An approach to merging machine learning models in an ensemble for load estimation

Catherine Cheung[1], Calista Biondic[1], Zouhair Adam Hamaimou[1] and Julio J. Valdés[2]

[1] *Aerospace Research Centre, National Research Council Canada,*
1200 Montreal Rd, Ottawa, ON, Canada K1A 0R6

[2] *Digital Technologies Research Centre, National Research Council Canada,*
1200 Montreal Rd, Ottawa, ON, Canada K1A 0R6

Abstract

In this work, an approach to merging machine learning models into an ensemble is presented for helicopter load estimation. Several machine learning techniques were explored, including Random Forest, long-short term memory recurrent networks, multivariate adaptive regression splines, and 1-D convolutional neural networks. Considerable variation in the model results was evident when changing the random seed and hyperparameter configuration. Individual models were evaluated using key metrics and ranked, using rank sum and rank product, to obtain a subset of high performing models. An ensemble was constructed enabling a number of machine learning models to be leveraged in the load estimation. Two ensemble methods were attempted, a simple average and a weighted average based on rank sum and rank product. The resulting output is more robust, more highly correlated, and achieves similar or lower error values as compared to the top individual models. 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.

Keywords: machine learning, ensembles, load estimation, HUMS, IVHM

Introduction

The popularity of machine learning and artificial intelligence solutions has dramatically increased in all applications, including in the domain of Health and Usage Monitoring Systems (HUMS). Machine learning approaches have been tested in many HUMS applications, such as regime recognition and load estimation. While there is tremendous potential for machine learning methods to be accurate and useful for these applications, the limitations of these methods are not always clearly expressed nor well understood. Furthermore there is a growing number of machine learning models and their countless variations that could be implemented in each application. 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]. 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.

In this work, an approach to merging machine learning models into an ensemble is presented in the context of helicopter load estimation. Several machine learning techniques with varying hyper parameters and random seeds are explored. These techniques include Random Forest, long-short term memory recurrent networks, multivariate adaptive regression splines, and 1-dimensional convolutional neural networks. Rank sum and rank product approaches to selecting suitable subsets of models are outlined as well as two initial options for creating an ensemble from these subsets.

Machine learning methods

For regression problems, such as load estimation, there are a large number of machine learning methods that can be used. These methods range from linear and polynomial regression, to artificial neural networks, to regression trees and random forests, and so on. Without a doubt, there are other applicable approaches and algorithms that could be considered, and in the future there will certainly be additional ones that are conceived and developed. For each of these algorithms or model types, there are a number of hyperparameters or architecture settings that can be selected. In addition, there are other initialization values that are specified, such as the initial random seed, which are unique to the model that is created at that time.

In this work, we used a variety of machine learning models to estimate main rotor yoke loads from 28 flight state and control system input parameters, described further in Ref. 1. In particular, multivariate adaptive regression splines (MARS) [2], random forest (RF) [3], long-short term memory (LSTM) recurrent networks [4], and 1-dimensional convolutional neural networks (1D-CNN) [5, 6] were implemented for load signal estimation. To evaluate the accuracy of the load signal predictions, root mean squared error (RMSE) and the correlation coefficient between the observed target signal and the predicted signal were calculated. Other additional metrics could be and have been used in evaluating the performance of each method, however, in this work we selected these two metrics to focus on. Certainly, the selection of appropriate metrics is a key consideration in driving the performance of the models and is an area that the authors intend to continue to investigate further in future work.

For each model type, multiple models were built with differing hyperparameters. Within each individual model, different random seeds were also tested to glean insight into their impact on the variability and stability of the model results. Table 1 lists the four model types and the hyperparameter settings that were explored. The total number of individual models that were generated added up to 108 models, and over 600 unique models once different random seeds are considered. We then looked at how we could visualize and evaluate all of these models to appropriately choose the best models to use for load estimation.

Table 1: Hyperparameter configurations for 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]
			Optimizer	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]
			Optimizer	Adam
			Loss function	Mean-squared error

Subsets of models

It is clear that there are many models that can be developed for a particular problem. While it may be logical that a single model with its unique set of settings trained on data to estimate loads in one particular location may not be the best model to estimate loads in other locations, it is not clear if we should expect the same type of model to be the only model to use for estimation but customized for the different loads to estimate. It is commonly seen in other HUMS-related research that a single machine learning model type is selected and that is the only model considered for solving the problem. We were interested in keeping a wide variety of model types in our portfolio, especially given that there might be newer models developed in future that would be worth exploring. One way to address this issue is to retain a number of high performing models in an ensemble and leverage all of these individual models. While all 108 individual models could be included in an ensemble, an effort to trim that total to a smaller subset of high performing models was made through investigating rank sum and rank product.

