This technique is known as sorted initialisation and is useful for small ensembles, where there is a tendency to overfit the validation set. The five top ranking models initially populate the ensemble before greedy stepwise selection begins. This forms a robust initial ensemble and limits the possibilities of overfitting during selection.

Figure 4 describes the general methodology for implementing ensembles in a regression problem, such as load estimation. The first step was to generate the set of base individual models from which the ensemble will be created. From this set, the selection of subsets of models is then carried out in the ensemble pruning stage. Finally, combining the individual models from the pruned subset is done in the ensemble integration step.

4.3.1 Ensemble generation

The first step of the ensemble methodology is generating a large array of individual models by varying one or more of the inductive algorithms, their hyperparameters, the input features and the training data. Regardless of the generation methods, the model pool needs to be as diverse as possible to optimise the ensemble's performance. A diverse set of models have uncorrelated errors that cancel each other through stochastic determinism [25].

In this work, a heterogeneous model pool was generated using five distinct machine learning algorithms (MARS, RF, LSTM, NN-MLP and 1D-CNN) each trained multiple times using different hyperparameter configurations. In addition, three different load estimation methods were used, in particular the baseline prediction method, signal approximation method and extended signal approximation method as described earlier in Section 3.0.

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

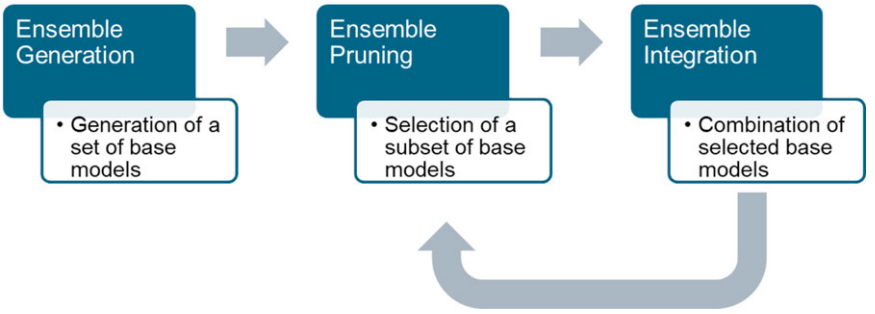

Figure 4. *General ensemble methodology.*

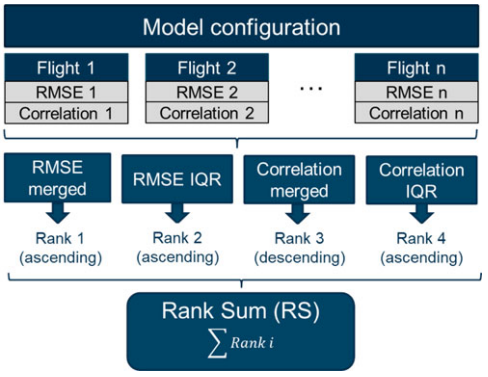

Figure 5. *Ranking individual models for forward selection.*

4.3.2 Ensemble pruning

The generated models vary significantly in performance. Common ensemble methodologies implement a pruning step to select a subset of high-performing models to retain for ensemble building. Pruning improves ensemble performance and reduces computational costs significantly. The chosen evaluation metrics must reflect the overall load estimation performance of a model and its local stability across different flights. Certainly, many metrics capture such information and further investigation is required to identify the optimal set or combination of metrics to consider.

To evaluate the accuracy of the load signal predictions, the RMSE and the correlation coefficient between the merged observed target signal and the predicted signal were calculated. The merged signal was created by concatenating all of the validation set flight data. To evaluate the stability of a model, the IQR of the RMSE and correlation distributions across flights was also calculated. In other words, the variation of the RMSE and correlation values across all flights in the validation set was used to generate the range for the IQR metric.

All four metrics except for the correlation coefficient were ranked in ascending order. The rank sum, the sum of the four ranks, is the evaluation metric for individual models, shown in Fig. 5. The rank sums do not decide the subset of models to include but influence the decision heavily, as explained in the following section.

4.3.3 Ensemble integration

Models are selected from a random subsample of the pool of models in a process known as bagging that reduces the chances of overfitting the ensemble to the validation set. Bagging is drawing a random sample of models from the model library to use for selection [16]. This process increases accuracy all

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

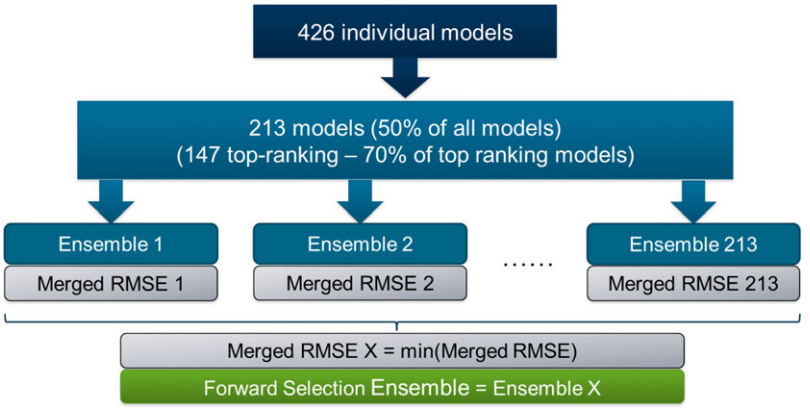

Figure 6. *Forward selection iteration.*

while reducing computational costs. Ordered bagging extends the idea further to ensure a fraction of the bag consists of high-performance models [26]. In this work, the selection occurs among a bag containing 50% of the model pool with at least 70% of the bag consisting of top-ranking models by rank sum, as shown in Fig. 6.

We started with the top five ranking models, then we added one model at a time to this ensemble. Because the bag of individual models is refreshed in each iteration, the same model can be included at multiple iterations. This tunes the weights of the final ensemble composition.

This process is repeated until the RMSE stops improving when a model is added, illustrated in Fig. 6. We deliberately chose RMSE as the criteria to minimise for the construction of the ensemble. Other metrics could be chosen instead, but RMSE makes sense for this effort since it was already one of the key metrics we were evaluating.

5.0 Experimental settings

5.1 Training, validation and testing sets

Flight data from a four-bladed twin-engine helicopter was available for this work. The helicopter had additional instrumentation installed for these flight trials covering critical components. The main rotor yoke beam bending was the target load. The recorded data from 28 flight state and control system parameters, which are part of standard helicopter instrumentation, were used as inputs for the models. The input FSCS parameters and target load were converted to z-scores, so that all variables had a mean of zero and standard deviation of one, according to the z-score transform, z_i in Equation (8):

$$z_i = \frac{x_i - \bar{x}}{\sigma} \tag{8}$$

where x_i is a variable with mean ($\bar{x}$) and standard deviation (σ).

The sampling rate used in this work for the load and 28 FSCS parameters was 50 Hz. While the FSCS parameters included in the input set correspond to flight data recorder parameters, this 50 Hz sampling rate was higher than the typical rates of a recording from the flight data recorder. More details about the flight data are provided in [2]. There were just under 6 million data points over 30 flight hours recorded from 57 flights available for this work. For Phase 1, 13 flights were used for training the models and 13 flights for testing as detailed in Table 1.

