

PAPER • OPEN ACCESS

## First application of machine learning algorithms to the position reconstruction in Resistive Silicon Detectors

To cite this article: F. Siviero *et al* 2021 *JINST* **16** P03019

View the [article online](#) for updates and enhancements.



**IOP | ebooks**<sup>TM</sup>

Bringing together innovative digital publishing with leading authors from the global scientific community.

Start exploring the collection—download the first chapter of every title for free.

## First application of machine learning algorithms to the position reconstruction in Resistive Silicon Detectors

F. Siviero,<sup>a,b,\*</sup> R. Arcidiacono,<sup>b,c</sup> N. Cartiglia,<sup>b</sup> M. Costa,<sup>a,b</sup> M. Ferrero,<sup>b,c</sup> F. Legger,<sup>b</sup> M. Mandurrino,<sup>b</sup> V. Sola,<sup>b</sup> A. Staiano<sup>b</sup> and M. Tornago<sup>a,b</sup>

<sup>a</sup>Università degli Studi di Torino, Torino, Italy

<sup>b</sup>INFN Sezione di Torino, Torino, Italy

<sup>c</sup>Università del Piemonte Orientale, Vercelli, Italy

E-mail: [federico.siviero@edu.unito.it](mailto:federico.siviero@edu.unito.it)

**ABSTRACT:** RSDs (Resistive AC-Coupled Silicon Detectors) are n-in-p silicon sensors based on the LGAD (Low-Gain Avalanche Diode) technology, featuring a continuous gain layer over the whole sensor area. The truly innovative feature of these sensors is that the signal induced by an ionising particle is seen on several pixels, allowing the use of reconstruction techniques that combine the information from many read-out channels. In this contribution, the first application of a machine learning technique to RSD devices is presented. The spatial resolution of this technique is compared to that obtained with the standard RSD reconstruction methods that use analytical descriptions of the signal sharing mechanism. A Multi-Output regressor algorithm, trained with a combination of simulated and real data, leads to a spatial resolution of less than 2  $\mu\text{m}$  for a sensor with a 100  $\mu\text{m}$  pixel. The prospects of future improvements are also discussed.

**KEYWORDS:** Particle tracking detectors (Solid-state detectors); Data processing methods; Timing detectors

ARXIV EPRINT: [2011.02410](https://arxiv.org/abs/2011.02410)

\*Corresponding author.

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Signal sharing</b>	<b>2</b>
<b>3</b>	<b>Position reconstruction</b>	<b>6</b>
<b>4</b>	<b>Position reconstruction using Multi-Output regression</b>	<b>7</b>
<b>5</b>	<b>Results</b>	<b>12</b>
5.1	Discussion of the results	15
5.1.1	The optimum value of <i>training noise</i>	15
5.1.2	Systematic uncertainties	15
5.2	On the expected spatial resolution	15
<b>6</b>	<b>Conclusions and future plans</b>	<b>16</b>

---

## 1 Introduction

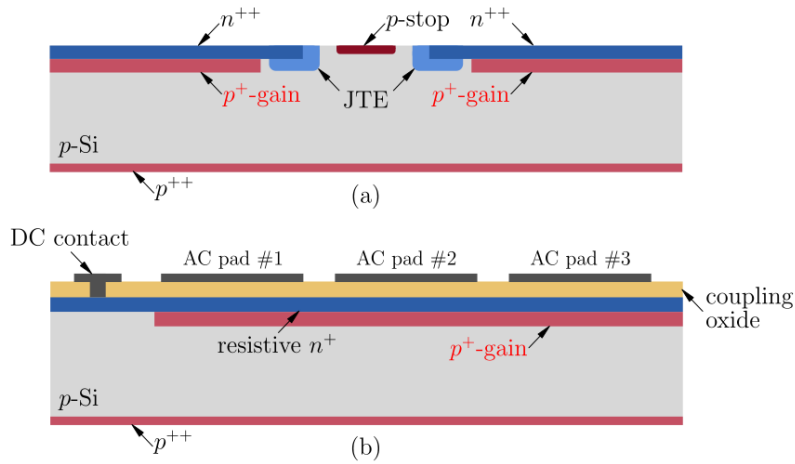
RSDs (Resistive AC-Coupled Silicon Detectors) are n-in-p silicon sensors based on the LGAD (Low-Gain Avalanche Diode) technology, featuring a continuous gain layer over the whole sensor active area [1].

The RSD design is an evolution of the LGAD design, as it combines internal gain with internal signal sharing. The presence of internal gain assures the excellent timing performances of standard LGADs while internal signal sharing enables the ability of precisely reconstruct the impact position using the information provided by the shared signals.

Moreover, the RSD design ensures a 100% fill-factor, meaning that the ratio between the sensor active and total areas is equal to 1, whereas standard LGADs reach a fill-factor ranging from 70% to 90% [2].

A comparison between the RSD and LGAD designs is shown in figure 1.

Signal formation in RSD is based on the principle of AC-coupled resistive read-out. In this design, the n-doped read-out layer is manufactured to be resistive, i.e., its doping level has been decreased with respect to the standard design. When the signal induced by an ionizing particle on the  $n^+$  layer discharges to ground, it uses the lowest impedance path. In resistive read-out, this path is via the AC-pads, grounded by the frontend electronics. This mechanism naturally induces signals on multiple AC pads. The AC read-out is obtained through metal pads capacitively coupled to the detector's bulk via a dielectric layer deposited between the metal pads and resistive  $n^+$  layer. The dielectric layer thickness and the  $n^+$  resistivity are the parameters governing the signal induction on the pads [3].



**Figure 1.** Cross-section of an LGAD (a) compared to an RSD (b).

The sensors measured in this work were part of the first RSD production (RSD1) manufactured at Fondazione Bruno Kessler (FBK). RSD1 features pixelated sensors with different geometries: in this contribution,  $3 \times 3$  pixel matrices are considered, all with the same  $n^+$  layer resistivity and dielectric thickness. 4 different geometries are explored: 100–200  $\mu\text{m}$ , 50–100  $\mu\text{m}$ , 150–300  $\mu\text{m}$ , 200–500  $\mu\text{m}$ , where the first number indicates the pad size and the second one the sensor pitch [4]. The difference between the sensor pitch and the pad size will be named *interpad* in the following. As an example, the *interpad* of the 200–500  $\mu\text{m}$  geometry is 300  $\mu\text{m}$ .

## 2 Signal sharing

Signal formation in RSD is different from that of a standard LGAD. The signal, formed on the  $n^+$  resistive layer due to the drift of  $e/h$  carriers in the bulk, spreads toward ground. The AC metal pads, grounded via the read-out electronics, offer to fast signals a path to ground with an impedance far lower than that provided by the  $n^+$  resistive sheet. For this reason, the fast signal (about 1 ns for 50- $\mu\text{m}$  thick sensor) becomes visible on the AC pads, charging the capacitors formed by the AC metal pads and the  $n^+$  layer, with a delay that increases with distance from the impact point. If the metal pad is right above the impact point, there is no delay. The main advantage of this read-out scheme is that the signal has the same characteristics of a standard LGAD, thus allowing precise timing, and it is shared among multiple pads, which is the key element for precise position reconstruction.