Individual model results

The next step in this task was to use the evaluation metrics to determine which models to retain for ensemble building. As a visualization tool, 2D boxplots were created to simultaneously examine both metrics and highlight their differences for the various models, configurations and random seeds. Figure 1 shows the 2D boxplot corresponding to 5 of the 14 configurations of the 1D-CNN. 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 interquartile range (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.

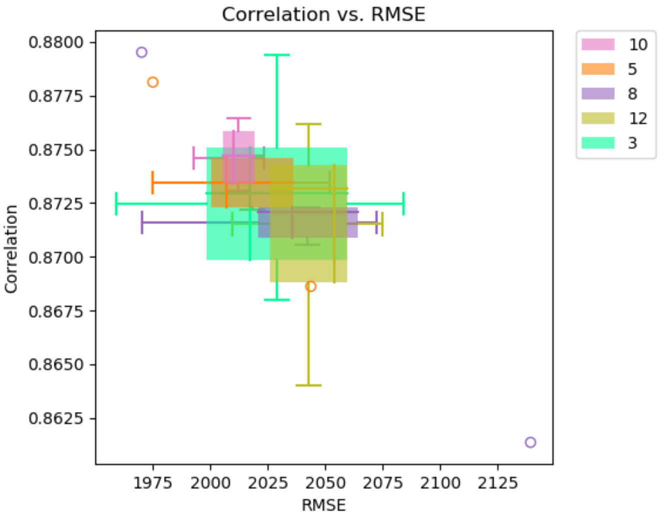

Figure 1: 2D boxplot of five 1D-CNN models showing RMSE and correlation

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. 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. The top models would ideally be located in the top left corner of the plot. 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. The models were ranked on their performance in four areas: RMSE IQR, correlation IQR, RMSE median and correlation median. For both IQR rankings and the RMSE median, the model would be ranked higher if the value was smaller. The opposite is true for the correlation median, as it is considered to rank higher at higher values. In this work, all four of the rankings, RMSE IQR, correlation IQR, RMSE median and correlation median, are important for a successful model.

Rank Sum and Rank Product

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 [7]. Rank sum can be calculated using Eqn 1, while rank sum is expressed in Eqn 2:

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

$$RP(op) = \left(\prod_{i=1}^{k} r_{op,i}\right)^{1/k} \tag{2}$$

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.

From the 108 individual models developed, Table 2 shows the top 30 individual models 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 are 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, 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 2, 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 prioritized, as designed, often above RMSE and correlation median values. While low variation is important, having all of 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 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, in that 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 prioritization 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.

Table 2: Rank sum and rank product for top 30 models

Model	RMSE				Correlation				Rank sum	Rank Pro-d-uct
	median		IQR		median		IQR			
	value	rank	value	rank	value	rank	value	rank		
4_MARS	2078	14	0	1	0.865	14	0.000	1	30	3.74
10_1D-CNN	2010	2	14	14	0.875	1	0.002	16	33	4.60
5_1D-CNN	2007	1	36	17	0.873	3	0.002	15	36	5.26
0_MARS	2217	19	0	2	0.853	17	0.000	2	40	6.00
8_MARS	2244	20	0	3	0.845	21	0.000	3	47	7.84
8_1D-CNN	2036	8	43	21	0.872	8	0.001	14	51	11.71
6_1D-CNN	2028	5	40	20	0.872	9	0.005	18	52	11.28
3_1D-CNN	2018	3	61	24	0.873	6	0.005	19	52	9.52
4_1D-CNN	2030	6	63	25	0.873	5	0.004	17	53	10.63
12_1D-CNN	2054	11	33	16	0.873	4	0.005	22	53	11.16
1_1D-CNN	2026	4	40	19	0.873	7	0.006	23	53	10.52
9_RF	2258	23	0	4	0.843	24	0.000	4	55	9.69
8_RF	2258	22	0	9	0.843	22	0.000	10	63	14.45
2_1D-CNN	2039	9	95	32	0.874	2	0.005	20	63	10.36
6_RF	2261	25	0	8	0.842	25	0.000	6	64	13.16
11_1D-CNN	2035	7	53	22	0.870	11	0.008	26	66	14.49
3_RF	2270	28	0	5	0.841	29	0.000	5	67	11.94
5_RF	2268	27	0	7	0.841	27	0.000	9	70	14.64
7_RF	2259	24	0	13	0.843	23	0.000	13	73	17.48
0_RF	2285	31	0	6	0.839	31	0.000	7	75	14.17
4_RF	2265	26	0	12	0.842	26	0.000	12	76	17.66
13_1D-CNN	2051	10	81	30	0.869	12	0.008	25	77	17.32
1_RF	2278	30	0	10	0.840	30	0.000	8	78	16.38
7_1D-CNN	2074	13	68	28	0.866	13	0.007	24	78	18.36
2_RF	2270	29	0	11	0.841	28	0.000	11	79	17.70
9_1D-CNN	2055	12	104	33	0.871	10	0.013	28	83	18.25
1_MARS	2249	21	63	26	0.847	20	0.016	29	96	23.72
9_MARS	2294	32	18	15	0.837	32	0.005	21	100	23.83
5_MARS	2092	15	220	55	0.863	15	0.031	41	126	26.69
3_MARS	2078	14	233	60	0.849	18	0.023	34	129	28.11