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

Table 1. *Summary of data used for training and testing for Phase 1*

Dataset	Number of data points	Number of flights
Training	907,472	13
Testing	1,395,616	13

Table 2. *Summary of data split for training, validation and testing for Phase 2*

Dataset	Number of data points	Number of flights	Proportion	Application
Training	1,418,461	19	24.5%	Training individual models
Validation	2,256,718	23	39.0%	Ensemble pruning: ranking models, forward selection
Testing	2,100,244	15	36.5%	Evaluation of models, ensembles
Total	**5,775,423**	**57**	**100%**	

The most common machine learning frameworks use three separate datasets:

1. The training set (or learning set) is used to train the individual models. The training process optimises the predictive performance of an algorithm according to its structure and hyperparameter configuration.
2. The validation set (or evaluation set) is needed to evaluate individual models and include or omit them from the ensemble according to their performance on unseen data. This data set is disjoint from the training set to avoid overfitting the ensemble, i.e. limiting its utility to the training set.
3. The testing set (or generalisation set) is withheld until the ensemble is developed. It is useful to evaluate the behaviour of the ensemble in unseen scenarios.

Table 2 summarises the size of each dataset as a percentage of the entire dataset and as an absolute number of flights or data points. The training set was used to train the individual models. The validation set was used for ensemble pruning – this was the data set for which the four metrics were calculated and evaluated that ranked the models. The validation set was also used for the forward selection process. The testing set was only used at the end once all the ensembles were created to evaluate the performance on unseen data.

5.2 Phase 1 – model configurations

In the first phase of this work on ensembles, there was a desire to understand how varying the hyperparameters and initialising random seeds would impact the variability and stability of the model results. Four machine learning algorithms were implemented to generate individual models for load estimation, in particular MARS, RF, LSTM and 1D-CNN. For each model type, multiple models were built with differing hyperparameters. Within each individual model, different random seeds were also tested. Table 3 lists the four model types and the hyperparameter settings that were explored in this phase of the work. In total, 108 model configurations were trained for this part of the work, and taking into account different initialising random seeds, 663 individual models were generated.

5.3 Phase 2 – model configurations

In the second phase of this work, the different load prediction methods (BPM, SAM, extSAM) described in Section 3.0 were explored with different hyperparameter settings. A greater variety of machine

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

Table 3. *Hyperparameter configurations for Phase 1 load estimation model types*

Model	No. of random seeds	No. of configurations	Hyperparameter	Values
MARS	5	12	Max terms	[10,20,30]
			Max degree	[1,2,3,4]
Random forest	6	10	Number of trees	[10,20,40,60,80,100, 150,200,300,400]
LSTM	6	72	Number of nodes	[5,10,15,20,25,30]
			Number of layers	[15,20,25,30,35,40]
			Activation function	[LeakyReLU, relu]
			Optimiser	Adam
			Loss function	Mean-squared error
1D-CNN	6	14	Number of nodes	[5,10,15,20,25,30,35]
			Number of layers	1
			Activation function	[LeakyReLU, relu]
			Optimiser	Adam
			Loss function	Mean-squared error

learning models were used in this phase of the work to estimate loads in the main rotor yoke from 28 flight state and control system parameters. A range of hyperparameter configurations for each of these model types were explored and are detailed in Table 4. In total, 426 individual models were trained for this phase of the work.

6.0 Results

6.1 Individual model results

6.1.1 Phase 1 – effect of varying hyperparameter values and initialising random seeds

In the first phase of this work, 108 model configurations were developed according to the parameter settings in Table 3. As a visualisation tool, 2D boxplots were created to simultaneously examine both metrics and highlight their differences for the various models, configurations and random seeds. Figure 7 shows the 2D boxplot corresponding to 5 of the 14 configurations of the 1D-CNN models. Each colour corresponds to a different configuration or option, meaning a different hyperparameter grouping, encompassing the results from all the random seeds for that option. The thicker lines in the boxes identify the median for both metrics across the various random seeds, the shaded area shows the IQR indicating 25th to 75th percentile results, the whiskers extending from each box correspond to 1.5 times IQR, and outliers not captured within the whiskers are plotted as outlined circles.

From the 2D boxplots, the differences in the models can be seen, in particular the wide range of results for different hyperparameter settings of the same model type, as illustrated by the location of the different coloured boxes. Some of the models have quite long whiskers and large IQR areas, meaning they have less stability in response to variation of the random seed.

From the 108 model configurations, Table 5 shows the top 30 in order according to overall rank sum along with rankings in the four categories: RMSE IQR, correlation IQR, RMSE median and correlation median. The rank product for these models is also provided. It is evident that between model 26 and 27, indicated by the thick line, there is a noticeable increase in the value of the rank sum and rank product,

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

Table 4. *Hyperparameter configurations for Phase 2 load estimation model types*

Model	No. of configurations	Hyperparameter	Values	Load estimation methods
MARS	7	Max terms	30, 50, 70, 90, 110, 130, 150	BPM, SAM, extSAM
Random forest	9	Number of trees	50, 100, 150	BPM, SAM,
		Random seed	32, 42, 52	extSAM
1D-CNN	36	Filter size	4, 5	BPM, SAM,
		Number of nodes	20, 30	extSAM
		Optimiser	Adam, SGD, RMSProp	
		Random seed	3, 4, 5	
NN-MLP	54	Number of layers	2, 4, 6	BPM, SAM,
		Number of nodes	16, 32, 64	extSAM
		Optimiser	Adam, SGD, RMSProp	
		Random seed	3, 4	
LSTM	36	Number of layers	2, 3	BPM, SAM,
		Number of nodes	10, 20	extSAM
		Optimiser	Adam, SGD, RMSProp	
		Random seed	3, 4, 5	

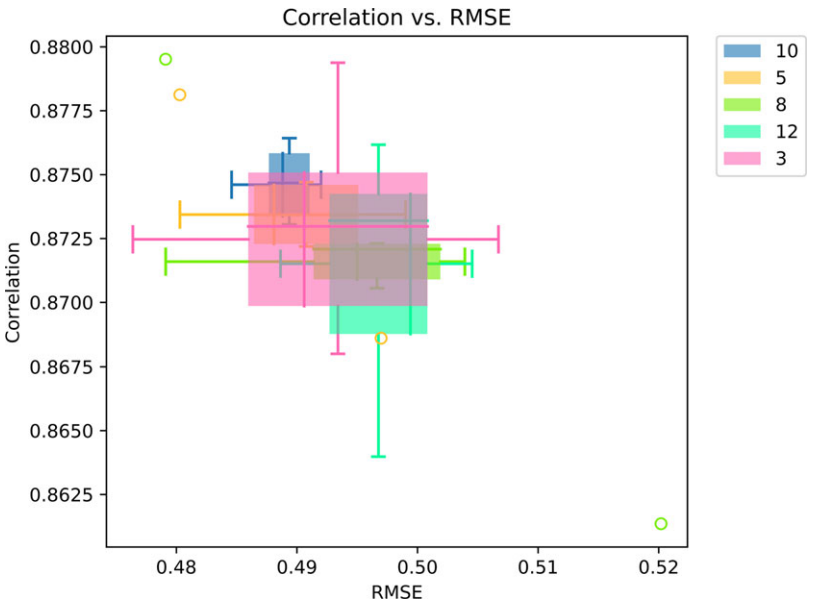

Figure 7. *Phase 1 – 2D boxplot of five 1D-CNN models showing RMSE and correlation.*

