Predicting NO$_2$ concentrations

Authors

- **Tessa Clarizio**
  · tessac2

- **Marcel Briguglio**
  · marcelbriguglio

- **Sudheer Salana**
  · sudheersalana1

- **Xiaokai Yang**
  · xiaoky97

- **Grace Fiala**
  · sfiala2

- **Hope Hunter**
  · hhunter3

Abstract

The application of machine learning models to inform environmental quality and change is continuously growing. Machine learning models with varying levels of complexity and interpretability can be used to improve understanding of air quality in regions where monitored data is not available. This research presents a neural network model that uses GIS-derived land use characteristics and modeled satellite data to estimate NO_2 concentrations at monitored sites across the contiguous United States. This neural network model was selected as the highest performing model of six, consisting of four neural network, one multiple regression, and one random forest models. Model performance was evaluated based on the root mean square error of the test predictions, with the best model having a root mean square error of 2.95 (training data mean = 11.83 ppb). Our model shows a relatively high root mean square error; however, most of this error comes from outliers. To further improve model predictions, more data is needed.

1. Introduction

NO_2 is defined by the U.S. Environmental Protection Agency (EPA) as a criteria air pollutant, meaning it poses a risk to human and environmental health. The primary National Ambient Air Quality Standard (NAAQS) for NO_2 is set at a 53 ppb annual average [1]. NO_2 can cause respiratory irritation and can aggravate respiratory diseases such as asthma [2]. NO_2 can also react with other chemicals in the atmosphere to form both particulate matter (PM) and tropospheric ozone (O_3) (US EPA, n.d., B). PM and O_3 are also criteria air pollutants and are harmful to human health. NO_2 also contributes to the formation of acid rain, smog, and nutrient pollution in coastal waters [2]. The primary source of NO2 emissions is fossil fuel combustion, particularly from traffic and power plants [2].

Therefore, understanding and predicting the spatial variability of NO_2 emissions is of great importance to public health. But the prediction of air quality can be complicated due to the number of factors that affect local air quality and the limited availability of monitored air quality data.

Bechle et al (2015) explores the usage of satellite data and GIS derived land use characteristics to better understand NO_2 concentrations and air quality in regions without air quality monitors [3]. This research incorporated WRF-Chem modeled DOMINO satellite column and surface estimates of NO_2 concentrations to create 132 monthly NO_2 concentration estimates based on scaled yearly land use regression data. The resulting dataset consisted of observed concentrations at monitored sites across the United States, land usage at 22 different radii from each site, and estimated NO_2 concentrations based on the developed model.

While this study focused on using land-use regression to create a model, there are a variety of modeling options that can be used to predict air quality. Machine learning models are a particularly useful tool to interpret and find relationships in complex data.

This report proposes a machine learning model to predict NO_2 concentrations spatially based on data provided by Bechle et al (2015). First, a literature review was undertaken to understand what machine learning models have typically performed well in predicting air quality. Next, an exploratory data analysis (EDA) was performed on the Bechle et al (2015) dataset. Finally, multiple linear regression, neural network and random forest models were built and results were compared to see which method had the lowest mean-squared error (MSE).

2. Methods

2.1 Literature Review

There are a number of studies examining how machine learning models can be used to predict air quality. Seven studies were examined as part of this literature review, and can be broadly categorized into 2 areas: predicting $PM_{2.5}$ and predicting the Air Quality Index (AQI)/ Air Pollution Index (API). One exception is that one of the studies examining AQI also predicted NO_x concentrations.

2.1.1 PM2.5

Chen et al (2018) explored the use of random forest (RF) models to predict $PM_{2.5}$ concentrations spatially in China and compared them to multiple linear regression (MLR) and generalized additive models [4]. While the study began with a large number of predictors, these were narrowed down to ground-based measurements, satellite retrieved AOD data, urban cover data and meteorological data. The random forests model had the greatest predictive power of all the models considered, with a root mean squared error (RMSE) of 28.1 µg/m^3 on a daily scale (R^2 = 83%), improving to 10.7 µg/m^3 (R^2 = 86%) and 6.9µg/m^3 (R^2 =86%) on monthly and annual time-scales, respectively.

Xu et al (2018) likewise considered a number of machine learning models for $PM_{2.5}$ prediction in British Columbia, Canada [5]. 8 models were examined in this study: 1) MLR, 2) Bayesian Regularized Neural Networks (BRNN), 3) Support Vector Machines with Radial Basis Function Kernel (SVM), 4) Least Absolute Shrinkage and Selection Operator (LASSO), 5) Multivariate Adaptive Regression Splines (MARS), 6) RF, 7) eXtreme Gradient Boosting (XGBoost), and 8) Cubist.The predictors included humidity, temperature, albedo, normalized difference vegetation index (NDVI), height of the planetary boundary layer (HPBL), wind speed, distance to the ocean, elevation, and calendar month beside the ground level monthly averaged $P_{2.5}$ data collected from 63 stations between 2001 to 2014

as well as 3km resolution aerosol optical depth (AOD) data from Moderate Resolution Imaging Spectroradiometer (MODIS). This study found that the cubist model had the highest accuracy (RMSE=2.64 µg/m^3 and R^2=0.48) and the the MLR had the lowest accuracy (MSE = 3.24 µg/m^3 and R^2=0.22). The predictors with the most influence were monthly AOD and elevation.

Enebish et al (2020) considered 6 different machine learning models for PM$_{2.5}$ prediction in Mongolia: 1) RF, 2) gradient boosting, 3) SVM with a radial basis kernel, 4) MARS, 5) generalized linear model with elastic net penalties (a type of MLR), and 6) generalized additive model [6]. These models were run for annual data, cold season and warm season. Parameters considered were air pollution monitoring data, meteorology, land use and population. Across all time periods, the RF had the best R^2 and RMSE values. Over the entire period using the hold-out test set, RF had a RMSE of 12.92 (R^2 = 0.96), and the cold season and warm season had RMSE of 21.23 (R^2 = 0.92) and 7.44 (R^2 = 0.84), respectively.

2.1.2 AQI/API

Azid et al (2014) used a multilayered perceptron feed-forward artificial neural network model to predict API, using daily measurements of NO$_2$, SO$_2$, CO, PM$_{10}$ and O$_3$ over a period of 7 years in Malaysia [7]. The best RMSE and R^2 occurred when the hidden nodes were set to 6, and were 0.618 and 10.017, respectively.

