LSDM Project 4 Report

Dataset 1

A Network backup Dataset, which comprises of simulated traffic data on a backup system over a network is taken. The system monitors the files residing in a destination machine and copies their changes in four-hour cycles. At the end of each backup process, the size of the data moved to the destination as well as the duration it took are logged, to be used for developing prediction models. We define a workflow as a task that backs up data from a group of files, which have similar patterns of change in terms of size over time. The dataset has around 18000 data points with the following columns/variables:

* Week index
* Day of the week at which the file back-up has started
* Backup start time: Hour of the day
* Workflow ID
* File name
* Backup size: the size of the file that is backed up in that cycle in GB
* Backup time: the duration of the backup procedure in hour

We load the dataset in csv format into a pandas data-frame and analyse it.

1. (a)  For the first twenty-day period (x-axis unit is day number) plot the backup sizes for all workflows (color coded on the y-axis).

The backup sizes are grouped by workflow as well as the different days and the resultant plot looks like this:



(b)  Do the same plot for the first 105-day period.



(c)  Can you identify any repeating patterns?

After plotting it over all the days we see that there is a pattern and the backup sizes repeat every 7 days. This implies that there is a similarity and close correlation between the backup sizes for every week for each workflow.

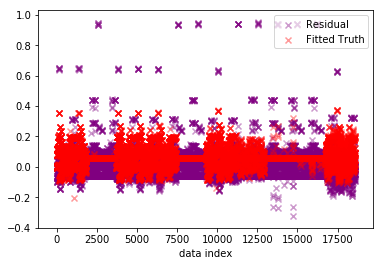
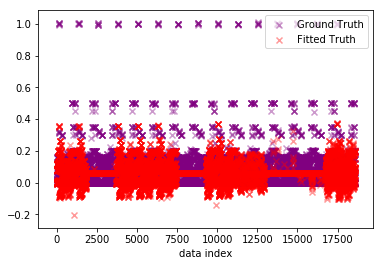
2)(c) Now use a neural network regression model (one hidden layer) with all features one-hot encoded.

We use a Neural network regressor with varying number of Parameters such as:

• Number of hidden units between 2, 5, 10, 50, 100, 150, 200, 300,400,500,600.

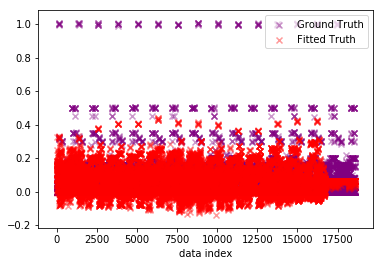
• Activation function ‘relu’, ‘logistic’ and ‘tanh’.

The train and test RMSE values are as follows::



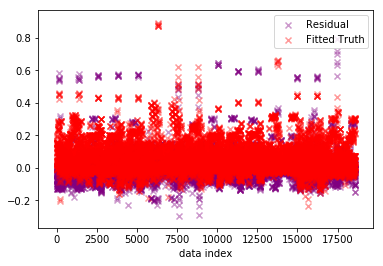
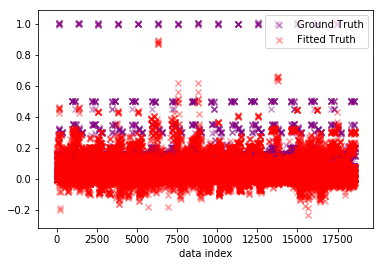
Train RMSE for 2 Hidden units and Activation: relu is = 0.08841820675813686

Test RMSE for 2 Hidden units and Activation: relu is = 0.08966545008477501



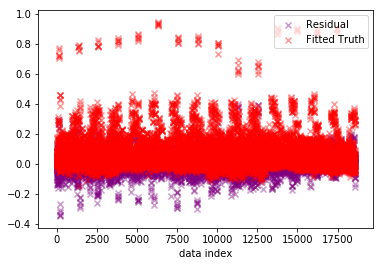
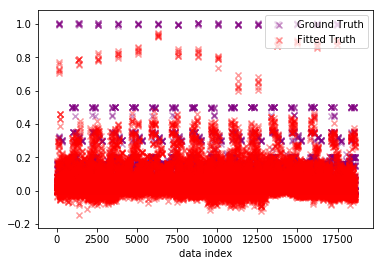
Train RMSE for 5 Hidden units and Activation: relu is = 0.07395021276068021

Test RMSE for 5 Hidden units and Activation: relu is = 0.07428355351162526



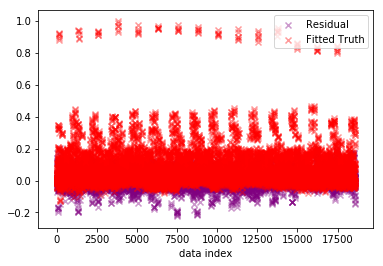
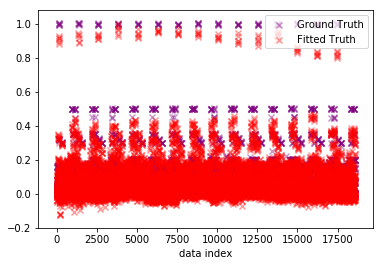
Train RMSE for 10 Hidden units and Activation: relu is = 0.06441313996531813

Test RMSE for 10 Hidden units and Activation: relu is = 0.06514280980240812



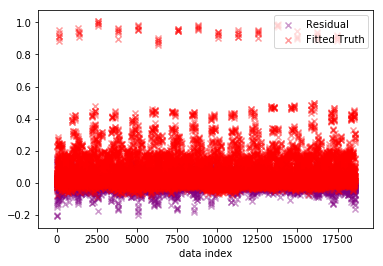
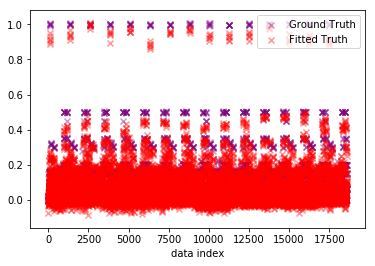
Train RMSE for 50 Hidden units and Activation: relu is = 0.03978897567782655

