Radioactive hot-spot detection using machine learning algorithms

Miguel Tomé Ramos e Barros
migueltrbarros@tecnico.ulisboa.pt
Physics Department, Instituto Superior Técnico, Lisboa, Portugal
(Dated: January 2021)

The detection of radioactive hot-spots has been a challenge for the security sector, especially in situations involving chemical, biological, radiological and nuclear (CBRN) threats. This work proposes a solution based on Machine Learning techniques, with a focus on Artificial Neural Networks (ANNs), so observations of radiological intensity counts and corresponding localizations can be used to estimate not only the number of unknown radioactive sources present in a given scenario, but also their location and activity at the same time. For this, a simulator is used to generate a training data set for the training process, and so, using the model already trained through a Divide and Conquer algorithm, fast and accurate predictions are achieved, ensuring the reliability of such an ANN-based approach.

The proposed solution is then tested in scenarios with multiple sources, with obstacles included, and with non-point-like sources. Unlike most existing algorithms, which begin failing in scenarios with those conditions, ANNs have shown that are capable of performing an accurate hot-spots detection, with a low number of limitations. Additionally, experimental results, done in lab environments and real scenarios located at old deposits of radioactive ore, have shown that the algorithm is scalable for very large regions, as well as for very short scenarios. Thus, ANNs have demonstrated the capability of being an emerging tool with the potential to make a difference in the nuclear field, by helping in the development of novel techniques and new solutions in order to safeguard human lives.

Keywords: Radioactive Hot-spots, Machine Learning, Artificial Neural Networks, Divide and Conquer

I. INTRODUCTION

Radioactivity is a field that can bring a lot of benefits for the human being, namely in science, medicine and industry areas. However, it also involves many issues and dangers, related to chemical, biological, radiological and nuclear (CBRN) threats [1] and the naturally occurring radioactive materials (NORMs) [2], which can arise from anywhere (although there are potential locations). Therefore, the detection of dirty bombs in public places, the autonomous inspection of nuclear sites and the remote control of places, like the one that followed the Chernobyl disaster in 1986, are examples of applications that have been developed in this field in the past decade. To perform an inspection in such radioactive scenarios, portable devices, such as radiological sensors, can be handled, in a backpack, or even in mobile robots, like an unmanned aerial vehicle (UAV). For example, in [3] the UAV helps in detecting radioactive material through search strategy methods within a specific area, whereas in [4], it is used to build a 2D radioactive heatmap.

Then, despite the success of some state-of-the-art algorithms, in detecting a single or a small number of radioactive point sources, most of them start failing as the scenarios get more complex. A high number of sources, the presence of obstacles, and the existence of sources with non-trivial shapes are some of the main limitations. In addition, even in simpler scenarios, when larger or shorter dimensions are considered, or when the data acquisition approach is different from the one considered, most existing works are not capable of keeping providing good results.

For this reason, in this work, a solution based on Artificial Neural Networks (ANNs) is proposed in order to estimate not only the distribution and the positions of the sources, but also the number and the respective activities of each of them. The reasons for this selected solution are the reduction of computing time through recent improvements in both software and hardware, the huge amount of current available open source frameworks, and also the fact that, although there is still not much nuclear data, according with the World Institute for Nuclear Security [5], it is intended that, in the near future, there will be large amounts of nuclear data. Thus, the main contributions of this work are as follows:

- A novel approach for detecting radioactive hot-spots, that is efficient in (i) handling multiple sources, without requiring any information about the number of sources; (ii) predicting the parameters of the detected sources (location and activity); (iii) considering scenarios with obstacles included, without any information given in advance and without training data composed of such scenario conditions; (iv) handling regions of the scenario with-

out any observation and still get accurate predictions; (v) approaching non-point-like sources with interesting results; and (vi) handling both large and short-scale scenarios, through the application of an algorithm based on the Divide and Conquer (DAC) method.

- Quantitative results are provided and show that the selected approach is capable of getting accurate predictions, with a low number of false negatives (missed detections) and false positives (false alarms) rates, as well as precise estimations regarding the exact location and activity of the detected sources.

The remainder of this work is organized as follows: the state of the art is presented in Section 2. In section 3, it is formulated the problem statement and, in section 4, the selected approach is described. In section 5, it is detailed the setup and data assumptions, as well as the way the simulator is used. The data preprocessing, the implementation of the selected approach, and the different tests performed to select the best models are detailed in section 6. Finally, chapter 7 provides the performance results with real and simulated scenarios, and chapter 8 summarizes the conclusions.

II. STATE OF THE ART ALGORITHMS

Earlier detection systems are designed for a single source, or separated distributed sources [6, 7], where the radiation field could be described by several unimodal distributions and the cumulative effect of multiple sources could be neglected. Whereas, more recently, many approaches have also been adopted to locate multiple sources, taking into account the cumulative effect in statistic models [8, 9]. Most of them assume that the number of sources N is known *a priori*, being in this case a parameter estimation problem, where the goal is to find the most probable source parameters (location and strength) of each source. Thus the number of estimated parameters increases with N, and then the algorithm complexity also increases. On the other hand, in works where the number of sources N is unknown, the parameters of the sources are estimated for a range of possible values for N, using in the end the ones with highest probability. This can lead to inaccurate predictions, since a large range of possible N values and a superposition of signal strengths from different sources are examples of factors that make existing algorithms more complicated, and even more when these factors are jointly considered. For instance, if we consider the problem of non-point-like sources, a sensor can record a high measurement induced by a single strong source or by a combined multiple weak sources. This means that we may get similar results that seem to be valid for distinct values of N.

According to [10], ANNs are the dominant learning-based algorithm used in nuclear and radiological science, mainly in applications of reactor health and monitoring, radiation protection (characterization and identification of radionuclides), and optimization. However, regarding the concrete problem of radioactive hot-spot detection, one has not found approaches in the literature of the last decade (only [11] from 1995 which detects only one source), which can be confirmed by the reviews [10, 12]. This may be related to the fact that, as we will see later, to build an ANN model, a training process has to be performed and some conditions must be fixed, such as the conditions of the scenario (e.g. its dimensions and the strength of the sources) and the way the observations are acquired (sensors distribution or the path taken by the UAV). Otherwise, new training processes must be performed and the time necessary for this may be a barrier to the development of such ANN-based approaches. Besides that, the computational costs in relation to both software and hardware to train the model, and the need to get a huge and varied training data set, can also be some of the constraints to obtain accurate results.