Gu et al (2020) focuses on predicting the AQI in Shenzen, China [8]. The dataset consists of 365 sets of daily pollution data over one calendar year (2018), and the purpose was to develop a model to predict AQI. Pollution measurements included PM$_{2.5}$, PM$_{10}$, SO$_2$, CO, NO$_2$ and O$_3$. Two SVM models were developed: smart adaptive particle swarm optimization and particle swarm optimization, SAPSO-SVM and PSO-SVM, respectively. Additionally, a back propagation (BP) neural network model was developed. SAPSO-SVM had a test set classification accuracy of 91.62%, and PSO-SVM 88.56%. For the BP-neural network model, ten iterations of the algorithm best fit the test data set, where the percent error ranged from 18.41% for PM$_{2.5}$ to 30.29% for SO$_2$. While Gu et al stated that both models were a good fit for the data, by using different statistical comparisons to explain model fit, it is not clear which of the two models has a better predictive ability, although it appears to be SAPSO-SVM. The paper listed a number of limitations associated with its neural network, particularly the limited data points.

Singh et al (2013) used ensemble learning methods to predict air quality index in Lucknow, India [9]. They trained four different models: single decision tree (SDT), decision tree forest (DTF), decision treeboost (DTB) and SVM. While decision trees can be different from random forest, it appears in Singh's methodology that the DTF and DTB involve randomization with replacement from the training dataset to create separate models, which are then used to predict the entire data from the subsets. This is consistent with RF models, as essentially RF are ensemble decision trees. The parameters included in the model are 5 years of data on: daily air quality measurements (SO$_2$, NO$_2$, suspended particulate matter and respirable suspended particulate matter) meteorology (air temperature, relative humidity, wind speed, evaporation, and daily sunshine period). The DTF and DTB models outperformed the SVM models. DTB performed the best, with a RMSE of 4.38 (R^2 = 0.92).

Liu et al (2019) developed SVM and RF models to predict hourly AQI in Beijing, China and hourly NO$_x$ in an Italian city [10]. Parameters included historical hourly averaged AQI concentrations for PM$_{2.5}$, O$_3$, SO$_2$, PM$_{10}$ and NO$_2$ in Beijing (five years), and hourly averaged responses for CO, non-methane hydrocarbons, benzene, NO$_x$ and NO$_2$ in the Italian city (1 year). The SVM performed better in predicting AQI in Beijing with a RMSE 7.666 (R^2=0.9776), but the RF model performed better in predicting NO$_x$ concentrations in the Italian city (RMSE = 83.67, R^2 = 0.8401).

2.1.3 Comparison of PM2.5 and AQI/API studies

The main difference between the PM$_{2.5}$ and the AQI studies is that studies examining PM$_{2.5}$ tended to only examine one pollutant, whereas AQI studies consisted of measuring and modeling a number of different pollutants. Therefore, some AQI models were more interested in classification than predicting a specific pollutant spatially or temporally. As a result, different parameters tended to be included in the model depending on if it was predicting PM$_{2.5}$ or AQI. For example, meteorological data tended to be included in PM$_{2.5}$ studies, but not in studies examining API/AQI. Additionally, different types models tended to perform best depending on the target prediction.

A common limitation of all of the studies is the volume of missing data. In Chen et al (2018), the model had only two years of ground-based measurements to train the model on (2014-2016), and then predicted PM$_{2.5}$ concentrations for a ten year period (2005 to 2014) [4]. Xu et al, 2018 also discussed the challenge of missing data, averaging hourly and daily measurements where available to monthly concentrations to use in model development [5]. Enebish et al, 2020 discussed there being few air quality monitoring stations and insufficient data to well represent the high seasonal variability of PM$_{2.5}$ concentrations [6]. Additionally, Gu et al (2020) only used pollution data from one year within one region in China[8], and Liu et al (2019) used pollution data from one Italian city over one year [10].

The models in each of these studies is summarized in Table 2.1 below:

Table 2.1 Models used in literature

PM$_{2.5}$	Both PM$_{2.5}$ and AQI
MLR (Xu et al, 2018; Enebish et al, 2020; Chen et al, 2018)	**RF** (Chen et al, 2018; Xu et al, 2018; Singh et al, 2013; Liu et al, 2019; Enebish et al, 2020)
LASSO (Xu et al, 2018)	**Neural Network** (Azid et al, 2014; Xu et al, 2018, Gu et al, 2020)

PM$_{2.5}$	Both PM$_{2.5}$ and AQI
MARS (Xu et al, 2018; Enebish et al, 2020)	**SVM** (Xu et al, 2018; Gu et al, 2020; Liu et al, 2019; Enebish et al, 2020; Singh et al, 2013)
Gradient Boosting (Xu et al, 2018; Enebish et al, 2020)	
Cubist (Xu et al, 2018)	
Generalized additive model (Enebish et al, 2020; Chen et al, 2018)	
Mixed effects models (Chen et al, 2018)	

As we can see above more models were used to predict PM$_{2.5}$ than AQI, and the ones that were used in AQI studies were also used in predicting PM$_{2.5}$. The best-predicting models in each study are shown in Table 2.2, alongside their RMSE and R^2 values.

Table 2.2 Best models in each study

Study	Target Prediction	Best Model	RMSE	R^2
Chen et al (2018)[4]	Annual average PM2.5	Random Forest	6.9	0.86
Xu et al (2018)[5]	Monthly average PM2.5	Cubist	2.6	0.48
Enebish et al (2020) [6]	Annual average PM2.5	Random Forest	12.9	0.96
Azid et al (2014)[7]	Daily AQI	Neural network	10.0	0.62
Gu et al (2020)[8]	Daily AQI	SVM	n.a.	n.a
Singh et al (2013)[9]	Daily AQI	Random Forest	4.4	0.92
Liu et al (2019)[10]	Hourly AQI	SVM	7.7	0.98
Liu et al (2019)[10]	Houly NOx	Random Forest	83.7	0.84

Table 2.2 demonstrates that RF models tend to provide the most accurate prediction when considering a single pollutant, with 3/4 studies looking at PM$_{2.5}$ or NO$_x$ having RF as the best predicting model. When examining AQI, SVM models tend to work best, with 2/4 studies finding SVM provides the best prediction.

The RMSE and R^2 values vary significantly for each study. This can be attributed to the different geographic areas considered, varying spatial resolutions, amount of uncertainty in the data sources, prediction type and different parameters included in the model.