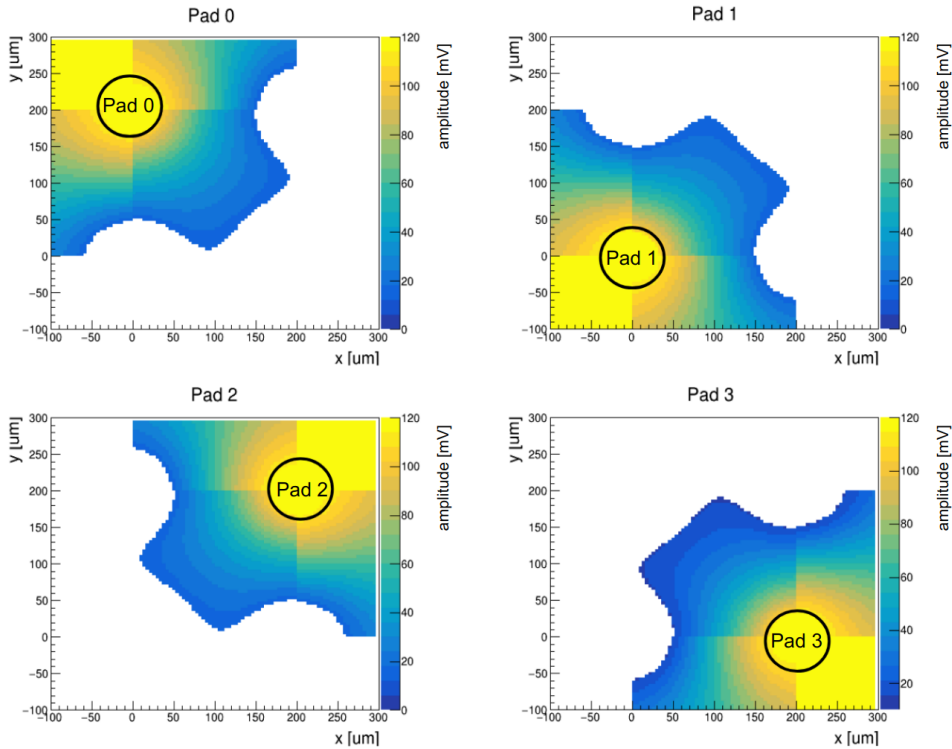
When a particle crosses the detector, nearby pads see a signal with an amplitude and delay depending on the hit position: closer pads will see a larger and less delayed signal than the further ones. Both delay and attenuation are caused by the propagation of the signal in the resistive layer. Signal sharing in RSD is such that there is only one position on the detector surface that produces a given signal split among 3 or more pads. Therefore 3 pads is the minimum number of active pads required by the reconstruction program to unambiguously identify the particle hit position. As a consequence, all regions within the sensor active area where less than 3 pads are active are inefficient, meaning that the hit position cannot be uniquely identified by the reconstruction program.

Two analytical models of signal sharing in RSD are presented in [6]. The first model, the so-called RSD Main Formula (MF), assumes that the impedance from the particle impact point to the read-out metal pad increases logarithmically with distance. The fraction of the total signal seen by each read-out pad as a function of its distance can be expressed by the following expression:

$$S_i(\alpha_i, r_i) = \frac{\frac{\alpha_i}{\ln(r_i)}}{\sum_{j=1}^n \frac{\alpha_j}{\ln(r_j)}} \quad (2.1)$$

where  $S_i$  is the fraction seen by the  $i$ -th pad;  $\alpha_i$  is the angle of view of the pad from the hit position;  $r_i$  is the distance between the hit position and the metal pad;  $n$  the number of active pads.

Using this model, it is possible to compute the fraction of the signal seen on each pad for any RSD geometry. Figure 2 shows how a signal with an amplitude of 120 mV (corresponding to a gain of  $\sim 15$ , which is a typical working point for LGAD devices) is shared in a  $2 \times 2$  RSD sensor with 100–200  $\mu\text{m}$  geometry (circular pads have been drawn for simplicity, although the RSD1 prototypes have square pads).



**Figure 2.** Amplitude seen by each of the 4 read-out pads of an RSD as a function of the impact position, considering a total signal amplitude of 120 mV. The colored axis represents the signal amplitude seen by the corresponding read-out pad when a particle hits the sensor in a given  $x$ - $y$  position. The geometry of this sensor is 100–200  $\mu\text{m}$ : the regions circled in black are the 100  $\mu\text{m}$  metal read-out pads, while the distance between the centers of two pads (the pitch) is 200  $\mu\text{m}$ .

The second analytical model of signal sharing in RSD differs from the MF model by assuming a linear attenuation of the signal with distance. This model is referred to as the “Linear Attenuation” (LA) model.

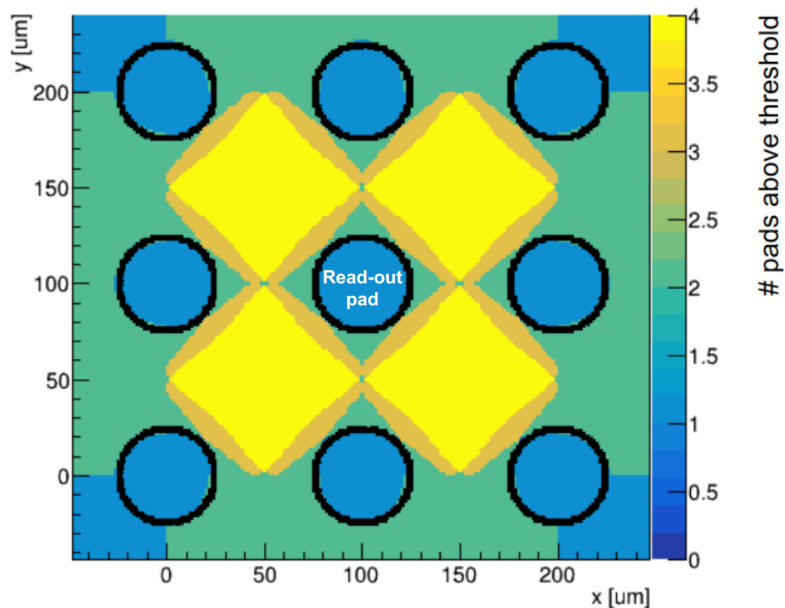
To assure the selection of real signals, only amplitudes above  $3\text{--}5\sigma$  noise level should be used in the reconstruction. This minimum amplitude also determines the maximum distance from the impact point at which the pad can be used in the reconstruction. In the present study, a threshold of 15 mV is used to safely reject the noise. The white areas in figure 2 indicates positions where the pad cannot be used in the reconstruction as the signal is below 15 mV.

It is important to note that, according to equation (2.1), when a particle hits directly a metal pad, the signal is visible only in that single pad as its resistance is null [7]. As a consequence, the spatial resolution for particles hitting the metal is that of a pixel detector with binary read-out:  $\sigma_x \sim \text{metal size}/\sqrt{12}$ .

Since it is different between metal and non-metal regions, the RSD spatial resolution results to be dependent on the hit position. In future productions, the design of the metal pads will be optimized in order to minimize such effect.

Assuming a given signal total amplitude, an RSD analytical model is able to predict the amplitude on each pad near the impact point. Using this information, the number of active pads (defined as those that see a signal above the minimum threshold) as a function of the hit position can be derived.