III. PROBLEM STATEMENT

This work aims to solve the problem of radioactive hot-spot detection (which includes its localization and quantification of the radiation level), by using an UAV with a Geiger-Müller Counter (GMC) and a Global Positioning System (GPS) receiver to get observations and then compute the number, location and activity of the source(s) present in the area of interest.

It is assumed an unknown number of radioactive sources, $N \geq 0$, with no information regarding the respective activities and locations, within an area of interest treated as a two-dimensional environment (since the distance to the ground is considered constant). Let $\mathcal{S} = [\boldsymbol{s}_1^T \dots \boldsymbol{s}_N^T]^T$ denote the set of radiations sources. Each source is considered as isotropic point-like source, which is parameterized by the vector parameter $\boldsymbol{s}_i = [\boldsymbol{p}_{s_i} a_{s_i}]^T \in \mathrm{R}^2 \times \mathrm{R}^+$, for $i = 1, ..., N$. The position of the i^{th} source is given in Cartesian coordinates, with $\boldsymbol{p}_{s_i} = (x_{s_i}, y_{s_i})$ in meters (m), and the activity, a_{s_i}, is measured in units of becquerel (Bq), which means the number of disintegrations per second. Also, the activity is assumed to be stable, i.e., the duration of any measurement is negligible when compared to the half-life of the source.

Then, the radiation intensity measurements are assumed as independent random variables each following a Poisson distribution, with an equal exposure time. Having a known number of measurements, M, let $\mathcal{O} = [\boldsymbol{o}_1^T \dots \boldsymbol{o}_M^T]^T$ denote the set of observations, with $\boldsymbol{o}_j = [\boldsymbol{q}_{o_j} c_{o_j}]^T \in \mathrm{R}^2 \times \mathrm{R}_0^+$, for $j = 1, ..., M$, where $M >> N$. Similarly, $q_{o_j} = (x_{o_j}, y_{o_j})$ represents the j^{th} measurement position, and its intensity (number of particles detected within the exposure time interval), measured in units of becquerel per squared meter (Bq/m^2), is given by

$$c_{o_j} = c_b + \sum_{i=1}^{N} \frac{a_{s_i} \exp(-\alpha d_{o_j,s_i})}{d_{o_j,s_i}^2} \tag{1}$$

where both the coefficient α, which is related to the ab-

sorption of radiation particles in the air, and the background intensity, c_b, are known *a priori*, and d_{o_j,s_i} is the Euclidean distance between positions $\boldsymbol{q}_{o_j}$ and $\boldsymbol{p}_{s_i}$.

When obstacles are considered, some of the previous expressions must get adapted, since they shield (totally or partially) the radiation rays. Also, the obstacles have different types of materials and shape, and can be located anywhere in the area of interest. So, assuming that each obstacle is made of homogeneous material, and the radiation source is on one side of the obstacle, the intensity that comes from the i^{th} source, a_{s_i}, becomes $a_{s_i}e^{-\mu l}$, where l is the thickness and μ the attenuation coefficient for the material of the obstacle. Finally, eq. (1) becomes

$$c_{o_j} = c_b + \sum_{i=1}^{N} \frac{a_{s_i} \exp\left(-\alpha d_{o_j,s_i} - \sum_{b\in\mathcal{B}_{o_j,s_i}} \mu_b l_b\right)}{d^2_{o_j,s_i}} \quad (2)$$

where $\mathcal{B}_{o_j,s_i}$ denotes the set of obstacles that intersect with the straight line between $\boldsymbol{q}_{o_j}$ and $\boldsymbol{p}_{s_i}$. Consequently, regarding each obstacle b, l_b is the total thickness along that intersection, and μ_b is the corresponding attenuation coefficient.

Thus, the localization problem is to use the vector of measurements, $\mathcal{O}$, in order to make predictions regarding the number of sources, $\hat{N}$, and respective parameters vector, $\hat{\mathcal{S}} = [\hat{\boldsymbol{s}}_1^T \dots \hat{\boldsymbol{s}}_{\hat{N}}^T]^T$, with $\hat{\boldsymbol{s}}_k = [\hat{\boldsymbol{p}}_{\hat{s}_k} \hat{a}_{\hat{s}_k}]^T \in \mathrm{R}^2\times\mathrm{R}^+$, for $k = 1, ..., \hat{N}$.

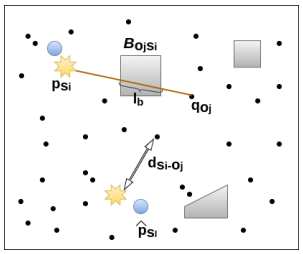

FIG. 1: Illustration of Problem Statement, with obstacles in gray color.

IV. SELECTED APPROACH

The proposed approach to detect radioactive hot-spots is the use of Machine Learning techniques, namely ANNs, and related architectures such as Convolutional Neural Networks (CNNs), essentially due to its ability in dealing with radiological data, which is non-linear, noisy and inconsistent. Besides, the training of such deep architectures has become even more efficient with improvements and breakthroughs in techniques related to training, optimization and regularization [13]. To overcome the problems and limitations involved in the development of an algorithm based on ANNs, which were mentioned in section I, the selected approach of this work aims to build an ANN model using a training data set composed only of simulated data, and adapting the simulator parameters according to the scenario conditions. Then, using an algorithm with the pre-trained ANNs, and taking advantage of the capacity of the regression and classification problems, it is possible to apply the model in both large and short-scale scenarios. Besides, the ANN abilities, and the way they are used in this selected approach, may solve some limitations of most existing algorithms, such as the existence of a maximum number of sources to get accurate detections (since some models degrade or the runtime explodes as the number of sources increases), the non-scalable applications of the models, and the obstacles information given *a priori*. Lastly, having the ANN model already trained, it is possible to use this approach in real-time (during a real surveillance, for instance), obtaining fast responses within an extremely short time (independently on the number of sources and the presence of obstacles), which can be highly demanded in radiological operations related with the detection of hot-spots.

Therefore, the ANN model adopted is inspired on Yolo state of the art [14], where a model based on classification and regression problems is applied for object detection tasks, with images used as input. Since the input data of this work is composed of intensity measurements acquired throughout the entire map, instead of a matrix of pixels, it is used a matrix with the intensity values. Then, the map is divided into $S \times S$ grid, and if a source is located in a grid cell b, then this grid cell is responsible for detecting that source. So, the classification outputs give the probability of having a source in each grid cell (it should be zero if no source exists in that cell, and one otherwise), making possible to compute the number of sources (assuming for now that each box has at most one source) and the regions (cells) where they are. Then, to compute the parameters of the sources (activity and location), the regression problem is applied in each cell, by associating 3 outputs (a_b, x_b, y_b) for each cell b.