Because the objective of this study is to predict a single variable (NO$_2$ concentrations), then the models used by PM$_{2.5}$ studies are the most relevant. Therefore, in our model analysis, we will use MLR, neural networks and RF.

2.2 Exploratory Data Analysis

The data for this study comes from the National Spatiotemporal Exposure Surface for NO$_2$: Monthly Scaling of a Satellite-Derived Land-Use Regression, 2000–2010, authored by Matthew J. Bechle, Dylan B. Millet, and Julian D. Marshall. The dataset has 255 observations of 134 variables: the IDs of the air quality monitors, the states in which they are located, the latitude and longitude of the monitors, the quantity of NO$_2$ observed in parts per billion, an approximation of DOMINO satellite data using the WRF-Chem model, the distance from a monitor to the coast in kilometers, the elevation of a monitor in kilometers, and several covariates about the land in the area regarding impervious surface, population, and major, residential, and total road length.

The target variable in this predictive case is NO$_2$ concentration in the continental United States. The observations from the data have a mean NO$_2$ concentration of 11.831 ppb with a standard deviation of 6.290 ppb. The NO$_2$ concentration data has a range from 0.309 to 31.016 ppb. The five highest values are considered outliers. The distribution of NO$_2$ concentration values is unimodal with a slight right skew.

Figure 2.1

The potential explanatory variables of air quality monitor IDs, states, latitude, longitude, and WRF-Chem+DOMINO each exist as single columns. The variables related to land-use of impervious surfaces, population, major road length, residential road length, and total road length each have several columns quantifying these values based on the radius from the air quality monitor. The impervious surfaces, population, and major road length variables have 22 columns each based on radii ranging from 100 to 10,000 meters. The variables minor road length and total road length have 30 columns each based on radii ranging from 100 to 14,000 meters.

The spatially-related data of latitude, longitude, and state of air quality monitors can be best interpreted by viewing this data plotted on a map. Each observation corresponds to a unique monitoring station at a distinct longitude and latitude, so there are 255 points. Monitors in this dataset come from 43 of the 50 states in the USA, excluding Alabama, Alaska, Hawaii, Mississippi, Montana, Oregon, Nebraska, and West Virginia. Many states have just one or two monitors, while nearly half of all of the monitors are in the three states California, Texas, and Pennsylvania. Observing Figure 2.2, there are many spatial gaps in the data and little discernible correlation between position and NO$_2$ quantity.

Figure 2.2

Figure 2.3 is a correlation matrix for the variables in this data set. The five land-use variables are aggregated such that each of the series of related covariates differentiated by radius are instead one column. As seen in the plot, each of the land-use variables has a moderately strong positive correlation with NO$_2$, as well as the variable WRF+DOMINO. The impervious surfaces, major road length, residential road length, and total road length variables each have a heteroskedastic relationship with NO$_2$ with higher variability occurring when paired with larger NO$_2$ values. Each of these relationships show a stronger correlation when measured with a Spearman's correlation as opposed to the Pearson's correlation used in the matrix. The three road-related variables are related and

highly correlated with each other. The variable of population is highly skewed and has an exponential relationship with NO_2. This correlation is stronger when a log-transformed population is paired with NO_2. The variables of distance to coast and truncated elevation have a very weak correlation with NO_2.
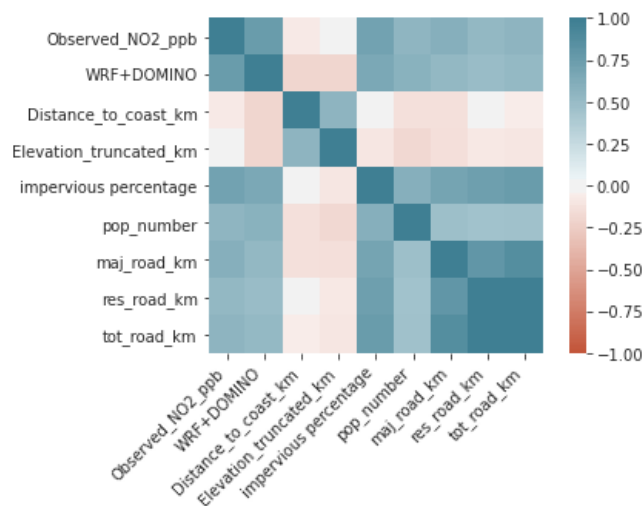

Figure 2.3 Correlation Matrix of Variables

The relationship between NO_2 and each of the five land-use variables changes as the radius changes. As seen in Figure 2.3, the overall correlation is moderately strong. However, the correlation between NO_2 and individual columns for each variable category with differing radii generally increases as the radius increases. The impervious surface correlation is highest at a radius of 7,000 meters ($\rho = 0.794$) and the four remaining land-use variables have their highest correlation at 10,000 meters: population (r = 0.721), major road length ($\rho = 0.770$), residential road length ($\rho = 0.754$), and total road length ($\rho = 0.771$).

2.3 Model

2.3.1 Multiple Linear Regression

Multiple linear regression (MLR) is a method of predicting a dependent variable using linear combinations of of any number of independent variables. MLR is widely used in statistical research because it is easy to interpret and understand. Additionally, multiple regression can also often result in a model with results comparable to that of a more complex model, such as a neural network or random forest model. The modeled equation takes the form of the following generalized equation:

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i + \varepsilon$

Where:

y = The dependent variable to be predicted,

β_0 = The constant intercept ,

$\beta_i x_i$ = The i-th constant multiplied by the i-th term

ε = The error term (difference between y predicted and y oberseved)

The multiple regression model used in this project first aggregates all 5 land use variables. Then, features with collinearity were removed. This was done by removing variables with a variance inflation factor over 5, and resulted in all road data being removed. The model was developed using ordinary least squares linear regression. Choosing the root mean square error as the loss function for the model, tests were run to select combinations of variables that resulted in the lowest root mean square error. The tests evaluated the effect of removing any combination of one, two, or 3 columns. These tests concluded that further removing columns only resulted in a higher root mean square error.

MLR Fit

The final model equation is presented in Table 2.3 with each feature and its respective coefficient.

