MIE1516 Final Project

Alex Kwan, 1001559057

Neural Networks for Retail Demand Forecasting

I. Motivation

One of Canada's largest retailers currently incurs $5M in annual overtime labour staffing costs at their four major distribution centres (DCs) due to inaccurate demand forecasts. This retailer is seeking a tool from my Capstone team that accurately predicts the outbound demand (ft^3) for their 4 DCs both one week (N+1) and two weeks (N+2) in advance.

Each week, current state forecasts are made by averaging the volume for a given DC and week from the previous 2 years without considering the spread between the 2 values. Figure 1 below demonstrates that for DC 5 in week 42, 2016 there was a 25.3% drop in volume compared to only a 3.0% drop for that same week in 2017. Clearly, taking the average to predict a -14.1% estimate for 2018 ignores the underlying trend.

2016	whse	Group	42
2016	03	REG	-7.1%
2016	05	REG	-25.3%
2016	07	REG	-17.1%
2016	08	REG	-15.3%

2017	whse	Group	42
2017	03	REG	-4.8%
2017	05	REG	-3.0%
2017	07	REG	-6.3%
2017	08	REG	-0.5%

Average	whse	Group	42
2016-2017	03		-6.0%
2016-2017	05		-14.1%
2016-2017	07		-11.7%
2016-2017	08		-7.9%

Figure 1: Current state forecasts are a simple average of the previous two years

Additionally, this retailer ships to 30 regions across Canada, with each region receiving goods from 3 of the 4 DCs as given in the table below:

DC	# Regions Served
4	30
5	30
7	12
8	18
	90 pairs

Table 1: 90 DC-Region pairs

Clearly, there is opportunity to increase the prediction granularity, by aggregating 180 DC-Region pair predictions into 8 predictions to predict N+1 and N+2 demand for each of the four DCs.

➔ (30 regions) * (each served by 3 DCs) * (2 weeks) = 180 predictions

II. Exploratory Data Analysis

This retailer provided 300 weeks of historical data (Jan 2013 to Sept 2018) including weather, POS (point of sales, or store sales), store inventory, store region, historical regular demand, and week N input (business data at DC/Region level for the current week). For NDA reasons, specific volume values are removed from graphs.

In Figure 2 below each graph provides the outbound average weekly volume for a DC over the entire historical data period. This is the current granularity used to make predictions, looking at for each DC and each week, what was the volume from last year, 52 weeks ago and two years ago, 104 weeks ago.

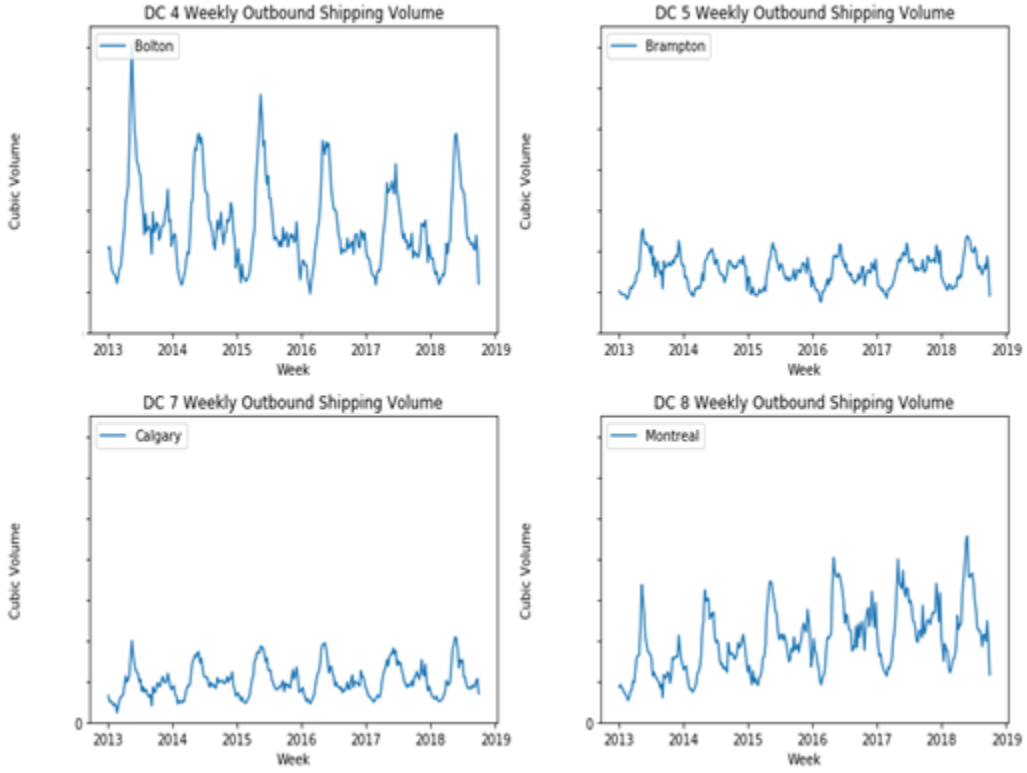


Figure 2: Aggregated outbound volume of each DC every week

One can see there is strong consistent seasonality for all 4 DCs, with volume peaking in the spring time and a smaller peak at the end of the year in the holiday season. Also, volumes are relatively stable year on year, with no downward or upward trend for any of the DCs except DC 8, which has a slight upward trend since this DC is expanding. Table 2 below demonstrates that achieving lower mean average percentage error for DC 4 predictions is the most important as it ships the most volume.

DC	Avg Weekly Volume (%)
4	37%
5	21%
7	14%
8	28%
	100%

Table 2: Average weekly DC volume split over the 300-week period

Furthermore, regions are not equal and overall 300 weeks of data and 30 Regions each served by 3 DCs means that there: 300 * 30 * 3 = 27,000 unique DC-Region weekly volume values which are each viewed as a sample. Figure 3 below plots the 4 DCs' volume to each region they serve. As expected, DC 4 ships the most to Regions.

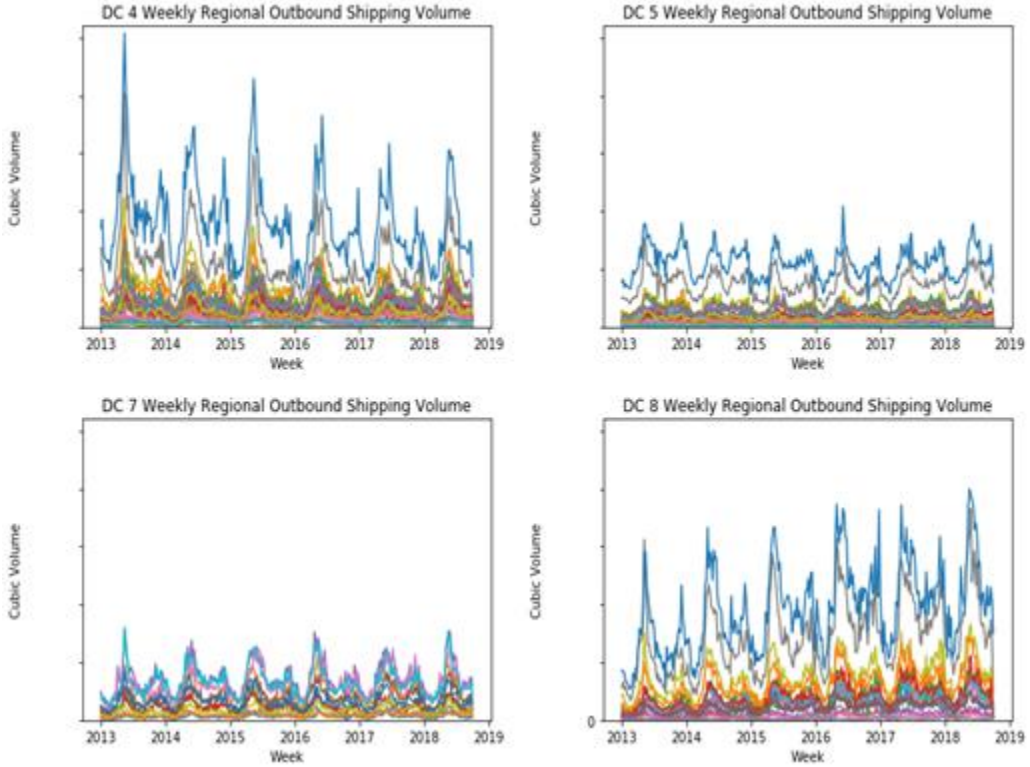


Figure 3: DC weekly volume to all regions they serve

Also 6 out of 30 regions account for 51% of shipping volume. The GTA, Region 27 itself accounts for 17.0% of total volume and the Greater Montreal Area, Region 24 accounts for 11.5%. Clearly, 10% prediction error on a high-volume region is much worse than 10% error on low volume regions.

III. Feature Engineering

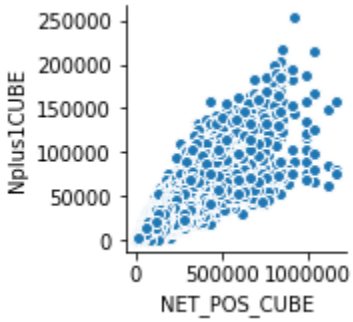
Features are used to predict the target variables N+1 CUBE and N+2 CUBE, which represent the amount of volume a DC ships to a region, one week and two weeks into the future.

