Plan for Machine Learning Model to Predict Soil Moisture Level During Rolling Compaction

proposed by
Pulkit Goyal

TABLE OF CONTENTS

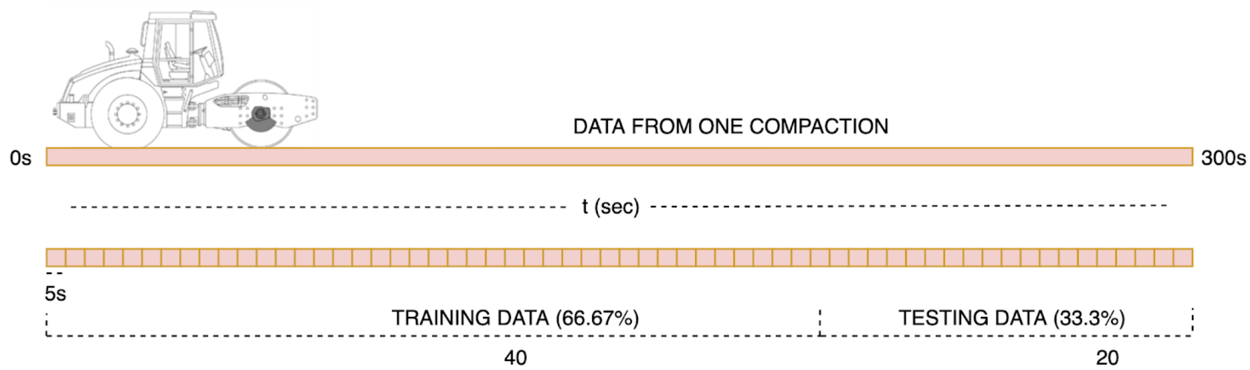
1. Abstract

For construction, compaction is a very crucial operation which is carried out to increase the soil density by applying stress to the landfill. These compactions are done until a maximum soil density is reached for a certain moisture level.

Traditional methods of determining the moisture level are very expensive, time taking, and tedious, thus impractical.

This project aims to deduce the soil moisture level using measurable physical parameters with a machine learning model to provide an alternative way of measuring the soil density.

2. Overview of the Dataset



> Exactly 300 seconds of data is gathered from one compaction run.
> Splitting the data into window sizes of 5 seconds will result in 60 data points.
> The frequency range of FFT, at sampling frequency of 5128Hz, will be [0.2Hz, 2564Hz]

The typical frequency range of the expected data, based on observation from the trail data, is assumed to be [50Hz, 1000Hz]. This lies within the frequency range of the (FFT) analysis.

The 60 data points from each run are split in the ratio 2:1 for training and testing respectively. Assuming an average of 4 compactions, a total of 160 training data points will be obtained per soil# type for training the model. This should be enough*.

** There is the option of increasing the data points by reducing the window size if this (60) is not sufficient. A bigger window size ensures that the data is not localised, and is a precise representation of the feature average. Also, since the conditions do not change much over a single run, the data points are almost similar and do not have much variation. So, having too many inputs of the same type during training will lead to overfitting.*

 # These data samples are obtained for a few different types of soils over a range of consecutive compaction runs.

> CONCERN
The planned data acquisition only collects data from compaction runs on one soil moisture level for a soil-type. This would likely cause the model to be applicable to only a particular moisture level for that soil-type. At other moisture contents, it would have bad performance.

To ensure the robustness and flexibility of the model, **it seems important to collect data of compaction runs at different moisture levels.**

3. Pre-Processing of Signal Data

- Given a data point (acceleration x-y-z + acoustic data), DFT will be computed to obtain its profile in the frequency domain.
- To clean the data, noise reduction will be performed.
- The idling vibration and sound frequency of the compaction roller will be subtracted.

3.1. Feature Extraction

The *k* dominant frequencies* and their amplitude will be used as inputs to the model.

** It is unsure how many frequencies (k) are required to accurately describe a profile. After looking at more data samples from runs on different soil-type and soil density, this can be assessed.*

Apart from using the constituent frequencies (pitch and amplitude) as a feature, a few other features that are characteristic of *timbre*, can be as input to the model. These include,

- spectral spread (a measure of bandwidth, tonality vs noisiness)
- kurtosis
- skewness

This paper lists a few other options available. -
http://recherche.ircam.fr/anasyn/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf

3.2. Feature Elimination

It is important to eliminate redundant features as having correlated input features leads to overfitting and over-complexity which in-turn leads to longer training time and poor performance.