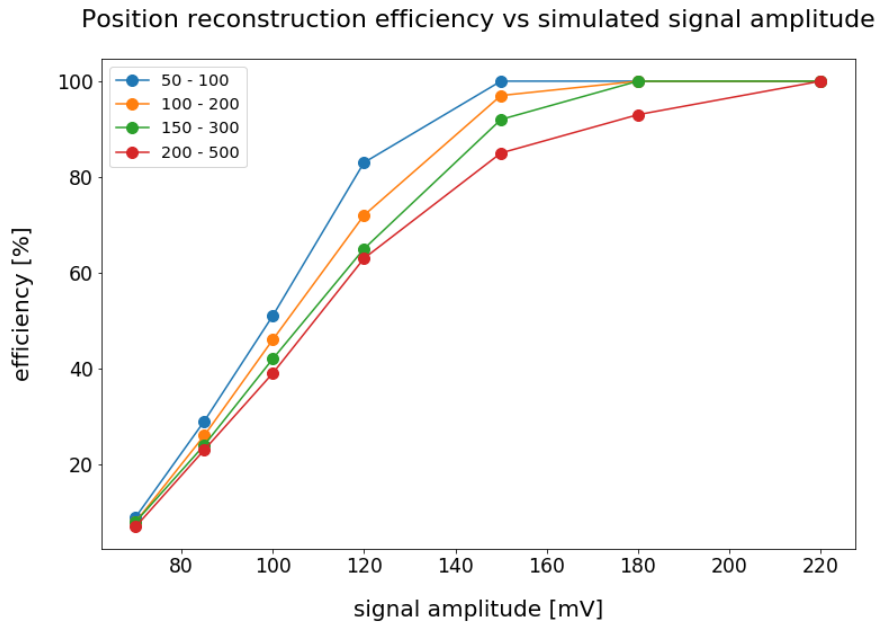
Figure 3 shows such 2D-map for a  $3 \times 3$  pad RSD with  $50\text{--}100\ \mu\text{m}$  geometry, considering a signal with a total amplitude of 120 mV and a threshold of 15 mV. Yellow and brown colors identify the efficient positions, where at least 3 pads see a signal; whereas in the blue and green parts only 1 or 2 pads are active, with consequent loss in position resolution.



**Figure 3.** MF prediction of the number of pads that see a signal above 15 mV, assuming a total signal amplitude of 120 mV (gain  $\sim 15$ ), in a  $3 \times 3$  RSD with  $50\text{--}100\ \mu\text{m}$  geometry. The colored axis represents the number of pads that see a signal larger than 15 mV when a particle hits the sensor in a given x-y position. The blue regions circled in black are the  $50\ \mu\text{m}$  metal read-out pads: only one pad sees the signal there, because signal sharing does not occur underneath the metal.

These maps are useful to determine the regions where the position cannot be reconstructed unambiguously using the signal sharing technique (only 1 or 2 pads active) and to calculate the position reconstruction efficiency, defined as the fraction of active area not covered by metal with at least 3 active pads.

A possible way to improve the performances of the reconstruction method is to increase the sensor gain: since the noise threshold is fixed to a given value, a larger signal will be above the noise threshold along a wider area. Figure 4 shows the position reconstruction efficiency as a function of the signal total amplitude. The efficiency improves by increasing the sensor gain and devices with larger pitch require higher gain to reach full efficiency, as expected. Hence, a proper tuning of the gain is key to reach full efficiency.



**Figure 4.** Position reconstruction efficiency as a function of the signal total amplitude for different RSD geometries.

The gain definition in RSD is similar to that of standard LGADs [19]:

$$\text{Gain}_{\text{RSD}} = \frac{Q_{\text{RSD}}}{Q_{\text{no-gain}}} \quad (2.2)$$

$Q_{\text{RSD}}$  is the total charge generated by a Minimum Ionizing Particle (MIP) and collected by the RSD at a given bias voltage;  $Q_{\text{no-gain}}$  is the charge collected in the same conditions by an equivalent sensor which does not have the gain layer (the RSD1 production features several of these no-gain sensors specifically for the purpose of measuring the gain level). The RSD gain is therefore defined as the collected charge normalized to the charge collected by a sensor equivalent to the sensor under test, but without gain.

In this work, the devices under test were operated at 3 different gain levels: 12, 17 and 24, corresponding to a reconstruction efficiency ranging from 20–30% for the lowest gain, to ~ 100% for the highest one.

For the purpose of this work, focusing on the precision of the reconstruction algorithm, only the events falling in regions of the devices with 100% 3-pads coverage are selected.

Considering a sensor with given *pitch* and the centers of its pads in:  $(0, 0)$ ,  $(0, \textit{pitch})$ ,  $(\textit{pitch}, 0)$  and  $(\textit{pitch}, \textit{pitch})$ , the regions used in the experimental measurements for each geometry are presented in table 1.

**Table 1.** X/y range of the central regions considered as experimental regions, assuming that the centers of the 4 read-out pads have coordinates:  $(0, 0)$ ,  $(0, \textit{pitch})$ ,  $(\textit{pitch}, 0)$  and  $(\textit{pitch}, \textit{pitch})$ .

Geometry	$x_{\min}$ [ $\mu\text{m}$ ]	$x_{\max}$ [ $\mu\text{m}$ ]	$y_{\min}$ [ $\mu\text{m}$ ]	$y_{\max}$ [ $\mu\text{m}$ ]
50–100 $\mu\text{m}$	30	70	30	70
100–200 $\mu\text{m}$	60	140	60	140
150–300 $\mu\text{m}$	100	200	100	200
200–500 $\mu\text{m}$	150	350	150	350

This study will be repeated in the near future, as fully efficient RSD sensors will soon become available.

### 3 Position reconstruction

A detailed description of the RSD position reconstruction techniques is given in [6], here we summarize the principles. In RSD, a point on the surface is uniquely identified by the relative amplitudes seen by the nearby pads and this property can be exploited to reconstruct very accurately the impact point position. In the analytical models of reconstruction, the impact point is identified by comparing the measured signal sharing to a lookup table that contains for each position (for example every  $1 \times 1 \mu\text{m}^2$ ) the predicted signal sharing. In [6], both MF and LA models were used to create the look-up tables.

The accuracy of the position reconstruction on RSD can be expressed as the sum of 3 terms:

$$\sigma_{\text{RSD}}^2 = \sigma_{\text{jitter}}^2 + \sigma_{\text{algorithm}}^2 + \sigma_{\text{sensor}}^2. \quad (3.1)$$

- $\sigma_{\text{jitter}}$ : this term represents the spatial uncertainty due to the electronic noise. The effect of the electronic noise is to change the amplitude by  $\sigma_{\text{el-noise}}$ , creating an uncertainty in the localization that can be written as:

$$\sigma_{\text{jitter}} = \frac{\sigma_{\text{el-noise}}}{\frac{dV}{dx}} = \frac{\sigma_{\text{el-noise}}}{\frac{\text{Amplitude}}{\text{Interpad}}} \quad (3.2)$$

- $\sigma_{\text{algorithm}}$ : the reconstruction code uses algorithms to infer the hit position from the measured signals. This can be done analytically, for example using the MF lookup table, or with more advanced techniques such as those presented in this paper. In all cases,  $\sigma_{\text{algorithm}}$  represents the uncertainty of the method selected;
- $\sigma_{\text{sensors}}$ : this term groups all sensor imperfections contributing to an uneven signal sharing among pads. The most obvious is a varying  $n^+$  resistivity: a 2% difference in  $n^+$  resistivity will turn an equal split between two pads into a non-equal split, yielding to an incorrect impact position reconstruction.



In the following part of this paper, the  $\sigma_{\text{RSD}}$  resulting from a reconstruction method based on a Multi-Output regressor is compared to that of the MF analytical method.

#### 4 Position reconstruction using Multi-Output regression

In RSD, each pad sees a modified version of the original signal. During the propagation on the  $n^+$  resistive surface, the signal becomes smaller, wider, with slower leading and falling edges, and is delayed. Each of these aspects is a valuable clue that should be used in the reconstruction of the hit position and time. For this reason, signal sharing among different pads leads naturally to the use of multivariate analysis techniques where many inputs contribute to the determination of the outputs.