One hot encoding was used for all discrete numeric variables: year, week, and region, as well as all categorical text variables: holidays and sale promotions in order to make the machine learning algorithms better interpret these variables. For example, one hot encoding allows the machine learning algorithms to learn separate factors for each week to model week-by-week seasonality as shown below.

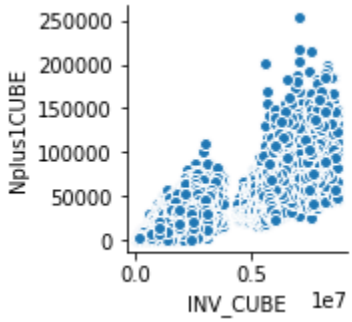
W18	0.077493
W19	0.083467
W20	0.078456
W21	0.065662
W22	0.054666

Figure 4: Correlation coefficients with N+1 CUBE for a subset of the weeks

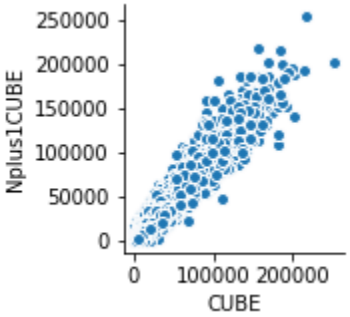
To reiterate, CUBE is the shipping volume from a specific DC to a specific Region in week N and the goal is to predict N+1 CUBE and N+2 CUBE. To determine what features are most correlated with N+1 and N+2 CUBE, a correlation coefficient was determined for each factor. A higher correlation means that factor's corresponding scatterplot with N+1 or N+2 CUBE has a tighter shape around a linear line. The following four scatter plots are correlation plots with N+1 CUBE



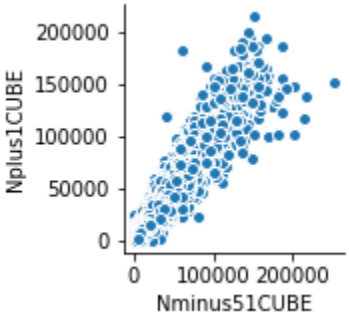
NET_POS_CUBE 0.884



INV_CUBE 0.845



CUBE 0.968

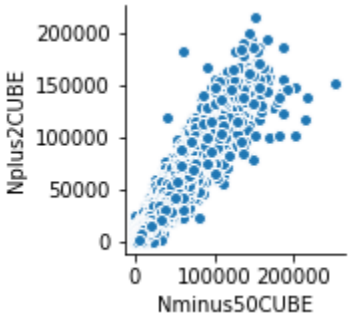


Nminus51CUBE 0.951

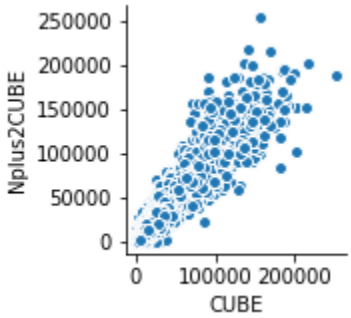
Figure 5: Scatter plots and correlation coefficients with N+1 CUBE for various volume features

Week N and Week N-51 CUBE are more correlated with N+1 CUBE than week N POS and week N INV (store inventory) as shown in the tighter spread in the bottom 2 scatterplots in Figure 5.

Moving onto N+2 CUBE the team found that using N-50 CUBE values are better than using Week N CUBE values to predict N+2 CUBE. This finding makes sense as N-50 is the 52 week shift of N+2 and the further one wants to predict in the future from Week N the more variability. Visually, this finding is supported as seen in the scatter plots below. Notice how in Figure 6 'CUBE' is less fitted to Nplus2CUBE than Nminus50CUBE and has a lower correlation value.



CUBE 0.949



Nminus50CUBE 0.951

Figure 6: Scatter plots and correlation coefficients with N+2 CUBE for various volume features

For temperature (°F), higher temperatures in week N, N-51 and N-50 have a positive significant correlation with week N+1 and N+2 CUBE values meaning that people shop less when it's cold outside, see Figure 7 and 8 below.

temperatureMin	0.199290
temperatureHigh	0.195058
tempMinNminus51Avg	0.214463
tempMinNminus50Avg	0.221633
tempHighNminus51Avg	0.206873
tempHighNminus50Avg	0.212798

Figure 7: Correlation coefficients with N+1 CUBE for various temperature features

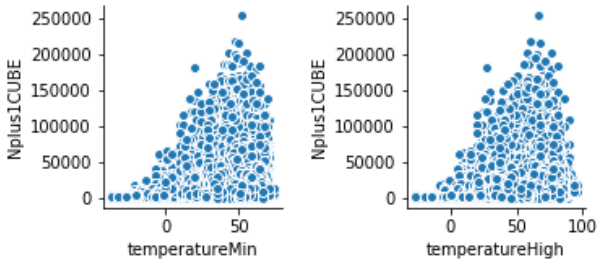


Figure 8: Week N min and high temperature scatter plots with N+1 CUBE

For inches of snow in week N, there was a negative correlation with N+1 CUBE (Figure 8) meaning that shoppers are less willing to go out since they must clear the snow off their car, shovel their driveway, and drive on unplowed roads.

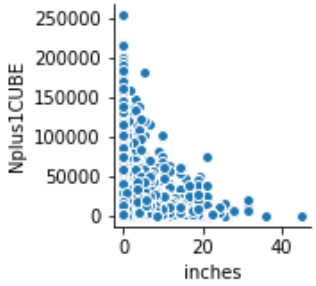


Figure 8: Week N inches of snow scatter plot with N+1 CUBE

For weather icons, non-precipitation icons had a low correlation with N+1 CUBE. For rain days there was a positive correlation with N+1 CUBE (Figure 9) meaning that when people can't go outside in the sunshine they opt to go shopping instead, while for snow days people prefer to stay home as shown by the negative correlation of snow days with N+1 CUBE.

rainDays	0.125651
snowDays	-0.127033
sleetDays	0.020128
clearIcon	0.013971
cloudyIcon	-0.034955
fogIcon	-0.075002
partlyIcon	0.051985
rainIcon	0.081438
sleetIcon	0.018667
snowIcon	-0.119066
windIcon	-0.017241

Figure 9: Correlation coefficients with N+1 CUBE for week N weather features

The team was also given the week of every holiday and sale from 2013 to 2019 including Quebec specific holidays. Therefore, unlike all the other features, sales and holidays are known to the retail company well before they occur.

The team used different time shifts of the sales and holidays with N, N+1, N+2, and N+3. The rationale is that perhaps store managers only order more stock after the sale/holiday not in anticipation of the sale/holiday. However, no holiday or promotion had a significant correlation factor above 0.1 with N+1 CUBE or N+2 CUBE. Nonetheless these features are still in the trainable feature set.

In summary, there are a total of 27,000 samples each with 219 features available for selection to be used for training to predict N+1 and N+2 CUBE for each of the four DCs as given below:

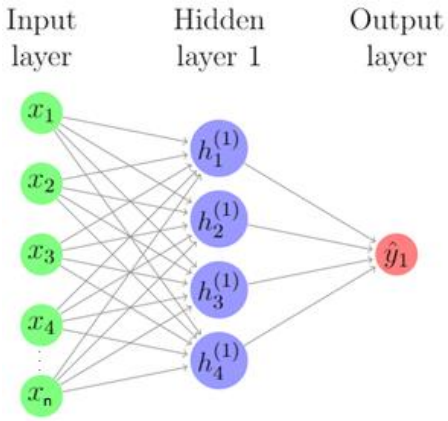
Feature Type	**Feature Group**	**Details**	**# Features**
Discrete & Categorical (One-hot encoded)	Year	2013 - 2018	6
	Week	1 – 52	52
	Region	17 – 46	30
	DC	4,5,7,8	4
	Statuary holidays	10 holidays, time-shifted N, N+1, N+2, N+3	40
	Promotions	12 promotions, time-shifted N, N+1, N+2, N+3	48
Continuous	Shipping volume (CUBE)	Time-shifted N-104, N-103, N-51, N-50, N-2, N-1, N	7
	Store inventory volume	Time shifted N, N-51, N-50	3
	Point of sales volume	Time shifted N, N-51, N-50	3
	Temperature	Min and max temperature, time shifted N, N-51, N-50	6
	Weather icons	Clear, cloudy, fog, overcast, rain, sleet, snow, and wind	8
	Precipitation	Rain, snow, and sleet days, time shifted N, N-51, N-50	9
	Inches of snow	Time shifted N, N-51, N-50	3
			219

Table 2 – List of features to train the neural network

All samples with NaN values were eliminated. Additionally, the entire training and testing set were standardized together with zero mean and unit variance [1]. The validation set predictions of the target variables are also scaled with the training set parameters of the target variables.

IV. Model Architectures

A single fully connected neural network architecture is proposed for all DC and target week predictions giving eight models: DC4Nplus1, DC4Nplus2, DC5Nplus1, DC5Nplus2, DC7Nplus1, DC7Nplus2, DC8Nplus1, and DC8Nplus2. The decision to use eight models and not four (one for each DC) with two target variables is so that the neural network nodes focus on a single target variable and not two (N+1 and N+2 CUBE). These eight models, known as the 'Global' models differ only in their samples for training and testing sets, with the 27,000 samples filtered by the respective DC. They all share the following architecture:



Where:

n = # of features up to 219 as given in Table 2

x = {x1,x2,…xn}

h_i= hidden nodes in the single hidden layer

$\hat{y}_1$ = N+1 CUBE or N+2 CUBE

Figure 10 – Neural Net Architecture

This architecture will serve as the baseline as the number of features, hidden units and hidden layers will also be adjusted. Additionally, eight more models where all 27,000 samples are used in the training/testing sets (meaning they aren't filtered by DC) will be known as the 'Monolithic' models. These Monolithic models may benefit from the increased number of samples assuming there are consistent relationships shared across all four DCs.