Test RMSE for 50 Hidden units and Activation: relu is = 0.04474060227356015



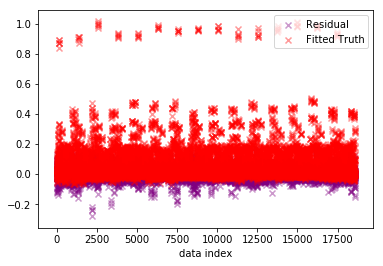
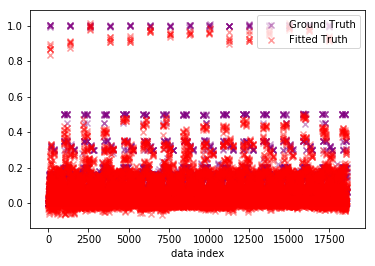
Train RMSE for 100 Hidden units and Activation: relu is = 0.029570327485074224

Test RMSE for 100 Hidden units and Activation: relu is = 0.03311019750392102



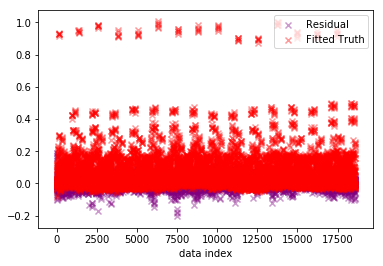
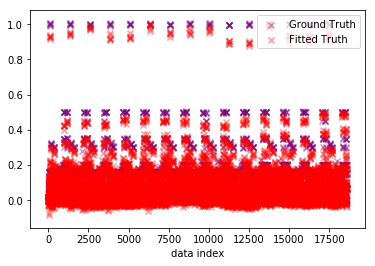
Train RMSE for 150 Hidden units and Activation: relu is = 0.024625735295600106

Test RMSE for 150 Hidden units and Activation: relu is = 0.028147194695741885



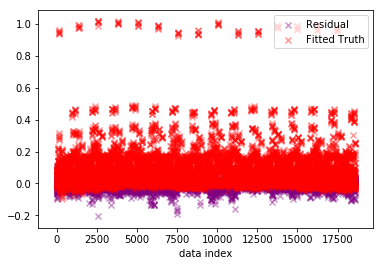
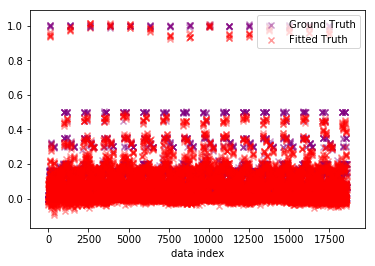
Train RMSE for 200 Hidden units and Activation: relu is = 0.020870801808297177

Test RMSE for 200 Hidden units and Activation: relu is = 0.025542429699539633



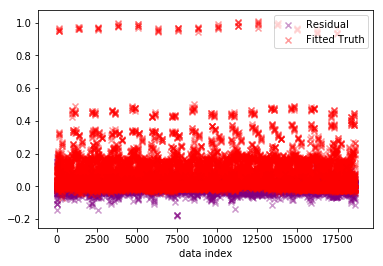
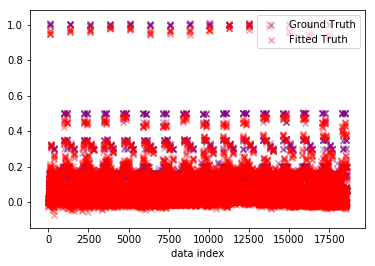
Train RMSE for 300 Hidden units and Activation: relu is = 0.019145436962833995

Test RMSE for 300 Hidden units and Activation: relu is = 0.02316916030857981



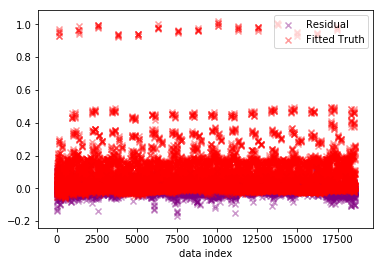
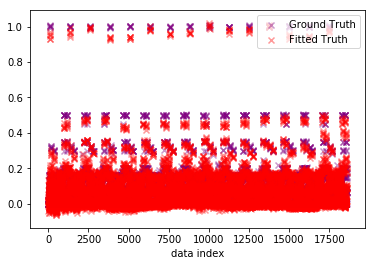
Train RMSE for 400 Hidden units and Activation: relu is = 0.018203174961728064

Test RMSE for 400 Hidden units and Activation: relu is = 0.022343818242190727



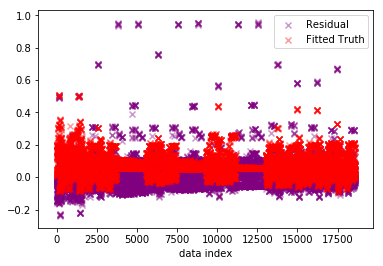
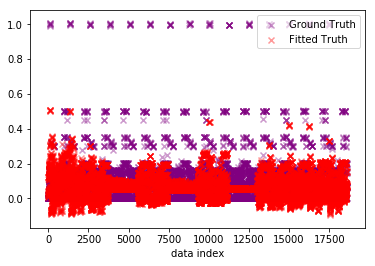
Train RMSE for 500 Hidden units and Activation: relu is = 0.015921106721378454

Test RMSE for 500 Hidden units and Activation: relu is = 0.01994274947484271



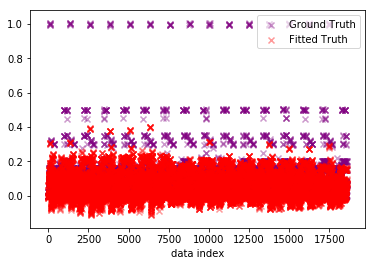
Train RMSE for 600 Hidden units and Activation: relu is = 0.017485400279973883

Test RMSE for 600 Hidden units and Activation: relu is = 0.02011599561065632



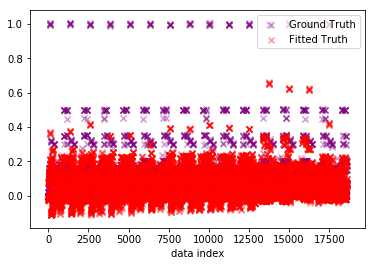
Train RMSE for 2 Hidden units and Activation: logistic is = 0.08129363332344416

Test RMSE for 2 Hidden units and Activation: logistic is = 0.08181500444787941



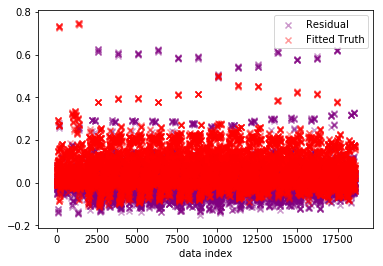
Train RMSE for 5 Hidden units and Activation: logistic is = 0.07124272219378454