Ensembles

Ensembles were constructed enabling a subset of models to be leveraged in the load estimation. There are several ways to construct an ensemble, including using other machine learning methods to combine results. In this paper, however, we initially explored two straightforward methods: a simple average and weighted average based on the rank sum and rank product.

Simple Average

For a simple average ensemble, the predictions of all the top models are averaged for each datapoint 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 5 or 6 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 3 shows the RMSE and correlation results for the simple average ensemble and the weighted average ensembles. Figure 2 illustrates the load signal predictions for several individual models and the resulting simple average ensemble. The 1,395,616 data points cover 39 test flights merged together totaling just under 24 flight hours.

Table 3: Ensemble model results for RMSE and correlation

Ensemble Model	RMSE	Correlation
Simple average	1968	0.879
Weighted average – rank sum	1948	0.881
Weighted average – rank product	1942	0.882

Weighted Average

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. The weightings for the rank sum ensemble used Eqn 3, while the rank product ensemble weightings followed Eqn 4.

$$y_{ensemble} = \frac{\sum_1^n \frac{y_{op}}{RS_{op}}}{\sum_1^n \frac{1}{RS_{op}}} \tag{3}$$

$$y_{ensemble} = \frac{\sum_1^n \frac{y_{op}}{RP_{op}}}{\sum_1^n \frac{1}{RP_{op}}} \tag{4}$$

where $y_{ensemble}$ is the load signal prediction, y_{op} is the load signal prediction from the individual model, RP_{op} is the rank product for that individual model, RS_{op} is the rank sum for that individual model, and n is the number of individual models in the subset.

The weighted average results for rank sum and rank product are included in Table 3 and Figure 2. A number of observations were made based on these results. Given that the individual models had a range of RMSE values from 1959 to 2284 and a correlation coefficient range from 0.839 to 0.880, the ensembles performed very well. Of the three ensembles, the simple average ensemble resulted in a slightly higher RMSE (1968) than the top individual model, but lower than the other 25 individual models, and a correlation coefficient just below the best model. The rank sum and rank product ensemble both 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 Figure 2 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, 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. In load estimation problems, underprediction of peak values is often a challenge since the number of peak values that appear in training are far outnumbered by off-peak data points. It is worth noting that the set of sensors from which the models receive the input were placed on the aircraft for other purposes, and the attempts to obtain accurate predictions from these sensors requires extracting as much of the relevant information as possible using the machine learning models.