On the other hand, analytical methods such as MF and LA use only the information on the shared amplitude, and they cannot use extra parameters, because the analytic description of how such parameters change with position is not very accurate. Thus, multivariate analysis techniques are intrinsically more powerful and can lead to better accuracy.

In order to efficiently reconstruct the hit position, a Multi-Output Regressor algorithm, using a Gradient Boosting Regressor as an estimator, has been developed and trained [9].

Gradient Boosting is a regression technique, used to develop a prediction model, which works in a forward, stage-wise manner [10, 11]. In a given analysis,  $N$  variables are selected to define an input vector  $\vec{x} = (x^0, x^1, x^2, \dots, x^N)$ . A set of  $n$  input vectors  $\{\vec{x}_1, \vec{x}_2, \vec{x}_3, \dots, \vec{x}_n\}$  will produce  $n$  output variables  $\{y_1, y_2, y_3, \dots, y_n\}$ . The purpose of Gradient Boosting is to find a function  $F(\vec{x}_i)$  such that:

$$F(\vec{x}_i) = y_i, \quad i = 1, \dots, n \quad (4.1)$$

$y_i$  is always a scalar, whereas the input variables defining the input vector  $\vec{x}_i$  can either be one ( $N = 1$ ) or more.

In the following, the index  $i$  will represent the  $i$ -th element of the input or output dataset, while  $k$  will indicate the  $k$ -th component of a vector.

Gradient Boosting proceeds in forward stages, in order to increasingly smooth the difference between the output variables and the values predicted by  $F$ . This is achieved by introducing at each stage of the algorithm a *weak learner*  $h$ , which represents the *residuals*, i.e. the discrepancy between  $F(\vec{x}_i)$  and  $y_i$  [12]:

$$h(\vec{x}_i) = y_i - F(\vec{x}_i) \quad (4.2)$$

The algorithm progressively reduces the *residuals*, in order to reach the best estimate for  $F$ .

The addition of  $h(\vec{x}_i)$  to  $F(\vec{x}_i)$  provides the functional form for the next stage of the algorithm. Therefore, at the  $m$ -th step ( $m$  represents the depth of the algorithm),  $F$  is equal to:

$$F_m(\vec{x}_i) = F_{m-1}(\vec{x}_i) + h_m(\vec{x}_i) = y_i \quad (4.3)$$

The algorithm starts with an *initial guess*  $F_0$  (a constant value), which usually is simply the average value of the output variables  $y_i$ :

$$F_0 = \sum_i \frac{y_i}{n} \quad (4.4)$$

Hence,  $F_m(\vec{x}_i)$  can be written as:

$$F_m(\vec{x}_i) = F_0 + \sum_{j=1}^m h_j(\vec{x}_i) = y_i \quad (4.5)$$

which clarifies the stage-wise nature of the algorithm. The sequential addition of *weak learners* is helpful because it produces an ensemble of weak predictive models which is much stronger than the single model, thus resulting in better performances.

Machine learning algorithms usually learn by means of a *loss function*, which evaluates how well a model fits the training data. In this work, the *loss function* is the *least squares*:

$$L(y_i, F(\vec{x}_i)) = \frac{(y_i - F(\vec{x}_i))^2}{2} \quad (4.6)$$

The sum over the number of elements in the input dataset of the *loss function*  $L$ , is the function  $J$ :

$$J = \sum_i L(y_i, F(\vec{x}_i)) \quad (4.7)$$

and it is straightforward to prove that:

$$-\frac{\partial J}{\partial F_{m-1}(\vec{x}_i)} = y_i - F(\vec{x}_i) = h_m(\vec{x}_i) \quad (4.8)$$

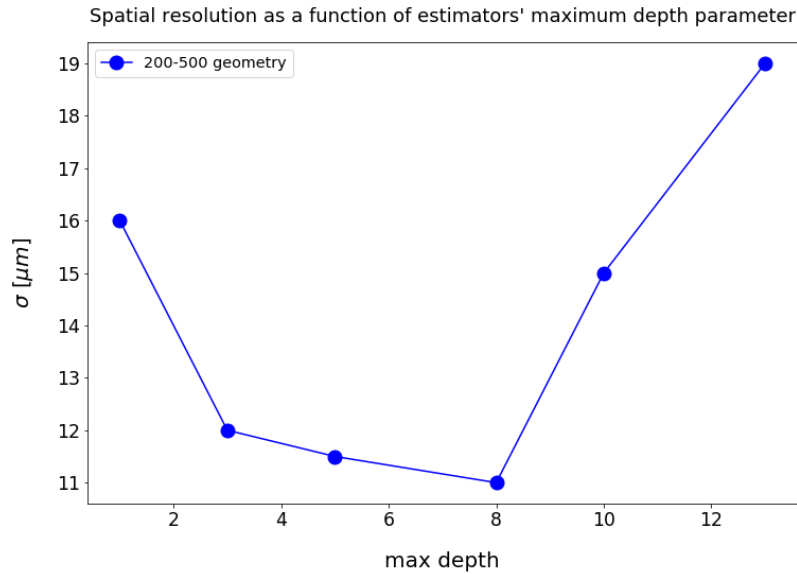
Therefore, the *residuals* can be treated, in this specific case, as the negative gradient of  $J$ . That is an important aspect, since, instead of minimizing the *residuals*, the algorithm can aim at minimizing a gradient, which is easily accomplished by the Gradient Descent technique [13]. Hence, the objective of Gradient Boosting is eventually the minimization, at each stage of the algorithm, of  $\frac{\partial J}{\partial F(\vec{x}_i)}$ .

In a typical regression problem, the output variable  $y_i$  is a scalar. Multi-Output is an evolution of the standard regressor, as it is able to predict an output variable which is a vector of  $N'$  real-valued numbers:  $\vec{y}_i = (y_i^{(1)}, y_i^{(2)}, \dots, y_i^{(N')})$ . In this work, multi-output regression has been implemented using the *Single-Target Method* [14]. This method requires the multi-output models to be divided in  $N'$  single predictive models ( $N'=2$  in this work) which are trained on transformed training datasets  $D^k$ :

$$D^k = \{(\vec{x}_1, y_1^k), \dots, (\vec{x}_n, y_n^k)\}, k \in \{1, \dots, N'\} \quad (4.9)$$

Each model predicts a single output variable and, finally, all predictions are concatenated in order to provide the final, multi-output variable. The drawback of this approach is that the potential relations between the predicted variables are not exploited [14]. For the purpose of the present work, that aspect has not been found to be an issue.

The Multi-Output Regressor algorithm developed in this work takes 8 input variables: the 4 amplitudes seen by the read-out pads ( $A_l$ ) and the same amplitudes normalized to the total amplitude ( $A_l/\sum_{l=1}^4 A_l$ ), while the x and y hit coordinates are the output. Such model has been trained with 100 boosting stages. The maximum depth (i.e. maximum number of stages) of the individual regression estimators was set to 8, with a learning rate of 0.1: such value provides the best resolution, as shown in figure 5, whereas larger values would cause the model to overfit, without really improving



**Figure 5.** Spatial resolution as a function of the individual estimators maximum depth used in the Multi-Output Regressor model for the device with 200–500 geometry.

the predictions (overfitting occurs when a machine learning algorithm generates data too similar or equal to the training dataset, leading to predictions which are not accurate [15]).

While the final goal is to train the algorithm using precise data collected at beam test, in this initial proof-of-principle study, the algorithm was trained using a mixture of simulation and experimental data.

