Position detection of particles using RSD Regression Problem

Monco Giovanni, Spada Erika
Politecnico di Torino
Student id: s315001, s318375
s315001@studenti.polito.it, s318375@studenti.polito.it

***Abstract*—The report aims to develop a machine learning pipeline using regression techniques for detecting the position of particles through RSD sensors.**
The proposed approach includes feature selection and feature engineering steps to better characterize the data and remove noisy features. Using different regression algorithms, the solution shows promising results overall.

I. Problem overview

The main purpouse of the project is detecting the positions of particles passing through a sensor called RSD (Resistive Silicon Detector). This sensor has a 2-dimensional surface and 12 metallic pads with a star shape used to measure a signal.

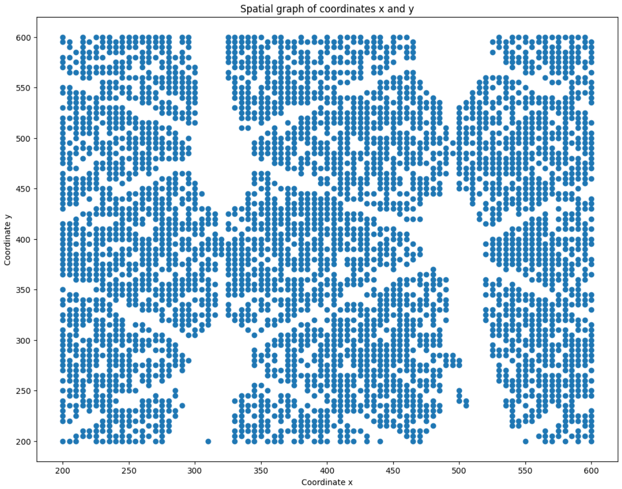

Fig. 1. Spatial distribution of the points.

Each i-esim pad extracts a category of measurements:

- pmax[i]: the magnitude of the positive peak of the signal (in mV);
- negpmax[i]: the magnitude of the negative peak of the signal (in mV);
- area[i]: the area under the signal;
- rms[i]: the root mean square (RMS) value of the signal;
- tmax[i]: the delay (in ns) from a reference time when the positive peak of the signal occurs.

The task is to build a data science pipeline that predicts for each passage of particles the (x,y) coordinates of the position. The dataset is composed by 514,000 records:

- 385,500 rows for the development set where (x,y) coordinates are known;
- 128,500 for the evaluation set.

At first, to visualise the data spatially, a scatter plot is made. Looking at Fig.(1), one can see, as anticipated in the text, that some areas are not covered by any points because pads or cables are present.

As mentioned in the text, although there are 12 pads, 18 measurements per category of features are present due to noisy data. Analysing the correlation of the measurements of each pad, as in Fig.(2), it immediately becomes clear that 6 matrices have an anomalous behaviour with respect to the other 12.

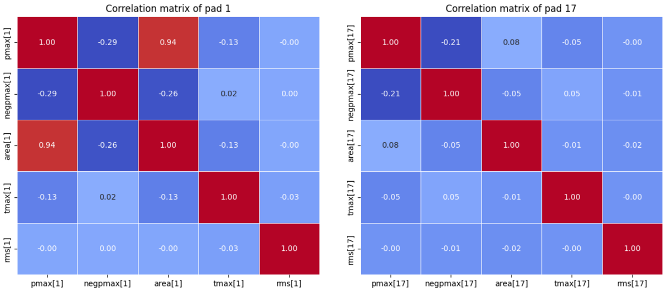

Fig. 2. Comparison between correlation matrix of correct pad 1 and noisy pad 17.

In particular, one would expect that for each pad the area would have a strong correlation with pmax, since the higher the value of pmax the more the area under the curve increases and vice versa the higher negpmax the more the area tends to shrink. Within the six outlier matrices, there is no significant correlation between the data and we therefore assume that the measurements pmax[i], negpmax[i], area[i] are wrong for pads 0,7,12,15,16 and 17. To extend the analysis further, the distributions of tmax and rms are plotted and it emerges that the values of tmax of pads 0,7,12,15,16 and 17 show different characteristics compared to the other twelve as in Fig.(3), while for rms, only the data of pads 16 and 17 are anomalous.

As a result of these considerations, it has been supposed that pads 0,7,12,15,16 and 17 are unreliable and therefore all the corresponding measurements are not usable because they are affected by noise.

