Magnesium Corrosion Data Exploration

Elisabeth Schiessler

February 16, 2021

1 The Dataset

The given dataset describes how certain additives influence magnesium corrosion of material *ZE41*. It contains 60 samples (additives/compounds). The first three columns of the dataset contain the compound name, the target value *inhibition efficiency ZE41 / %*, as well as a scaled version of the target column. Furthermore the dataset contains 1260 columns of features (molecular descriptors).

All data except for the compound name is given in decimal values, with varying ranges. As a preprocessing step, all descriptors are scaled to the interval $[0, 1]$ using linear (min-max) scaling per molecular descriptor. We directly use the scaled version of the inhibition efficiency, *LinIE ZE41*, as the target variable.

10% of the training data (i.e. 6 samples) are randomly selected and withheld from all analysis and training methods to be used as a validation set, c.f. table 1. This set is not varied at any point.

Table 1: Set aside validation set (randomly selected).

Index	Compound description
0	3-Amino-124-triazole
5	4-methylsalicylicacid
13	35-Dinitrosalicylicacid
36	maleicacid
45	para-toluicacid
54	salicylhydrocamicacid

2 Introduction

Our aim is to identify a subset of the provided features (molecular descriptors) that is most relevant in predicting the target value. This subset should ideally be small enough to reduce the required number of experiments/measurements, while still being able to provide sufficient information such that a deep learning model can reach an adequate predictive quality. We introduce several methods for feature selection, and identify most relevant subsets in two categories:

- Low number of features: the $n \in \{1, 2, \ldots, 6\}$ most relevant molecular descriptors
- Medium number of features: the top 5 % most relevant molecular descriptors

We then train a number of deep learning models based on the selected groups of molecular descriptors, and compare the results to a model trained on the full dataset.

3 Statistical Analyses

3.1 Individual Most Relevant Features

A popular means for feature selection is to train a Random Forest Regressor and inspect the features that are used as support, i.e. used for decision making in any of the calculated decision trees that make up the random forest. To counter deterministic influences, we train 100 times each on 100 different random seeds (i.e. 10.000 runs in total), each time permuting the order of the columns. The supporting features of all trained trees are stored, and those features that are chosen most often are deemed to be the most relevant ones. The result of this method can be seen in table 2.

To get some perspective on the relevance towards the target value of each individual feature selected by this method, the dataset is expanded by an additional column containing a random value, which is also reset for each calculation. The random column is included in the random forest's support in 19.3% of all runs, which places it within one standard deviation above average compared to all features.

Table 2: Most relevant features as determined by Random Forests, and how often they were included in the support. Key statistics also provided.

Feature	Included in % of runs	rank
VE2_G/D	20.26	1
Eig14_EA(dm)	20.25	2
Mor31m	20.10	3
TDB04u	20.03	4
HATS1e	19.99	5
⋮	⋮	⋮
SpMAD_EA(bo)	19.58	63
random	19.30	242
mean	19.00	-
std	0.35	-
minimum	17.76	1261

3.2 Groups of Most Relevant Features

Instead of searching for individual most important features, we can also look for groups of n features that together most are able to most accurately predict, the target value. We do so by repeated application of recursive feature elimination (RFE). RFE searches for subsets of features by repeatedly fitting a chosen regression model, and discarding the least relevant features until only the desired number is retained. As the underlying regression model we chose a random forest regressor. The RFE algorithm is run 100 times (with varying random seed to counter deterministic behaviour in any random seed generators).

When selecting a low number of features, the top 3 obtained lists of n most relevant molecular descriptors are then compared for each $n \in \{1, 2, \ldots, 6\}$. As we can see from table 3, selecting the most relevant one to five molecular descriptors is a relatively clear decision, albeit with almost a tie for fourth place. Afterwards the number of times a combination wins the competition drops significantly.

Comparing this list to the method used in section 3.1, out of the top 5 most relevant feature group, only P_VSA_MR_5 and Mor22s individually have a higher relevance than average, being used in 19.39 % and 19.02 % of all supports respectively.

For the second variant where we select a medium number of features, no two runs of RFE pick the exact same combination. A total of 513 features is used over all runs, 6 of which are included in each run. They correspond to the group of 6 features most often used when just looking for 6 indivdual features, see table 3. 150 features are just used once, 316 (i.e. 62%) are used 5 times or less. The top 63 features

Table 3: Top 3 combinations per number of selected features as found by RFE. Third column indicates how often the given combination won the competition against potential other contestants.

n	Top 3 combinations per n	Won (x/100)
1	P_VSA_MR_5	71
	LUMO / eV	21
	Mor04m	8
2	P_VSA_MR_5, LUMO / eV	54
	P_VSA_MR_5, Mor04m	39
	LUMO / eV, Mor04m	5
3	P_VSA_MR_5, LUMO / eV, Mor04m	83
	P_VSA_MR_5, LUMO / eV, E1p	4
	P_VSA_MR_5, LUMO / eV, Mor22s	4
4	P_VSA_MR_5, LUMO / eV, Mor04m, E1p	29
	P_VSA_MR_5, LUMO / eV, Mor04m, Mor22s	27
	P_VSA_MR_5, LUMO / eV, Mor04m, P_VSA_LogP_2	13
5	P_VSA_MR_5, LUMO / eV, Mor04m, E1p, Mor22s	21
	P_VSA_MR_5, LUMO / eV, Mor04m, E1p, P_VSA_LogP_2	11
	P_VSA_MR_5, LUMO / eV, Mor04m, Mor22s, P_VSA_LogP_2	6
6	P_VSA_MR_5, LUMO / eV, Mor04m, E1p, P_VSA_LogP_2, Mor22s	11
	P_VSA_MR_5, LUMO / eV, Mor04m, E1p, Mor22s, HOMO / eV	8
	P_VSA_MR_5, LUMO / eV, Mor04m, E1p, P_VSA_LogP_2, HOMO / eV	6

are included in at least 30 % of all runs, cf. figure 1. Depending on the desired cut-off, this frequency distribution may come in handy for further fine tuning of the deep learning models (e.g. when deciding the number of input features to be used).