Thus, in the end, we have an $S \times S \times 4$ tensor, where each grid cell can be represented as a four-dimensional vector:

$$z_b = [p_b \; x_b \; y_b \; a_b]^T \quad (3)$$

where p_b is the probability of having a source in the respective grid cell, b. If $p_b = 1$, and supposing that it is the i^{th} source the one present in this cell b, then (x_b, y_b) parameters are equivalent to the positions (x_{s_i}, y_{s_i}) and a_b refers to its activity a_{s_i}. Otherwise, if $p_b = 0$, then $(x_b, y_b, a_b) = (?, ?, ?)$ as done in Yolo approach, where "?" means "Don't care". Thus, despite a regression problem is going to be done even in cells without any source, by combining the results of both problems, we can select which outputs of regression problem matter (while the other ones must be neglected) and, in the end, we get the final estimations with regressions only on the cells where the classification problem have determined that there is a source, as represented in Fig. 2.

As we can see in eq. (3), only one source can be detected per each grid cell. Therefore, an algorithm based on the DAC method is applied, in such a way that the pre-trained ANN model can be used. This is a strategy typically used to solve a problem recursively by applying three steps according to the level of recursion: divide,

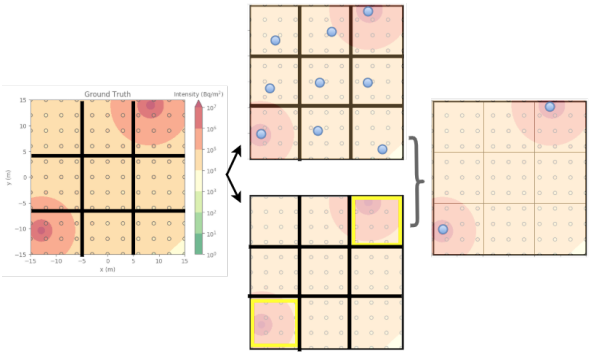

FIG. 2: Scheme of the classification and regression-based model.

conquer and combine. In the first step, "divide", the problem breaks into smaller sub-problems that are similar to the original problem (called "Division 0"), until the problem is small enough to be solved. These sub-problems are regions of the original map, so the input of the model is just the set of observations that is measured in this region. Only the outputs related to the classification problem are used, where p_b has now a different meaning, referring to the probability of having at least one source ($p_b = 1$) or not having any source ($p_b = 0$) in the corresponding cell b. Then, the "conquer" step is when the original problem is already divided into the smallest possible parts and, therefore, both the classification and the regression outputs are used. In Fig. 3 it is represented one division ("Division 1") where the "divide" step is applied, whereas the "conquer" step occurs on the last division ("Division 2"). Finally, the last step, called "combine", is when we are able to combine all the predictions of number, location and activity of the found sources, and, in the end, to solve the whole problem and compare the results with the real values.

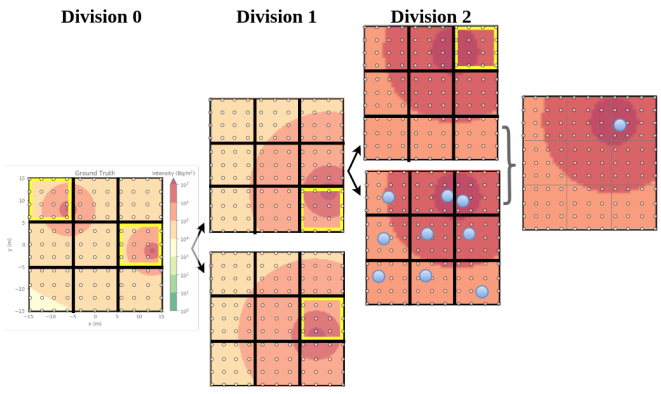

FIG. 3: Representation of the Divide and Conquer method, with an example of two divisions.

The number of divisions depends on the quantity and quality of the observations that are used as input of the ANN model over the successive divisions (since they must be dense and spatially well-distributed throughout the scenario); and on the dimensions of the scenario where the "conquer" step will be applied.

A. Quality Metrics

For each model considered, the neural network is trained using the training set, while its error is evaluated at the same time in the validation set. According to the Early-stopping technique [15], those parameters corresponding to the best iteration are saved, and are then used with the test data set in order to evaluate the performance of the model.

Therefore, there are some criteria to evaluate and then select the best models. Regarding the regression outputs, the Mean Squared Error (MSE) is used to measure the average of the squares of the errors, as follows:

$$MSE_a = \frac{1}{n}\sum_{i=1}^{n}(\hat{a}_i - a_i)^2 \tag{4}$$

which is used for the activity predictions $\hat{a}$, and with n being the total number of samples. For the location predictions $(\hat{x}, \hat{y})$, a similar expression is used:

$$MSE_{x,y} = \frac{1}{n}\sum_{i=1}^{n}\left[(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2\right] \tag{5}$$

where (a, x, y) represent the real values.

Regarding the classification outputs, the Precision and Recall concept [16] is used. The Precision measures how accurate are the predictions by giving the percentage of corrected identifications, while the Recall measures the efficiency of the model to find the proportion of positives correctly identified. Also, there are the terms true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), to compare the results from the model (associated to positive/negative) with the reality (related to true/false). Therefore, the corresponding rates are given by:

$$Recall = \frac{TP}{TP + FN}, \; Precision = \frac{TP}{TP + FP} \tag{6}$$

There are other important definitions to evaluate how good the prediction are, such as the Accuracy (eq. (7)), which corresponds to the measure of all the correctly identified cases, and the F1-score (eq. (8)), which makes a relationship between Recall and Precision. The Accuracy metric is also used during the training process for the classification outputs.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{7}$$

$$F1\text{-}score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{8}$$

V. SETUP AND DATA ASSUMPTIONS

A. Real Data

The assumed setup is depicted in Fig. 4 (left). Since the radiation sources are assumed to be stable, static and

located on the ground, the UAV is used to get measurements. More precisely, they are acquired by a drone, which carries on a GMC to measure the intensity, and a GPS receiver to measure the position. Two other sensors, a depth camera and a rangefinder, are also used to allow the mission to be planned at a given constant distance to the ground.

