Aim of study

The aim of this study is to create a generic and robust model to predict the runtime of the furnace. So we can determine the heat demand in each household and automatically adjust the temeprature on the thermostat accordingly. In this study we took into account the Ecobee and Weather data and combined them into one source file. A variety of methods and expermeints were used to create and determine the best performing model.

Independent and Dependent Variables

Using all the variables would very time consuming, lead to complex models, and overall is not good pratice. The following variables were chosen based on the previous papers, what has be seen during the EDA process and checking the features importances after fitting a random forest on entirety of the data. These variables are as follows:

1. Independent
    A. In-Out = difference between the indoor and outdoor temperature
    B. Delta = difference between the indoor and setpoint temperature
    C. T_step_heat = heating setpoint temperature
    D. Fan = furnace fan operation (either ON/OFF or minutes of runtime)
    E. Temp (°C) = outdoor temperature
    F. Thermostat_Temperature = indoor temperature
    G. Month = Month of the year
    H. Day = day of the week
    I. Hour = hour of day in 24 hour format
2. Dependent
    A. auxHeat1 = furnace run-time in seconds within the 5-minutes interval

Approaches Considered

Two main approaches were implemented. The first method being a neural network and it's variation along with regression models using Linear Regression, the Generalized Linear Model (GLM) and a Random Forest. In both cases the model were trained and tested with and without date time variables such as 'Month', 'Day, and 'Hour'. In the case of the neural neworks an ANN

and LSTM model were considered. Furthermore, the use of orbit, a library created by Uber for timer series data was also[1]

Model Performance Metrics

The most common measures in regression models are MSE, MAE, RMSE, and R-Squared. In mathematical statistics, the term "mean square error" (MSE) refers to the expected value of the square of the difference between the parameter estimate and the parameter value. The "average error" can be more easily measured using MSE. The extent of data change can be assessed using MSE. RMSE is the square root of MSE as well. The more accurately the prediction model describes the experimental data, the lower the values of MSE, MAE, and RMSE, and the higher the value of R-Squared.

Experimental Design

Data Cleaning/Pre-Processing

The majority of the data was cleaned and processed during the Exploratory data analysis phase. The only pre-processing that had to be done here was creating the ‘In-Out’ variable which is the difference between the indoor and outdoor temperature.

Other than that the data was standardized as all the features had a variety of ranges of values. All independent variables are scaled to have unit variance and a mean of zero.

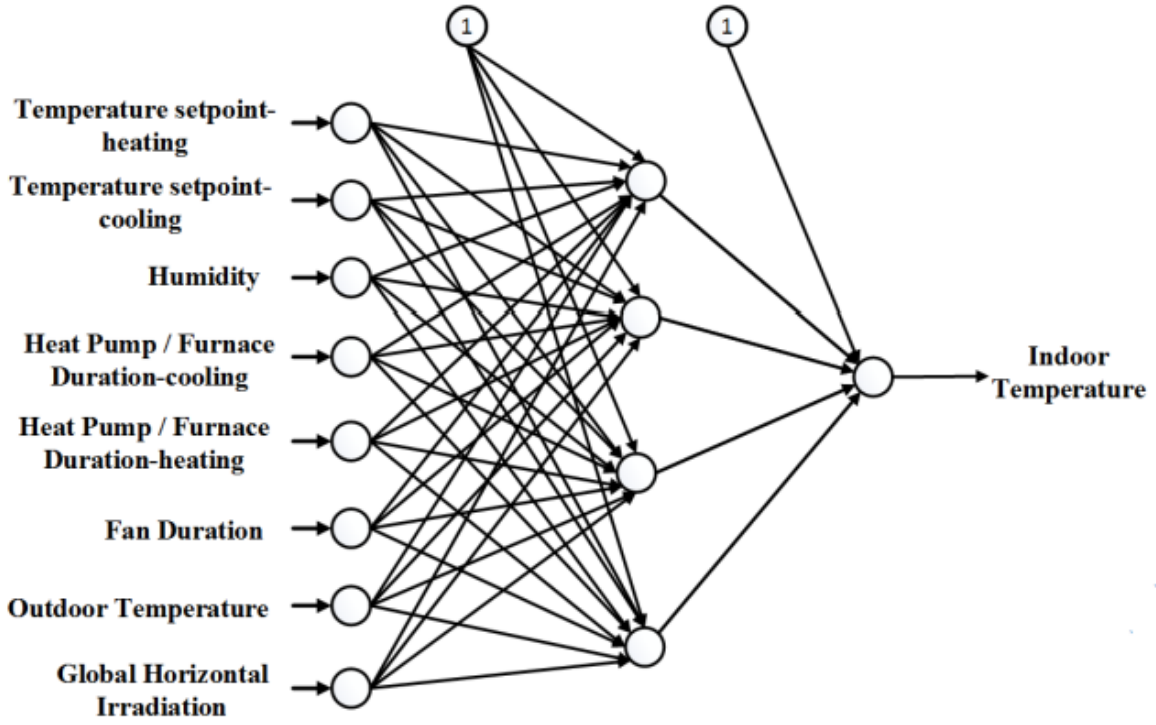
Training & Testing

The split used for training and testing was 80/20 80% (0.8) of the data was used for training the model while 20% (0.2) was used for testing the model.

ANN Architecture

The first verision of the ANN model followed the the Paper: "Predicting indoor temperature from smart thermostat and weather forecast data" which had 8 inputs and 1 hidden layer with 4 nuerons.

[1] Note that these models were considered and not all of them worked out during implementation due to whatever the case may be.



The only difference is that here 9 inputs are used instead of 8 as mentioned in the Independent and Dependent Variables section.

The second version of the ANN was a bit more complex having an input layer of 16 neurons and 2 hidden payers with 8 and 4 neurons respectively.

```
Model: "sequential_9"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_26 (Dense)            (None, 16)                160

 dense_27 (Dense)            (None, 8)                 136

 dense_28 (Dense)            (None, 4)                 36

 dense_29 (Dense)            (None, 1)                 5

=================================================================
Total params: 337
Trainable params: 337
Non-trainable params: 0
_________________________________________________________________
None
```

Hidden layers have ‘Relu’ as the activation function while the output neuron has a ‘linear’ activation function since this is a regression model.

Regression

A simple linear regression model was used. Which by using scikit learn library was easy to implement. The data was split using a random state of 42.

Random Forest

Also implemented via scikit learn library with 100 estimators (decision trees) and the data was split using a random state of 12.

Generalized Linear Model

The data was split using a random state of 12 and based on the range of values of our target variable and its power (0,1) the Tweedie (distrubtution) Regressor was used. That is why Poisson and Gamma Regressors was not used.