Test RMSE for 5 Hidden units and Activation: logistic is = 0.07196298431599246



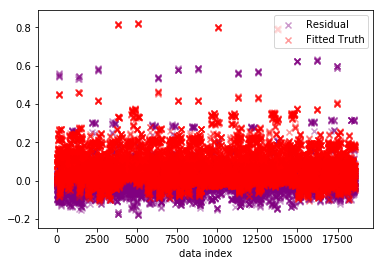
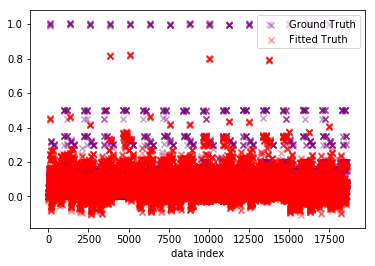
Train RMSE for 10 Hidden units and Activation: logistic is = 0.06331162459036646

Test RMSE for 10 Hidden units and Activation: logistic is = 0.06410434814447791



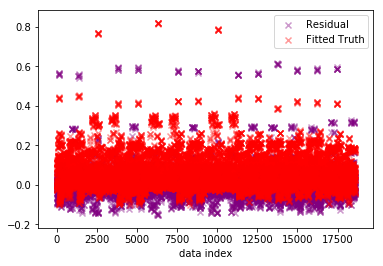
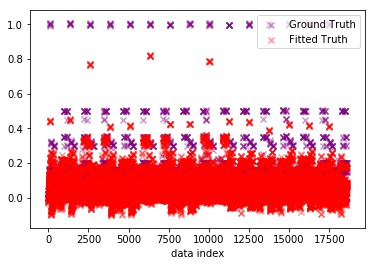
Train RMSE for 50 Hidden units and Activation: logistic is = 0.06395189947215296

Test RMSE for 50 Hidden units and Activation: logistic is = 0.063789293521998



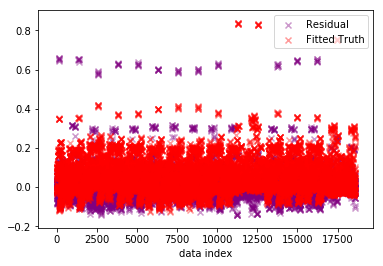
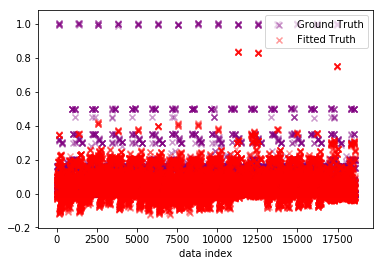
Train RMSE for 100 Hidden units and Activation: logistic is = 0.06020934634664528

Test RMSE for 100 Hidden units and Activation: logistic is = 0.060328617318963385



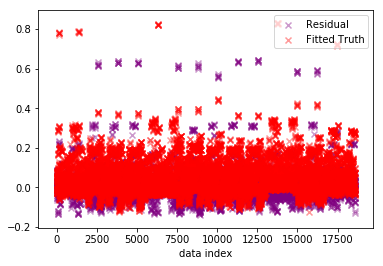
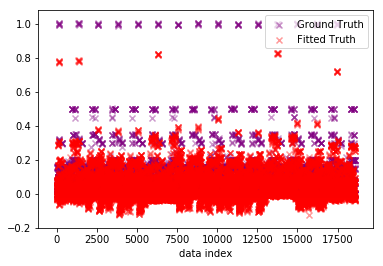
Train RMSE for 150 Hidden units and Activation: logistic is = 0.058490590401386426

Test RMSE for 150 Hidden units and Activation: logistic is = 0.05945787334048091



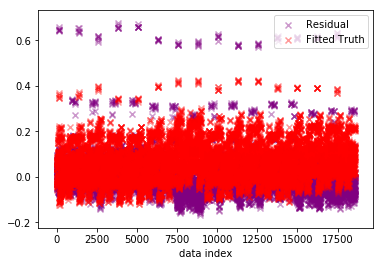
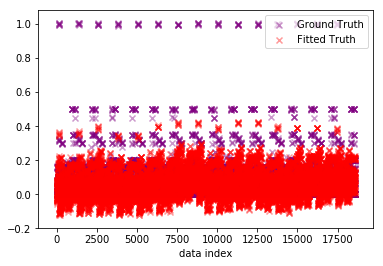
Train RMSE for 200 Hidden units and Activation: logistic is = 0.06401322406584897

Test RMSE for 200 Hidden units and Activation: logistic is = 0.06432432696257852



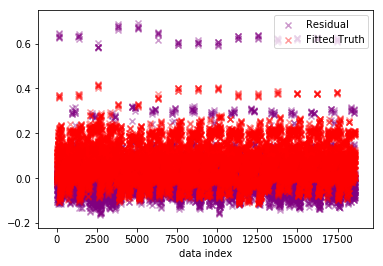
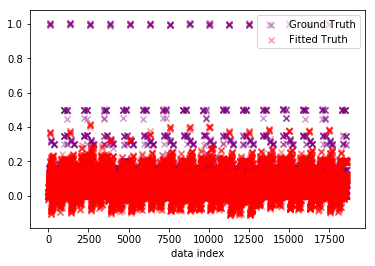
Train RMSE for 300 Hidden units and Activation: logistic is = 0.05825886675506815

Test RMSE for 300 Hidden units and Activation: logistic is = 0.05940629904072118



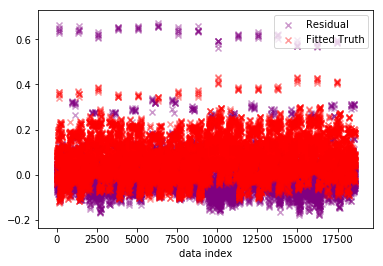
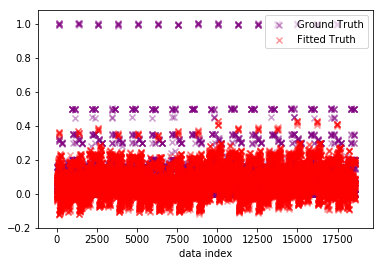
Train RMSE for 400 Hidden units and Activation: logistic is = 0.07223729117712266