The input signals for the algorithm training were obtained in the following way:

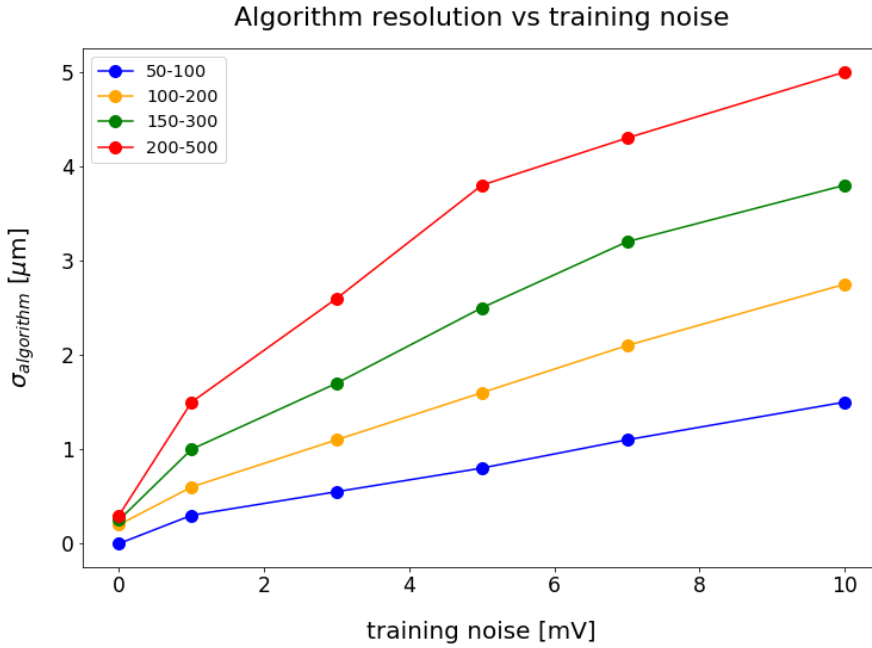
- The total signal amplitude (obtained by summing the amplitudes recorded by the 4 read-out pads) for each geometry and for 3 different gain levels was measured in laboratory using the laser setup described in the following section.
- After that, 100 different signal amplitudes were generated for each position within the experimental regions reported in 1; that was repeated for all geometries and for the 3 different gains.
- Then, using the RSD Main Formula, the sharing among the pads was analytically computed.
- Finally, a Gaussian noise smearing was added to each pad signal. The addition of noise, called *training noise*, and the use of 100 different amplitudes per position are introduced to prevent overfitting. The *training noise* prevents overfitting as it generalizes the machine learning model: its effect is comparable to training the algorithm with 100 different dataset, therefore it can also be seen as a form of data augmentation [16, 17].

The input signals produced in this way, combined with their relative x-y coordinates, form the *training dataset*, which is then used to train the Multi-Output Regressor algorithm.

The *training noise* is a key aspect because it is very helpful to make reliable predictions, but it also introduces an intrinsic uncertainty ( $\sigma_{\text{algorithm}}$ ) since a set of 4 amplitudes does not uniquely

define a position. Such uncertainty can be assessed by feeding to the trained algorithm the same set of amplitudes that were used for its training, this time without noise (this set will be called *test dataset*, to distinguish it from the *training dataset*), and then comparing the predicted hit coordinates with the known coordinates.

Figure 6 shows  $\sigma_{\text{algorithm}}$  as function of the *training noise* for the explored geometries, assuming a *training dataset* with a total signal amplitude of 120 mV. This analysis is performed using positions contained within a  $40 \times 40 \mu\text{m}^2$  square located in the middle of the area among the 4 pads. The term  $\sigma_{\text{algorithm}}$  scales linearly with the *training noise*, with larger geometries having a slightly worse resolution. Since the positions are within a squared region of fixed dimension for all geometries, such result does not depend on the width of the training region.



**Figure 6.**  $\sigma_{\text{algorithm}}$  as a function of *training noise* for RSDs with different designs. The signal amplitude is 120 mV for all geometries, corresponding to a gain  $\sim 15$ .

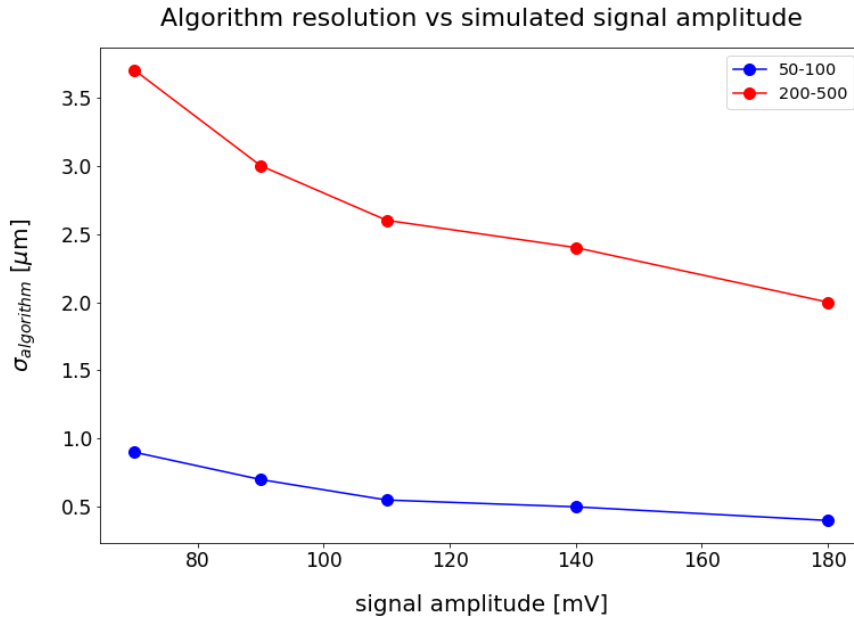
It is worth noticing that  $\sigma_{\text{algorithm}} \sim 0$  for all geometries when *training noise* equal to zero, underlining that, in absence of noise, the algorithm is able to predict the position with a negligible uncertainty, regardless the sensor’s pitch.

A possible explanation to the worse resolution in devices with larger pitch is the different attenuation of signal amplitude per unit of distance. A signal with total amplitude of 80 mV (gain 12) generated in a position equidistant from the centers of 4 different read-out pads will produce, according to equation (2.1), a 20 mV signal amplitude in each of the pads, regardless the device geometry. In a RSD with the 50–100  $\mu\text{m}$  design, such hit position will be at a distance  $d_{50-100} = 25 \mu\text{m} * \sqrt{2} = 35 \mu\text{m}$  from the pad metal, whereas in a RSD with the 200–500  $\mu\text{m}$  geometry it will be  $d_{200-500} = 150 \mu\text{m} * \sqrt{2} = 212 \mu\text{m}$ . Assuming a linear change of the signal amplitude with distance, if the signal moves by one micron in the 50–100 (200–500) design, the signal will change

by  $\frac{1}{35} \sim 3\%$  ( $\sim 0.5\%$ ). Hence, a *training noise* of 2 mV (a typical value used in this study) smears the reconstructed position differently depending on the sensor size:

- $\sigma_{\text{algorithm}}^{50-100}$ :  $\sim 2 \text{ mV} / 0.6 \text{ mV}/\mu\text{m} \sim 3.3 \mu\text{m}$
- $\sigma_{\text{algorithm}}^{200-500}$ :  $\sim 2 \text{ mV} / 0.125 \text{ mV}/\mu\text{m} \sim 20 \mu\text{m}$ .

Another consequence of this effect is shown in figure 7:  $\sigma_{\text{algorithm}}$  improves as a function of the simulated signal amplitude (i.e. the total amplitude used in the *training dataset*) because the amplitude change per  $\mu\text{m}$  is higher and, as a consequence, the effect of the *training noise* on the reconstructed position is milder.