V. Existing Baseline

MAPE is used to measure the forecast accuracy between the selected machine learning forecasting algorithms and the existing company solutions. Its use has been specified as a constraint by the client. A lower MAPE percentage is better and the MAPE formula is:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|Y_i - \widehat{Y}_i\right|}{Y_i}$$

Where:

- Y_t is the historical test data for time period t
- $\hat{Y}_t$ is the forecasted value for time period t
- T is the number of time periods

It should be noted that MAPE suffers from many issues:

1. It cannot be used if there are zero values in the historical data (which is why all samples with NaN values were removed)
    a. Would cause division by zero
2. MAPE favours under forecasting errors
    a. Under forecasting errors are capped at 100% whereas over forecasting errors are unbounded

3. MAPE is not scaled and thus cannot accurately measure forecast accuracy between methods using different data scales

As this problem is a time series, a simple 80/20 split into training and testing sets does not suffice as it would mean the model would train on the future to predict the past. The evaluation process is:

- For each of the 52 test weeks from Week 37, 2017 until Week 36, 2018, each methodology made an N+1 (one week into the future) and N+2 (two weeks into the future) prediction for each DC-Region pair using historical data dating back to Week 1 2013 up to the current test week minus 1.
- After each week's N+1 and N+2 predictions were determined, the subsequent test week prediction uses updated historical data with the test week that just passed included in the training data.

Meaning as the 52-week prediction horizon progresses, the number of samples in the training set increases and the number of samples in the validation set decreases. This process was to mimic the actual functionality of the current company's process. The Capstone team has already attempted various machine learning algorithms including Gradient Boosting for regression with results given below:

DC	**Current State**		**Gradient Boosting**	
	N+1	**N+2**	**N+1**	**N+2**
4	12.8%%	16.3%	5.6%	7.5%
5	10.1%	12.4%	4.7%	6.2%
7	15.6%	20.6%	7.9%	10.9%
8	14.0%	15.4%	6.8%	8.3%
Average	**13.1%**	**16.2%**	**6.2%**	**8.2%**

Table 3: N+1 and N+2 MAPE scores over 52-week testing horizon. GB performed the best.

Gradient Boosting outperforms the current state by an average of 6.9% for N+1 and 8.0% for N+2, which translates to annual savings $2.6MM.

Additionally, another constraint is that each algorithm takes less than 30 minutes to train and make eight weekly predictions, which Gradient boosting does. The neural network is also evaluated with the same criteria.

VI. Experiments

RMSprop was the chosen optimizer as given in the keras model guide for continuous regression [1]. Adam was also used [2] since it chooses an adaptive learning rate for every model parameter [2].

To achieve the best performance, the number of epochs and learning rate was adjusted so that the neural network would achieve early stopping since once the training loss starts to plateau, the models begin to overfit and test set error increases as shown below [3]:

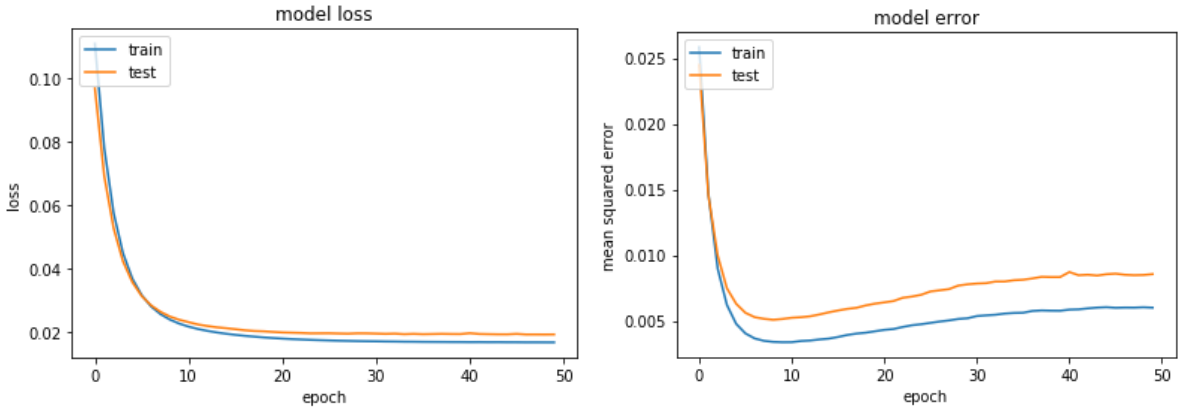


Figure 11 – Model error increases due to overfitting from too many epochs

For activation functions, ReLu was used since it doesn't suffer from the vanishing gradient problem but also tanh was tried as well since this is a shallow architecture and so the vanishing gradient problem should not be too significant. Also, tanh provides negative values and a stronger gradient which may be beneficial for placing low emphases on non-significant features [4].

L2 regularization is selected to prevent overfitting by adding an error term to discourage large weights [5].

As this is a regression problem, accuracy is not meaningful, and the metric of choice is mean squared error (MSE) whose formula is:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2$$

Where:

- Y_t is the historical test data for time period t
- $\hat{Y}_t$ is the forecasted value for time period t
- n is the number of time periods

Training loss is based on the training data from Week 1 2013 to the test week. For the Global models, the 27,000 samples are filtered for each respective DC so when making the first prediction in the 52-week window prediction window, the train and test splits are as follows:

DC	# DC Region Pairs (90)	# Samples (300 weeks)	# Training Samples (248 weeks)	# Testing samples (52 weeks)
4	30	9,000	7,440	1,560
5	30	9,000	7,440	1,560
7	12	3,600	2,976	624
8	18	5,400	4,464	936

Table 4 - Global models train and test splits for testWeek = 249

While for the Monolithic models the train and test splits for each DC are identical:

DC	# DC Region Pairs (90)	# Samples (300 weeks)	# Training Samples (248 weeks)	# Testing samples (52 weeks)
4	30	27,000	22,320	4,680
5	30	27,000	22,320	4,680
7	12	27,000	22,320	4,680
8	18	27,000	22,320	4,680

Table 5 - Monolithic models train and test splits for testWeek = 249

VII. Results

After numerous model combinations (see Appendix A) with both Global and Monolithic models, level of layers, number of nodes, and activation functions, the best model was determined to be a Global model as follows:

1F: Global Model	**Design & Hyperparameters Choice**	**Value**
Model	Training features (n)	All 219
	epochs	20
	Batch_size	64
Optimizer	Function	RMSprop
	Learning rate	0.0002
	Loss	Mean Squared Error
	rho	0.9
	decay	0.0
Hidden Layer 1	Hidden Nodes	8
	Initialization	Xavier (glorot_normal)
	L2 Regularization	0.01
	Dropout	N/A
	Activation Function	tanh
Output Layer	Nodes	1
	Activation function	linear

Table 6 – Best performing neural network configuration

The neural network easily meets the 30 minutes runtime constraint as a single week of eight predictions requires only 39 seconds and the total running time for all 52 weeks equals 2,047 seconds.

Comparing the neural network's performance to the other techniques, it is evident that the neural network does beat the current state but is unable to match the performance of the gradient boosting regressor as shown in Table 7 below.

DC	Current State		Gradient Boosting		Neural Network	
	N+1	N+2	N+1	N+2	N+1	N+2
4	12.8%%	16.3%	5.6%	7.5%	14.4%	13.2%
5	10.1%	12.4%	4.7%	6.2%	8.4%	9.6%
7	15.6%	20.6%	7.9%	10.9%	11.1%	15.3%
8	14.0%	15.4%	6.8%	8.3%	11.1%	11.0%
Average	**13.1%**	**16.2%**	**6.2%**	**8.2%**	**11.2%**	**12.1%**

Table 7 – Gradient Boosting consistently demonstrates the best performance across all DCs and target variables

Figures 12 and 13 below further detail that Gradient Boosting (GB) performs best year-round. For DC specific weekly comparisons please refer to Appendix B.