Another way to get measurements without using a drone is the one depicted in Fig. 4 (right). This alternative experiment is associated to a human operation walking in the scenario, equipped with a mobile phone connected to the GMC, and, as done with the drone, with an approximately constant distance between the sensor and the ground. Both experimental approaches are done with the aid of a mobile application, denominated MARIA (Mobile Application for Radiation Intensity Assessment), which is being developed at Instituto de Plasmas e Fusão Nuclear (IPFN) and is detailed in [17].

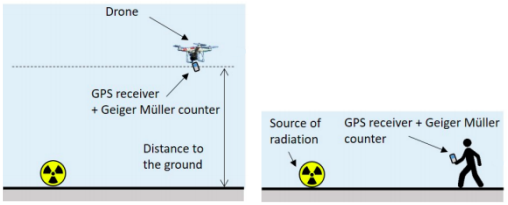

FIG. 4: Experimental setup with a drone flying over the scenario. Alternative with a person operation instead of a drone. Images from [18].

B. Simulated Data

In the MSc Thesis of Yoeri Brouwer [19], it was developed in Python a computer algorithm able to provide a radiological heatmap in 2D, given the initial conditions of the radioactive sources: number, activity and spatial distribution. Later, a similar simulator was also developed to consider the presence of obstacles.

Thus, taking into account the objectives of this work, a simulator based on the previous two simulators is used with some improvements implemented. It is adopted two different approaches (see Fig. 5) to get them: Random, by taking the "Poisson-disk sampling" method that uses Robert Bridson's algorithm [20]; and Grid approach, with a grid sampling based on Gaussian distribution, to make a discretization of the space and get a grid of pre-defined $G \times G$ fixed positions.

The generated training data has similar dimensions to the last division of the DAC method since it is here when both the regression and classification problems are used. Instead, in the previous divisions, only the classification problem is needed. Although the more distinct are the dimensions of the scenarios used during the training process, the lower is the accuracy obtained regarding the classification outputs, the model is still capable of getting accurate predictions, as we will see in subsection VI C by studying the threshold's influence.

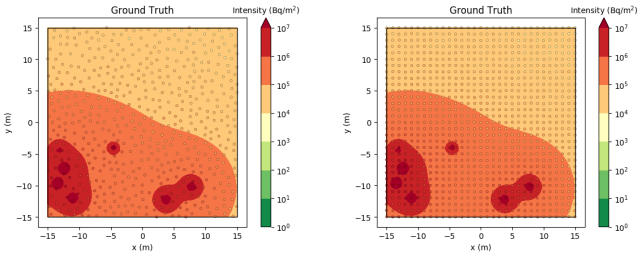

FIG. 5: Representation of Random (left) and Grid (right) approaches.

It is generated 20,000 simulations, each one with a different set of sources, and the corresponding configuration was as follows: scenario dimensions of $3.33 \times 3.33\ m$; sources activity within range $[10^6, 10^8]\ Bq$; one source at most in each of the $S \times S$ grid cells, with $S = 3$; and a maximum total number of possible sources of 4 for each scenario. Regarding the observations, it is defined 11×11 data points for the Grid approach, and about 200 for the Random approach.

VI. NEURAL NETWORKS IMPLEMENTATION

The neural networks algorithm was implemented in Python supported by Jupyter Notebook [21]. It was used the framework Keras [22], running on top of the open-source Machine Learning platform TensorFlow [23], which is supported by Graphics Process Unit (GPU). All the computations associated to the neural networks implementation were performed through the CUDA compatible Nvidia GeForce MX150 and an Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz × 8 processor.

A. Data Preprocessing

Firstly, the simulated data acquired from the simulator is transformed so we have the input vector, with a format $(N_{\text{sim}}, 3 \times M)$, where N_{sim} represents the number of simulations with a set of source(s) and $3 \times M$ means the total number of values $(c_{o_j}, x_{o_j}, y_{o_j})$ of all the measurements, with $j = 1, ..., M$; and the target vector, with a format $(N_{\text{sim}}, 3 \times N)$, with $3 \times N$ being the total number of values $(a_{s_i}, x_{s_i}, y_{s_i})$ of all the parameters of the sources, with $i = 1, ..., N$.

Then, both vectors are transformed to fit the respective input and output layers of the ANN model. The input is a square matrix composed of intensity values, whose corresponding measurement positions in the map match with the positions of each of the $s \times s$ matrix entries. Since the drone's positions have always uncertainty associated and the measurements may be acquired in a random way, then the positions (x_{o_j}, y_{o_j}) should be part of the input too to get more accurate results. To do so, as represented in Fig. 6, a three-dimensional matrix is used as input, where each dimension is composed by the parameters c_{o_j}, x_{o_j} and y_{o_j}. For the Grid approach, we have $G = s$, which means that each fixed point of the grid has already

a correspondence to each entry of the input matrix. For the Random approach, the data must be clustered in each of the $s \times s$ input cells, so only one input is obtained through the average pooling or max pooling methods (see how it is done in the thesis).

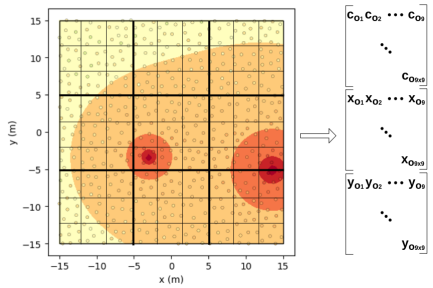

FIG. 6: ANN model's input, with $s = 9$ and Random approach used. For the Grid approach, $s = 11$ is used.

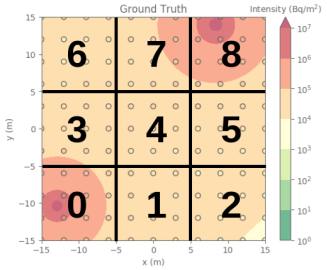

FIG. 7: Map division with 3×3 grid cells.