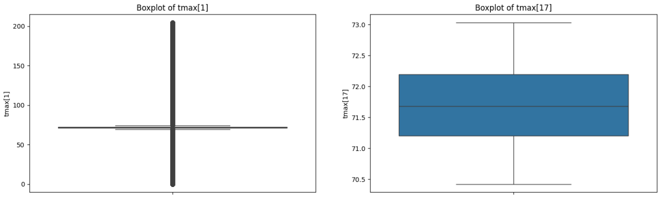

Fig. 3. Comparison between tmax feature boxplots of correct pad 1 and noisy pad 17.

Last, it's possible to notice that the dataset is well balanced, as there are in total 3885 positions and for each of them there are 100 records. In addition, all the features are in float format and have no missing values.

II. Proposed approach

A. *Preprocessing*

At first, all the features of the wrong pads are removed from the dataset.

Next, a check is made for incorrect data regarding the positivity of pmax and the negativity of negpmax. It turns out that 3 rows have incompatible values with the definition of the physical quantities.

The presence of outliers for each feature is checked by means of the spatial distribution of the points according to a feature at time, as shown in Fig.(4). 400 outliers of the features negpmax[i] with i=1,3,4,5,6,8,9,10,11,13,14 are detected and removed.

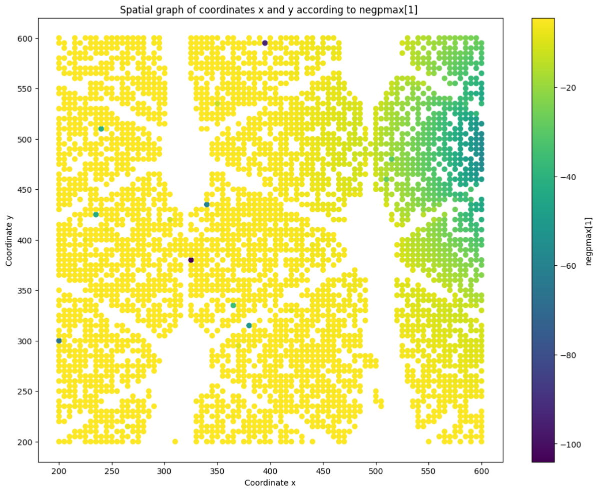

Fig. 4. Detection of outliers of the feature negpmax[1].

Afterwards, the feature range[i] = pmax[i] - negpmax[i] is introduced for each pad, bringing a total of 12 new features. This quantity is considered relevant since it covers the entire range of magnitude values of each signal, as confirmed by the feature importance later.

Finally, to reduce the complexity of the dataset and improve computational performance, feature selection is performed to identify a subset of features to be used to train the model. In particular, due to the use of methods based on decision trees (as analysed later on), feature importance is calculated and only columns with higher value of it are kept. This results in the rms and tmax features making a negligible contribution in terms of feature importance, so they are removed from the dataset, for a total elimination of 24 features.

The final dataset used to train the model presents 385097 rows and 50 columns (2 targets and 48 features).

B. *Model selection*

To handle the size of the dataset and model possible non-linear relationships, algorithms such as Linear Regressor or Ridge were discarded a priori, in preference to more complex methods such as SVR and Random Forest. Furthermore, since previous studies show the effectiveness of the Random Forest [1], other ensemble methods such as the ExtraTreesRegressor have been explored.

The principal characteristics of the chosen algorithms are summarised below:

- SVR: it is an algorithm that can apply a non-linear transformation to the data using a kernel function. Its goal is to find the hyperplane that maximizes the margin of errors between predicted and actual values. We chose to evaluate it because it is generally one of the best performing machine learning algorithms, but as we did not examine it in detail, its hyperparameter tuning was less accurate.
- Random Forest Regressor: it is a powerful ensemble technique that combines multiple decision trees trained on different portions of the training set and subsets of features to make accurate predictions. This technique is widely used to enhance performance and avoid overfitting. Although it may be less interpretable in regression tasks than in classification problems, it can provide valuable insights into feature importance, as discussed in section II-A. Random forests, like decision trees, work on one feature at a time, so normalization is not required. We chose to use this model due to its ability to handle complex data with easily.
- Extra Trees Regressor: this is an alternative to random forests that uses extremely randomized trees. Similar to random forests, a random subset of candidate features is used. However, instead of searching for the most discriminative thresholds, these are randomly generated for each candidate feature and the best of these randomly generated thresholds is chosen as the splitting rule [2]. This powerful alternative to random forest is used because it has been shown to reduce the variance of the model.

It is worth mentioning that the x and y coordinates are predicted independently, which necessitates training two distinct regressors. To accomplish this, we used the MultiOutputRegressor [3] that fits one regressor for each target.

For all regressors, the best configuration of hyperparameters has been determined using a grid search, as discussed in the following section II-C.