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

Table 5. *Phase 1 – Rank sum and rank product for top 30 models*

	RMSE				Correlation					
	median		IQR		median		IQR			
Model	value	rank	value	rank	value	rank	value	rank	Rank sum	Rank Product
4_MARS	0.505	14	0.000	1	0.865	14	0.000	1	30	3.74
10_1D-CNN	0.489	2	0.004	14	0.875	1	0.002	16	33	4.60
5_1D-CNN	0.488	1	0.010	17	0.873	3	0.002	15	36	5.26
0_MARS	0.539	19	0.000	2	0.853	17	0.000	2	40	6.00
8_MARS	0.546	20	0.000	3	0.845	21	0.000	3	47	7.84
8_1D-CNN	0.495	8	0.012	21	0.872	8	0.001	14	51	11.71
6_1D-CNN	0.493	5	0.011	20	0.872	9	0.005	18	52	11.28
3_1D-CNN	0.490	3	0.016	24	0.873	6	0.005	19	52	9.52
4_1D-CNN	0.494	6	0.017	25	0.873	5	0.004	17	53	10.63
12_1D-CNN	0.499	11	0.009	16	0.873	4	0.005	22	53	11.16
1_1D-CNN	0.492	4	0.011	19	0.873	7	0.006	23	53	10.52
9_RF	0.549	23	0.000	4	0.843	24	0.000	4	55	9.69
8_RF	0.549	22	0.000	9	0.843	22	0.000	10	63	14.45
2_1D-CNN	0.496	9	0.026	32	0.874	2	0.005	20	63	10.36
6_RF	0.550	25	0.000	8	0.842	25	0.000	6	64	13.16
11_1D-CNN	0.495	7	0.014	22	0.870	11	0.008	26	66	14.49
3_RF	0.552	28	0.000	5	0.841	29	0.000	5	67	11.94
5_RF	0.552	27	0.000	7	0.841	27	0.000	9	70	14.64
7_RF	0.549	24	0.000	13	0.843	23	0.000	13	73	17.48
0_RF	0.556	31	0.000	6	0.839	31	0.000	7	75	14.17
4_RF	0.551	26	0.000	12	0.842	26	0.000	12	76	17.66
13_1D-CNN	0.499	10	0.022	30	0.869	12	0.008	25	77	17.32
1_RF	0.554	30	0.000	10	0.840	30	0.000	8	78	16.38
7_1D-CNN	0.504	13	0.018	28	0.866	13	0.007	24	78	18.36
2_RF	0.552	29	0.000	11	0.841	28	0.000	11	79	17.70
9_1D-CNN	0.500	12	0.028	33	0.871	10	0.013	28	83	18.25
1_MARS	0.547	21	0.017	26	0.847	20	0.016	29	96	23.72
9_MARS	0.558	32	0.005	15	0.837	32	0.005	21	100	23.83
5_MARS	0.509	15	0.059	55	0.863	15	0.031	41	126	26.69
3_MARS	0.505	14	0.063	60	0.849	18	0.023	34	129	28.11

indicating a possible natural cut-off point for models to include. Certainly, other cut-off points could be considered. Therefore, in both cases, rank sum and rank product, 26 individual models were selected for a subset of top performing models, which provides some diversity in the individual models making up the ensemble in terms of configuration and algorithm.

From the results in Table 5, the rank sum and rank product scores seemed to favour essentially the same models with minor differences in ordering. The first 26 models contain the same models for rank product and rank sum, but not quite in the same order. It is evident that models with small IQR were prioritised, as designed, often above RMSE and correlation median values. While low variation is important, having all the top models selected based on IQR metrics and not the medians does not seem ideal. The 1D-CNN models performed extremely well with high correlation, low RMSE and low IQRs, and therefore scored well with both rank sum and rank product. All but one of the 1D-CNN configurations scored in the top 26. The MARS and RF models had higher RMSE and lower correlation, but their IQR were smaller, allowing them to score well through this method. None of the LSTM models that

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

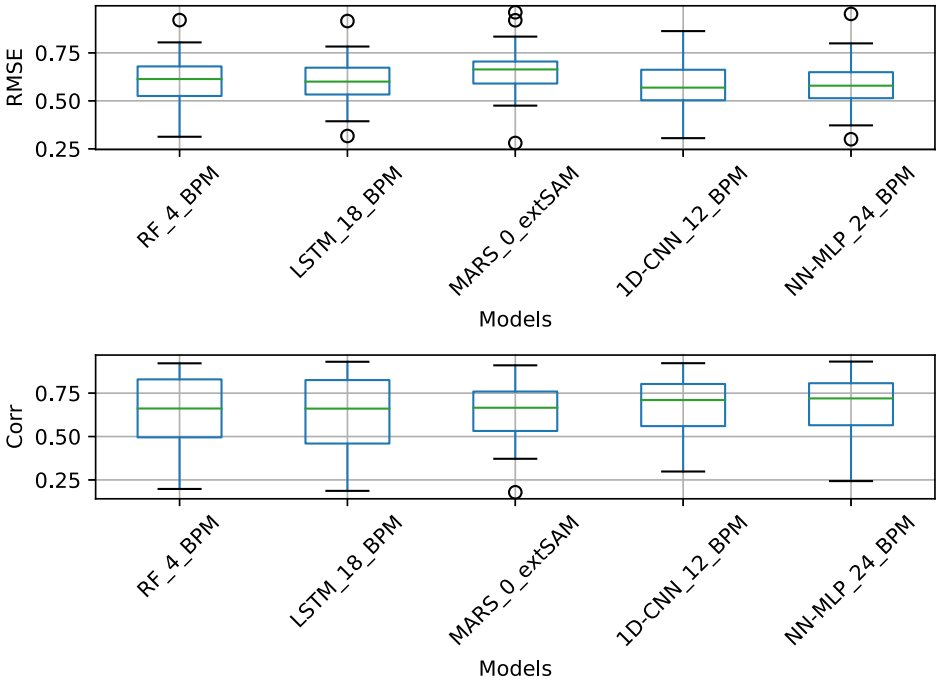

Figure 8. *Phase 2 – boxplots showing interquartile range of RMSE and correlation coefficient for 23 flights in validation set.*

were attempted performed well enough to appear in this list, tending to have higher RMSE and lower correlation values, so perhaps, a broader exploration of its hyperparameter values might be necessary.

There are some limitations with the ranking approach, as very small differences in RMSE or correlation values could lead to very different rankings if many models achieve similar performance. Likewise, if there are significant differences in RMSE and correlation but not many models in that range, the ranking could be deceptively similar. In this work, if multiple models achieved the identical values in any of the metrics, they were given the same ranking with the next model assigned the next ordinal ranking, known as dense ranking. Other strategies to resolve tied rankings were briefly explored, such as competition ranking to leave a gap in ranking numbers, but that approach resulted in excessive prioritisation of small IQR values. The equal weighting of the four categories was perhaps not ideal, so in future we would consider modifying the weighting of the categories or perhaps a multi-step process could be implemented. It would be prudent in this step to consider other metrics as well. Despite the limitations, the top individual models with the lowest RMSE and highest correlation were properly identified through this process.