The target vector must be divided into three distinct vectors: one for the classification problem and the other two for the regression problem, with one referring to the activity values of the sources (a_{s_i}) and the other to the position values (both x_{s_i} and y_{s_i} parameters). For the classification target, it is followed the scheme with 3×3 boxes represented in Fig. 7 in order to encode the positions of the sources so it is possible to associate each classification output neuron to a specific box of the map. This way, using the example from Fig. 7, where the boxes 0 and 8 are the ones with a source, the classification target is as follows:

$$[p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8] = [1, 0, 0, 0, 0, 0, 0, 0, 1] \tag{9}$$

whereas the regression target is

$$[x_0, nan, nan, ..., nan, x_8, y_0, nan, nan, ..., nan, y_8] \tag{10}$$

for the location values, with $x_{0,8}$, $y_{0,8} \in \mathbb{R}$, and

$$[a_0, nan, nan, nan, nan, nan, nan, nan, a_8] \tag{11}$$

for the activity values, with $a_{0,8} \in \mathbb{R}^+$, and 'nan' representing the expression "Don't care".

B. Neural Networks Model

The model is implemented based on CNNs, where the initial convolutional layers extract features from the input while the fully connected layers (composing a Multilayer Perceptron (MLP)) predict the classification and regression outputs. The size of the dense layers, the number of hidden layers, the number of respective neurons, and the number of convolutions (as well as the kernel size, the stride and the depth) have been subject of extensive experiments. In the end, the best hyperparameters found are represented in Fig. 8, where it was observed that adopting small differences in these hyperparameters according to the classification, localization and activity outputs, better results are achieved. Being a multi-label classification problem, the loss function used during the training process is the Binary Cross Entropy function, which is given by the following expression:

$$\mathcal{L}_p = -\frac{1}{N_{\text{sim}}} \sum_{i=1}^{N_{\text{sim}}} [p_i \log(\hat{p}_i) + (1 - p_i) \log(1 - \hat{p}_i)] \tag{12}$$

Based on the MSE metric, the loss functions for the activity and the location are, respectively, as follows:

$$\mathcal{L}_a = \frac{1}{N_{\text{sim}}} \sum_{i=1}^{N_{\text{sim}}} p_i (\hat{a}_i - a_i)^2 \tag{13}$$

$$\mathcal{L}_{x,y} = \frac{1}{N_{\text{sim}}} \sum_{i=1}^{N_{\text{sim}}} p_i \left[(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2 \right] \tag{14}$$

with the output values $(\hat{p}_i, \hat{a}_i, \hat{x}_i, \hat{y}_i)$. So, if there is no source in a certain cell, multiplying by p_i ensures that the regression predictions of that cell do not get penalized.

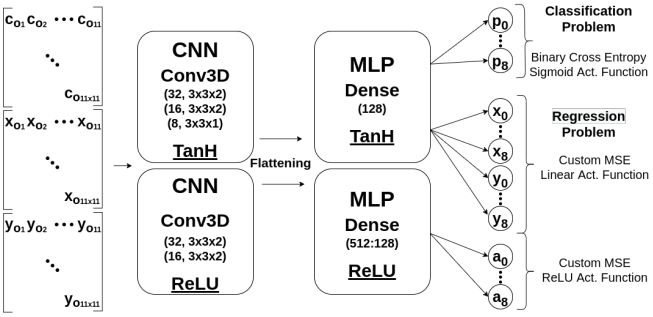

FIG. 8: Scheme of the final ANN model.

C. Training Process and Results

During the training process, there are some parameters that must remain fixed, namely the background intensity, the range of activity values of the sources, the scenario dimensions (width and length) used during the last division of the DAC process, and the approach used for the observations acquisition. Otherwise, if any of these parameters change considerably, then a new training process must be performed.

The optimizer used during training is the Adam algorithm [24], with a learning rate $\eta = 10^{-4}$, and a $L2$ (ridge regression) regularization [25] is implemented on every layer, with $\lambda = 0.001$. Then, having simulated $20,000$ samples in total, which leads to $\sim 16,000$ data for

training, it means that we get approximately 65 batches, using a batch size of 256. Then, 600 epochs are performed, where each epoch represents a loop made over all the batches.

In Table I the best values of accuracy and loss error are presented, related to the Early-stopping point, for the three different approaches regarding the way the observations enter the ANN. The accuracy is very high, which means that the model is detecting the region with a source almost perfectly. Regarding the position predictions, the error is lower than 0.02 m^2 (much shorter than the scenario dimensions of 3.33×3.33 m), which means that the model is able to get precise results. About the activity estimations, although it seems we are getting an error too high, around $\sim 10^{14}$ Bq^2, it is important to note that, since the target activities are within the range $[10^6, 10^8]$ Bq, and $\mathcal{L}_a$ is based on MSE function (see eq. (13)), when computing the loss error, all the quadratic errors of entire training data are being summed, leading to an expected huge loss error.

Approach	**Pooling**	**Acc (%)**	$\mathcal{L}_a$ $(10^{14}$ $Bq^2)$	$\mathcal{L}_{x,y}$ $(10^{-2}$ $m^2)$
Grid	-	97.33	0.594	1.025
Random	Avg	95.38	1.247	1.802
	Max	95.14	1.655	1.879

TABLE I: Early-stopping points of the different approaches.

Threshold study for DAC algorithm

An important characteristic of the classification problem is the threshold value, i.e the value of probability (classification output) from which it means that there is at least one source on the corresponding box. Although it is commonly assumed that the threshold should be 0.5, the choice of a threshold is a problem-dependent. Besides, we may assume different values of threshold depending on the division and the goals we have according to that division. Therefore, in Fig. 9, it is presented the plots of the quality metrics related to the classification problem, using the test data set. In the first divisions, the goal is to have a low FNR (rate of FN) as possible so as not to have lost sources at the beginning of the DAC process, and a low FPR (rate of FP) so unnecessary boxes are not approached later. In the last division, the goal is to obtain the best combination of accuracy and F1-score and with FPR practically zero. In intermediate divisions, the threshold must belong to the range in which accuracy and F1-score present the best results, and both FNR and FPR remain less than 10%. This way, a threshold value of 0.1 is selected for "Division 0", 0.9 for the last division, and for the intermediate divisions, the $[0.5, 0.8]$ range is the choice for the Grid approach. Although similar, the difference with the Random approach is explained in the thesis.