Test RMSE for 400 Hidden units and Activation: logistic is = 0.07230515012040861



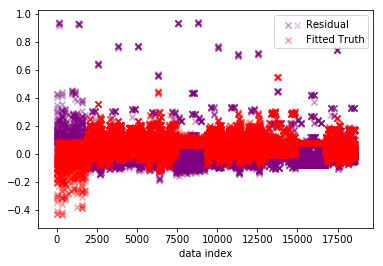
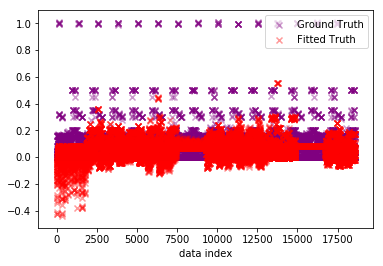
Train RMSE for 500 Hidden units and Activation: logistic is = 0.0710000216760998

Test RMSE for 500 Hidden units and Activation: logistic is = 0.07108686735399951



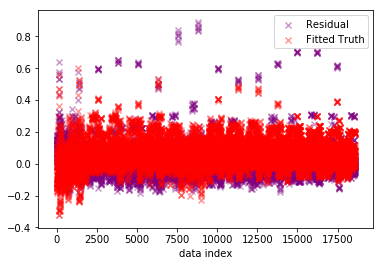
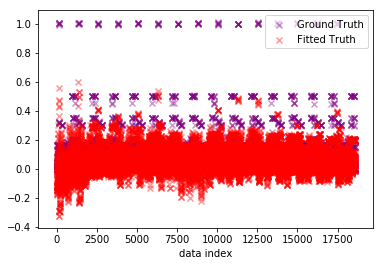
Train RMSE for 600 Hidden units and Activation: logistic is = 0.07295413702694545

Test RMSE for 600 Hidden units and Activation: logistic is = 0.0730056947915052



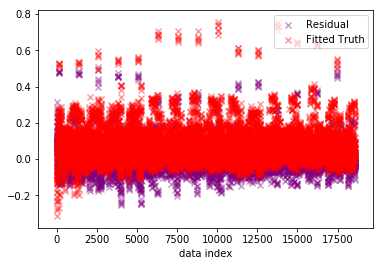
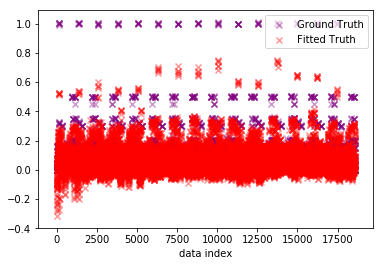
Train RMSE for 2 Hidden units and Activation: tanh is = 0.08384408154139526

Test RMSE for 2 Hidden units and Activation: tanh is = 0.08677019825291143



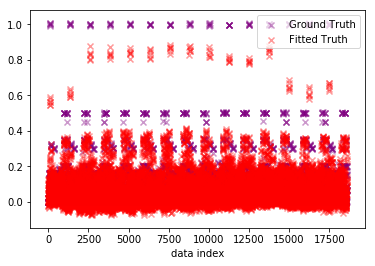
Train RMSE for 5 Hidden units and Activation: tanh is = 0.07254169423883083

Test RMSE for 5 Hidden units and Activation: tanh is = 0.07548201366490202



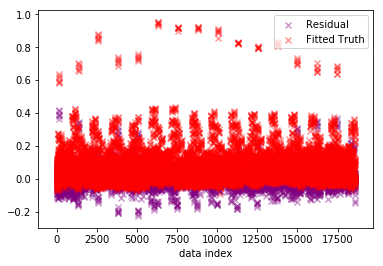
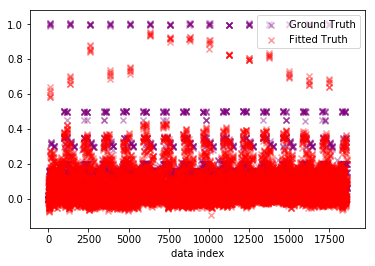
Train RMSE for 10 Hidden units and Activation: tanh is = 0.05289582551686063

Test RMSE for 10 Hidden units and Activation: tanh is = 0.05621955219771206



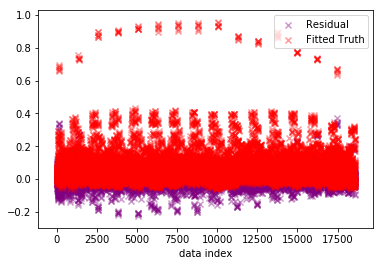
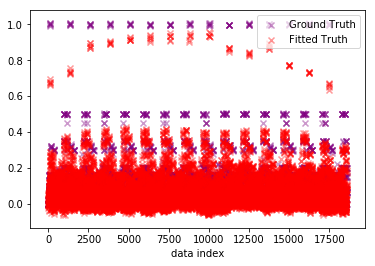
Train RMSE for 50 Hidden units and Activation: tanh is = 0.04015225549561528

Test RMSE for 50 Hidden units and Activation: tanh is = 0.04350508048443337



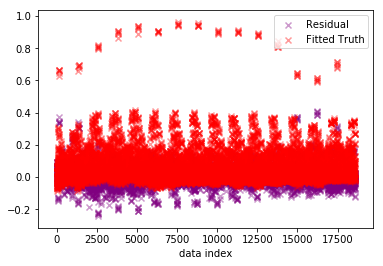
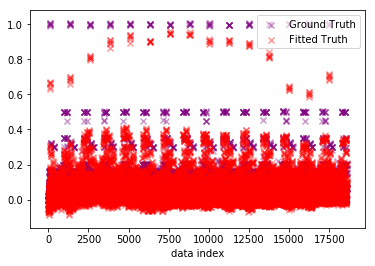
Train RMSE for 100 Hidden units and Activation: tanh is = 0.03516152109154982

Test RMSE for 100 Hidden units and Activation: tanh is = 0.039019092933521



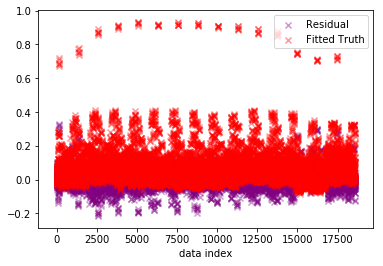
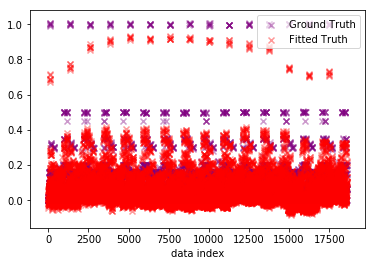
Train RMSE for 150 Hidden units and Activation: tanh is = 0.033334197351370796