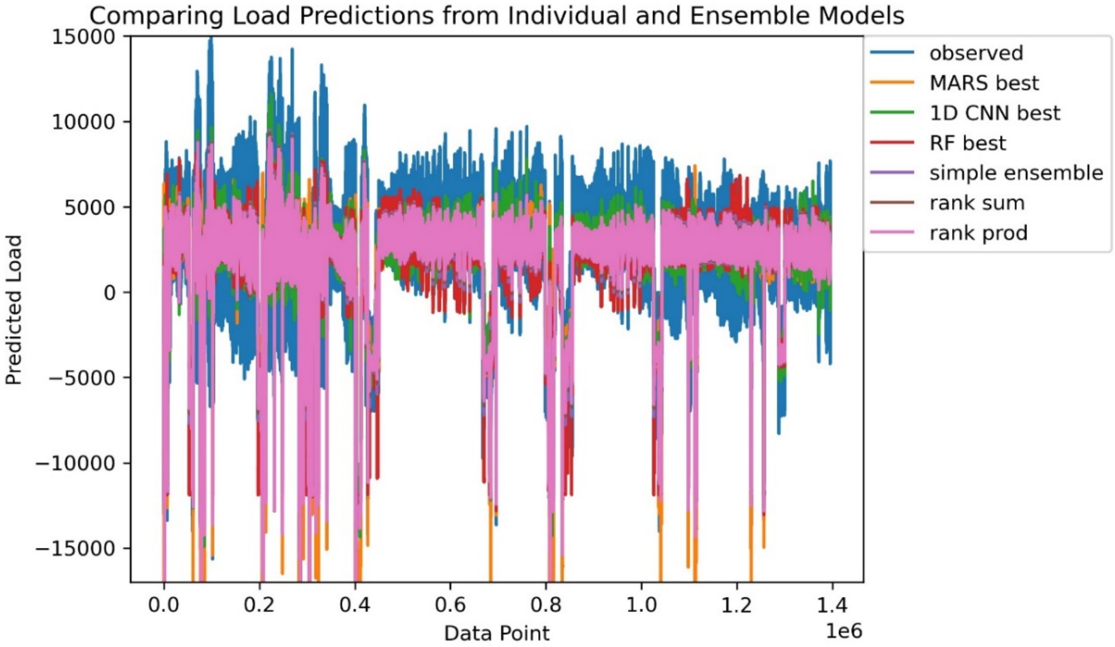

Figure 2: Load signal predictions for several individual models and the three ensembles

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 future, we plan to continue to explore other metrics that may lead to better overall models and therefore better ensembles in the end.

Concluding Remarks

An approach to merging models from various machine learning techniques, including Random Forest, long-short term memory recurrent networks, multivariate adaptive regression splines, and 1-dimensional convolutional neural networks, into an ensemble is outlined in this paper in the context of helicopter load estimation. The resulting 108 individual models covering a range of hyperparameter configurations were evaluated primarily by their root mean squared error and correlation with the target signal. Considerable variation in the model results was evident when changing the random seed and hyperparameter configuration.

These 108 individual models were then evaluated to determine their stability across random seeds, comparing median values and interquartile ranges from boxplots of their RMSE and correlation. Using rank product and rank sum, the individual models were ranked to down select a subset of 26 high performing load estimation models. An ensemble of the subset of models was then constructed enabling a number of machine learning models to be leveraged in the load estimation. A simple average ensemble and weighted average ensemble related to the rank sum and rank product results were trialled. There were some notable benefits to using an ensemble of individual models, in particular introducing some diversity in the individual models making up the ensemble in terms of configuration and algorithm, and a smoother load signal prediction with higher correlation. Our results found that in this application, all three ensemble methods obtained low RMSE and high correlation. There were several individual 1D-CNN models that performed very well. The output of the ensembles performed similar to, if not better than these models, and overall should provide a more robust and consistent load estimate.

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. It is evident that more effort in the future is required to obtain better individual models, however the results of this initial work aimed at developing an approach for managing, selecting and leveraging a large number of machine learning models are promising. Future work is anticipated in the following areas in order to further improve the load estimates:

- inclusion of other machine learning model types for load signal estimation,
- further exploration of hyperparameter values for each model type,
- further refinement of subset selection and the criteria for subset selection, and
- further consideration of other ensemble methods, including dynamic approaches.

Acknowledgments

This work was funded in part by Defence Research and Development Canada. The authors would like to acknowledge the contributions of Davis To in advancing this work.

References

1. Cheung, C., Sehgal, S., Valdés, J.J., “A machine learning approach to load tracking and usage monitoring for legacy fleets”, *Proc. of the 30th Symposium of the International Committee on Aeronautical Fatigue and Structural Integrity,* Krakow, Poland, June, 2019.
2. Friedman, J., “Multivariate adaptive regression splines (with discussion)”, *Annals of Statistics,* Vol 19, No. 1, pp. 1-141, 1991.
3. Breiman, L., “Random forests”, *Machine learning*, Vol. 45, pp. 5-32, 2001.
4. Hochreiter, S., Schmidhuber, J., "Long short-term memory", *Neural Computation,* Vol 9, No. 8, pp. 1735–1780, 1997.
5. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., “Gradient-based learning applied to document recognition”, *Proc. of the IEEE,* Vol. 86, No. 11, pp. 2278-2324, Nov 1998.
6. Damien, A., et al., “TFLearn: Deep learning library featuring a higher-level API for TensorFlow”, 2016, https://github.com/tflearn/tflearn.
7. Breitling, R., Armengaud, P., Amtmann, A., Herzyk, P., “Rank Products: A simple, yet powerful, new method to detect differentially regulated genes in replicated microarray experiments”, *Federation of European Biochemical Societies Letters*, Vol. 573, pp. 83-92, 2004.