Considering Missing Observations

During the data acquisition step, it may happen that some regions of the scenario do not have any observation, which leads to the problem that some of the $s \times s$ entries

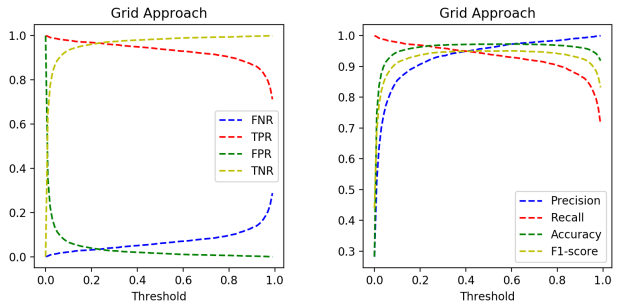

FIG. 9: Threshold study for the Grid approach.

of the input matrix do not get any value. In order to simulate this situation, the input layer becomes also a dropout layer, where a given rate of random input neurons are neglected. The goal is to extrapolate information from empty observations based only on those available. The problem of having regions of the map without any observation is more likely to happen when we collect data according to the Random approach. Therefore, using the average pooling, tests are made to assess the performance gain with rates 0, 0.05, 0.10 and 0.15 in scenarios with 0, 10, 15 and 20 random inputs set to zero. The results are shown in Fig. 10 and, analyzing the accuracy and F1-score metrics, we see that, initially with all the observations, the dropout rate of 0 provides the best results, but when we have 15 and 20 missing observations, the dropout rates of 0.05 and 0.10 produce the best results, whereas the dropout rate of 0 becomes the worst in the end. Regarding the activity and localization of the sources, which are respectively evaluated by eqs. (4) and (5), the dropout rate of 0 provides similar results compared to the remaining rates only with all observations, whereas, for 10 or more missing observations, the results improve as the dropout rate increases.

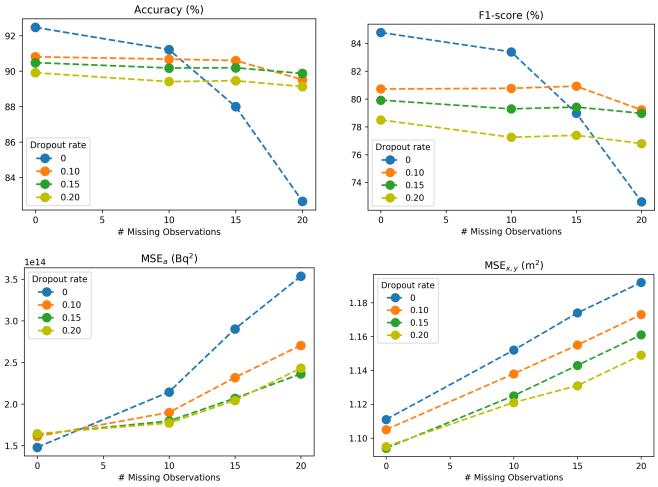

FIG. 10: Plots of Accuracy, F1-score, MSE_a and $\text{MSE}_{x,y}$, as a function of the number of missing observations.

VII. RESULTS

A. Simulation Results

Starting with the simulation results, two scenarios are approached (see Fig. 11), both with 6 sources but with different complexities, related to the proximity between the sources.

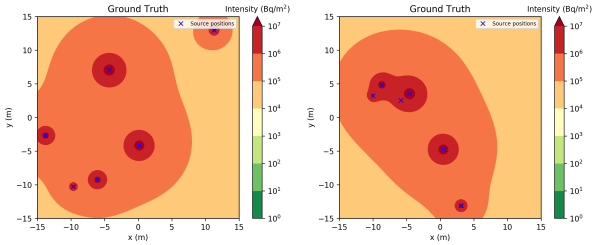

FIG. 11: Simulated scenarios **1.a)** and **1.b)**.

In relation to the scenario **1.a)**, the final result represented in Fig. 12 (right) shows that all the 6 sources are detected, and also the location and activity predictions are both very accurate, which is confirmed by the relationship between the 'o' (predictions) and the 'x' (targets) symbols and the similarity between colors, respectively. Regarding the scenario **1.b)**, from Fig. 12 (right), we see that now only 5 sources are detected and there is one missed detection. This non-detected source is so weak and there may be an overlay of the closest source's intensity. Besides, there are 5 false positives, where 4 of them are associated with target sources. Nevertheless, in practical terms, the "user" can also identify where to confirm the existence of hot-spots, in terms of location and must take into account that the predictions are independent of each other, i.e. the estimated activity is not distributed among the different predicted sources, and so the real value does not decrease.

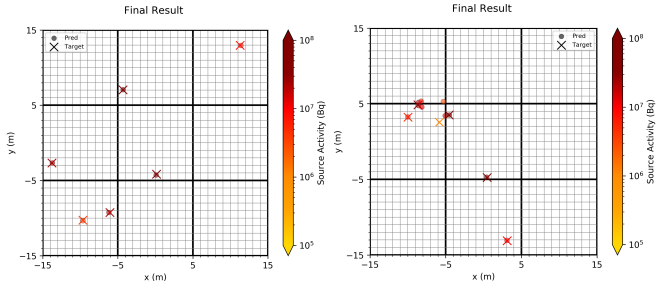

FIG. 12: Result of scenarios **1.a)** (left) and **1.b)** (right).

One of the main problems of this work is related to the fact that it is not known how many sources the ANNs are able to detect. Therefore, follows a scenario (called **1.c)**), with a high number of sources, namely 16, as represented in Fig. 13 (left). Analyzing the final result depicted in Fig. 13 (right), we see that all the 16 sources are detected, even with sources so close to each other and some of them much weaker than others, although some false positives arise. Regarding the parameters of the detected sources, we can verify that both location and activity estimations

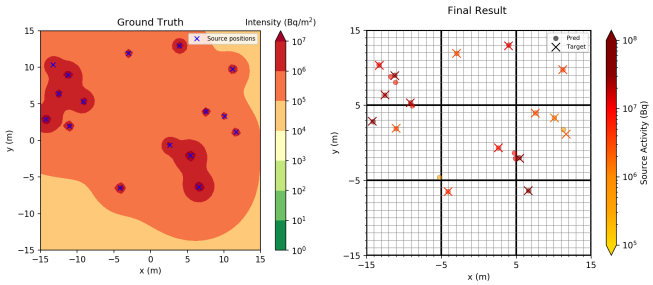

FIG. 13: Scenario **1.c)** (left) and final result (right).

are very close to the real values.

Considering the presence of Obstacles

In order to test the presence of obstacles in a scenario, two obstacles (with $\mu = 18.4m^{-1}$) were added to the previous scenario **1.b)**. Recall that there was not a new training process, and so the weights of the model are still the same. The configuration and the position of the obstacles are represented in Fig. 14 (left).