Table 2.3 Coefficients used in final MLR model

Variable	Value
———–	——–
Constant intercept	2.9128368899112775
WRF+DOMINO	0.69
Distance_to_coast_km	-0.0013
Elevation_truncated_km	1.1e+01
radius	0.00013
Impervious	0.11
Population	0.00011

The model fits the data well for lower NO_2 concentrations. However, the model predicts high concentrations very poorly. This is due to the distribution of the data used to build the model: There are very few observations of high NO_2 values. The model fit is plotted in Figure 2.4
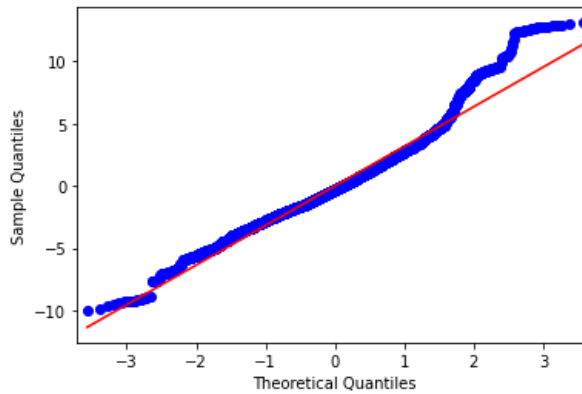
Figure 2.4 Linear Model Fit

Overall, the model was a fair predictor of NO_2 concentrations, with an R^2 value of 0.775. Of the models presented, it had the second lowest root mean square error (3.08) when applied to the test data. However, this model is not appropriate for the data it is predicting due to the fact that the data exhibits heteroscedacity, meaning that the variance of errors is not constant. This violates this assumption needed for using a linear model, and as a result the predictions are less accurate. Further, building a regression model based on aggregated land use columns resulted in multiple predictions for each monitor. These predictions at each monitor were similar and an average value was taken. However, this is not a good practice for predicting values with precision. Various MLR models with non-aggregated columns were tested, but the model built on aggregated data outperformed in all tests.

The multiple regression model presented, although comparable to other models used, is not an accurate, appropriate model for predicting NO_2 concentrations based on the available data.

2.3.2 Neural Networks

Artificial neural networks are based on the design philosophy of the neural connections in our brain. They consist of a group of interconnected nodes joined through edges. The edges transmit the signals (which in a machine learning model would be a real number) from one neuron to another just like a synapse functions in a brain. A typical neural network has many layers. Each layer is composed of a set of neurons and each layer transforms and processes data in a different way based on the hyperparameters of the model.

There are different kinds of neural networks. The most commonly used of which are Feed Forward Neural Network, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long-short term memory neural network (LSTM), Gated Recurrent Unit (GRU). There are several other advanced ones such as General Adversarial Networks, Auto-encoders and Deep Belief Neural networks. However, we restrict our discussion to the five types listed here.

We will first discuss, briefly, the CNN, RNN, LSTM and GRU before moving on to the Feed formal neural network which is the neural network model used in this project.

2.3.2.1 CNN

Largely used for classifying image and audio data, the convolutional neural network, has three layers called convolutional layer, pooling layer and a fully-connected layer which is similar to the regular neural network. Additionally, similar to a regular neural network, there is an input and output layer. The convolutional layer takes the input from the input layer and convolutes the data and sends it downstream for further processing. In simpler terms, in CNNs an input image is taken and a filter is applied on it repeatedly acorss the image to create a feature map which can be used to identify and classify the input image [11].

2.3.2.2 RNN

These neural networks are used to predict the temporal trend of the data. In RNNs, the 'memory' of the previous inputs is stored and used for further processing of future inputs and outputs.This relationship between input and output data could be both unidirectional (moving in the forward direction) or bidirectional (moving both forward and backward) where future data could also be used to improve the current inputs and outputs. These neural networks could use a variety of relationships between their inputs and outputs and could be described as one to one, one to many, many to one and and many to many relationships. Sigmoid, ReLu, Tanh are the common activations used in RNNs [12].

2.3.2.3 LSTM

LSTMs are a type of RNN which solve the common problems that RNNs face, that is their inability to efficiently handle short-term memory over a lengthy series of steps. LSTMs solve this problem using a memory cell and gating units and consists of a set of gates called input, output and forget.The input gate is responsible for monitoring and deciding the kind and quantity of data that is allowed to enter the cell, the memory gate is responsible for deciding the proportion of data that should be 'forgotten' and which

information is useless and to be discarded. Finally, the output gate is responsible for deciding the amount of data that is passed as output from the cell [13].

2.3.2.4 GRU

GRU is similar to LSTM with the exception that an output gate is absent. Instead it has a reset and update gates. The reset gate works as a combination of input and forget gate and the update gate works as an additional forget gate. Their performance is equivalent and sometimes even better than LSTMs, especially when dealing with some special data such as those involved in language processing[14].

2.3.2.5 Neural networks used in this project

In this project, majority of the models used were simple feed forward neural network models. This decision was based on two factors:

1. The literature search revealed that neural network models are one of the most widely used models especially in pollution prediction problems

2. Neural networks are simpler as compared to random forests but at the same time slightly more complex than linear regression and hence they incorporate some machine learning components

We used keras, which is an opensource library of functions that allows the use of the tensorflow library for machine learning models in python. The building blocks of a neural network model such as layers, objective functions,activation functions and optimizers, are all provided by keras.

A regular neural network consists of an input layer, hidden layers and an output layer. We used a sequential model which consists of few layers stacked upon one another, linearly [15]. Each layer has multiple cells and we could define the number of input cells for these hidden layers. We can also define the type of activation function for each layer. We used an average of 3 to 5 hidden layers in our neural network models.

We used a combination of three different activation functions:

a. Linear Activation - a linear activation uses the weights to multiply the inputs providing an output which is linearly proportional to the input. This function is also termed as no activation function as there is no further transformation being operated on the values.

b. ReLU - Rectified Linear Activation Unit - even though very similar in appearance to a linear function, ReLU allows back-propagation of errors. This function always gives an output of 0 for negative values and behaves linearly for values greater than 0.

c. Sigmoid activation - by implementing this activation function, the outputs are normalized and bound between 0 and 1. A sigmoid function provides a smooth gradient curve and much clearer predictions.

Further, the compile function is used to compile the model created, which takes in a variety of arguments. We used an Adam optimizer which uses the combination of squared and moving average gradients to individually compute the learning rates for each parameter. As such it provides the benefits of both RMSProp and Stochastic Gradient Descent.

We chose a learning rate of 0.0005 to 0.005. Learning rate is an essential hyperparameter that influences the weights of a model and the extent to which they are updated. A very low learning rate could lead to a model being very slow and eventually unable to reach the desired result. At the same time, a very high learning rate could worsen the model by making it unstable and inaccurate and eventually lead to inferior quality of weights.