3.3 Random Search

A third alternative is to randomly choose n features and train a deep learning model on it. Repeating this process often enough (e.g. 100 times for 100 different random seeds), and retaining the combinations of features that yields the best result will produce the best scoring model. However this method has to be viewed with some caution, as we can no longer check each trained model with regard to how much overfitting has occurred, and no fine-tuning of the produced models can happen due to sheer amount of repetitions. We thus perform this variant only for $n = 5$ to demonstrate its abilities and provide a baseline for result comparison.

4 Deep Learning Models

We evaluate the results of the statistical analyses by training a number of deep learning models, and compare their performance on the validation set (cf. 1). To allow for better comparability, we train three different model types based on the number of selected features that are used as model input:

- MS - 3-5 input features
- MM - 63 input features
- ML - full model trained on the whole dataset

All models contain 3 hidden layers which use `relu` activation. They are trained for 25 epochs each using an Adam optimizer, and mean squared error as the loss function. Since the dataset is very small (only 54 training samples), for each model the input data is first passed through a Gaussian noise layer with $\mu = 0$

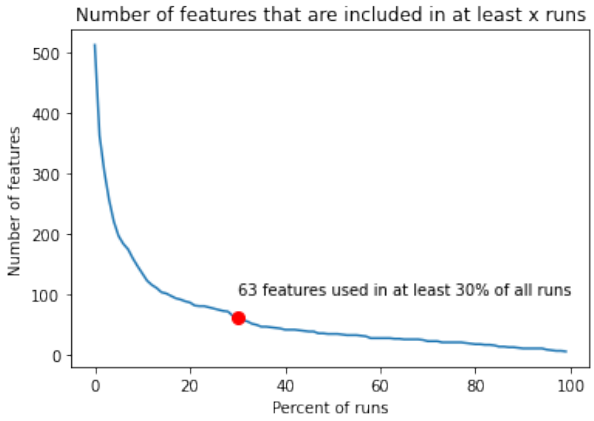

Figure 1: The recursive feature extraction (RFE) identifies a total of 513 features when tasked with selecting the 63 most relevant features 100 times.

and $\sigma = 0.1$. This layer adds some random noise (with a Gaussian distribution) in each epoch, which acts as a data augmentation technique and helps to improve generalization of the model as well as to reduce overfitting. The Gaussion noise layer is not active when predictions are made on the (previously unseen) validation data.

Hyperparameters that are varied based on model type are the number of units in each hidden layer, as well as the learning rate for the Adam optimizer, cf. table 4.

Table 4: Chosen hyperparameters for the different model types.

Model type	Layer sizes	Learning rate
MS	50-20-10	0.01
MM	50-20-10	0.05
ML	100-50-10	0.005

5 Results and Discussion

Based on the analytic methods and model types presented in sections 3 and 4 respectively, we train a total of 8 different models, using varying sets of features as inputs:

- MS type models:
  - M3a: top 3 individual most relevant features (determined by Random Forests)
  - M3b: top 3 grouped most relevant features (determined by RFE)
  - M5a: top 5 individual most relevant features
  - M5b: top 5 group most relevant features
  - M5c: 5 randomly selected input features (only averages are reported)
- MM type models:
  - M63a: top 63 individual most relevant features
  - M63b: top 63 group most relevant features

- ML type models:
  - MF: full model trained on the whole dataset

As expected, the individual most relevant low number of features (M3a, M5a) does not necessarily work best as a group and thus the associated models yield very poor results. The 3- and 5-parameter models trained on the selected group features (M3a, M5a) on average perform better than models trained on 5 randomly selected features, as does the model trained on the full dataset (MF).

The medium sized models are quite similar in their performance, but achieve a slightly worse validation loss overall.

The dataset size should be kept in mind for any considerations regarding prediction quality. With only 60 samples, 10% of which were withheld from all training and evaluation processes, we can not expect any result to generalise well on unknown data. Adding a Gaussian noise layer helps to improve prediction quality for unknown data.

Finally the choice of validation set can also play a huge role in prediction quality. For a more reliable result, all steps (including statistical analyses on training sets) should be repeated several times for a number of different validation sets, and then cross validated, to get a better feel for general behaviour.

Table 5: All trained models by type, with validation loss and individual performance on the validation set. All predicted values are rounded to nearest integer and compared to the target variable *inhibition efficiency ZE41 / %*.

Type	Model	Validation Loss	Validation results						
			Index	0	5	13	36	45	54
			True value	-157	39	38	12	-6	-17
MS	M3a	0.0486		-15	-47	-20	-18	-48	-52
	M3b	0.0140		-90	6	104	14	-6	-6
	M5a	0.0786		47	-55	1	-8	-54	-58
	M5b	0.0112		-99	9	59	69	-14	-20
	M5c	0.0341		(Average statistics only)					
MM	M63a	0.0250		-48	-31	33	-6	-31	-32
	M63b	0.0220		-199	44	148	-8	-11	25
ML	MF	0.0145		-66	10	67	22	-19	-11