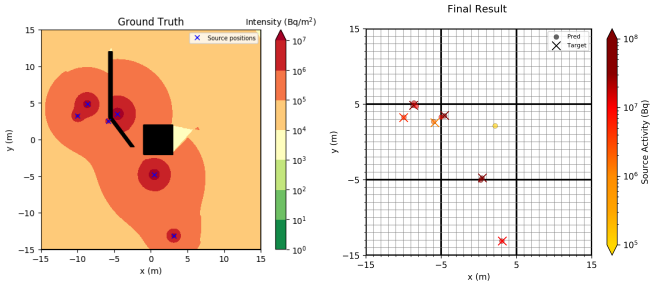

FIG. 14: Scenario **1.d)** (left) and final result (right).

In Fig. 14 (right) it is represented the final result. All the 6 sources are now detected, including the one that was not detected without the obstacles. This also means that, even with a more complex environment that was not part of the training data set at all, the ANN model is capable of detecting the sources, presenting even better results in relation to the model's accuracy. This may be related to the shielding created by the obstacles, which reduces the interference between two or more sources, and leads to better observations for the individual sources. Regarding the disadvantages, we see that, although there are less false positives, new false positives may occur near to the position of the obstacle(s), due to the fact that there may be gaps of intensity that lead to wrong interpretations by the ANNs. Still, this false positive has a very low activity (less than 10^5 Bq). Therefore, the presence of obstacles does not provide significant negative effects, and may even provide good effects (namely by reducing false negatives and false positives).

Non-point-like source Approach

To test non-point-like sources, the simulator is used in such a way that a non-point-like source is actually an arrange of multiple sources in a certain configuration. In Fig. 15 two different scenarios are presented. On the left, with scenario **1.e)**, there are 36 sources in a L-shaped

configuration with the activity of 5×10^6 Bq and with a distance of 10 cm from each other. On the right, with scenario **1.f)**, we have the same number of sources with the same values of activity, but the distance between each other is just 3 cm. Then, in Fig. 16, the final results show

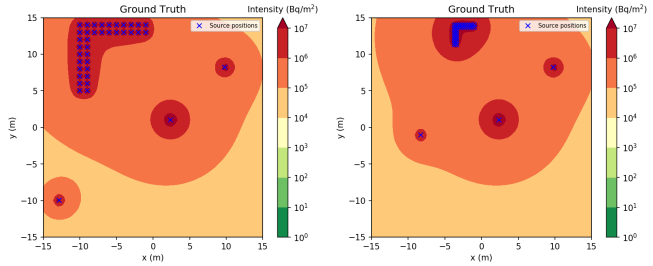

FIG. 15: Scenarios **1.e)** (left) and **1.f)** (right).

that, in both cases, the point-like sources are detected with accurate predictions in relation to the corresponding parameters, and so there is not any negative impact. Regarding the non-point-like source, in scenario **1.e)** it is detected almost the total number of actual sources (28 of 36), and the activity values are close to the actual value (5×10^6 Bq). On the other hand, in scenario **1.f)**, there is only 3 predictions, which is closer to what was desirable (i.e., only one prediction), located in points with almost the same distance between each other, and activity values within the range $[8, 9] \times 10^7$ Bq.

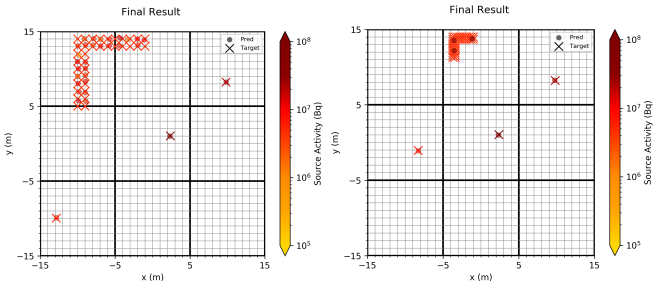

FIG. 16: Result of scenarios **1.e)** (left) and **1.f)** (right).

B. Lab Results

At Instituto Superior Técnico, some lab tests (see setup in Fig. 17), in short-scale scenarios, were carried out with real sources with activity values within the range $[10^2, 10^3]$ Bq. So, a new training process was performed, using the Grid approach with 11×11 inputs.

In Fig. 18 it is depicted the lab scenario **2.a)** with, on the left, the data points acquired by MARIA, with CPM (counts per minute) unit. The conversion to Bq/m^2 is explained in the thesis). Some isolated points can be visualized with intensity ~ 150 CPM due the fact that the sources are covered by lead which is not entirely homogeneous, and so the sources are not as isotropic as desirable. Applying the ANN model (on the right), the final result shows that false positives may appear (although most of them with very low estimated activity), which may be related to the previous isolated data points. Regarding the

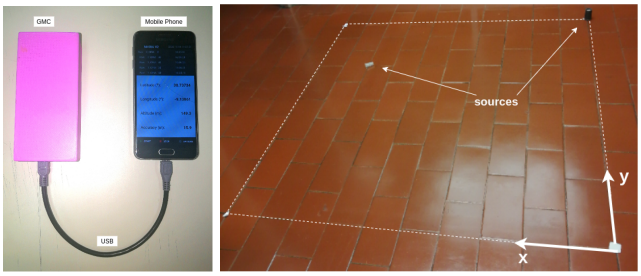

FIG. 17: Lab setup: Mobile phone running MARIA (left) and scenario's representation (right).

true positives, accurate results are obtained, since both estimations of the position and activity of the sources are very close to the targets (represented by circles).

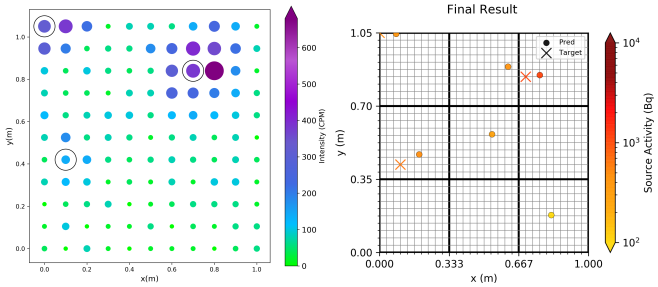

FIG. 18: Data from lab scenario **2.a)** (left) and the final result (right).