Test RMSE for 150 Hidden units and Activation: tanh is = 0.03644276370583232



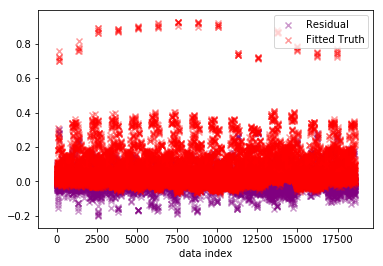
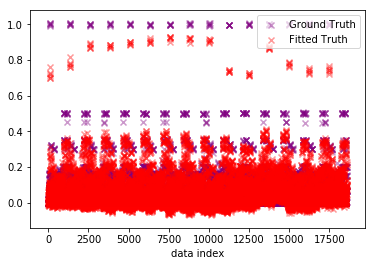
Train RMSE for 200 Hidden units and Activation: tanh is = 0.037627751351045534

Test RMSE for 200 Hidden units and Activation: tanh is = 0.03909618707659477



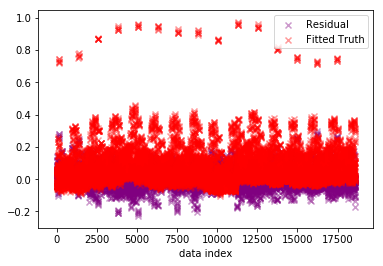
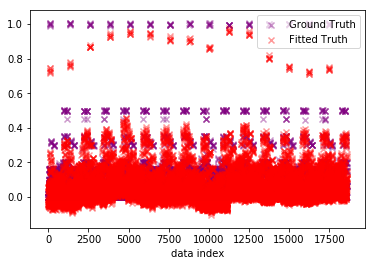
Train RMSE for 300 Hidden units and Activation: tanh is = 0.03633941199984668

Test RMSE for 300 Hidden units and Activation: tanh is = 0.03836566965459148



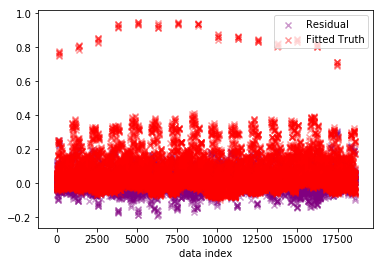
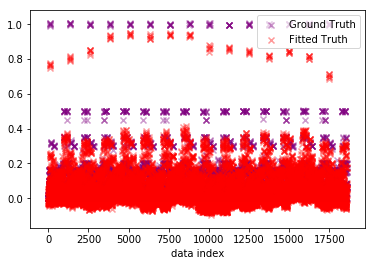
Train RMSE for 400 Hidden units and Activation: tanh is = 0.03690641486271769

Test RMSE for 400 Hidden units and Activation: tanh is = 0.038025863046630054



Train RMSE for 500 Hidden units and Activation: tanh is = 0.043550648183044015

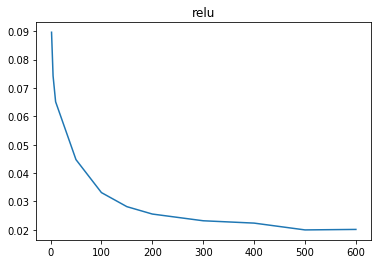
Test RMSE for 500 Hidden units and Activation: tanh is = 0.04271434931253474

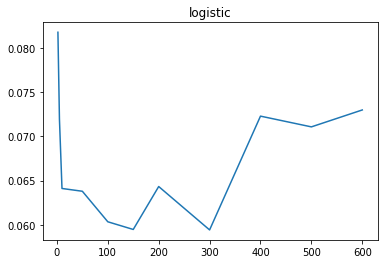
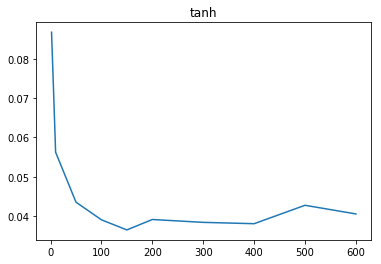


Train RMSE for 600 Hidden units and Activation: tanh is = 0.04034371559907949

Test RMSE for 600 Hidden units and Activation: tanh is = 0.04050965039850504

The test-RMSE vs the number of hidden units for each activation function is plotted as shown below.



The best combination is achieved using Relu with higher number of Hidden units ( stays almost same between 500-600 hidden layers).

(d) Predict the Backup size for each of the workflows separately.

1. Using linear regression model. Explain if the fit is improved?

The backup sizes for each of the workflows separately are predicted using Linear with the following train and test RMSE:

work\_flow\_0: RMSE\_train = 0.033652 RMSE\_test = 0.033992

work\_flow\_1: RMSE\_train = 0.101801 RMSE\_test = 0.102025

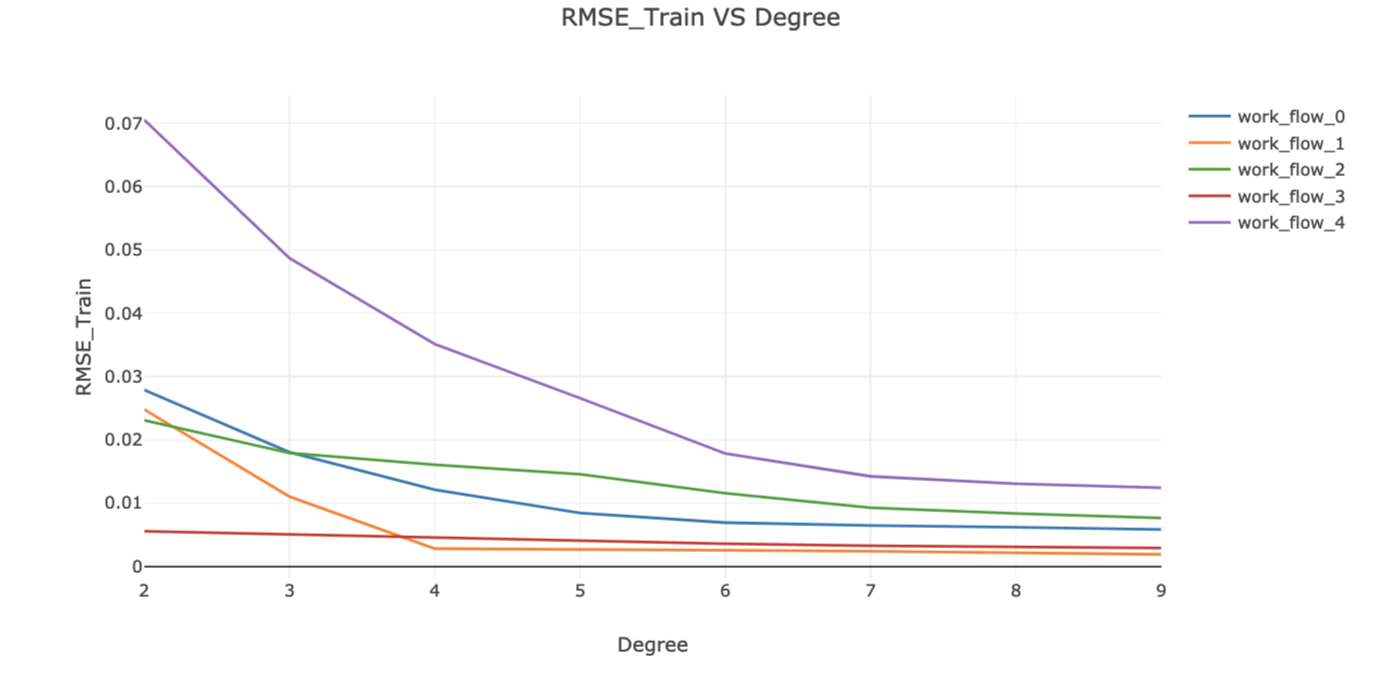
work\_flow\_2: RMSE\_train = 0.025513 RMSE\_test = 0.025605

work\_flow\_3: RMSE\_train = 0.005846 RMSE\_test = 0.005863

work\_flow\_4: RMSE\_train = 0.101027 RMSE\_test = 0.101600

1. Try fitting a more complex regression function to your data. You can try a polynomial function of your variables. Try increasing the degree of the polynomial to improve your fit. Again, use a 10 fold cross validation to evaluate your results. Plot the average train and test RMSE of the trained model against the degree of the polynomial you use. Can you find a threshold on the degree of the fitted polynomial beyond which the generalization error of your model gets worse? Can you explain how cross validation helps controlling the complexity of your model?

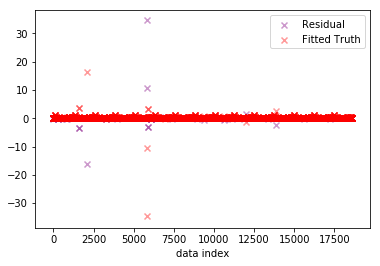
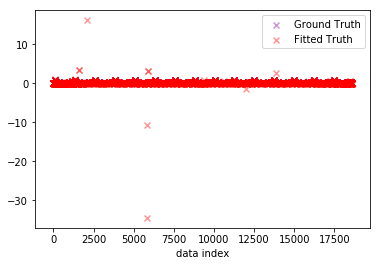
We calculate the train and test RMSE for linear regression using different degrees of polynomial ranging from 2 to 9. On plotting them together, for each workflow separately, we see the following results:



We see that the threshold for the polynomial is at 7 since after that our model does not show any change.

The test RMSE is also plotted for all the different degrees of polynomial.

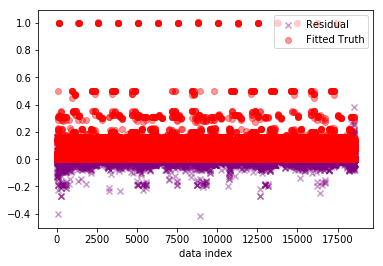
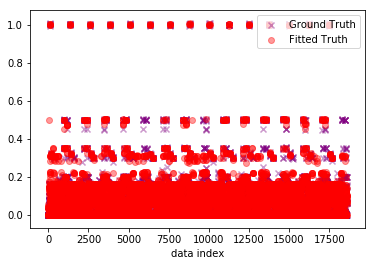
We use cross 10 fold cross-validation to fit our model with polynomial degree=7. The Ground truth vs fitted truth and ground truth vs residual are plotted as shown below.



Using a higher degree polynomial may lead to the problem of overfitting since the regressor learns the data “too well” as the variance of the estimator increases.To prevent that from happening we use 10 fold cross validation. This helps in controlling the complexity of the model and avoids the overfitting problem.

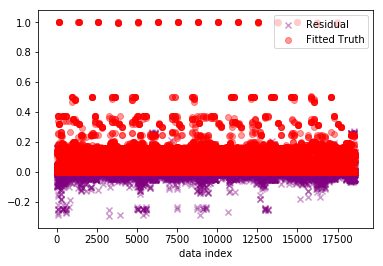
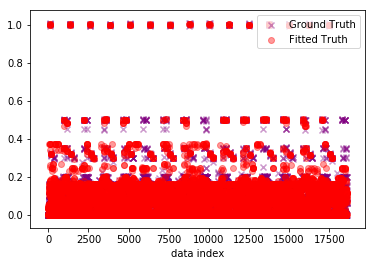
(e) Use k-nearest neighbor regression and find the best parameter.