The crude features that are proposed to be inputs to the model are,

- Acceleration in three directions
- Acoustic data
- Weather data (Temperature and Humidity)
- Soil class

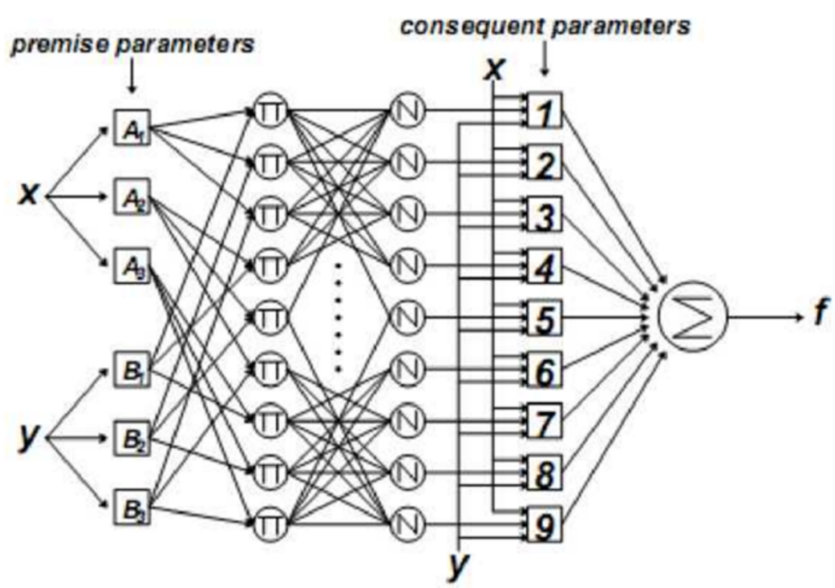
It is quite likely that the acceleration and acoustic data are correlated to each other.
Using PCA, the correlation between the two features can be analysed, and if they are strongly correlated, it is advisable to remove the feature with less variance from the input features.

> CONCERN (cont.)
The relation between the soil density and vibration profile are likely strongly dependent on the soil moisture. So, **it seems crucial to have the soil moisture as an input feature**, which seems to be missing in the data collection plan.

4. Proposed Model - ANFIS

ANFIS (Adaptive Neuro-Fuzzy Inference System) is basically a Sugeno fuzzy inference system represented under a neural network framework. The idea behind it is that it allows training of a fuzzy system using the same data driven principles used in ANN.

★ The original paper is here - https://www.dca.ufrn.br/~meneghet/FTP/anfis%2093.pdf.
★ An example of implementation of an ANFIS system for a similar application can be read here - https://ti.arc.nasa.gov/m/pub-archive/archive/SPIE_02_3_r.pdf.
★ A simple overview of fuzzy system and ANFIS can be read here - https://shodhganga.inflibnet.ac.in/bitstream/10603/24154/7/08_chapter%203.pdf



Two-input first-order Sugeno fuzzy model with nine rules

➔ Layer 1: *Input layer*
➔ Layer 2: *Fuzzification layer* that computes the membership value of the real inputs (*y* and *y*) in predefined fuzzy sets (A1, A2 and A3, A4). In this particular example, there are 3 (A_n) sets for first input and 3 (B_n) sets for the second input.
➔ Layer 3: *AND layer* consisting of 3 x 3 (A_n x B_n) nodes. These represent all the combinations between membership sets of the inputs. In this layer, the outputs from the last layer are simply multiplied to obtain a “net firing strength” of that rule.
➔ Layer 4: *Normalization layer* consisting of 3 x 3 (A_n x B_n) nodes. This layer normalises its corresponding input based on other outputs from the previous layer.
➔ Layer 5: *Fuzzy inference layer* consisting of 3 x 3 (A_n x B_n) nodes. This layer computes the “inference value” based on the original inputs and the corresponding weights from the previous layer.
➔ Layer 6: *Defuzzification layer* with one node, which computes the crisp value by simply adding the outputs of the last layer.