C. Real Results

Finally, in this section we have one of the real scenarios (located at old deposits of radioactive ore) tested, with dimensions that are very larger than the ones considered in the previous subsections, and represent a real context where the proposed solution of this work is intended to be used. Fig. 19 (up) shows the considered scenario **3.a)** with the radiological heatmap provided by MARIA.

The ANN model is used with the Random approach and Fig. 19 (bottom) shows the final results, after applying the DAC algorithm. Since the activity and position of the targets are unknown, instead of visualizing these parameters, a plot of all observational data with an intensity higher than 100 CPM is done, so the location of possible hot-spots can be seen. Thus, we see that the three main hot-spots seen in the heatmap (with red color) are detected and, besides that, there are also a few sources identified in other points of the map.

VIII. CONCLUSIONS

The results of the present work show that the usage of ANNs provides good results in detecting radioactive hot-spots in simple and more complex scenarios, where the most existing algorithms present some limitations. The main downsides are related to the long time required for training, the computational costs (i.e., the software and

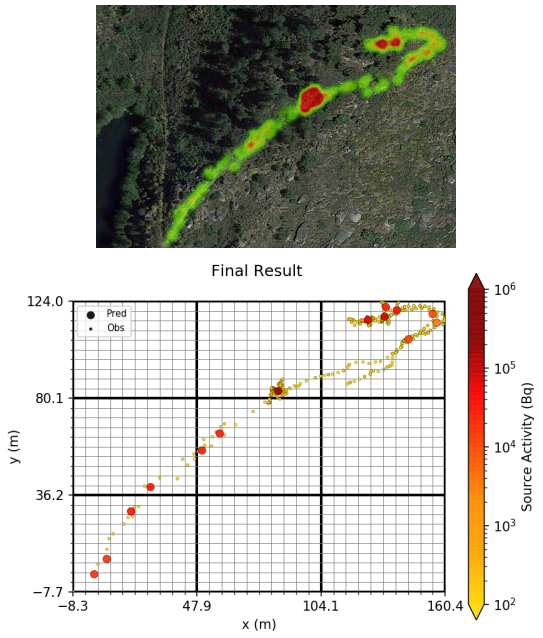

FIG. 19: Radiological heatmap of scenario **3.a)** (up), and final result (bottom).

hardware needed to train the ANN model), and the use of a training data set composed only of simulated data.

Nevertheless, having the ANN model already trained, it can be used through the DAC algorithm in order to obtain fast and accurate predictions, which can be very useful in real-time operations (e.g. for a real surveillance). Simulation results verified the accuracy of the algorithm, involving realistic complexities. The dimensions of the various tested scenarios had variations from 1 m^2 to more than 10^4 m^2, which shows that the model is scalable to large regions and capable of dealing with a high number of sources. Besides, the use of a dropout input layer may improve the results when missing observations must be considered. The presence of obstacles was also tested and, without requiring any prior information about them, the algorithm shows to be capable of obtaining good results and even improving in some cases, by reducing the number of missed detections. Lastly, non-point-like sources were approached, by arranging multiple sources in a certain configuration, and the performance of the model was dependent on the distance between each of these sources. This is, the shorter the distance, the lower number of detected sources was associated with a non-point-like (target) source.

Thus, we can affirm that Machine Learning, namely ANNs, has indeed conditions to make such a difference on the nuclear problem of radioactive hot-spots detection, taking into account the reproducibility of the presented model in the most varied scenarios, as well as the expected improvements in computing time and in Machine Learning frameworks with developments in both software and hardware.

[1] "Cbrn risk mitigation and security governance programme," `http://www.unicri.it/topics/cbrn/`, accessed: 2019-12-10.

[2] "International atomic energy agency. naturally occurring radioactive material," `https://www.iaea.org/topics/radiation-safety-norm`, accessed: 2019-11-22.

[3] H. S. Cho and T. H. Woo, Annals of Nuclear Energy **94**, 138 (2016).

[4] K. Oyama, H. Wakabayashi, *et al.*, 2012 IEEE 36th Annual Computer Software and Applications Conference Workshops , 607–612 (July 2012).

[5] World Institute for Nuclear Security, "Data analytics for nuclear security," (Jan 2015).

[6] C. Q. Wu *et al.*, IEEE Transactions on Industrial Informatics **15**, 2308 (2019).

[7] N. S. V. Rao *et al.*, in *2008 11th International Conference on Information Fusion* (2008) pp. 1–8.

[8] M. Morelande *et al.*, 10th International Conference on Information Fusion , 1 (07 2007).

[9] H. E. Baidoo-Williams, "Maximum likelihood localization of radiation sources with unknown source intensity," (2016), arXiv:1608.00427 [math.OC].

[10] G. Fernandez *et al.*, Nuclear Engineering and Design **359** (2019), 10.1016/j.nucengdes.2019.110479.

[11] E. Wacholder *et al.*, Nuclear Technology **110**, 228 (1995).

[12] J. Ma and J. Jiang, Progress in Nuclear Energy - PROG NUCL ENERGY **53**, 255 (2011).

[13] M. R. Minar and J. Naher, "Recent advances in deep learning: An overview," (2018).

[14] J. Redmon *et al.*, CoRR **abs/1506.02640** (2015).

[15] Y. REN and G. BAI, Chinese Journal of Aeronautics **24**, 25 (2011).

[16] K. M. Ting, "Precision and recall," in *Encyclopedia of Machine Learning*, edited by C. Sammut and G. I. Webb (Springer US, Boston, MA, 2010) p. 781.

[17] "Maria," `https://www.ipfn.tecnico.ulisboa.pt/MARIA/`, accessed: 2019-12-17.

[18] Y. Brouwer, A. Vale, B. Gonçalves, and H. Fernandes, EPJ Web of Conferences **225**, 06005 (2020).

[19] Y. Brouwer, *Radiological Monitor for Mobile Robots Operating in Scenarios with Nuclear Threats*, Master's thesis, Instituto Superior Técnico, Portugal (2019), supervised by Alberto Vale and Rodrigo Ventura.

[20] R. Bridson, (2007), 10.1145/1278780.1278807.

[21] T. Kluyver *et al.*, in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, edited by F. Loizides and B. Schmidt (2016) pp. 87 – 90.

[22] F. Chollet *et al.*, "Keras," `https://keras.io` (2015).

[23] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," (2016).

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," (2014), arXiv:1412.6980 [cs.LG].

[25] C. Cortes, M. Mohri, and A. Rostamizadeh, "L2 regularization for learning kernels," (2012).