A loss function refers to the quantity that the model is trying to either minimize or maximize and used as a metric to define the fit of the model. Both absolute error and root mean squared error were chosen as the loss functions.

The final step was to train our model so created using training dataset and the keras function model.fit. An important parameter for fitting the model is 'epoch' which indicates the number of times the training dataset is run through our model. For example, if we choose 500 epochs we are allowing the training data to run through the model for 500 times. The number of epochs determine how well the model familiarizes itself with the model. Greater the number of epochs the better the model understands your data.

However, it must be ensured that excessive epochs in proportion to the complexity of the model and the dataset is avoided as this could result in a situation wherein the model no longer tries to find a pattern among the data but rather simply memorises it. In such situations, the model overfits the data. We used a range of epochs between 350- 1000, based on several trials, where in each case, a different epoch was chosen and the final RMSE was monitored. The final model used the most suitable epoch for the given set of features and other hyperparameters.

Usually, training dataset is split into training and validation dataset to resolve the problem of overfitting if encountered. A validation dataset also allows for considerable improvements in model before the model is fit using the test data. A common practice is to split the training data in such a way that 80%-90% of the data is used to train the model and 10-20% to validate. However, our best

performing neural network model did not use any validation data citing the limited number of datapoints available for the competition.

Our neural networks achieved a RMSE values in the range of 2.92-6.6, which implies that, in most cases, our model overfit the training data. The best performing neural network model is described in detail in the Results section of the report. The details about the neural networks are summarised in Table 2.3.

Table 2.3 Summary of 3 Neural Networks Used in the project (the fourth and the best performing is explained in detail in the Results Section)

Features	Hyperparameters	Group Member	Final RMSE Achieved
WRF+DOMINO, percentage impervious surface, population, major road length, resedential road, total road: All at buffer 10000 m	learning rate = 0.0004; batch size = 120; hidden layers = 4; epochs = 350	Sudheer	6.60
WRF+DOMINO, impervious percentage, population, major road length, resedential road, total road: All at buffer 10000 m	learning rate = 0.001; hidden layers = 1; epochs=2000	Hope	4.23
WRF+DOMINO, percentage impervious surface at buffers 3000,3500,4000,5000,6000,7000,8000,10000 m	learning_rate = 0.005; epochs = 400; hidden layers = 4; batch_size = 100	Tessa	4.33

2.3.3 Random Forest

Trees and tree algorithm is among the most widely applied machine learning algorithms. It includes random forest, gradient boosting decision trees, XGBoost, etc. The fundament of trees algorithm is decision tree, which can be divided into classification tree and regression tree. In this project, the label is the observed NO_2 concentration, thus, regression tree algorithm can be applied to make the prediction.

Random forest is a combination of tree predictors based on model aggregation ideas. It is realized by creating an ensemble of trees by generating random vectors that govern the growth of trees and letting them vote for the most popular label [16].

Random forest have the advantages of low overfit, low noise affect. However, for random forest regression, a relatively large number of features are required to reduce the test set error[16].

Variable selection and feature engineering are very important in data preprocess of random forest. The two main objectives are to find variables highly related to the response variable, and to find a small number of variables sufficient to a good prediction of the response variable[17].

Based on previous exploratory data analysis, the raw dataset contains valuable land-use variables in series including impervious surfaces, population, major road length, residential road length, and total road length within different buffers. It also include elevation, distance to coast, latitude, longitude, linear combinations of satellite data and WRF-Chem output. The variables within their series are highly correlated, so a few representative ones should be selected out based on the second criteria mentioned before. Here we select one for each kind of land-use variables. It is interesting that high correlations have been found between some of these land-use variables, like road lengths. Thus, we only selected the road length having the highest relationship with the variable to be predicted. Latitude and longitude are binned and one-hot coded. Around 20% of the raw data were selected out for prediction validation.

Random forest model in this project was created with scikit-learn RandomForestRegressor library. GridSearchCV was imported to conduct hyperparameter optimization. The model output generally had a MSE around 3.2 in the validation data. However, on the test data it did not show a good performance (MSE = 4.2). One possible reason is that random forest regression is weak on predicting values with magnitudes beyond the train dataset, meaning that it cannot confidently handle noise or outliers. It also prefers high dimensional, large-scale data, however, in this project the dataset is on a relatively low scale.

In conclusion, random forest is possibly not a very appropriate algorithm to make prediction in regression problems from dataset in a low-scale and a low-dimension.

3. Results

3.1 Model Devolpment

The choice of algorithm to use is one of the most important and consequential choices in the process of developing a model. Using the information obtained in the exploratory data analysis, the most important factors that will dominate this choice is the fact there is only 255 samples in the training data and that the data is skew right; meaning that an algorithm needs to be able to make predictions based on finite data and inputs but can evaluate each sample independently. A neural network model is the most applicable to this situation to make the best predictions for most of the data but can also avoid over and underfitting.

The inputs used in a neural network determines the effectiveness of the model; and as with all datasets some data is more important than others, while some have no bearing on the dependent variable. The spatial variance of the monitors and outliers in the data means that less complexity in the inputs will reduce overfitting.

Table 3.1 Inputs used in Neural Network Model

Data Parameter	Buffer Length (m) if applicable
WRF+ DOMINO	N/A
Distance to the coast (km)	N/A
Truncated elevation (km)	N/A
Percentage of Impervious surfaces	1500
Population (in Thousands)	4000
Length of Major Roads	1500
Length of Total Roads	14000

Table 3.1 shows the 7 inputs used in the model. The use of a single length of each input parameter was done to avoid dependencies and the reducing the complexity of the model. The one exception is major road parameter, which is directly imputed once and indirectly added in the total road parameter; this was done due to cars being a major localized source of NO2. The first 3 inputs are considered point inputs, describing the monitor location and unique monitor atttributes; whereas the last 4 inputs are dependent on the surronding area and is defined by the buffer length around the area. This may cause areas dense with monitors may have similar inputs but different NO_2 concentrations; to avoid this complication buffer length are kept a small as possible. Another reason to keep the buffer areas small is that some monitors in the training data are close to the United States border, because this outside the scope of this project any parameter outside the continental United States is not counted towards the total (ie. a major road connecting a US city to a mexican city would only have the highway in the United States count aganist the major roads parameter). This may cause some errors in the trainng of the nueral network, so the inputs try to minimize this case.