6.1.2 Phase 2 – varying load estimation method and hyperparameter values

For Phase 2, a total of 426 individual models, as described in Table 2, were trained to estimate main rotor yoke loads from 28 flight state and control system parameters. Figure 8 shows a boxplot of the interquartile ranges of RMS error and correlation coefficient in the validation set of 23 flights for the top model of each machine learning algorithm. These metrics were selected to prioritise models that generated consistent predictions across a range of flights, instead of just an overall accuracy measure. The RMSE plots reveal more outlier flights, represented by black circles, than the correlation coefficient plots. This suggests that the RMSE is a more useful metric for detection of anomalous results. On the other hand, the correlation coefficient distributions have larger IQRs relative to the entire range than the

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

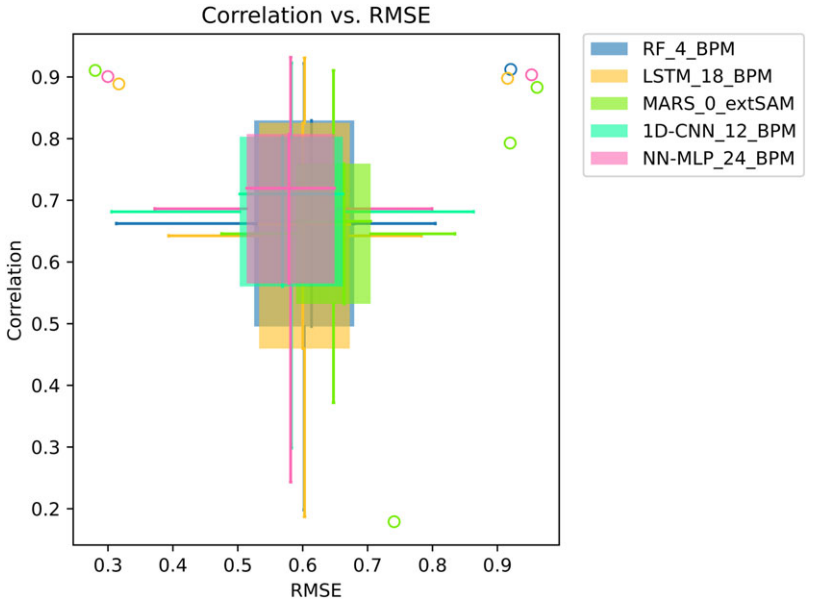

Figure 9. *Phase 2 – overall correlation vs overall RMSE of top models in the validation set.*

RMSE plots. This indicates that the correlation coefficient is a more sensitive metric to data variations making it useful for ranking models with similar RMSEs.

The models' RMSE and correlation coefficient distributions across 23 flights from the validation set are displayed in a 2D boxplot format in Fig. 9. The more accurate and reliable models are those that have high correlation, low RMSE and small IQRs in both metrics. The performances of the top models from each algorithm are very similar but not identical, as seen by the similar sized and positioned 2D-boxplots. This indicates that certain models will perform better than others for certain flights. This makes ensemble techniques that combine multiple models preferable over the common approach of selecting the best individual model.

The metrics of the top 25 ranked individual models are listed in Table 6. These metrics include the RMSE and correlation coefficient of the merged validation set, as well as, the interquartile ranges of the RMSE and correlation coefficient over the set of 23 individual flights in the validation set. The ranks for each metric and the overall rank sum are also provided. The list in Table 6 is dominated by neural network models using the baseline prediction method. Several 1D convolutional neural networks also appear in the list of top ranked models. The simple average ensemble and weighted average ensemble were based on these top 25 ranked models.

The NN-MLP and 1D-CNN models obtained the lowest RMSE values and highest correlation values. MARS models using extSAM load prediction method had the lowest RMSE IQR values, while NN-MLP models had the lowest correlation IQR values, as well as low RMSE IQR values. The BPM models tended to obtain the best scores for the ranked metrics. The models using the SAM load prediction method had both lower correlation and higher RMSE. The models using extSAM load prediction method had significantly larger IQR scores that disadvantaged the rank sums of certain models.

The scores for the testing data are provided in Table 9 and in fact were much improved over the validation set, with narrower IQR, higher correlation coefficients and lower RMSE values. From the testing scores, there were more 1D-CNN models that performed very well in contrast to the validation results, where almost all the high scoring models were NN-MLP.

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

Table 6. *Phase 2 – Validation scores of top-ranking individual models*

Model	RMSE	Corr	RMSE IQR	Corr IQR	RMSE rank	Corr rank	RMSE IQR rank	Corr IQR rank	Rank sum
NN-MLP_24_BPM	0.577	0.748	0.156	0.242	19	1	25	29	74
NN-MLP_28_BPM	0.578	0.740	0.163	0.225	27	14	41	6	88
NN-MLP_47_BPM	0.582	0.738	0.156	0.233	36	16	26	16	94
NN-MLP_11_BPM	0.574	0.741	0.166	0.251	10	7	52	40	109
NN-MLP_30_BPM	0.586	0.736	0.155	0.238	48	19	23	25	115
NN-MLP_23_BPM	0.574	0.740	0.179	0.234	12	10	98	19	139
NN-MLP_34_BPM	0.586	0.733	0.167	0.227	49	31	55	7	142
NN-MLP_49_BPM	0.584	0.735	0.160	0.254	44	25	34	42	145
NN-MLP_38_BPM	0.581	0.744	0.142	0.276	33	5	11	96	145
1D-CNN_12_BPM	0.570	0.745	0.182	0.243	2	3	111	30	146
NN-MLP_16_BPM	0.581	0.730	0.166	0.220	35	57	53	3	148
NN-MLP_46_BPM	0.587	0.732	0.164	0.227	55	41	46	8	150
NN-MLP_18_BPM	0.588	0.732	0.162	0.231	57	40	39	14	150
NN-MLP_42_BPM	0.577	0.740	0.145	0.283	21	12	12	109	154
1D-CNN_31_BPM	0.574	0.743	0.167	0.270	9	6	58	83	156
NN-MLP_52_BPM	0.571	0.740	0.187	0.231	3	11	132	15	161
NN-MLP_32_BPM	0.587	0.740	0.164	0.256	54	15	45	48	162
NN-MLP_39_BPM	0.603	0.726	0.129	0.233	87	71	1	17	176
NN-MLP_43_BPM	0.577	0.733	0.168	0.259	20	34	60	62	176
NN-MLP_7_BPM	0.584	0.735	0.161	0.260	45	24	37	71	177
1D-CNN_5_BPM	0.584	0.738	0.158	0.274	42	17	31	89	179
1D-CNN_13_BPM	0.570	0.745	0.182	0.267	1	4	109	77	191
NN-MLP_1_BPM	0.596	0.730	0.162	0.234	75	61	40	20	196
1D-CNN_23_BPM	0.581	0.730	0.168	0.256	32	54	61	50	197
NN-MLP_31_BPM	0.591	0.730	0.165	0.240	66	59	50	26	201

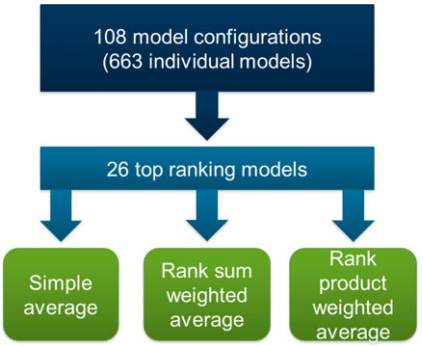

Figure 10. *Phase 1 ensemble construction.*

6.2 Ensemble results

6.2.1 Phase 1 – simple average and weighted average ensemble results

In this work, ensembles were constructed enabling a subset of models to be leveraged in the load estimation. We initially explored two straightforward methods: a simple average and weighted average based on the rank sum and rank product. From the 108 model configurations, 26 top ranking models were included in the subset from which the ensembles were created, as described in Fig. 10.

