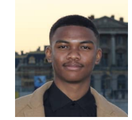
CIVIL_226

Predicting temperature around nuclear waste canister

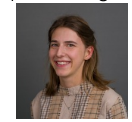
By ST_DMF

Frank Djidingar Monbe



316435

Sophie Tiberghien

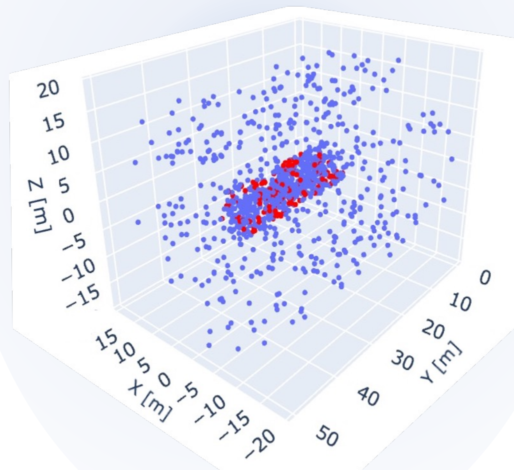


282240

1 The Problem

Taking care of nuclear waste in Switzerland by burying them underground in a canister ;

Using measures from known sensors to predict the temperature evolution of unknown sensor ;



2 Data pre-treatment

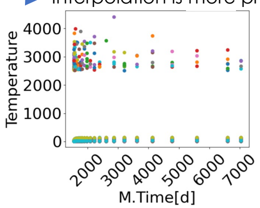
The data is split into two sets: training (900 sensors) and testing (146 sensors), for each set the relative humidity, pressure, coordinates (X, Y, Z) , materials and radius are available for **32 times**, and the temperature is available only for the training set.

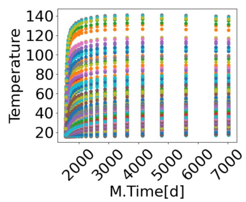
Plotting the temperature against time helps us to identify the outliers and missing data. All the temperatures > 2000 °C were removed.

Filling missing values → Mean of the temperature for the sensor

Filling missing values → Interpolation with the previous and next time

→ Interpolation is more precise





3 Models

Two techniques were tested to predict the temperatures.

K-Nearest Neighbours (KNN):

This method is using the temperature of the closest sensors to predict the unknown ones.

Features : Coordinates (X, Y, Z), Humidity, Pressure and radius

Characteristics = Loop for the 32 times, K=3, Euclidian $d(p,q) = \left(\sum_{i=1}^{n}(p_i - q_i)^2\right)^{1/2}$,distance weighting

Neural Network (NN):

This method is passing the data into layers to predict the temperatures

Features : Coordinates, Humidity, Pressure, radius and time

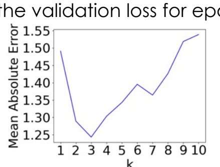
Characteristics = Batch size = 32, Epoch = 300,
Optimizer = Adam $W := W - \alpha \frac{V_{corr\,dw}}{\sqrt{S_{corr\,dw}} + \varepsilon}$
Loss function = L1 (Mean absolute error), Lr = 0.001 $L_1 = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
Scheduler (reducing the Lr by factor 10 if the loss is not improving for 10 epochs)
Activation function = ReLU

Architecture = (7, 256) → (256,512) → (512,256) → (256, 64) → (64,1)

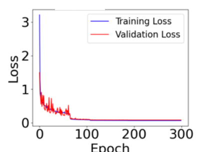
4 Validation

In order to see if the method could be efficient on unseen data, we split the training (80% train – 20% test). For KNN, we plotted the mean square error against the number of neighbours. For NN, we plotted the training loss against the validation loss for epochs.

KNN :



NN :



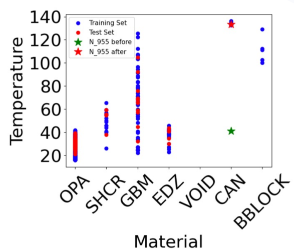
The best k =3, the training loss and validation loss are converging (The epoch should be reduced).

5 Data post-treatment

For both methods, we have plotted different graphs to verify if the relation of the different features were similar for both sets.

When plotting the material against temperature we have realized that the temperature for material CAN was severely underestimated ★. The graph is for time 2090 but this is valid for all the times

→ For sensor N_955, using mean of the temperature for each time for the CAN sensor ★

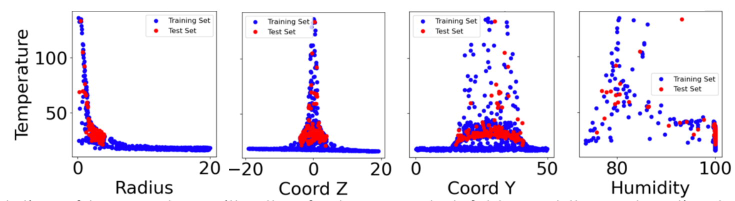


6 Results / Discussions

	L1 score	L2 score	Computational time
KNN	1.2294	7.3497	instantaneous
NN	0.2263	0.1748	15 min

The **Neural Network** method delivers better, more accurate prediction. However, the computational time is way larger.

Finally, the predicted temperature are plotted to compare them we other features for one time (here 2090).



The comparison of the relation of temperature with other features was helpful to spot the underestimated or overestimated temperatures.

To improve the result, a better way to deal with the sensor in CAN material needs to be implemented. Furthermore, other techniques should be tested, like random forest to better identify the interesting features or spline interpolation by using the relation of the features and temperature to predict the latter. Using ensemble method by combining several models could also improve the results