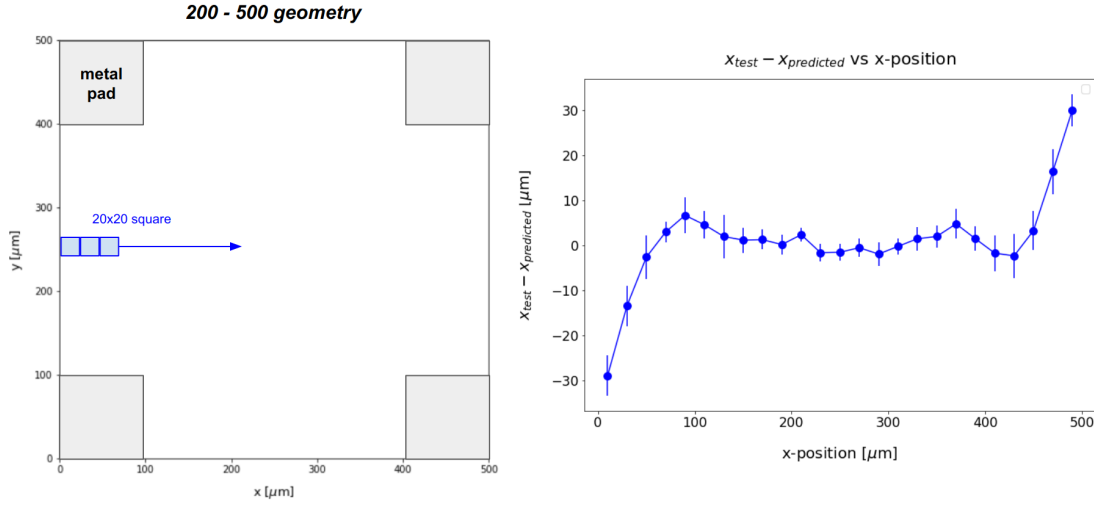


**Figure 7.**  $\sigma_{\text{algorithm}}$  as a function of the simulated signal amplitude for RSDs having 50–100 and 200–500  $\mu\text{m}$  geometries. The *training noise* is set to 2 mV.

An equivalent smearing is also generated by the electronic noise:  $\sigma_{\text{jitter}}$ , see equation (3.2), is higher for larger geometries as the *interpad* increases.

Smaller geometries are therefore benefiting twice: the reconstruction algorithm is more precise and the position jitter is smaller. In the same way, increasing the gain has a double effect: it decreases the contributions of the reconstruction method and of the jitter term.

Up to now,  $\sigma_{\text{algorithm}}$  has been computed using as test area a  $40 \times 40 \mu\text{m}^2$  square positioned centrally among pads. This study has been extended to the whole sensor area for the 200–500  $\mu\text{m}$  geometry. The position resolution as a function of the x-axis has been computed in a sliding window of  $20 \times 20 \mu\text{m}^2$ . This process is shown in the left side of figure 8 while the right side of the picture shows, for each x position, the offset and the value of  $\sigma_{\text{algorithm}}$ . In the x-intervals 0–100  $\mu\text{m}$  and 400–500  $\mu\text{m}$ , the offset worsens significantly since the signals in those areas are split among 6 pads while the test is performed considering only 4 pads. In the central area, 100–400  $\mu\text{m}$ ,  $\sigma_{\text{algorithm}}$  is almost constant and the offset small.



**Figure 8.** Mean values of  $x_{\text{test}} - x_{\text{predicted}}$  distribution as a function of x-position for a 200–500 sensor. The simulated signal amplitude is 120 mV and the *training noise* is 2 mV.

## 5 Results

The performances of the Multi-Output regressor algorithm has been measured using data collected with a laser setup. The events have been acquired shooting a 1064 nm picosecond laser on several  $3 \times 3$  RSD pixel detectors with different geometries (100–200  $\mu\text{m}$ , 50–100  $\mu\text{m}$ , 150–300  $\mu\text{m}$ , 200–500  $\mu\text{m}$ ) and reading the signal of the 4 surrounding pads with a fast oscilloscope (4 GHz analog bandwidth, 40 GS/s). The laser data are ideal for this measurement since they have been obtained using a Transient Current Setup (TCT), comprising a movable x-y stage, that measures the laser shot position [18] with 2  $\mu\text{m}$  precision. It is therefore possible to use the positions provided by the TCT setup as reference positions to assess the accuracy of the reconstruction algorithm. The laser intensity was set to generate the same number of  $e-h$  pairs of 3 Minimum Ionizing Particles (MIPs).

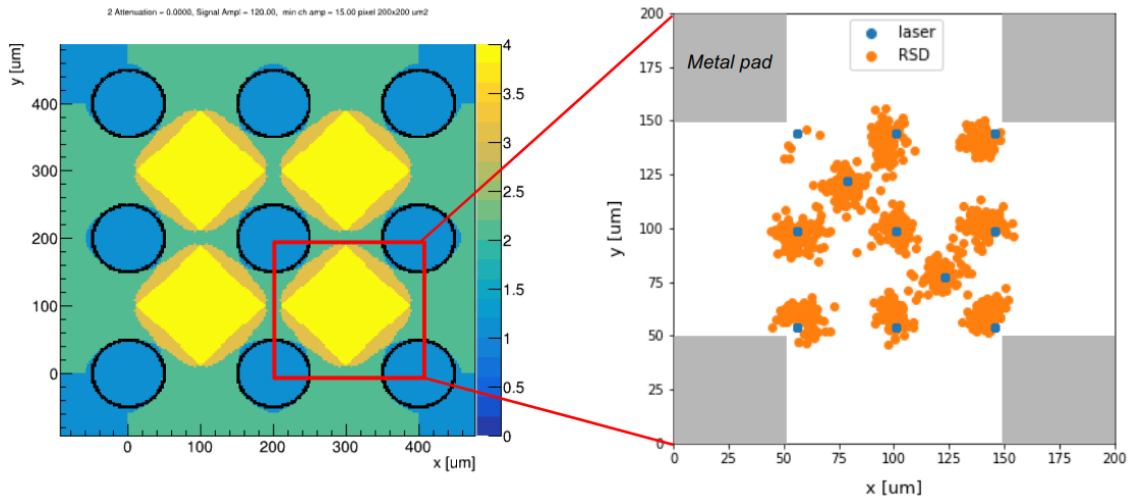
Figure 9 (left side) shows the number of active pads as a function of x-y position for a 100–200  $\mu\text{m}$  RSD and the red square indicates the area tested with the laser. Similar regions have been considered in all measured sensors.

In order for the algorithm to be fully efficient, the laser data have been collected only from positions within the experimental regions reported in table 1, where at least 3 pads are active.

Figure 9 (right side) shows the positions obtained with the ML algorithms (orange) compared to the true laser shots positions (blue); a total of 100 laser shots were fired in each position. Similar maps have been obtained for the other geometries as well. As it is clearly visible on the plot, the reconstructed points cluster tightly around the laser positions.

The width of the distribution  $x_{\text{RSD}} - x_{\text{Laser}}$  ( $\sigma_{\text{measured}}$ ) provides the squared sum of the RSD measured resolution ( $\sigma_{\text{RSD}}$ ) and the TCT setup resolution ( $\sigma_{\text{TCT}} = 2 \mu\text{m}$ ), hence:

$$\sigma_{\text{RSD}} = \sqrt{\sigma_{\text{measured}}^2 - \sigma_{\text{TCT}}^2} \quad (5.1)$$



**Figure 9.** Left side:  $x$ - $y$  map of the measured  $3 \times 3$  RSD with 100–200  $\mu\text{m}$  geometry. The region considered during the measurement is highlighted in red. Right side: reconstructed RSD positions (orange) for different laser shots positions (blue).

