To produce a predictive neural network, the ‘deepLearning’ function from the H2O package was used. This allowed the training of numerous networks with different parameter values, therefore allowing for tuning of the parameters. Due to the relatively large range of parameters available for tuning, a random search was carried out rather than a grid search. Parameters were randomly allocated for each candidate model from the following ranges:

Number of hidden layers: 1-4.

This was determined through experimentation on subsets of the overall dataset. Networks with more than 4 hidden layers performed poorly while requiring excessive computation time.

Number of nodes per hidden layer: INSERT REFERENCE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hidden layer | 1 | 2 | 3 | 4 |
| Number of nodes | XX | XX | XX | XX |

Activation function: Rectifier, Rectifier With Dropout.

The rectifier type activation function is typical for regression models. The ‘rectifier with dropout’ activation function was also tried. REFERENCE

Epochs: 10.

Compromise between computation speed and accuracy. REFERENCE

Input dropout ratio: XX

L1: XX

L2: XX

Distribution = Gaussian REFERENCE

Other distributions were tried during the experimentation phase, however, the best performing models invariably used a Gaussian distribution.

Adaptive learning rate = TRUE.

Provided reduced RMSE for given computation time during experimentation on subset of overall dataset.

Following experimentation on a small subset of the data to determine the above ranges for tuning parameters, the full training run was carried out on EC2. Computation time was limited to 4 hours, which resulted in the generation of XX candidate models. This is significantly higher than the number of models generated by the SVM and Random Forest algorithms, however the benefit is offset by the larger parameter search space.