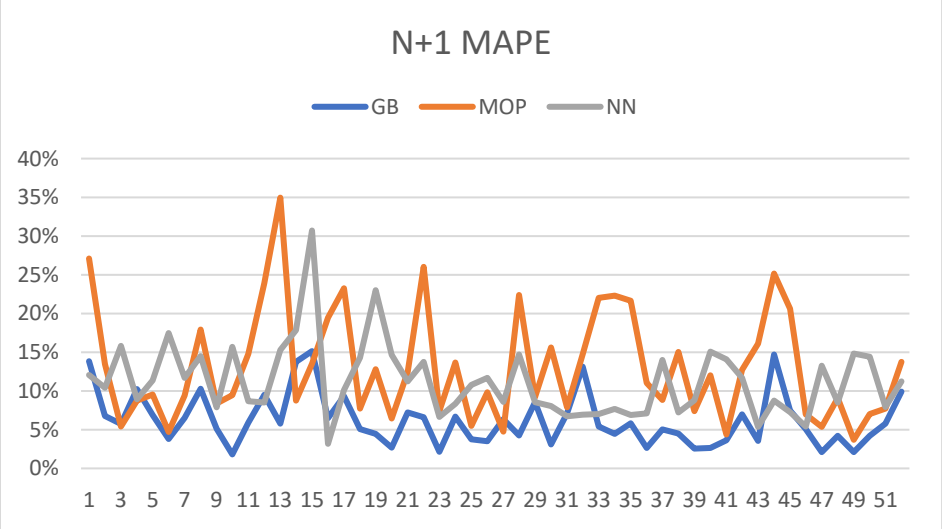


Figure 12 – N+1 MAPE comparison

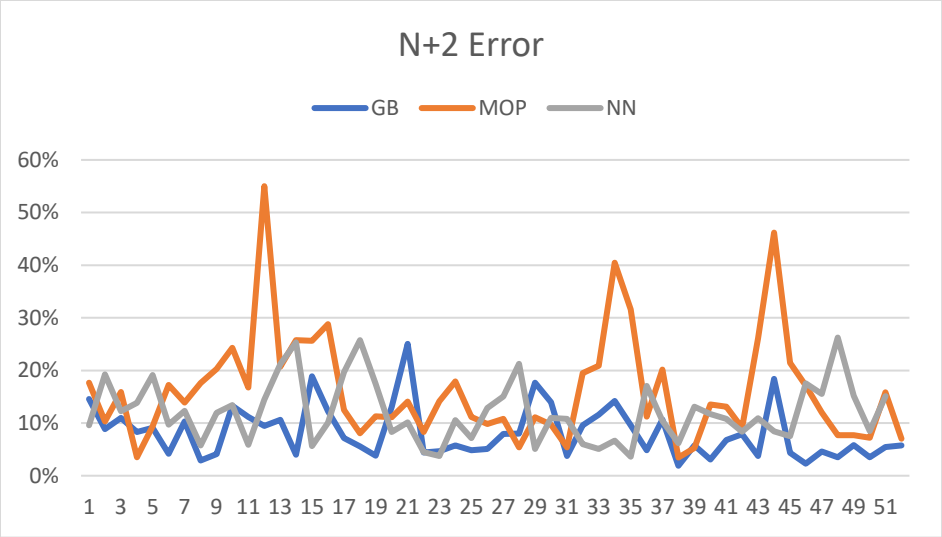


Figure 13 – N+2 MAPE comparison

VIII. Conclusion and Next Steps

Overall, the neural networks are unable to reach the performance of the gradient boosting baseline. A thorough, systematic approach is recommended to tune the neural network configuration. This approach could use TensorBoard and consider: four different architectures for each DC, training feature selection, different optimizers, a scheduled learning rate, activation functions, number of hidden layers, number of hidden notes, and regularization.

IX. References

[1] R. Gandhi and R. Gandhi, "Improving the Performance of a Neural Network," *Towards Data Science*, 17-May-2018. [Online]. Available: https://towardsdatascience.com/how-to-increase-the-accuracy-of-a-neural-network-9f5d1c6f407d. [Accessed: 11-Apr-2019].

[2] Sebastian Ruder, "An overview of gradient descent optimization algorithms," *Sebastian Ruder*, 29-Nov-2018. [Online]. Available: http://ruder.io/optimizing-gradient-descent/index.html#rmsprop. [Accessed: 12-Apr-2019].

[3] "Getting started with the Keras Sequential model," *Guide to the Sequential model - Keras Documentation*. [Online]. Available: https://keras.io/getting-started/sequential-model-guide/. [Accessed: 12-Apr-2019].

[4] A. S. V and A. S. V, "Understanding Activation Functions in Neural Networks," *Medium*, 30-Mar-2017. [Online]. Available: https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0. [Accessed: 12-Apr-2019].

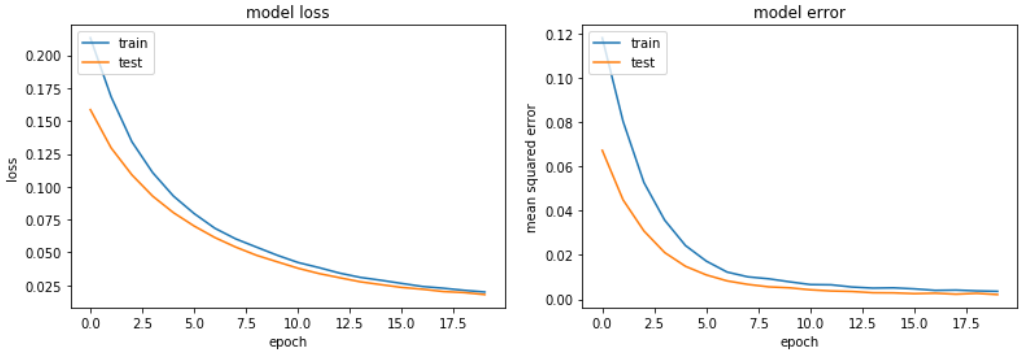
[5] J. McCaffrey10/05/2017, "Neural Network L2 Regularization Using Python," *Visual Studio Magazine*. [Online]. Available: https://visualstudiomagazine.com/articles/2017/09/01/neural-network-l2.aspx. [Accessed: 12-Apr-2019].

X. Appendices

Appendix A – Experimental Results

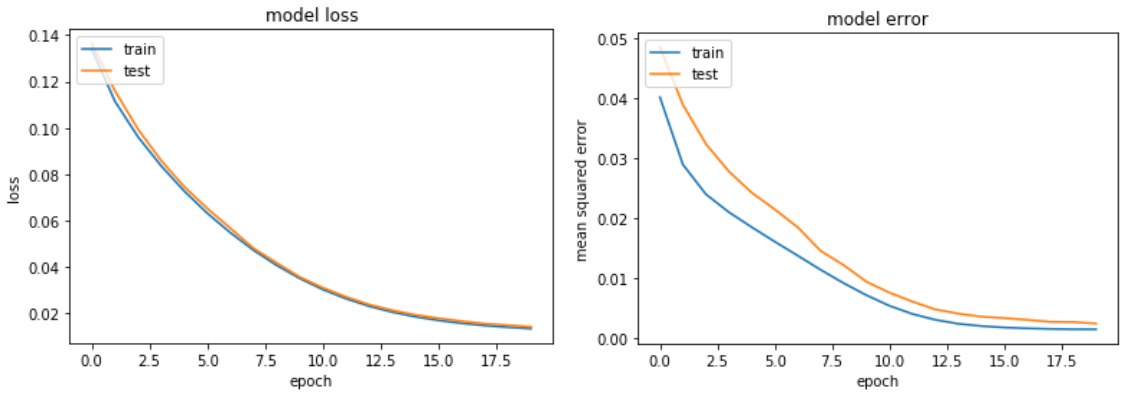
1A: Global model	Design & Hyperparameters Choice	Value
Model	Training features (n)	All 219
	epochs	30
	Batch_size	64
Optimizer	Function	RMSprop
	Learning rate	0.0002
	Loss	Mean Squared Error
	rho	0.9
Hidden Layer 1	Hidden Nodes	4
	Initialization	Xavier (glorot_normal)
	L2 Regularization	0.01
	Dropout	0.25
	Activation Function	Relu
Output Layer	Nodes	1
	Activation function	Linear

```
{'DC4Nplus1MAPE': 22.703668717516575,
 'DC4Nplus2MAPE': 24.000398498679036,
 'DC5Nplus1MAPE': 11.353596137027738,
 'DC5Nplus2MAPE': 12.983112434631645,
 'DC7Nplus1MAPE': 18.886698883274974,
 'DC7Nplus2MAPE': 19.895106485082113,
 'DC8Nplus1MAPE': 12.212126763306138,
 'DC8Nplus2MAPE': 15.952144644282711}
```



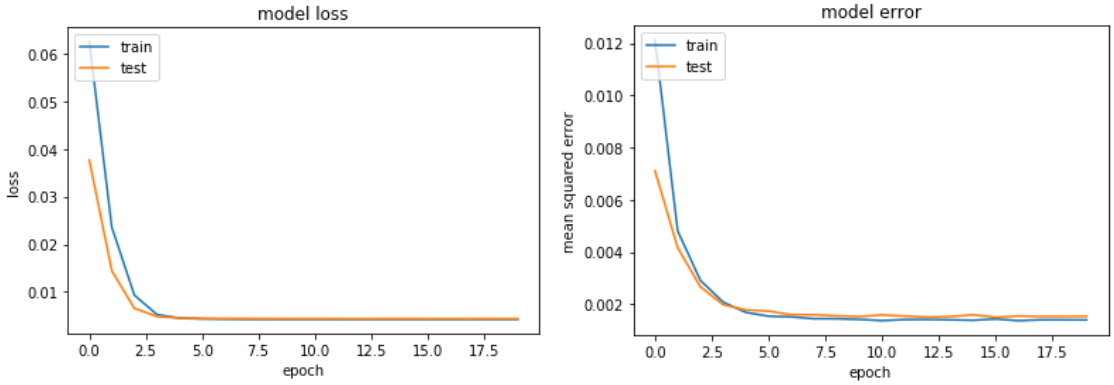
1B: Global model - Same as 1A but without dropout

```
{'DC4Nplus1MAPE': 14.530016039644774,
 'DC4Nplus2MAPE': 16.55478644084172,
 'DC5Nplus1MAPE': 8.502068188781088,
 'DC5Nplus2MAPE': 10.846040697938033,
 'DC7Nplus1MAPE': 18.20815589684756,
 'DC7Nplus2MAPE': 17.585844196211127,
 'DC8Nplus1MAPE': 12.00334010653989,
 'DC8Nplus2MAPE': 15.43836585419849}
```