Table 2 shows:  $\sigma_{\text{RSD}}$ ; the laser mean signal amplitude (used as total amplitude to generate the *training dataset*); the *training noise* used in the *training dataset* that gave the best performance of the algorithm; the uncertainty of the ML algorithm ( $\sigma_{\text{algorithm}}$ ) and the expected total resolution ( $\sigma_{\text{expected}}$ , details in the following section).

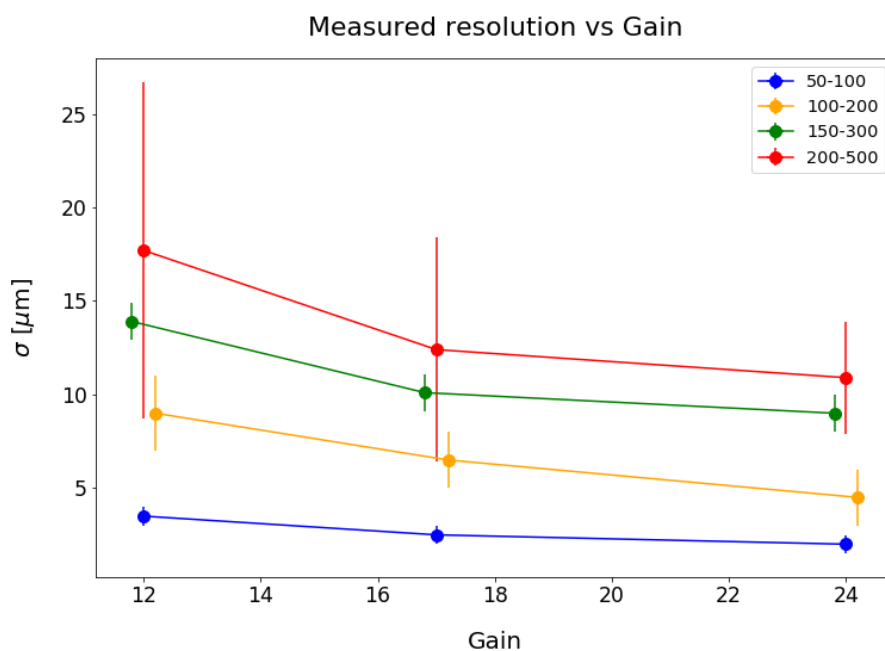
**Table 2.** Spatial resolutions of RSD sensors.

Geometry	$\sigma_{\text{RSD}}$ ( $\pm$ exp. $\pm$ syst.)	Amplitude	<i>Training noise</i>	$\sigma_{\text{algorithm}}$	$\sigma_{\text{expected}}$
50–100 $\mu\text{m}$	$2 \pm 0.1 \pm 0.5 \mu\text{m}$	190 mV	1 mV	0.2 $\mu\text{m}$	1.5 $\mu\text{m}$
100–200 $\mu\text{m}$	$4.5 \pm 0.2 \pm 1.5 \mu\text{m}$	160 mV	1 mV	0.5 $\mu\text{m}$	3.5 $\mu\text{m}$
150–300 $\mu\text{m}$	$9.0 \pm 0.4 \pm 1 \mu\text{m}$	225 mV	1 mV	0.6 $\mu\text{m}$	3.5 $\mu\text{m}$
200–500 $\mu\text{m}$	$10.9 \pm 0.2 \pm 3 \mu\text{m}$	135 mV	30 mV	27 $\mu\text{m}$	28 $\mu\text{m}$

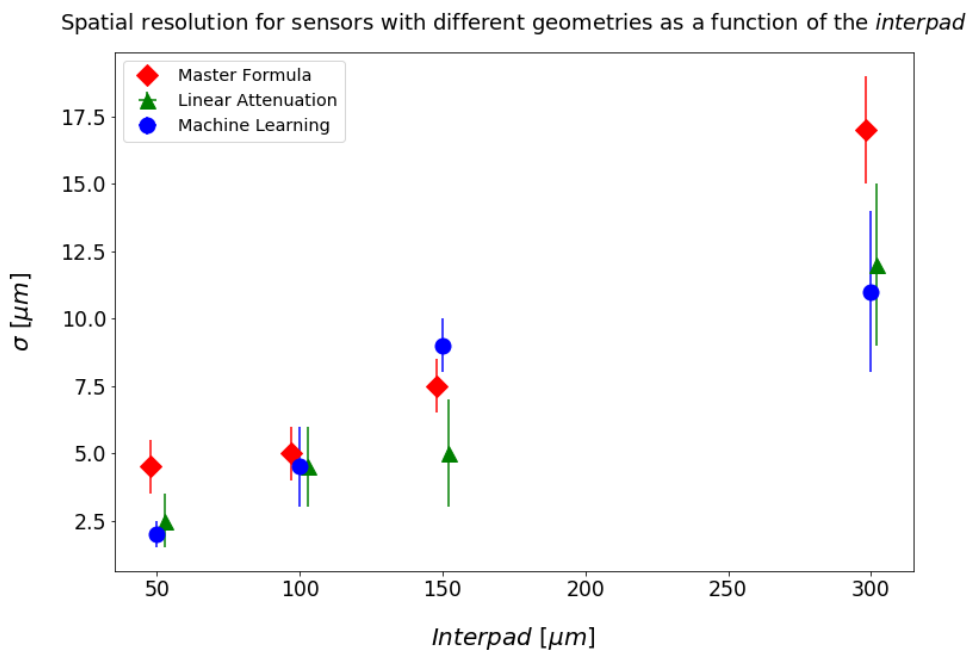
For the 50–100 device,  $\sigma_{\text{measured}}$  is reported instead of  $\sigma_{\text{RSD}}$ , since the uncertainty is completely dominated by  $\sigma_{\text{TCT}}$ .

The behaviour of the spatial resolution as a function of gain is shown in figure 10 for all 4 geometries. As expected, the resolution improves with increasing gain since both  $\sigma_{\text{jitter}}$  and  $\sigma_{\text{algorithm}}$  decreases.

In [6], two analytic methods (MF and LA) and the Discretize Positioning Circuit (DPC) are used to evaluate the RSD position resolution. Figure 11 compares these results with that of the ML method: the ML method is consistently the best or close to the best, even though its training has not been performed on real data. This demonstrates that an approach based on ML has the capability of improving the accuracy of RSDs beyond what is achievable with the more traditional reconstruction methods.



**Figure 10.** Measured spatial resolution for different geometries as a function of gain.



**Figure 11.** Spatial resolution as a function of the *interpad* size for the measured RSD sensors. Red diamonds (green triangles) show the results obtained with the MF (LA) analytical method, while blue circles those obtained with ML.



## 5.1 Discussion of the results

### 5.1.1 The optimum value of *training noise*

The results presented in table 2 were obtained with therein listed training noise. The best performance of the algorithm is achieved when the *training noise* is close to the experimental electronic noise, which was measured to be 1.5–2 mV. This result is expected, since the *training noise* has on the *training dataset* the same smearing effect that the electronic noise has on the experimental data, therefore it is not surprising that the algorithm predictions are most accurate when training and experimental noises are set equal or very similar.