We perform regression using the k nearest neighbor algorithm and try different values for k between 2-20 to see which gives us the best results. The values for Train and test RMSE are as follows:



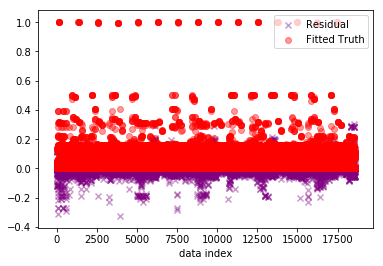
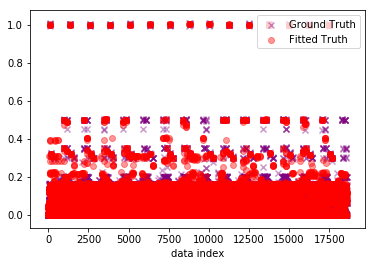
Train RMSE for 2 neighbours = 0.009748076291782953

Test RMSE 0.029600278177349778



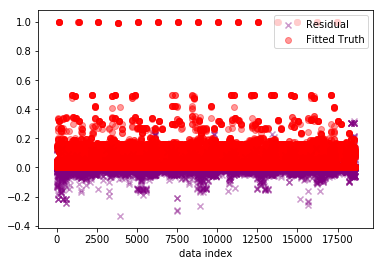
Train RMSE for 3 neighbours = 0.01247542330185669

Test RMSE 0.03231040379935868



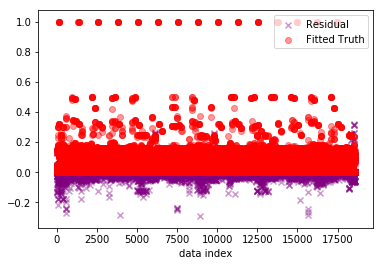
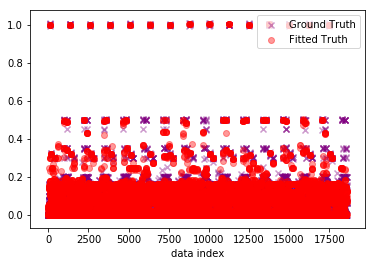
Train RMSE for 4 neighbours = 0.013970989877003091

Test RMSE 0.03152996229099001



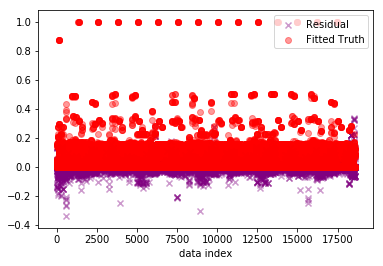
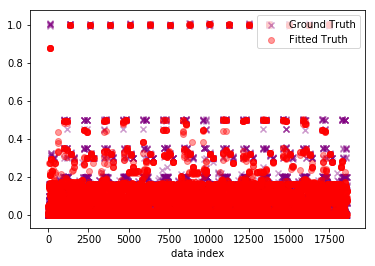
Train RMSE for 5 neighbours = 0.014389906251551485

Test RMSE 0.02976856605197938



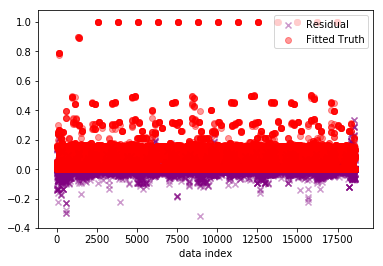
Train RMSE for 6 neighbours = 0.014138263062027584

Test RMSE 0.02819073268857089



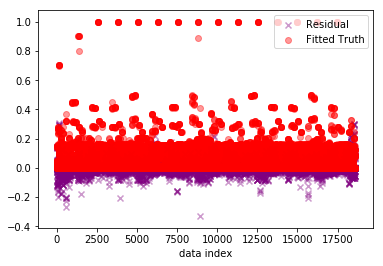
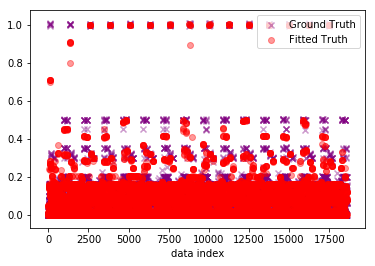
Train RMSE for 7 neighbours = 0.013924596210132053

Test RMSE 0.02716008946626608



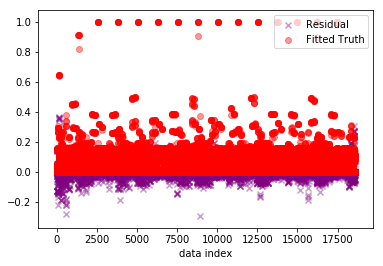
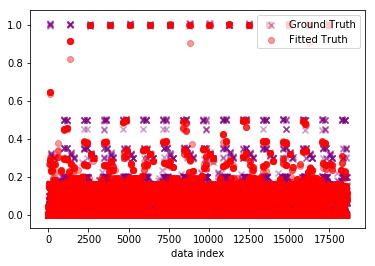
Train RMSE for 8 neighbours = 0.01404897680524547

Test RMSE 0.02688074794528538



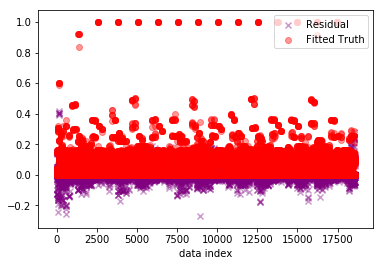
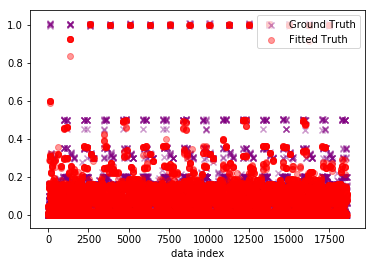
Train RMSE for 9 neighbours = 0.014606768106967784

Test RMSE 0.02741537965544601



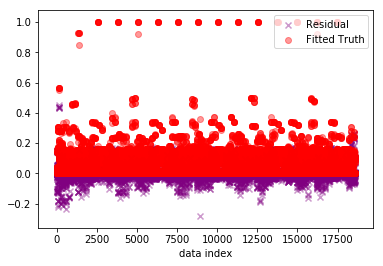
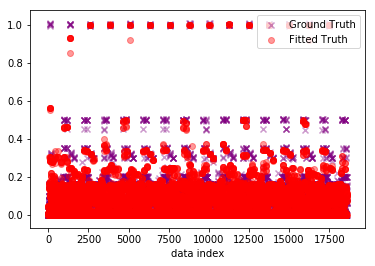
Train RMSE for 10 neighbours = 0.015598095784047753

Test RMSE 0.0287919553302838



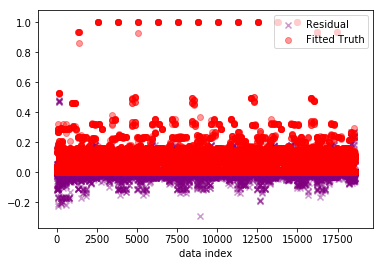
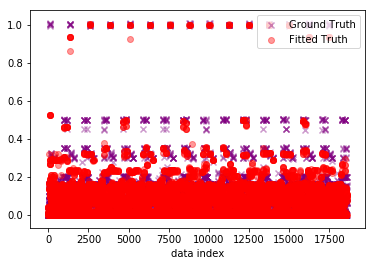
Train RMSE for 11 neighbours = 0.016786644891622453