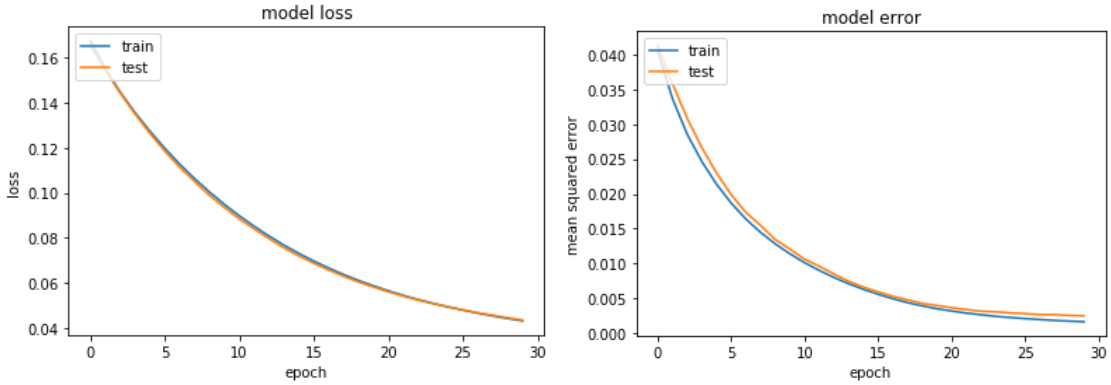
2B: Monolithic model - Same as 1B but uses 27,000 samples for all models

```
{'DC4Nplus1MAPE': 12.261503432644961,
 'DC4Nplus2MAPE': 14.30092847904775,
 'DC5Nplus1MAPE': 12.870995759785353,
 'DC5Nplus2MAPE': 14.429532407698217,
 'DC7Nplus1MAPE': 22.685280269657902,
 'DC7Nplus2MAPE': 25.39161321964874,
 'DC8Nplus1MAPE': 21.018442037572818,
 'DC8Nplus2MAPE': 25.349980308851706}
```



Global model: 1C – Second 4-node hidden layer. adjusting epochs to 30 and raising learning rate to 0.0001

```
{'DC4Nplus1MAPE': 20.085791998665524,
 'DC4Nplus2MAPE': 24.228947705303984,
 'DC5Nplus1MAPE': 10.248503095984246,
 'DC5Nplus2MAPE': 11.670300309109722,
 'DC7Nplus1MAPE': 23.755823980194045,
 'DC7Nplus2MAPE': 22.711438395809992,
 'DC8Nplus1MAPE': 14.820172252312199,
 'DC8Nplus2MAPE': 15.07025699567515}
```

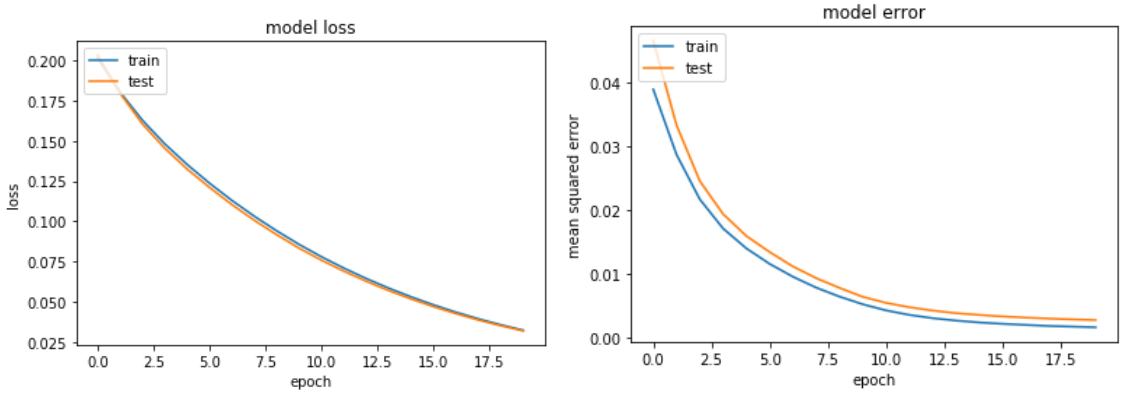


Global model: 1D – Only 1 hidden layer with 8 hidden units and ReLu activation, 20 epochs, learning rate = 0.0002

```
{'DC4Nplus1MAPE': 15.011068240554543,
 'DC4Nplus2MAPE': 16.71582546555331,
 'DC5Nplus1MAPE': 8.14530458572244,
 'DC5Nplus2MAPE': 10.365316122202998,
 'DC7Nplus1MAPE': 14.518696868055612,
 'DC7Nplus2MAPE': 16.03441523055706,
 'DC8Nplus1MAPE': 11.246145211356989,
 'DC8Nplus2MAPE': 13.697586081916215}
```

Global model: 1E – Same as 1D but with 16 hidden units in the single hidden layer

```
{'DC4Nplus1MAPE': 16.9688717963686,
 'DC4Nplus2MAPE': 18.05587111773634,
 'DC5Nplus1MAPE': 9.460946613254388,
 'DC5Nplus2MAPE': 10.55526244641739,
 'DC7Nplus1MAPE': 14.581158788630393,
 'DC7Nplus2MAPE': 15.448254791324226,
 'DC8Nplus1MAPE': 10.678498365233382,
 'DC8Nplus2MAPE': 13.163741367911639}
```

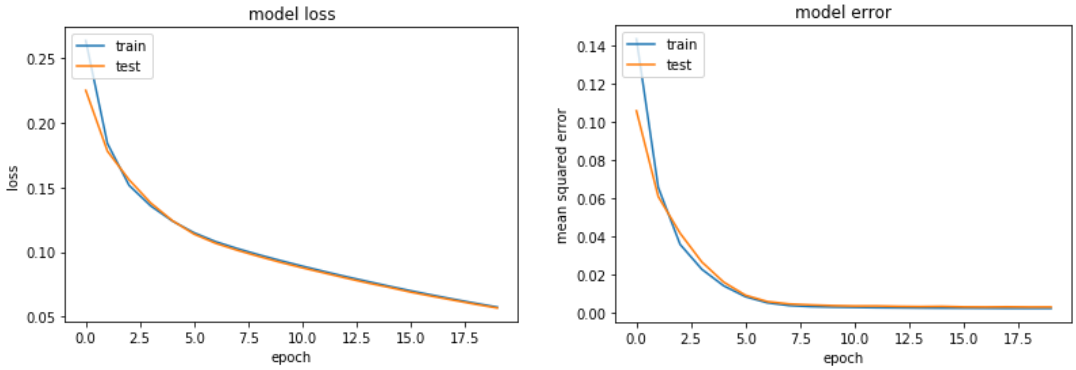


Global model: 1F – Same as 1E but with 8 units in the hidden layer, with tanh activation function

```
{'DC4Nplus1MAPE': 13.98835068673277,
 'DC4Nplus2MAPE': 14.741916909699654,
 'DC5Nplus1MAPE': 8.386490360218948,
 'DC5Nplus2MAPE': 10.906896757404121,
 'DC7Nplus1MAPE': 11.32565043575264,
 'DC7Nplus2MAPE': 15.836386224560478,
 'DC8Nplus1MAPE': 11.039584585253134,
 'DC8Nplus2MAPE': 12.735658644592164}
```

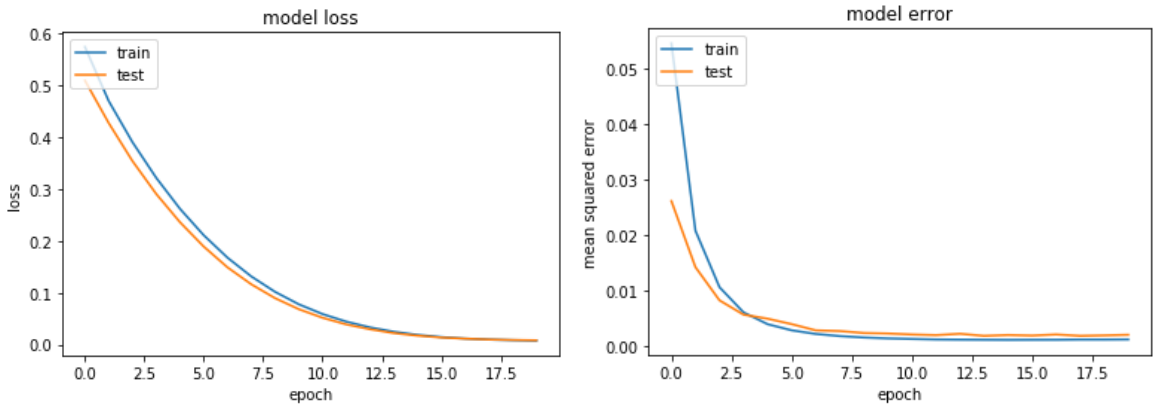
Global model: 1Fb – only n=10 features that have the highest correlations with the target variables

```
'Nminus50CUBE','Nminus51CUBE','Nminus2CUBE','Nminus1CUBE','CUBE','temperat
ureMin','tempMinNminus50Avg', 'tempMinNminus51Avg','rainDays','snowDays','
inches'
```




```
{'DC4Nplus1MAPE': 18.539222740947682,
 'DC4Nplus2MAPE': 20.292867708159225,
 'DC5Nplus1MAPE': 10.968694757200785,
 'DC5Nplus2MAPE': 15.699805760505464,
 'DC7Nplus1MAPE': 24.10087867137506,
 'DC7Nplus2MAPE': 25.43723146047054,
 'DC8Nplus1MAPE': 17.255550221253984,
 'DC8Nplus2MAPE': 20.69395893464669}
```