For the geometry 200–500, on the contrary, the best resolution was reached at a much higher *training noise*,  $\sim 30$  mV. A possible explanation for this difference is the lower accuracy of the MF algorithm for the 200–500 geometry. As reported in figure 11, the use of the MF algorithm with the 200–500 leads to a large spatial resolution, indicating that this model fails to reproduce correctly the split among pads. Since the ML network has been trained with data generated using the MF model, also its resolution for the 200–500 geometry should be rather modest. However, the use of a large *training noise* is able to compensate for this initial inaccuracy. This case highlights the power of the ML method: it achieves better results than a pure analytical model thanks to the noise introduced to the system. For the 200–500 sensor, the resolution achieved using the ML method is 55% better than that with the MF.

### 5.1.2 Systematic uncertainties

The systematic uncertainties reported in table 2 have been assessed by changing the *training noise* in the vicinity of its optimal value and splitting each laser dataset in two subsets, and then comparing the results. Such uncertainties are rather small, demonstrating that the ML method is not very susceptible to simulation parameters.

## 5.2 On the expected spatial resolution

The trained ML algorithm can be used not only to compute its intrinsic uncertainty ( $\sigma_{\text{algorithm}}$ ), but also to give an expectation ( $\sigma_{\text{expected}}$ ) on the spatial resolution that will be obtained once the algorithm is tested to real data. This can be done in the same way described in section 4 for  $\sigma_{\text{algorithm}}$ , but this time using a *test dataset* with a 1.5 mV smearing applied, which represents the electronic noise.

While the smearing applied to the *training dataset* (i.e. the *training noise*) is responsible for  $\sigma_{\text{algorithm}}$ , the smearing applied to the *test dataset* is comparable to the effect of the jitter, as described in equation (3.1), therefore the expected resolution can be written as:

$$\sigma_{\text{expected}} = \sqrt{\sigma_{\text{algorithm}}^2 + \sigma_{\text{jitter}}^2} \quad (5.2)$$

Such equation does not take into account the third term of equation (3.1),  $\sigma_{\text{sensor}}$ , which depends on non-uniformities of the real detector, that cannot be simulated. However, since  $\sigma_{\text{expected}}$  is similar to  $\sigma_{\text{RSD}}$  in the 50–100 and 100–200 designs,  $\sigma_{\text{sensor}}$  is expected to be rather small in all geometries, because all tested sensors come from the same wafer and should therefore have similar non-uniformity terms.

$\sigma_{\text{expected}}$  is less accurate for 150–300, where it predicts a lower resolution, and 200–500, where conversely the expected resolution is much higher. The cause of such inaccuracies in the largest geometries should be sought in the imperfections of the MF, which might also be the reason for the large  $\sigma_{\text{algorithm}}$  of the 200–500 device. Both aspects will be fixed in the future by training the algorithm directly with real data taken at the beam test.

## 6 Conclusions and future plans

A Multi-Output regressor algorithm has been trained to reconstruct with great accuracy the hit position in RSDs. A spatial resolution ranging between 2  $\mu\text{m}$  and 10  $\mu\text{m}$ , depending upon the sensor geometry, has been measured.

Even though this first implementation of the Multi-Output regressor algorithm uses a limited set of input variables in the training phase, it is presently the most accurate reconstruction algorithm for RSD.

In the near future, the Multi-Output regressor algorithm will be trained with beam test data taken with a very precise tracker. In this way, the algorithm will be fed with a wider range of input features, experimentally measured, whose attenuation laws cannot be derived analytically. This will further improve the spatial resolution, especially in devices with large geometries, because it will be possible to train a much deeper and complex network.

## Acknowledgments

We warmly thank the INFN Computing Center of Torino for providing support and computational resources and kindly acknowledge the following funding agencies and collaborations: INFN — Gruppo V; Horizon 2020, grant UFSD 669529; Dipartimenti di Eccellenza, University of Torino (ex L. 232/2016, art. 1, cc. 314, 337); Ministero della Ricerca, PRIN 2017, progetto PRIN2017L2XKTJ, 4DInSiDe. Ministero della Ricerca, FARE, progetto R165xr8frtfare.

## References

- [1] G. Pellegrini et al., *Technology developments and first measurements of Low Gain Avalanche Detectors (LGAD) for high energy physics applications*, *Nucl. Instrum. Meth. A* **765** (2014) 12.
- [2] G. Paternoster et al., *Latest Developments of Low Gain Avalanche Detectors at FBK*, Presentation at the *TREDI2019: 14th “Trento” Workshop on Advanced Silicon Radiation Detectors*, Trento Italy (2019), [https://indico.cern.ch/event/777112/contributions/3314805/attachments/1802242/2939922/TREDI\\_19\\_paternoster.pdf](https://indico.cern.ch/event/777112/contributions/3314805/attachments/1802242/2939922/TREDI_19_paternoster.pdf).
- [3] M. Mandurrino et al., *Analysis and numerical design of Resistive AC-Coupled Silicon Detectors (RSD)*, *Nucl. Instrum. Meth. A* **959** (2020) 163479.
- [4] M. Mandurrino et al., *Demonstration of 200-, 100-, and 50-  $\mu\text{m}$  Pitch Resistive AC-Coupled Silicon Detectors (RSD) With 100% Fill-Factor for 4D Particle Tracking*, *IEEE Electron Device Lett.* **40** (2019) 1780.
- [5] N. Cartiglia et al., *Signal formation and designed optimization of Resistive AC-LGAD (RSD)*, at *TREDI2020: 15th “Trento” Workshop on Advanced Silicon Radiation Detectors*, Vienna Austria (2020).

- [6] M. Tornago et al., *Resistive AC-Coupled Silicon Detectors: principles of operation and first results from a combined analysis of beam test and laser data*, submitted to *Nucl. Instrum. Meth. A*, [arXiv:2007.09528](https://arxiv.org/abs/2007.09528).
- [7] N. Cartiglia and M. Mandurrino, *Innovative Silicon Sensors for Future Trackers*, *CERN Detector Seminar*, CERN, Geneva Switzerland (2020).
- [8] N. Cartiglia et al., *Silicon Sensors for Future Particle Trackers*, *Nucl. Instrum. Meth. A* **979** (2020) 164383 [[arXiv:2003.13990](https://arxiv.org/abs/2003.13990)].
- [9] <https://scikit-learn.org/stable/modules/generated/sklearn.multioutput.MultiOutputRegressor.html>.
- [10] J.H. Friedman, *Greedy Function Approximation: A Gradient Boosting Machine*, *Ann. Statist.* **29** (2001) 1189.
- [11] J.H. Friedman, *Stochastic gradient boosting*, *Comput. Statist. Data Anal.* **38** (2002) 367.
- [12] <https://towardsdatascience.com/tree-based-methods-regression-trees-4ee5d8db9fe9>.
- [13] <https://towardsdatascience.com/gradient-descent-explained-9b953fc0d2c>.
- [14] H. Borchani, G. Varando, C. Bielza and P. Larrañaga, *A survey on multi-output regression*, *WIREs Data Mining Knowl. Discov.* **5** (2015) 216.
- [15] D. Hawkins, *The problem of overfitting*, *J. Chem. Inf. Comput. Sci.* **44** (2004) 1.
- [16] C.M. Bishop, *Training with Noise is Equivalent to Tikhonov Regularization*, *Neural Comput.* **7** (1995) 108.
- [17] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, Cambridge U.S.A. (2016).
- [18] G. Kramberger, *Advanced Transient Current Technique Systems*, *PoS(Vertex2014)032*.
- [19] H.F.W. Sadrozinski, A. Seiden and N. Cartiglia, *4D tracking with ultra-fast silicon detectors*, *Rept. Prog. Phys.* **81** (2018) 026101 [[arXiv:1704.08666](https://arxiv.org/abs/1704.08666)].