The tunable parameters of an ANFIS system are

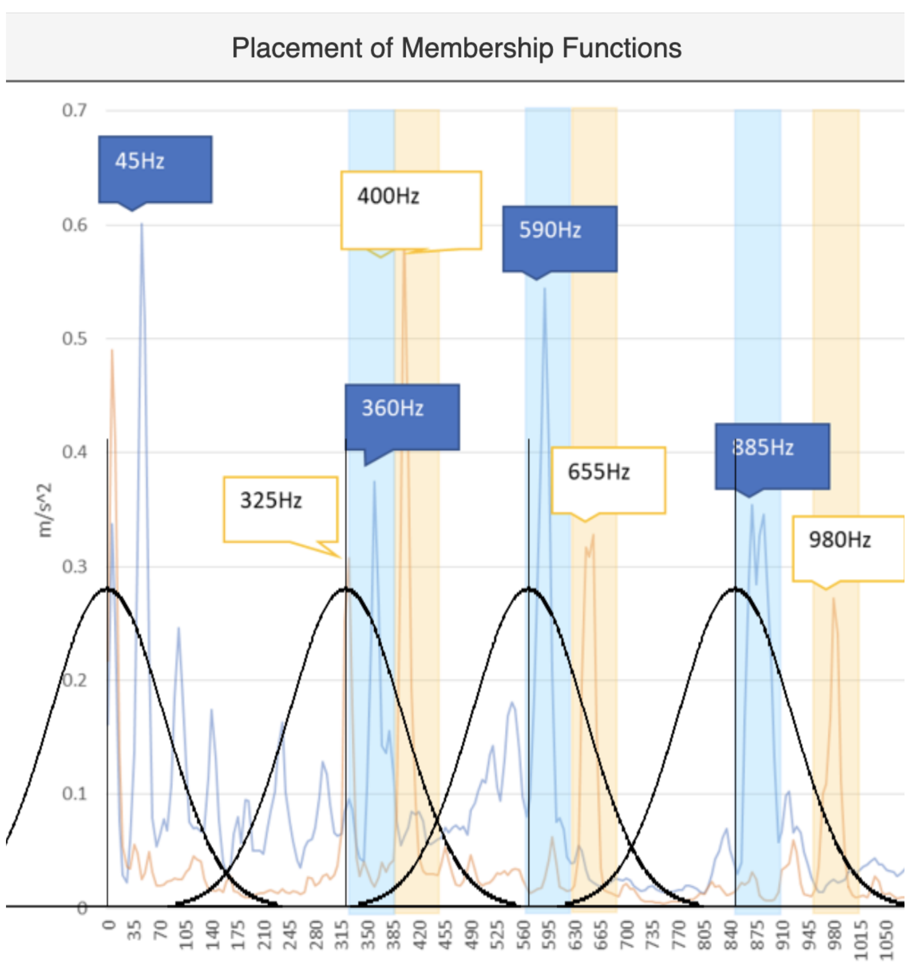
- a_i, b_i, and c_i = premise parameters (properties of the membership functions)
- p_i, q_i, and r_i = consequent parameters (properties of the inference system)

Research shows that ANFIS models train well with a hybrid training algorithm combining the least squares method (in a forward pass) to optimize the consequent parameters and the gradient descent method (in a backward pass) to optimize the premise parameters.

4.1. Motivation Behind Selecting ANFIS

A fuzzy logic system, in its core, works by finding the belongingness/nearness or “membership” of an input data point to predefined membership sets in order to interpolate an output. So, in cases where it is known that the input values are distributed around certain fixed values, a fuzzy system can naturally represent the system.

In our case, the dominant frequencies after each compaction are expected to lie around certain values. The premise parameters can thus be initialised efficiently, as these dominant frequencies can perfectly represent the mean of the membership functions. This gives the impression that an ANFIS system can effectively model this system.



4.2. ANFIS Compared to ANN (in general)

While a conclusive study has not been published yet, the majority of studies to compare the performance of the two have stated that ANFIS typically has a slightly better performance than ANN. [https://journals.sagepub.com/doi/pdf/10.1177/1847979018768421]

Another notable advantage is that ANFIS models can generate linguistic rule sets that can be understood by humans unlike ANN models which behave as black boxes.

An ANFIS system can be given accurate information about the properties of the system by selecting optimal membership functions because of its intuitive structure that maps directly to the system properties.

5. Complete Model (tentative)