C. Hyper-parameters tuning

The tuning of hyper-parameters is an essential process for training the model, but at the same time computationally expensive. In this context, where a grid search is necessary to explore the best solution of hyper-parameters to apply to the models used, two main problems arise:

1) The size of the dataset $\longrightarrow$ To handle this problem, the grid search is applied to a subset of the dataset with approximately 40 % of the size of the initial dataset. In terms of performance, it proves to be the optimal choice as it drastically reduces computational time, but in terms of accuracy, with the same hyper-parameters, it only presents a deterioration of 0.14 compared to the original dataset (results obtained by checking the leaderboard score). Furthermore, in order to maintain a representative and balanced subset, to create it, the first 40 rows of the original dataset are selected for each x and y equal, thus always resulting in 3855 points, with 40 measurements each.
2) The division of the development set in two non-overlapping parts $\longrightarrow$ In order to have a complete separation between the two sets, all events recorded for a specific pair of coordinates have been assigned to either the training or test set.

The tuning is run out with an 80/20 split and the criteria used to assess the performances of the models is the average (Euclidean) distance of the predictions from the targets, i.e.:

$$d = \frac{1}{n} \sum_{i} \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} .$$

The hyper-parameter combinations for each algorithm are summarised in TABLE 1 below.

Model	Hyperparameter	Values
SVR	kernel	{poly, rbf, sigmoid}
	gamma	scale
	C	$\{1, 10, 50, 100, 500\}$
Random Forest	n_estimators	$\{100, 300, 500\}$
Extra Trees	max_depth	{15, 30, 40, None }
	min_samples_split	$\{2, 5, 10\}$
	min_samples_leaf	$\{1, 2, 4\}$
	max_features	{ sqrt, log2 }
	bootstrap	{True, False }

TABLE I

III. Results

The tuning of hyperparameters shows the following best performing configurations and the matching results on the 20% test set:

Model	Hyperparameter	Best Values	Score
SVR	kernel	rbf	**5.301**
	gamma	scale	
	C	100	
Random Forest	n_estimators	500	**4.992**
	max depth	30	
	min_samples_split	2	
	min_samples_leaf	1	
	max_features	sqrt	
	bootstrap	False	
Extra Trees	n_estimators	500	**4.852**
	max depth	40	
	min_samples_split	2	
	min_samples_leaf	1	
	max_features	sqrt	
	bootstrap	False	

TABLE II

It can be deduced from the table that both ensemble methods perform better than SVR, and in particular the Extra Trees Regressor ranks as the best algorithm considered.
Then, using the combination of the best hyper-parameters, the ensemble methods have been trained using no longer the subset, but the whole dataset, and have been used to predict the coordinates of the evaluation set. The public scores obtained for Random Forest Regressor and Extra Trees Regressor are respectively 4.698 and 4.636 .
They confirm the results obtained through the grid search.

IV. Discussion

The proposed strategy achieves significantly better results than the defined baseline of 6.629. Below are summarised some aspects that might merit further consideration in order to improve the results obtained:

- Further noise removal through the elimination of rows.
  Since each position has 100 measurements, it was first thought to remove rows for each position that had values of some features that were distant from those with the same coordinates. However, unexpectedly this process worsened the accuracy of the predictions and was therefore abandoned.
  Certainly, the exploration of new methods to remove noise and 'lighten' the dataset would bring great benefits in terms of performance and computational costs.
- Deeper Grid Search analysis.
  A thorough analysis could be conducted by expanding the range of hyperparameters and their possible values.
- Other advanced machine learning techniques.
  Finally, advanced machine learning techniques such as the Gradient Boosting Regressor [4] and neural networks can be considered and explored.

References

[1] F. Siviero, F. Giobergia, L. Menzio, F. Miserocchi, M. Tornago, R. Arcidiacono, N. Cartiglia, M. Costa, M. Ferrero, G. Gioachin, M. Mandurrino, and V. Sola, "First experimental results of the spatial resolution of rsd pad arrays read out with a 16-ch board," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 1041, p. 167313, 2022.

[2] *https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html.*

[3] *https://scikit-learn.org/stable/modules/generated/sklearn.multioutput.MultiOutputRegressor.html*.

[4] F. Siviero, R. Arcidiacono, N. Cartiglia, M. Costa, M. Ferrero, F. Hegner, M. Mandurrino, V. Sola, A. Staiano, and M. Tornago, “First application of machine learning algorithms to the position reconstruction in resistive silicon detectors,” *Journal of Instrumentation*, vol. 16, p. P03019, 03 2021.