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

Table 7. *Simple average and weighted average results*

Ensemble model	RMSE	Relative RMSE	Correlation
Simple average	0.479	13.2%	0.879
Weighted average – rank sum	0.474	13.0%	0.881
Weighted average – rank product	0.472	13.0%	0.882

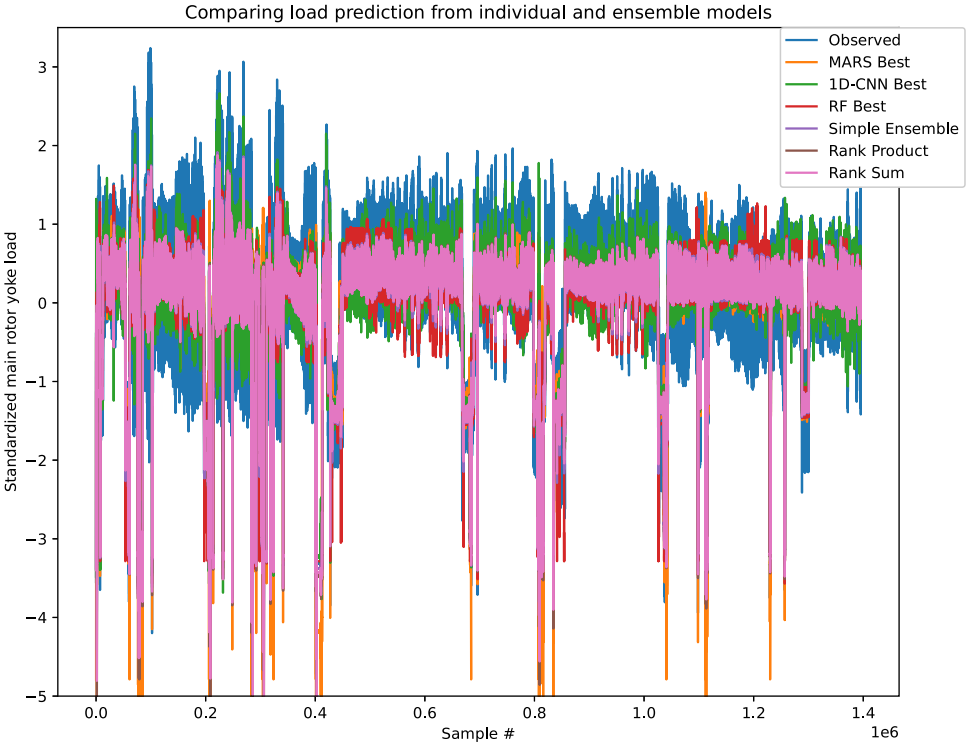

Figure 11. *Phase 1 load signal predictions for simple and weighted averages and several individual models.*

For a simple average ensemble, the predictions of all the top models are averaged for each data point in the test set. The RMSE and correlation are then calculated based on the new ensemble prediction values. For the ensemble, we used the individual model from the particular configuration with the random seed that yielded the lowest RMSE and highest correlation, as opposed to using the full set of five or six individual models with different random seeds. Since the rank sum and rank product subsets chose the same set of 26 models to include, the simple average ensemble is the same.

Table 7 shows the RMSE, relative RMSE, and correlation results for the simple average ensemble and the weighted average ensembles. The RMSE values (Equation (1)) are based on z-score values of the load variable (Equation (8)), and therefore have a theoretical range of 0 to infinity. A relative RMSE metric can be defined by the following equation (Equation (9)):

$$Relative\ RMSE = \frac{RMSE}{|max\ (y_{train})|} \tag{9}$$

where y_{train} is the true load signal used to train the models. This metric has its range in [0,1].

Figure 11 illustrates the load signal predictions for several individual models and the resulting simple average ensemble. The 1,395,616 data points cover 13 test flights merged together totaling just under 24 flight hours.

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

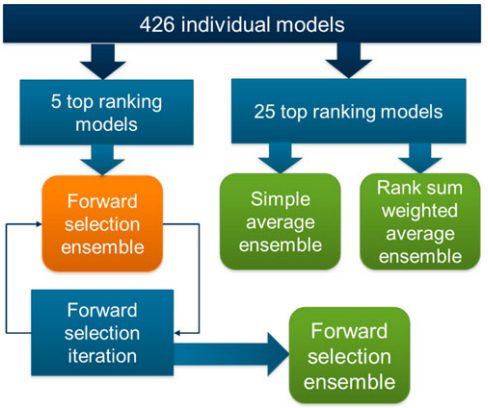

Figure 12. *Phase 2 ensemble construction.*

A number of observations were made based on these results. Given that the individual models had a range of RMSE values from 0.488 to 0.558 and a correlation coefficient range from 0.837 to 0.875, the ensembles performed very well. The simple average, the rank sum and rank product ensembles all resulted in lower RMSE values and higher correlation values than the best individual model. Overall, the rank product ensemble achieved the best performance based on RMSE and correlation metrics.

From Fig. 11, though, which plots the best individual models and the three ensemble predictions, it is evident that the models fall short with respect to the target observed signal amplitude, but they seem to be in phase and the gaps are also followed. These latter features show that the models are capturing relevant information, but improvements in the amplitude estimation of the load signal are still required. Although correlation and RMSE values between the ensembles and top individual models are similar, the top 1D-CNN model prediction seems to visually follow the observed signal more closely. The lower peaks are generally quite well estimated by all models, but the upper peak loads are underestimated. Because the predictions are averaged, the load signal often is smoother. However, if all individual models tend to underpredict peak values, the ensemble will similarly underpredict these peaks.

Likely the initial decision to include 26 top models could be revised to include fewer models and therefore remove some models that detract from the load signal prediction. With the results and the example provided in this work, it is evident that the most appropriate evaluation of the models may not be fully encapsulated in the RMSE and correlation metrics. If the focus of load estimation is cycle counting for damage estimation and load exceedance tracking, the synchronicity of the model with the target signal may not be as important as accurately capturing the peaks and valleys of the signal. In previous work, this observation led to consideration of other metrics related to the load exceedance curve in addition to RMSE and correlation of the load signal. Therefore, in the future, we plan to continue to explore other metrics that may lead to better overall models and therefore better ensembles in the end.

6.2.2 Phase 2 – forward selection ensemble results

The total number of models included in the ensemble influences model performance significantly. Therefore, identifying how many models to include is the first step of the forward selection ensemble process. The top five ranking individual models initially populated the ensemble before the forward selection began. At each step of the selection process, a bag of 213 models, 50% of the model pool, was evaluated to determine which model to select, as was shown in Fig. 6. At least 70% of the bag consisted of top-ranking models by rank sum to prioritise performance and stability. The ensemble construction process for Phase 2 is depicted in Fig. 12.