Test RMSE 0.03020264925764733



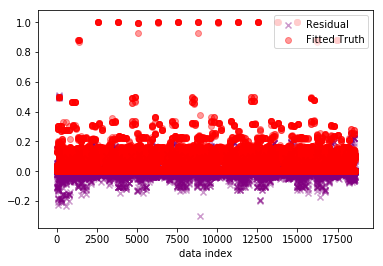
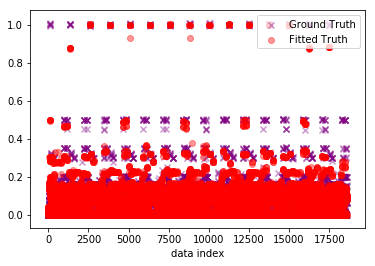
Train RMSE for 12 neighbours = 0.01769035993907844

Test RMSE 0.03180150905484143



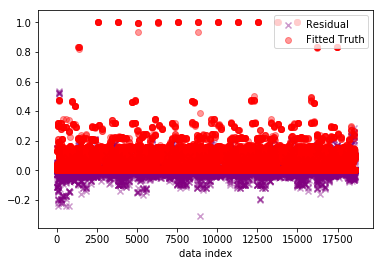
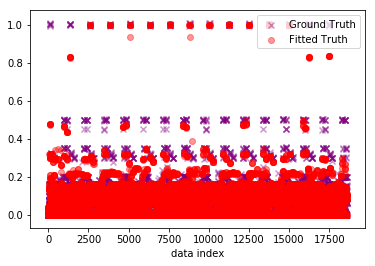
Train RMSE for 13 neighbours = 0.018406657099892268

Test RMSE 0.03314988428446555



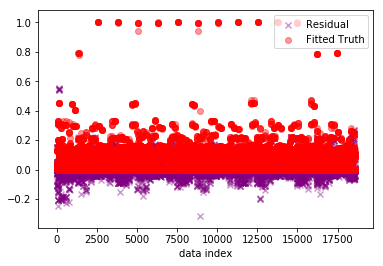
Train RMSE for 14 neighbours = 0.018889848655368486

Test RMSE 0.03412053758530458



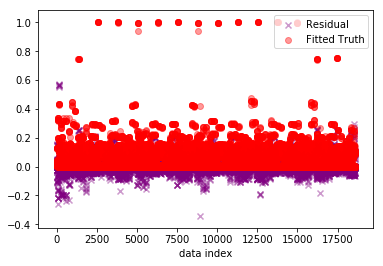
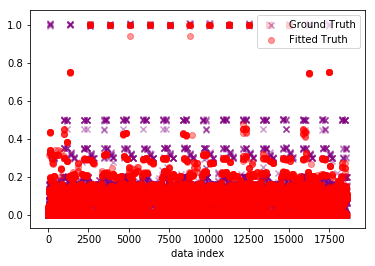
Train RMSE for 15 neighbours = 0.01938564607118386

Test RMSE 0.03523866991795542



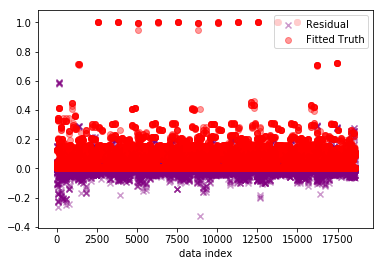
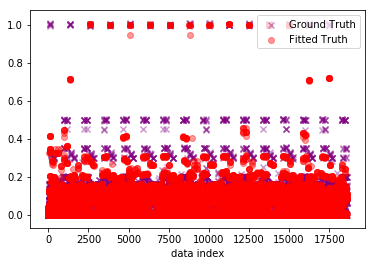
Train RMSE for 16 neighbours = 0.019957930152601736

Test RMSE 0.03606861473277587



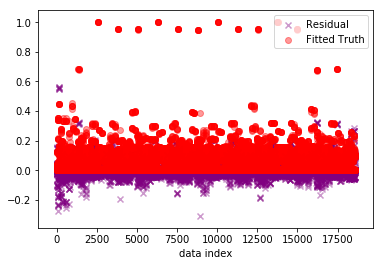
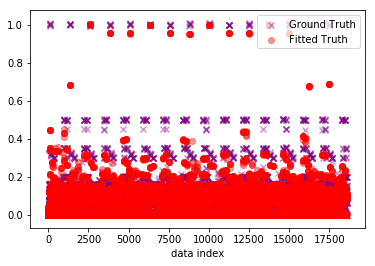
Train RMSE for 17 neighbours = 0.020644355824624067

Test RMSE 0.03675349188264677



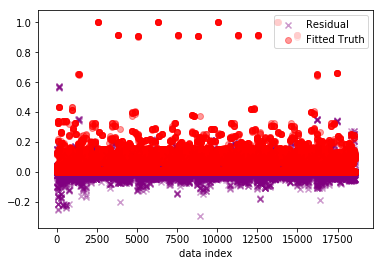
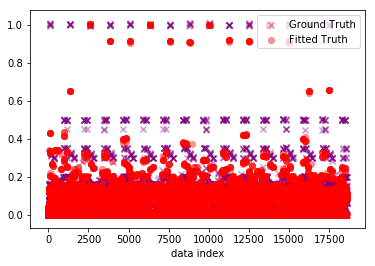
Train RMSE for 18 neighbours = 0.02140477529275484

Test RMSE 0.03754348848235362



Train RMSE for 19 neighbours = 0.022087493364981926

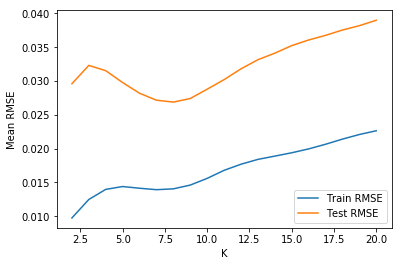
Test RMSE 0.03818646171496168



Train RMSE for 20 neighbours = 0.022640725819767043

Test RMSE 0.03898975461461182

Plotting the mean RMSE train and test over all K, we get a plot as follows:



The lowest value of RMSE is obtained for k=2.