The buffer area dor the non-point inputs were determined based on R^2 for each parameter. As a main goal was to decraese the buffer area, the all relative maximas as well as any buffer area which is smaller than the relative maxima and the R^2 was increasing. This led to very small buffer areas creating the best features to include in the model; with the only exception being the amount of total roads but this features was specfically added to identify very large metropolises with large suburbs.

```python
train_dataset =
tf.data.Dataset.from_tensor_slices((x_dependent_train,y_ind_train)).batch(batch_size=255)

model = tf.keras.Sequential()


model.add(tf.keras.layers.Dense(units=128))

model.add(tf.keras.layers.Dense(units=16))

model.add(tf.keras.layers.Dense(units=8))

model.add(tf.keras.layers.Dense(units=4))


model.add(tf.keras.layers.Dense(units=1, input_shape=(7,)))


model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=.0005),
    loss='mean_squared_error',
)

history = model.fit(train_dataset,epochs= 10000)
```

The code block above shows the architecture of the neural network used that achieved the best performance. The main hyperparameters (batch size, number of epochs, and learning rate) were choosen to allow the training on the outliers and allow for minimizing the loss function but to allow quick convergence. The number of units per hidden layer and number of hidden layers were choosen to be in base 2 to increase computational speed and to prevent overfitting on the traing data. Many choices made regarding were governed by two main motives; reducing the complexity of the model and ensuring the model captures the entire range of concentrations in the training data. To avoid overfitting the model was designed with reduced complexity in the model, like the inputs, to allow monitors with similar inputs to have the output similar concentrations. Unlike other models used there was no validation data used to confirm that the training process did not overfit the data, this was done intentionally for two reasons. The first reason is that the data set was relatively small for training a neural network and that extracting 10-20 percent for validation may cause mistraining of the neural network, this fact is also compounded by the fact the observed NO_2 concentration is heavily skewed. The code block also shows that activations were used in the hidden layers of the model; this was done to allow the extreme outliers in the data be learned and trained from properly in the model but not impact other nodes in each layer.

3.2 Model Analysis

The final model was further analyzed using two major parameters: squared error and absolute error. Squared error penalizes predictions further from the observed concentrations, which is beneficial for detecting outliers and anomalies in the data. The squared error is also related to the root mean squared error, the measured used to determine the best model in the Kaggle competition meaning it can provide insights on the individual monitors in the training data.

The problem is that the loss function used was the mean squared error, meaning that only using this metric can may cause us to fail to observe other possible problems while training the data. Absolute error as a criterion for analysis fills these gaps in knowledge, such as bias in the training data or the geographic distribution of the error.
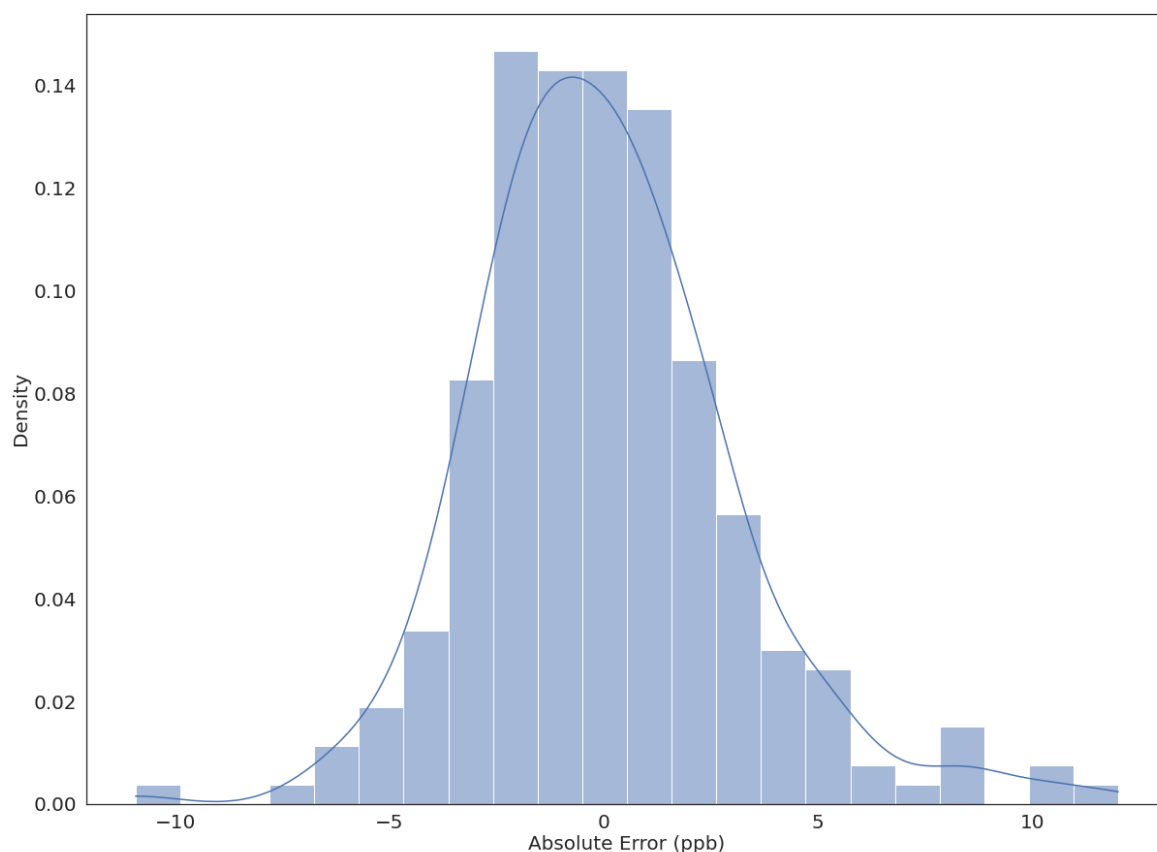

Figure 3.1. Training Data Absolute Error Histogram

Figure 3.1 shows the distribution for the absolute error on the training data. The distribution is nearly gaussian, which is expected with a large sample size. The distribution is also centered on 0 and has a very small skew, meaning that there is no bias in the when the model was training. The major difference between the distribution and a Gaussian distribution is the kurtosis.

The kurtosis was found to be greater than 3, meaning that the data is more concentrated near the mean with some extreme outliers. This is also an interesting because since the loss function of the model was mean squared error, these outliers should be minimized,

but it is actually better to have weights that make these monitors be incorrectly predicted than to minimize their error.