Figure 13 shows the progression of the ensemble development tracking the RMSE and correlation with each iteration. The ensemble performance metrics were calculated from the validation set samples

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

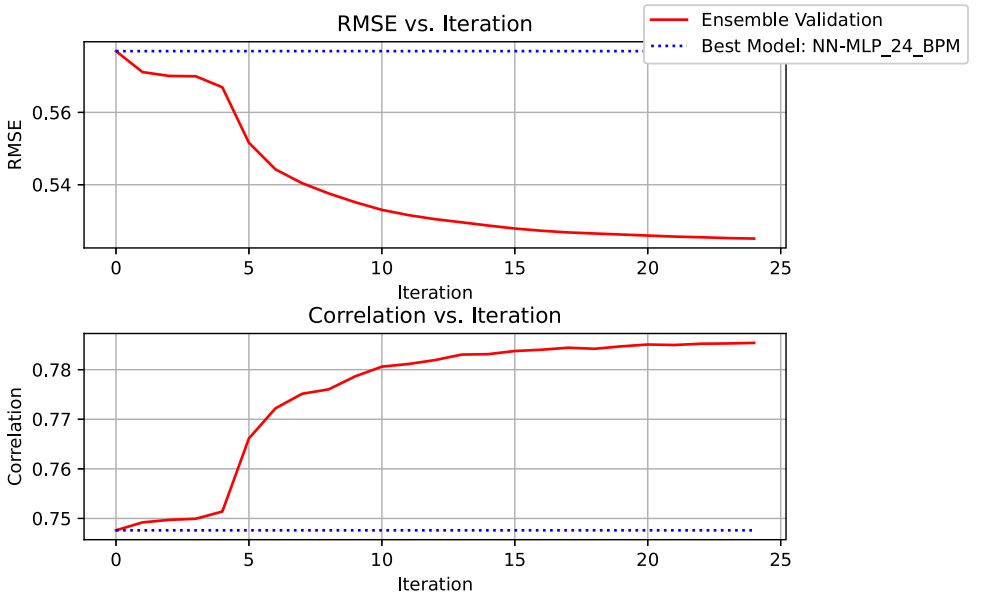

Figure 13. *Phase 2 – validation and testing progression of RMSE and correlation as models are added to the ensemble. The overall top model is indicated by the blue dotted line.*

(blue line) and from the testing set samples (red line) at each iteration; however, the forward selection process only relied on validation results while building the ensemble. The metrics of the top individual model are indicated by dashed lines as a reference point. The first five models added to the ensemble at the first iteration match the performance of the best individual model performance due to the initialisation process. The ensemble performance tends to improve significantly after the first five iterations as the ensemble selects models that keep improving its performance regardless of the model's individual performance.

By inspecting the forward selection validation progression from Fig. 13, the optimal number of models was determined to be 25 models corresponding to the point after which the metrics reached a plateau. The forward selection was then repeated using this number as the target number of models to select. The results of the forward selection process are detailed in Table 8.

Unlike in Phase 1 where the number of models was selected at a noticeable jump in the rank sum and rank product scores, in Phase 2, the decision was based on the performance of the ensemble evaluated with different numbers of models. For comparison, a simple and weighted average based on rank sum ensemble were created from the 25 models in Table 6. The composition of the forward selection ensemble was slightly different, as detailed in Table 8.

While a majority of the models chosen by the forward selection algorithm are NN-MLP and 1D-CNN models, several MARS models and an LSTM model were also included (Table 8). The composition of the 25 models in this ensemble is quite different from the rank sum models in Table 6, which were predominantly NN-MLP models using BPM. There is much more diversity in the forward selection models, including four of the five model types and both BPM and extSAM load prediction methods. There were several models that were picked more than once for the forward selection ensemble.

The testing set scores of these three ensembles and the top individual model for each machine learning type are detailed in Table 9. Figure 14 shows the load predictions from the three ensembles and the top individual model. The NN-MLP model listed was the top ranked individual model across the four metrics. It should be noted that the top RF model had better RMSE and correlation but was ranked lower because of IQR values.

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

Table 8. *Forward selection testing results*

Model	Count	Weight	RMSE	Corr	RMSE IQR	Corr IQR
NN-MLP_24_BPM	1	0.04	0.492	0.837	0.061	0.060
NN-MLP_28_BPM	1	0.04	0.469	0.850	0.075	0.063
NN-MLP_47_BPM	1	0.04	0.490	0.835	0.086	0.063
NN-MLP_11_BPM	1	0.04	0.511	0.824	0.098	0.077
NN-MLP_30_BPM	1	0.04	0.487	0.838	0.092	0.058
MARS_1_extSAM	1	0.04	0.530	0.817	0.068	0.082
NN-MLP_6_extSAM	2	0.08	0.478	0.787	0.103	0.082
1D-CNN_25_BPM	4	0.16	0.517	0.824	0.060	0.054
MARS_0_extSAM	2	0.08	0.529	0.818	0.057	0.085
NN-MLP_13_extSAM	1	0.04	0.573	0.789	0.123	0.080
NN-MLP_16_extSAM	1	0.04	0.714	0.685	0.167	0.112
1D-CNN_19_BPM	1	0.04	0.504	0.828	0.081	0.083
MARS_6_extSAM	1	0.04	0.530	0.817	0.068	0.082
1D-CNN_16_BPM	2	0.08	0.497	0.839	0.083	0.072
NN-MLP_25_BPM	1	0.04	0.492	0.839	0.100	0.057
NN-MLP_0_extSAM	1	0.04	0.561	0.800	0.092	0.080
NN-MLP_52_BPM	1	0.04	0.475	0.847	0.095	0.082
1D-CNN_12_extSAM	1	0.04	0.553	0.799	0.120	0.111
1D-CNN_14_extSAM	1	0.04	0.560	0.794	0.131	0.072
Ensemble	**25**	**1**	**0.449**	**0.864**	**0.069**	**0.059**

Table 9. *Phase 2 – Testing results for simple, weighted, forward selection ensemble*

Ensemble model	RMSE	Relative RMSE	Correlation	RMSE IQR	Correlation IQR
Top RF model	0.478	16.2%	0.846	0.101	0.086
Top LSTM model	0.488	16.6%	0.836	0.069	0.092
Top MARS model	0.529	18.0%	0.818	0.057	0.085
Top 1D-CNN model	0.488	16.6%	0.839	0.080	0.051
Top NN-MLP model and overall top individual model	0.492	16.7%	0.837	0.061	0.060
Simple average	0.464	15.8%	0.855	0.074	0.068
Weighted average – rank sum	0.460	15.6%	0.857	0.097	0.063
Forward selection	0.449	15.2%	0.864	0.069	0.059

The simple average ensemble performed better than the top ranked individual model, with lower RMSE and higher correlation. The weighted average based on rank sum improved slightly on the simple average ensemble. The forward selection ensemble obtains even more improvement in the scores than the other two ensembles. While the simple and weighted average ensembles selected models based on their performance across all metrics, the forward selection ensemble only optimised the RMS error. The three other evaluation metrics still saw improvements as a result of this optimisation. Based on the improvements obtained in these metrics, it is evident that the use of ensemble integration techniques will improve load estimates.