Global model: 1G – Same as 1F but with 32 units in the hidden layer



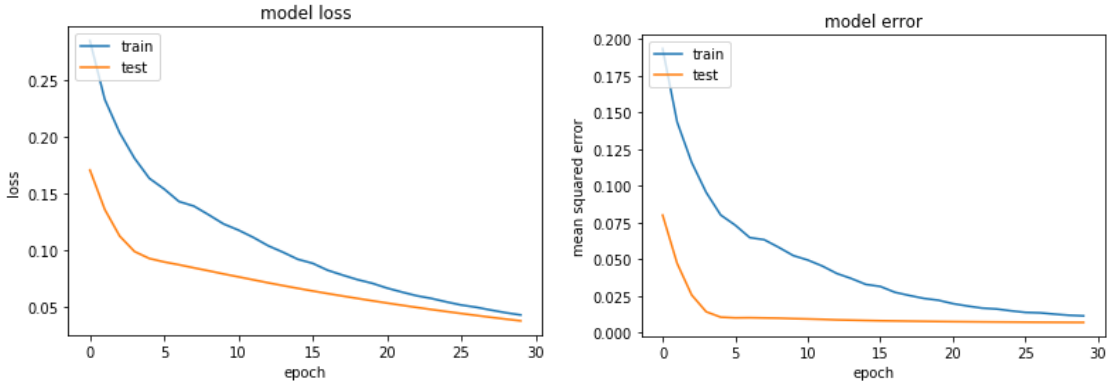

```
{'DC4Nplus1MAPE': 17.925223654870383,
 'DC4Nplus2MAPE': 18.157045432649166,
 'DC5Nplus1MAPE': 9.326582265689202,
 'DC5Nplus2MAPE': 10.993002171865651,
 'DC7Nplus1MAPE': 10.982004536861968,
 'DC7Nplus2MAPE': 15.536007405983979,
 'DC8Nplus1MAPE': 11.572605956329737,
 'DC8Nplus2MAPE': 13.146600745085186}
```

Global model: 1H – Same as 1F but with decay 0.01

```
{'DC4Nplus1MAPE': 21.1265616393783,
 'DC4Nplus2MAPE': 23.66695817595834,
 'DC5Nplus1MAPE': 11.608540817243215,
 'DC5Nplus2MAPE': 12.248894198380912,
 'DC7Nplus1MAPE': 15.956937992301004,
 'DC7Nplus2MAPE': 19.1925019088458,
 'DC8Nplus1MAPE': 12.144280929651961,
 'DC8Nplus2MAPE': 13.870279878659744}
```

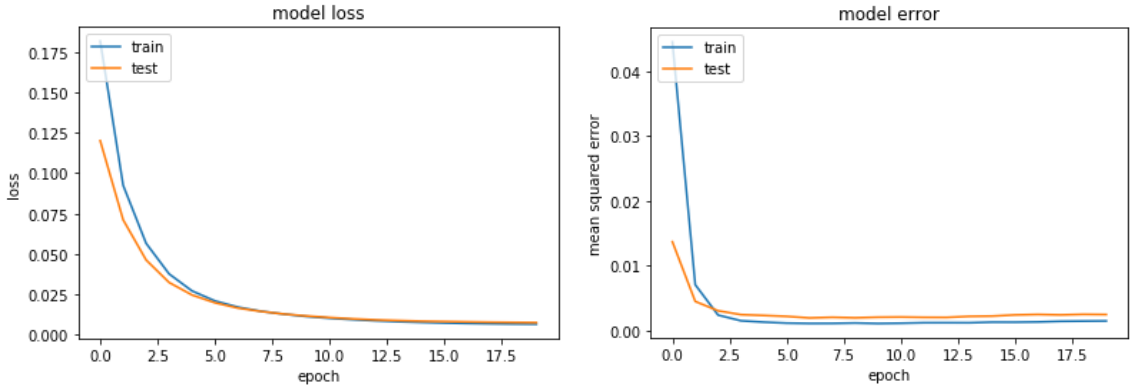
Global model: 1I – Same as 1F, but with only 4 nodes in hidden layer, and learning rata = 0.0005

```
{'DC4Nplus1MAPE': 19.55298009233726,
 'DC4Nplus2MAPE': 23.355019105720416,
 'DC5Nplus1MAPE': 9.841946377153054,
 'DC5Nplus2MAPE': 11.45953018657448,
 'DC7Nplus1MAPE': 17.694744816477023,
 'DC7Nplus2MAPE': 20.32092715873825,
 'DC8Nplus1MAPE': 12.13721493992767,
 'DC8Nplus2MAPE': 13.928699986189748}
```

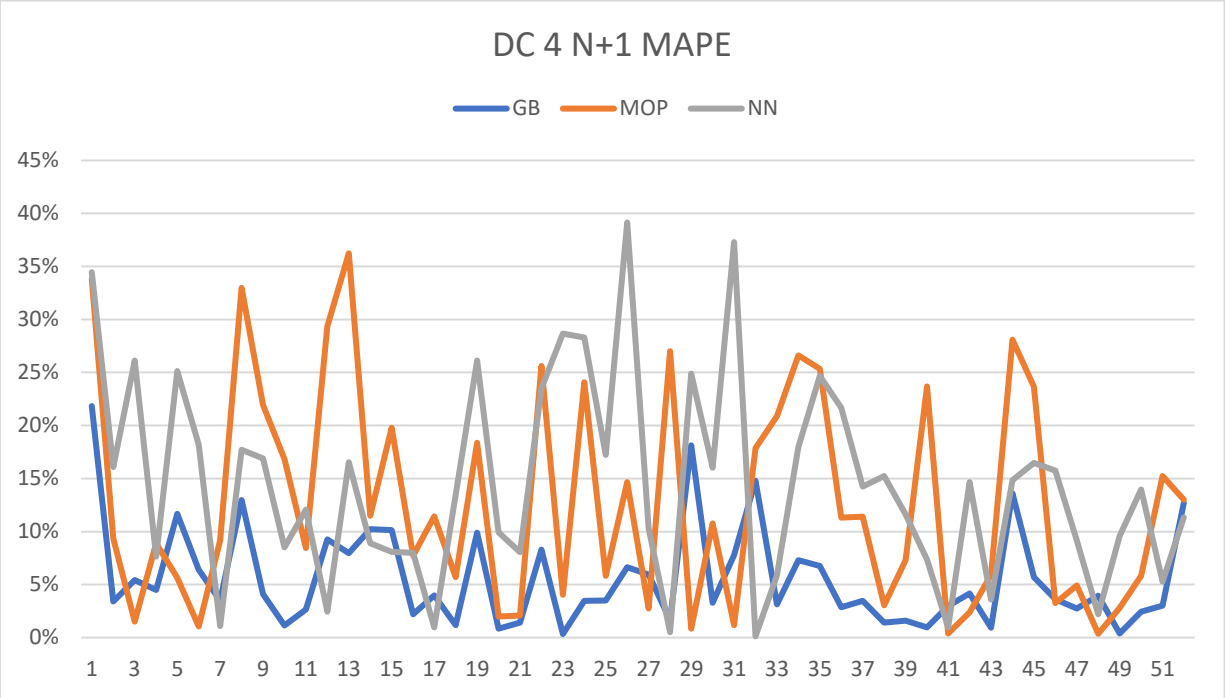


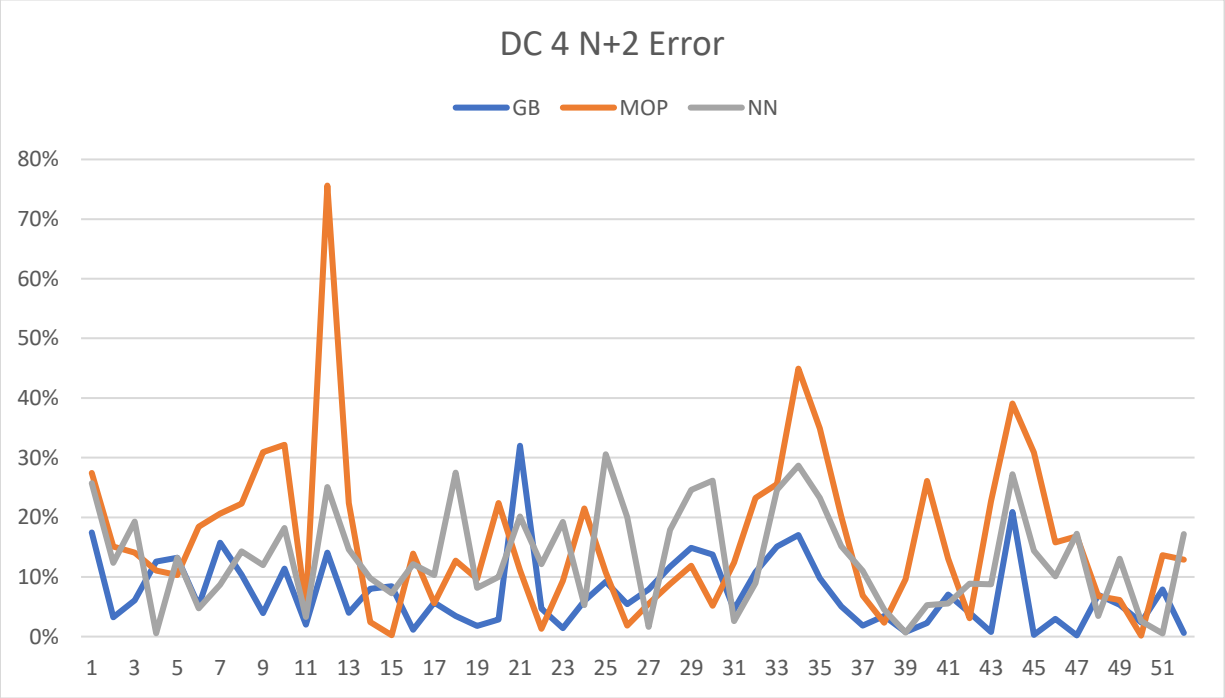
Global model: 1J - Same as 1F but with Adam optimizer