Table 3.2 5 Number Summary and mean of Absolute Error

Parameter	Value
Mean	0.002
Standard Deviation	3.056
Minimum	-10.976
25%	-1.973
50%	-0.206
75%	1.514
Maximum	11.994

Table 3.2 shows that the the majority of monitors predicted value is within 2 ppb of the actual value, with outliers being greater than 5.99 and less than -6.45 ppb.
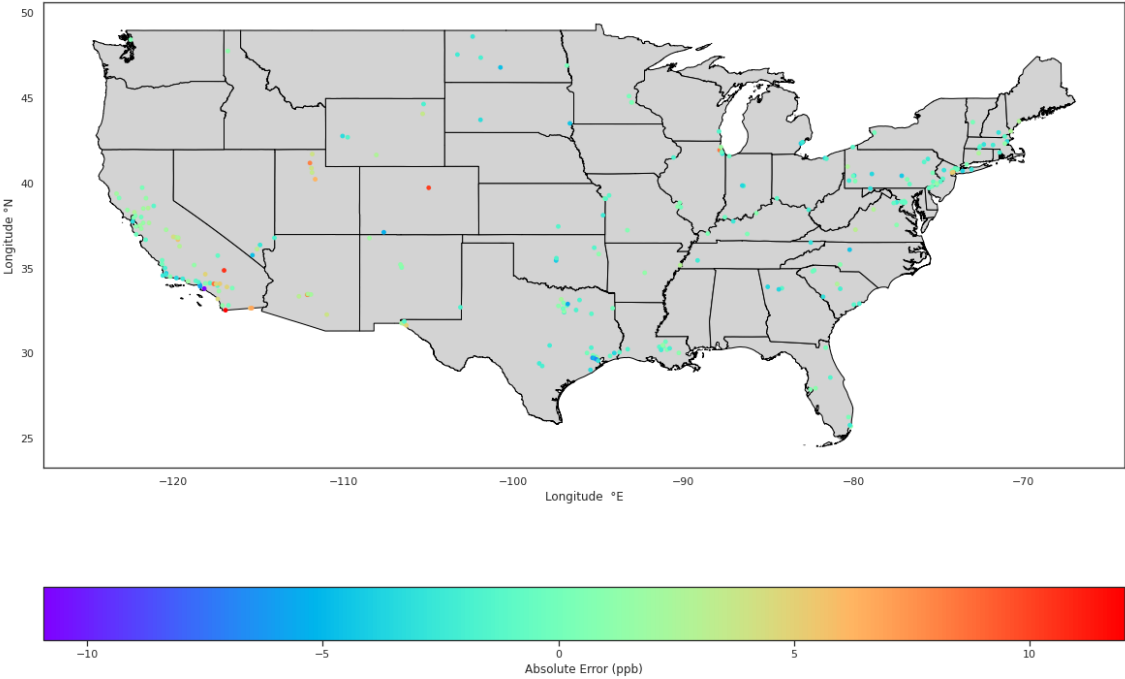

Figure 3.2. Training Data Absolute Error Geographic Distribution

Figure 3.2 shows that the absolute error is not spatially homogenous, with most of the outliers being in the West. Further analysis shows out of the 255 training points, 12 of those were outliers. As root mean squared error is the major performance metric of our model the extreme outliers are of particular note becuase they are weighted heavily in this metric.

All the previous analysis was done on the training data but to ensure our model is robust, the model needs to be run on other data to ensure it valdity as a model.
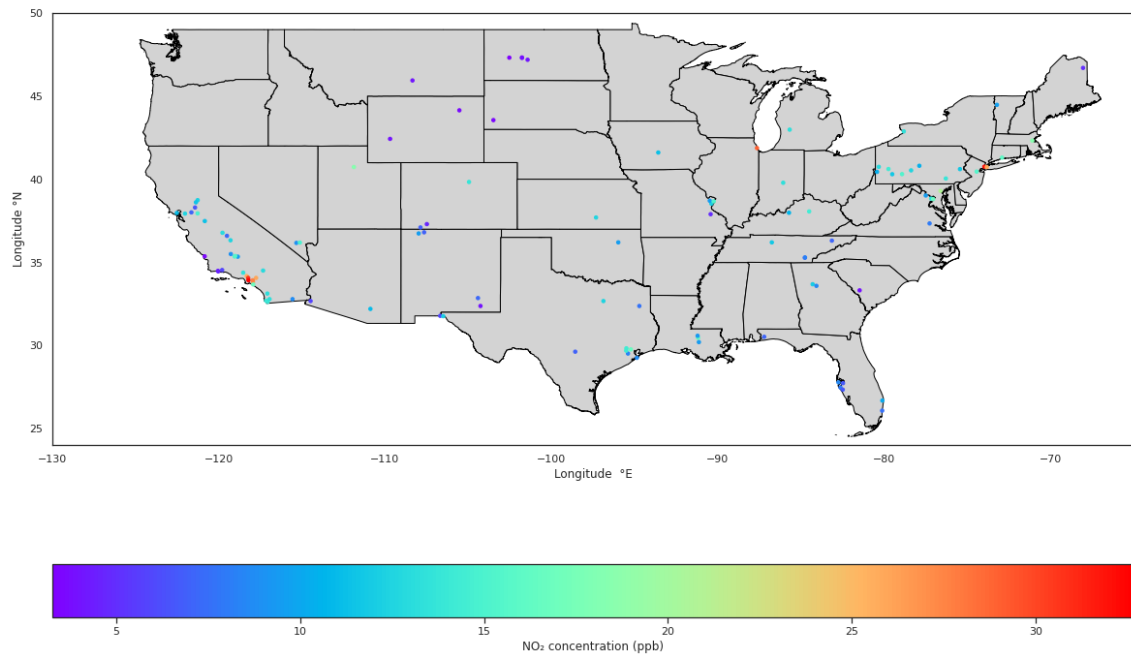
Figure 3.3. Testing Data NO$_2$ Concentrations Map

Figure 3.3 shows the predicted values of the known testing data. The model was run on 165 samples (139 of the testing data and 26 hidden values) and the root mean squared error was found to be 2.925, which is lower than the root mean squared error on the training data (3.05). This means that the final model is robust and can be applicable across the United States.