The RMSE and correlation coefficient IQRs estimate the expected variation in performance of a model across flights. It is expected that ensembles have lower IQRs because the inclusion of a diverse set of individual models reduces the bias overall. In other words, ensembles are theoretically more stable.

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

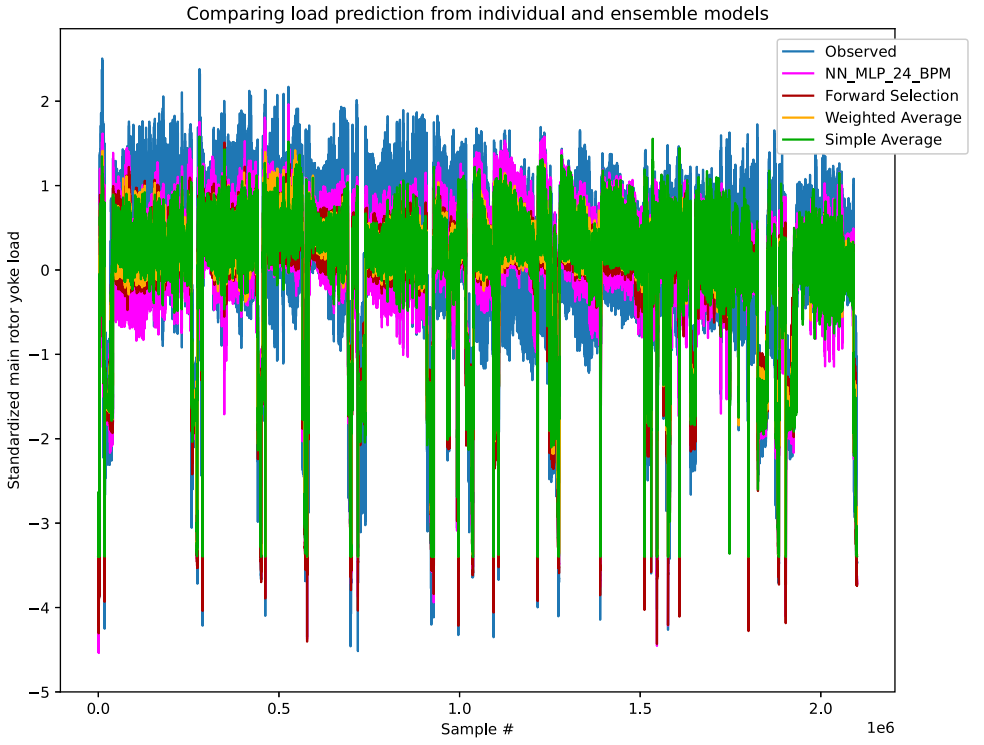

Figure 14. *Phase 2 – load signal estimates for the ensembles and the best individual model.*

However, the limited number of flights in the validation and testing sets make the IQRs insufficient to draw absolute conclusions about the stability of the ensembles. Nevertheless, relative to the individual models, all three ensembles achieve top-ranking IQR scores.

This work reinforces the idea that careful selection of metrics is crucial when using machine learning algorithms because these metrics drive the optimisation process. The selection of metrics impacts not only the training of individual models, but also the ensemble integration process with forward selection, as well as, ranking models for the rank sum weighted average and simple average ensembles. In addition, the selection of hyperparameter values and the range of their values considered also has an important impact and will limit the scope of models considered and sets boundaries for the pruning and integration stages of the ensemble process. It is important to note that the selection of these parameters is somewhat arbitrary.

In the construction of the simple and weighted average ensembles, the impact of equal weighting of the ranking metrics is apparent. This drawback was noted in Section 6.1.1, highlighting the importance of carefully selecting metrics and whether equal weighting was appropriate. In contrast, the forward selection ensemble process is more deliberate in selecting models that will improve the overall performance of the ensemble, whereas the simple and weighted average ensembles make the assumption that the top individual models will automatically form the best ensemble. For diversity reasons and including models that have different features, considering the ensemble as a whole is more valuable. The result is that the composition of the forward selection ensemble is quite different from the rank sum list, and greater improvement in the performance metrics, RMSE and correlation, is obtained.

The plots of the load signal predictions in Fig. 14 reflect the non-zero RMSE values and that the ensembles are not fully matching the target observed signal. It is worth noting that the set of sensors providing inputs to the models were not installed for the purpose of load estimation, therefore the efforts to achieve accurate predictions from these sensors requires using the machine learning models to extract

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

as much relevant information as possible. Underprediction of peak values has been a difficult challenge for machine learning-based load prediction, since the training data has a much higher proportion of off-peak data points than peak values. Since there is averaging involved in constructing the ensembles, if all the individual models tend to under predict peak values, the ensemble will similarly under predict these peaks. While the ensembles generated in this work provide improved load prediction models, there is still plenty of opportunity to improve on the results obtained.

7.0 Concluding remarks

In this work, a large pool of helicopter load estimation models was generated to estimate main rotor yoke loads from 28 flight state and control system parameters. It is important to note that these existing sensors were not originally installed for the purpose of estimating component loads, and attempts to obtain accurate predictions from these sensors requires extracting as much relevant information as possible using machine learning models. Five machine learning model types were included, in particular random forest, multi-variate adaptive regression splines, multi-perceptron neural networks, one-dimensional convolutional neural networks, and long-short term memory recurrent networks. Three load estimation methods were explored, namely NRC's baseline prediction method, signal approximation method, and extended signal approximation method. A wide variety of hyperparameter configurations and random seed values were investigated. Individual models were evaluated based on root mean squared error and correlation with the target signal, as well as the interquartile ranges of these metrics. Considerable variation in the model results was evident when changing the random seed and hyperparameter configuration. Subset selection based on rank sum and rank product were trialled, and then simple average or weighted average ensembles were constructed from the reduced set of models. There were some notable benefits to using a simple ensemble of individual models, in particular introducing diversity in the individual models making up the ensemble in terms of configuration and algorithm, and a smoother load signal prediction with higher correlation. While individual model outputs can vary significantly, the effects can be mitigated using a thoughtful approach to evaluating models and creating subsets and ensembles of models.

From the 426 individual models that were trained, 25 top models were selected based on the ranking metrics. A simple average ensemble, a weighted average ensemble based on rank sum, and a forward selection ensemble were generated. All ensembles achieved improved performance for these four metrics compared to the best individual models, with the forward selection ensemble obtaining the lowest RMSE, highest correlation and closest load signal prediction visually of all models. While the simple average and weighted average ensembles relied predominantly on NN-MLP models using BPM, there was much more diversity in the forward selection models, including three model types and both BPM and extSAM load prediction methods.

It is evident that there is opportunity for further improvement of the load estimation models to obtain better individual models and therefore better ensemble models. The forward selection approach of considering models based on overall ensemble performance was notable for its improved performance over other ensemble integration methods. The role of metrics in the optimisation process in training individual models and building ensembles was another important takeaway from this work.