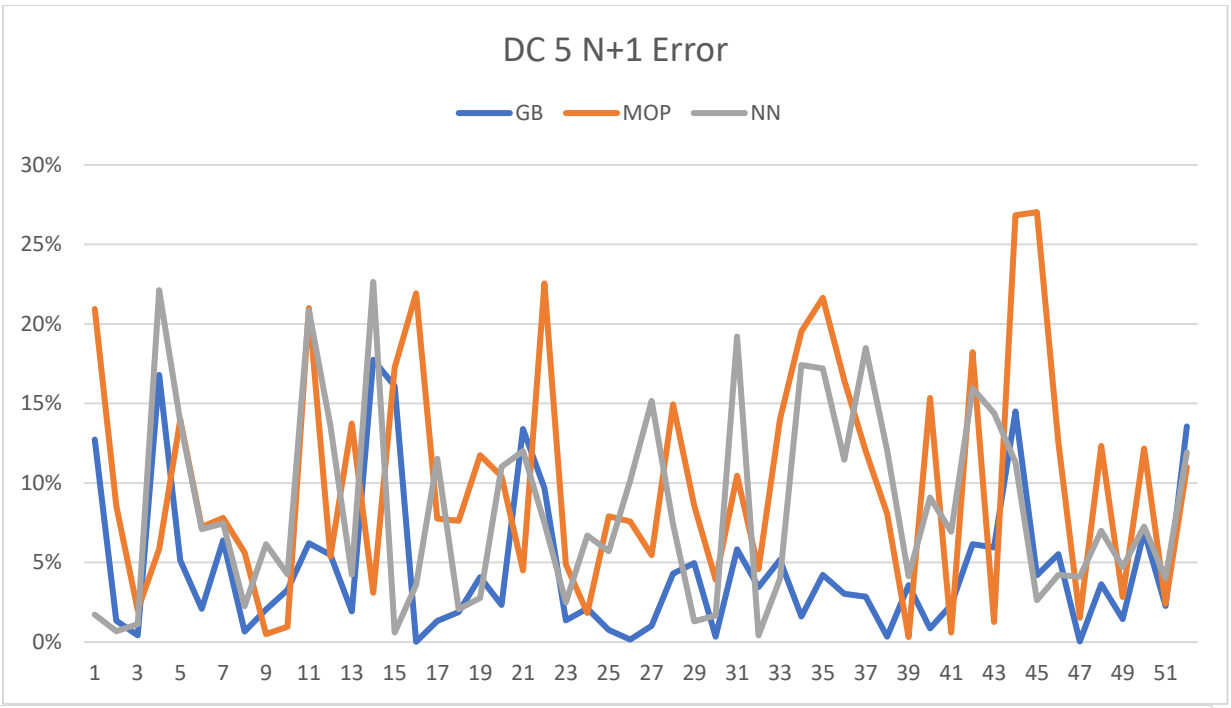
```
{'DC4Nplus1MAPE': 17.375650500838592,
 'DC4Nplus2MAPE': 17.947893849656936,
 'DC5Nplus1MAPE': 8.897933454262812,
 'DC5Nplus2MAPE': 10.570488055823226,
 'DC7Nplus1MAPE': 12.749352625218261,
 'DC7Nplus2MAPE': 14.7907757968298,
 'DC8Nplus1MAPE': 11.071693255299138,
 'DC8Nplus2MAPE': 12.881049989423191}
```

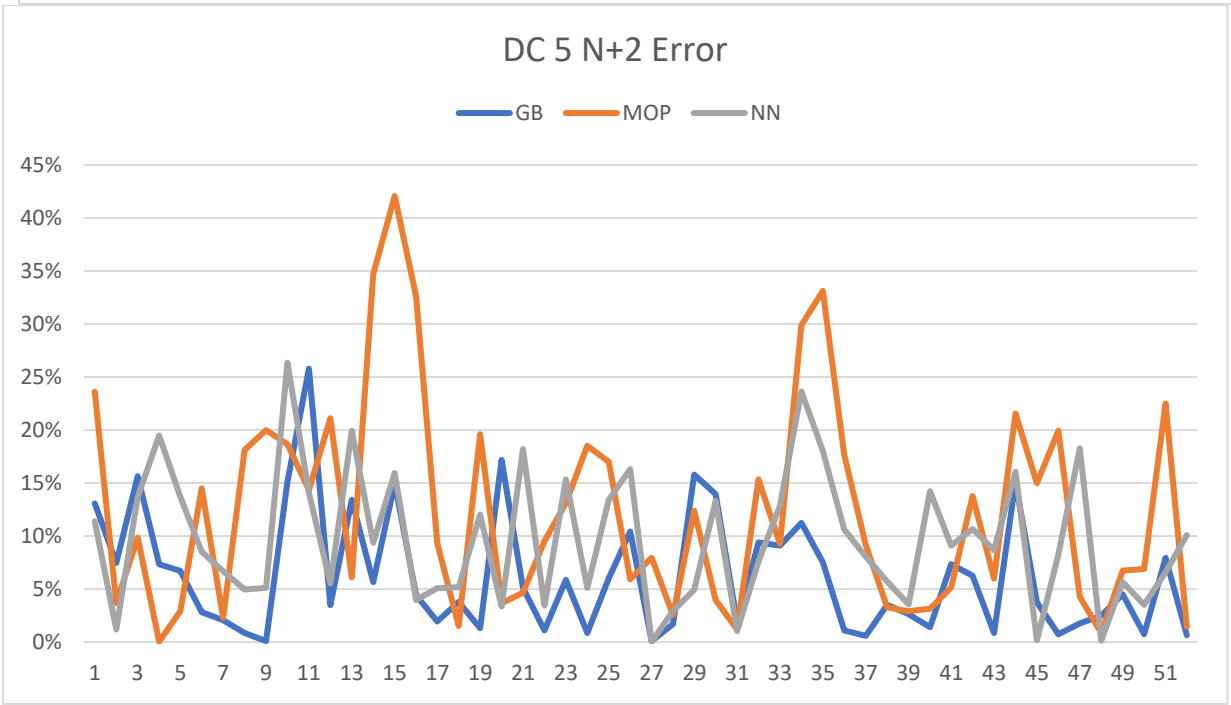


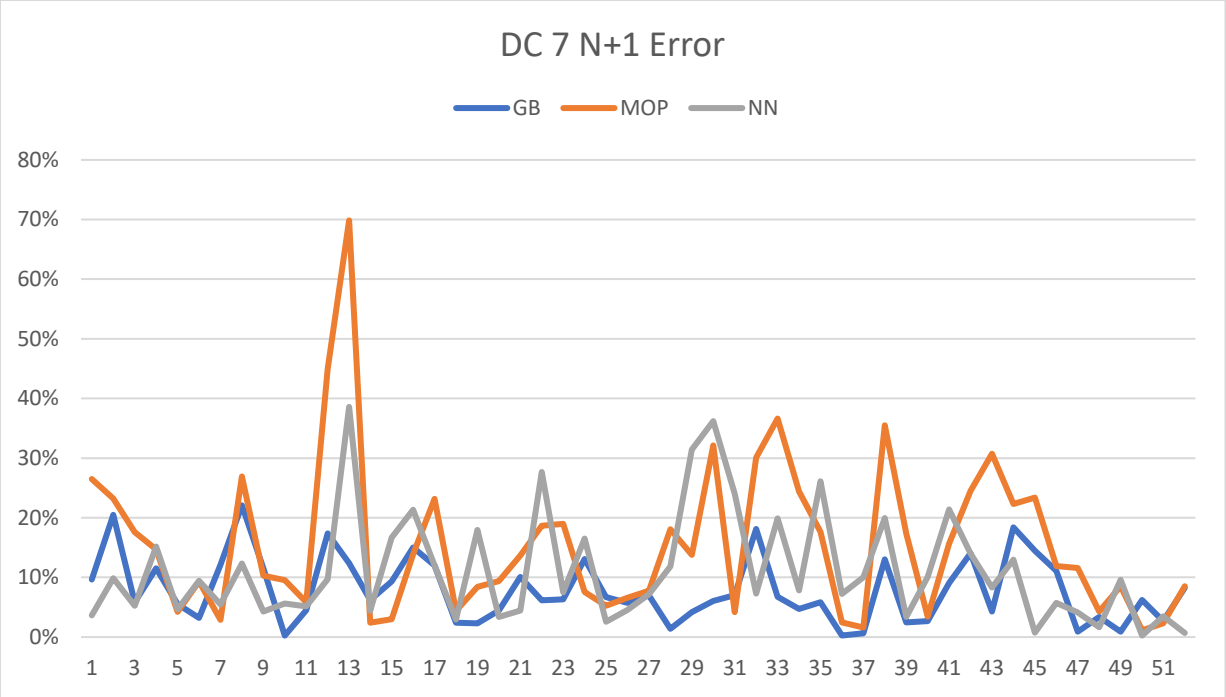
Appendix B – DC Specific MAPE Comparison