4. Discussion

4.1 Models Presented

Our team presented six different models for predicting NO$_2$ concentrations with varying degrees of complexity: one multiple linear regression, four neural networks, and one random forest model. The models were compared based on their test data predictions' root mean square errors. This found that, while most models preduced comparable results, a neural network model was the best predictor of NO$_2$ concentrations and was the most applicable model. Multiple linear regression models were tested, but despite having a relatively low root mean square error, the heteroscedacity of the data makes for a non-applicable model. A random forest model is also presented. The model was a good predictor on the validation data, but showed poor performance on testing data. The random forest model was found to not be appropriate for predicting NO$_2$ concentrations when using few data points.

4.2 Model Performance

Every model tested resulted in a relatively high root mean square error when compared to the mean value of the data (11.83 ppb). This may be due to a combination of factors. The model parameters could be better fine-tuned; however, we attribute much of this error to the size of the dataset used for training the model. Analysis on the dataset shows that the outliers accounted for approximately 40% of the standard error, meaning that the absolute error may be a better metric to evaluate the model performance.

The full dataset contained less than 500 observations across the United States. These observations were not evenly distributed across the US, resulting in some states with no data and a handful of states with many data points. The data used also had very few observations of high NO$_2$ concentrations available for training the model. The best performing neural network model created using this data was a poor predictor of NO$_2$ concentrations in western states.

The best performing model showed no evidence of overfitting. The model performed slightly better on the testing data than it did on the validation data, supporting the conclusion that the model is robust.

4.3 Limitations and Future Research

Still, further research is needed. This may include incoportating more data sources to paint a more detailed picture of how NO_2 concentrations vary across the United States. The data used in this report was based on land use information. Other recent studies have found that incorporating meteorological data can improve the model predictions. Although these factors are present implicitly through the WRF+DOMINO data column, our model may benefit from data like temperature and humidity being explicit. Also, NO_2 comes from multiple sources. This dataset includes information about roads, but does not include information regarding distance from nearby power plants or industry. This may be causing the model to underpredict the NO_2 concentration at stations nearby to other sources of NO_2.

Because of the spatial variability and limited data, this model is only applicable to some regions within the United States. And, due to the limited amount of US data and the uneven distribution of this data across the country, it may have been better to focus on modeling air quality across different regions in the country instead of across the entire contigous United States.

Further, neural networks are limited due to their difficulty of determining a learning rate, slow speed of convergence, and lack of precision [8]. Given a larger amount of data, it may be useful to revisit the random forest models, as it was shown to perform well in previous air quality studies.

References

1. **Primary National Ambient Air Quality Standards (NAAQS) for Nitrogen Dioxide**
   OAR US EPA
   US EPA (2016-07-01) https://www.epa.gov/no2-pollution/primary-national-ambient-air-quality-standards-naaqs-nitrogen-dioxide

2. **Basic Information about NO2**
   OAR US EPA
   US EPA (2016-07-06) https://www.epa.gov/no2-pollution/basic-information-about-no2

3. **National Spatiotemporal Exposure Surface for NO2: Monthly Scaling of a Satellite-Derived Land-Use Regression, 2000–2010**
   Matthew J. Bechle, Dylan B. Millet, Julian D. Marshall
   Environmental Science & Technology (2015-10-20) https://doi.org/10.1021/acs.est.5b02882
   DOI: 10.1021/acs.est.5b02882

4. **Redirecting** https://doi.org/10.1016/j.scitotenv.2018.04.251

5. **Redirecting** https://doi.org/10.1016/j.envpol.2018.08.029

6. **Predicting ambient PM 2.5 concentrations in Ulaanbaatar, Mongolia with machine learning approaches**
   Temuulen Enebish, Khang Chau, Batbayar Jadamba, Meredith Franklin
   Journal of Exposure Science & Environmental Epidemiology (2020-08-03) https://www.nature.com/articles/s41370-020-0257-8
   DOI: 10.1038/s41370-020-0257-8

7. **Prediction of the Level of Air Pollution Using Principal Component Analysis and Artificial Neural Network Techniques: a Case Study in Malaysia**
   Azman Azid, Hafizan Juahir, Mohd Ekhwan Toriman, Mohd Khairul Amri Kamarudin, Ahmad Shakir Mohd Saudi, Che Noraini Che Hasnam, Nor Azlina Abdul Aziz, Fazureen Azaman, Mohd Talib Latif, Syahrir Farihan Mohamed Zainuddin, … Mohammad Yamin
   Water, Air, & Soil Pollution (2014-07-21) https://doi.org/10.1007/s11270-014-2063-1
   DOI: 10.1007/s11270-014-2063-1

8. **Prediction of air quality in Shenzhen based on neural network algorithm**
   Kuiying Gu, Yi Zhou, Hui Sun, Lianming Zhao, Shaokun Liu
   Neural Computing and Applications (2020-04-01) https://doi.org/10.1007/s00521-019-04492-3
   DOI: 10.1007/s00521-019-04492-3

9. **Redirecting** https://doi.org/10.1016/j.atmosenv.2013.08.023

10. **Air Quality Index and Air Pollutant Concentration Prediction Based on Machine Learning Algorithms**
    Huixiang Liu, Qing Li, Dongbing Yu, Yu Gu
    Applied Sciences (2019-01) https://www.mdpi.com/2076-3417/9/19/4069
    DOI: 10.3390/app9194069

11. **An Introduction to Convolutional Neural Networks**
    Keiron O'Shea, Ryan Nash
    arXiv:1511.08458 [cs] (2015-12-02) http://arxiv.org/abs/1511.08458

12. **A Critical Review of Recurrent Neural Networks for Sequence Learning**
    Zachary C. Lipton, John Berkowitz, Charles Elkan
    arXiv:1506.00019 [cs] (2015-10-17) http://arxiv.org/abs/1506.00019

13. **Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network**
    Alex Sherstinsky
    Physica D: Nonlinear Phenomena (2020-03) http://arxiv.org/abs/1808.03314
    DOI: 10.1016/j.physd.2019.132306

14. **Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling**
    Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio
    arXiv:1412.3555 [cs] (2014-12-11) http://arxiv.org/abs/1412.3555

15. https://doi.org/10.1016/S0169-7439

16. **Random Forest**
    Leo Breiman
    Machine Learning (2001)
    DOI: doi:10.1023/a:1010933404324

17. **Variable selection using random forests**
    Robin Genuer, Jean-Michel Poggi, Christine Tuleau-Malot
    Pattern Recognition Letters (2010-10) https://doi.org/czr7p4
    DOI: 10.1016/j.patrec.2010.03.014