Developing an approach for managing, selecting and leveraging a large number of machine learning models as well as other solutions is valuable, particularly in the rapidly advancing field of ML and AI. Future work will look at including other machine learning algorithms and exploring other objective functions, such as considering individual measures and combinations of other metrics, that could provide improved load estimates, particularly at the peaks. The effect of using lower sampling rates consistent with current flight data recorder rates could be investigated. Accordingly, the inclusion of other metrics, such as maximum error as well as mixtures of measures, is planned in future work to increase the confidence in the load estimates.

Acknowledgements. This work was funded in part by Defence Research and Development Canada.

https://doi.org/10.1017/aer.2023.103 Published online by Cambridge University Press

Competing interests. The authors declare none.

References

[1] Cheung C., Sehgal S. and Valdés J.J. A machine learning approach to load tracking and usage monitoring for legacy fleets. ICAF 2019 – Structural Integrity in the Age of Additive Manufacturing: Proceedings of the 30th Symposium of the International Committee on Aeronautical Fatigue, June 2–7, A. Niepokolczycki and J. Komorowski (eds), Krakow, Poland, Springer Cham, 2019, pp 922–937. doi: 10.1007/978-3-030-21503-3.

[2] Cheung C., Rocha B., Valdés J.J. and Puthuparampil J. Improved load estimation and fatigue life tracking demonstrated on multiple platforms using the signal approximation method. *American Helicopter Society 72nd Annual Forum Proceedings*, Fairfax, VA, 2016, pp 1–14.

[3] Cheung C., Biondic C., Hamaimou Z.A. and Valdés J.J. An approach to merging machine learning models in an ensemble for load estimation. *Proceedings of the 12th DST International Conference on Health and Usage Monitoring (HUMS2021)*, Defence Science and Technology Group, Melbourne, Australia, 2021, pp 1–8.

[4] Cheung C. and Hamaimou Z.A. Ensemble integration methods for load estimation. *Vertical Flight Society 78th Annual Forum Proceedings*, Fairfax, VA, 2022, pp 1–9.

[5] Polanco F.G. *Estimation of Structural Component Loads in Helicopters: A Review of Current Methodologies*. Technical Note DSTO-TN-0239. Melbourne, VIC: DSTO Aeronautical and Maritime Research Laboratory, 1999, pp 1–31.

[6] Oldersma A. and Bos M.J. Airframe loads and usage monitoring of the CH-47D "Chinook" helicopter of the Royal Netherlands Air Force. In *ICAF 2011 Structural Integrity: Influence of Efficiency and Green Imperatives* (J. Komorowski, ed.), pp 473–493. Dordrecht: Springer Netherlands, 2011.

[7] Qing X., Liao Y., Wang Y., Chen B., Zhang F. and Wang Y. Machine learning based quantitative damage monitoring of composite structure. *Int. J. Smart Nano Mater.*, 2022, **13**, (2), pp 167–202.

[8] Mucha W. Comparison of machine learning algorithms for structure state prediction in operational load monitoring. *Sensors*, 2020, **20**, (24), p 7087.

[9] Del cistia Gallimard C., Beroul F., Denoulet J., Nikolajevic K., Pinna A., Granado B. and Marsala C. Harmonic decomposition to estimate periodic signals using machine learning algorithms: Application to helicopter loads. *2022 International Joint Conference on Neural Networks (IJCNN)*, IEEE, Padua, Italy, pp 1–8, 2022.

[10] Isom J., Fletcher J., Davis M., Cycon J. and Rozak J. Flight test of technology for virtual monitoring of loads. *American Helicopter Society 69th Annual Forum Proceedings*, Phoenix, AZ, 2013, pp 1–20.

[11] Raymond B. Jr., Davis, M., Kloda, J. and Templeton, D. S-92 main rotor blade life extension. *American Helicopter Society 72nd Annual Forum Proceedings*, West Palm Beach, FL, USA, May 2016, pp 1–10.

[12] Owen I., White M.D., Padfield G.D. and Hodge S.J. A virtual engineering approach to the ship-helicopter dynamic interface – A decade of modelling and simulation research at the University of Liverpool. *Aeronaut. J.*, 2017, **121**, (1246), pp 1833–1857.

[13] Y. Ling and Mahadevan S. Integration of structural health monitoring and fatigue damage prognosis. *Mech. Syst. Signal Process.*, 2012, **28**, pp 89–104. Interdisciplinary and Integration Aspects in Structural Health Monitoring.

[14] Cheung C., Rocha B., Valdés J.J., Kotwicz-Herniczek M. and Stefani A. Expanded fatigue damage and load time signal estimation for dynamic helicopter components using computational intelligence techniques. *American Helicopter Society 70th Annual Forum Proceedings*, Montréal, Québec, Canada, May 2014, pp 1–10.

[15] Friedman J.H. Multivariate adaptive regression splines. *Ann. Statist.*, 1991, **19**, (1), pp 1–67.

[16] Breiman L. Random forests. *Mach. Learn.*, 2001, **45**, (1), pp 5–32. doi: 10.1023/A:1010933404324.

[17] Hochreiter S. and Schmidhuber J. Long short-term memory. *Neural Comput.*, 1997, **9**, pp 1735–1780.

[18] Pal S. and Mitra S. Multilayer perceptron, fuzzy sets, and classification. *IEEE Trans. Neural Netw.*, 1992, **3**, (5), pp 683–697.

[19] Lecun Y., Bottou L., Bengio Y. and Haffner P. Gradient-based learning applied to document recognition, in *Proceedings of the IEEE*, **86**, (11), pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[20] Dietterich T.G. Machine-learning research. *AI Mag.*, 1997, **18**, p 97.

[21] Webb G. and Zheng Z. Multistrategy ensemble learning: reducing error by combining ensemble learning techniques. *IEEE Trans. Knowl. Data Eng.*, 2004, **16**, (8), pp 980–991.

[22] Mendes-Moreira J.A., Soares C., Jorge A.M. and De Sousa J.F. Ensemble approaches for regression: A survey. *ACM Comput. Surv.*, 2012, **45**, (1), Article 10 (pages 1–40). doi: 10.1145/2379776.2379786.

[23] Breitling R., Armengaud P., Amtmann A. and Herzyk P. Rank products: A simple, yet powerful, new method to detect differentially regulated genes in replicated microarray experiments. *FEBS Lett.*, 2004, **573**, (1), pp 83–92.

[24] Tsoumakas G., Partalas I. and Vlahavas I. A taxonomy and short review of ensemble selection. *Proceedings ECAI'2008 Workshop on supervised and unsupervised ensemble methods and their applications (SUEMA'2008)*, 2008, pp 41–46.

[25] Dietterich T.G. Ensemble methods in machine learning. *Multiple Classifier Systems*, Berlin, Heidelberg, Springer Berlin Heidelberg, pp 1–15, 2000.

[26] Martnez-Muñoz G. and Suárez A. Pruning in ordered bagging ensembles. *Proceedings of the 23rd International Conference on Machine Learning, ICML '06*, New York, NY, USA, Association for Computing Machinery, pp 609–616, 2006.

Cite this article: Cheung C., Seabrook E., Valdés J.J., Hamaimou Z.A. and Biondic C. (2023). Ensemble approaches for leveraging machine learning models in load estimation. *The Aeronautical Journal*, **127**, 2082–2104. https://doi.org/10.1017/aer.2023.103