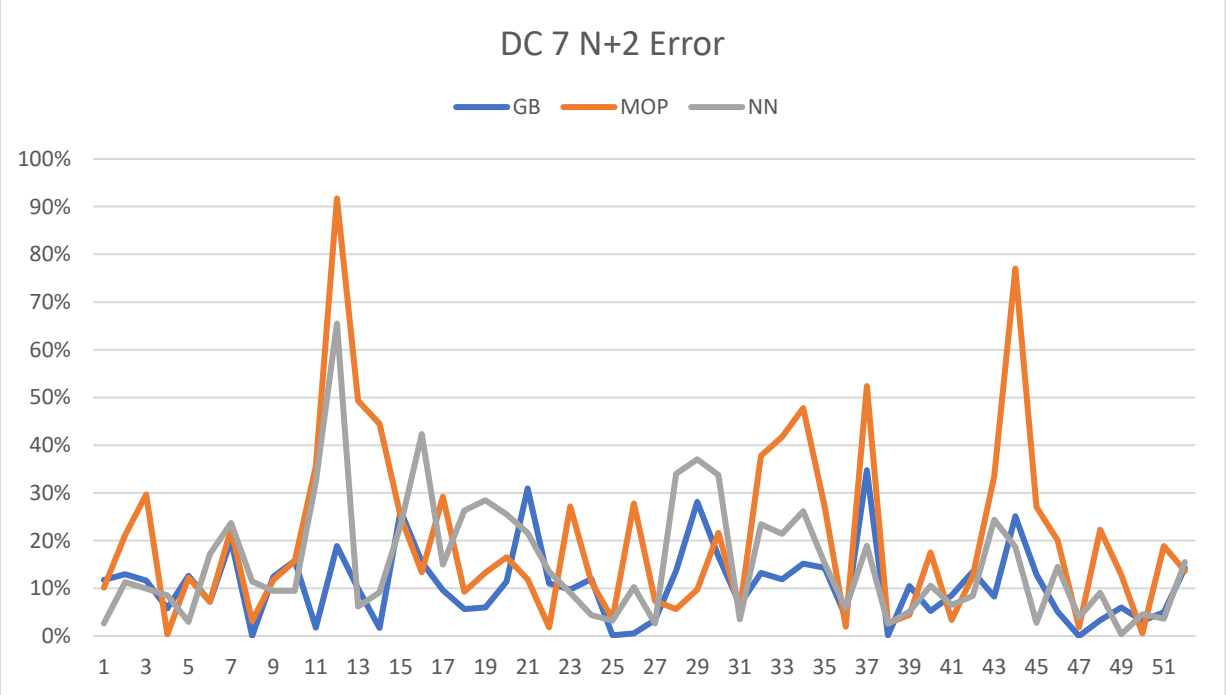


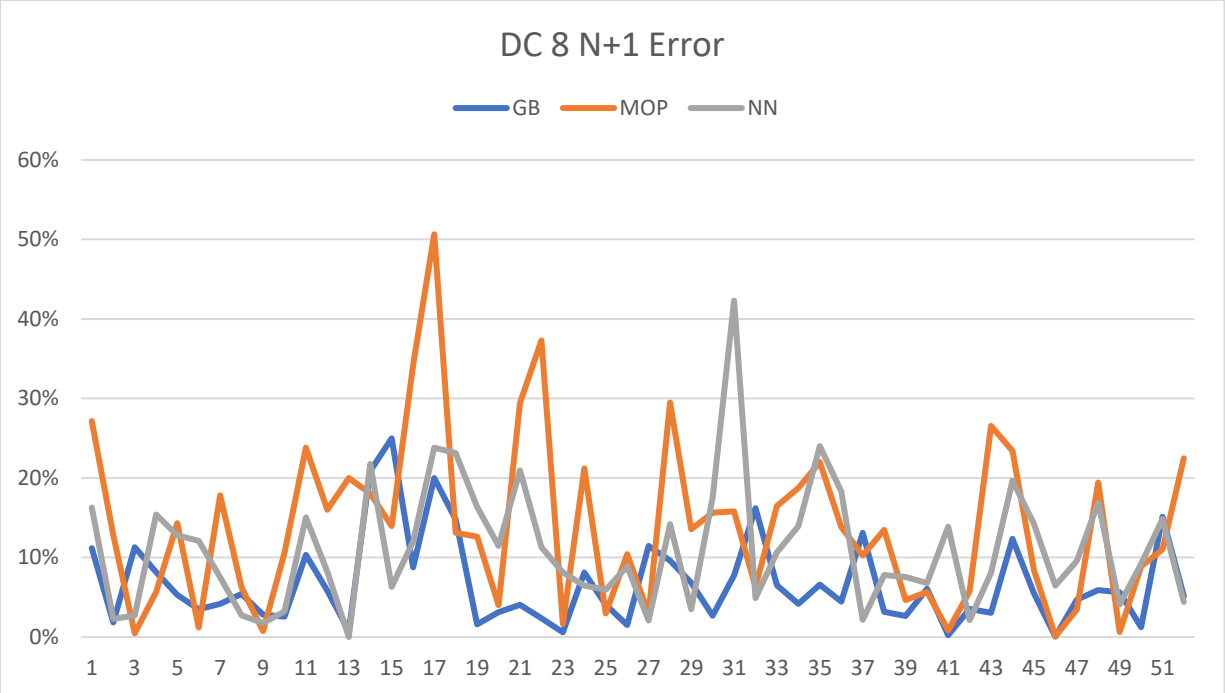


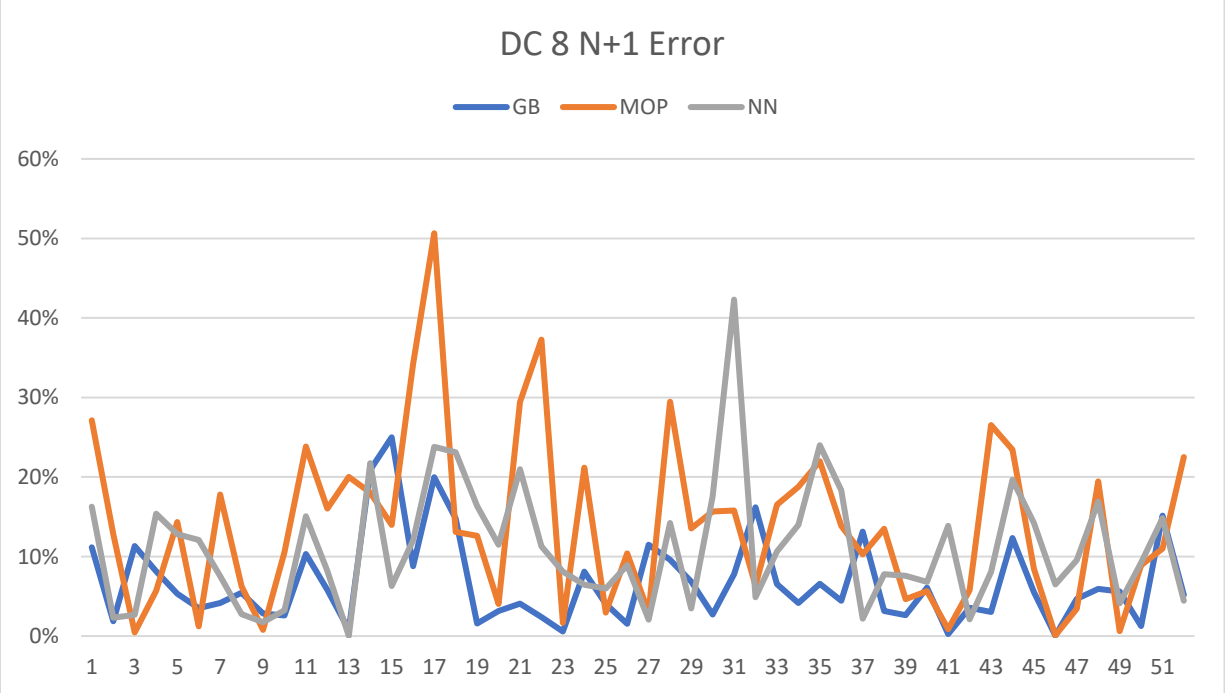

DC 5 N+1 Error
GB
MOP
NN
30%
25%
20%
15%
10%
5%
0%
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51


DC 5 N+2 Error
GB
MOP
NN
45%
40%
35%
30%
25%
20%
15%
10%
5%
0%
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51


DC 7 N+1 Error
GB
MOP
NN
80%
70%
60%
50%
40%
30%
20%
10%
0%
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51


DC 7 N+2 Error
GB
MOP
NN
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51


DC 8 N+1 Error
GB
MOP
NN
60%
50%
40%
30%
20%
10%
0%
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51


DC 8 N+1 Error
GB
MOP
NN
60%
50%
40%
30%
20%
10%
0%
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51