ORNL/TM-1999/320

APPLICATION OF STOCHASTIC AND ARTIFICIAL INTELLIGENCE METHODS FOR NUCLEAR MATERIAL IDENTIFICATION

Sara Pozzi[*]
Department of Nuclear Engineering
Polytechnic of Milan
Italy

F. J. Segovia
Facultad Informatica
Universidad Politecnica de Madrid
Madrid, Spain

August 13, 1999

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831-6004
managed by
LOCKHEED MARTIN ENERGY RESEARCH CORP.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-96OR22464

[*] Sara Pozzi was a student visitor at the Oak Ridge National Laboratory.

This page intentionally left blank.

CONTENTS

This page intentionally left blank.

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF TABLES

This page intentionally left blank.

ACKNOWLEDGMENTS

We would like to thank R. B. Perez, T. E. Valentine, J. T. Mihalczo, and J. K. Mattingly for their suggesting the topic and for providing help for the use of the MCNP-DSP Monte Carlo Code. Many thanks go to Matteo Cantoni for his help in the implementation of the neural network-genetic algorithm approach and to Professor Marzio Marseguerra for his valuable intellectual contribution and experienced guidance. Both these individuals are from the Politecnico, Milano, Italy. One of us, Sara Pozzi, would like to express her gratitude towards the staff of the Instrumentation and Controls Division for their help and kindness during her stay at the Oak Ridge National Laboratory.

This page intentionally left blank.

1. INTRODUCTION

Nuclear materials safeguard efforts necessitate the use of non-destructive methods to determine the attributes of fissile samples enclosed in special, non-accessible containers. To this end, a large variety of methods has been developed at the Oak Ridge National Laboratory (ORNL) and elsewhere.[1, 2] Usually, a given set of statistics of the stochastic neutron-photon coupled field, such as source-detector, detector-detector cross correlation functions, and multiplicities are measured over a range of known samples to develop calibration algorithms. In this manner, the attributes of unknown samples can be inferred by the use of the calibration results.

The sample identification problem, in its most general setting, is then to determine the relationship between the observed features of the measurement and the sample attributes and to combine them for the construction of an optimal identification algorithm. The goal of this paper is to develop an artificial intelligence (AI) approach to this problem whereby neural networks (NN) and genetic programming (GP) algorithms are used for sample identification purposes. To this end, the time-dependent MCNP-DSP[3] Monte Carlo code has been used to simulate the neutron-photon interrogation of sets of uranium metal samples by a ^{252}Cf-source. The resulting sets of source-detector correlation functions, $R_{12}(\tau)$ as a function of the time delay, τ, served as a data-base for the training of the AI algorithms.

The organization of this paper is as follows: Section 2 describes the Monte Carlo simulations of source-detector cross correlation functions for a set of uranium metallic samples interrogated by the neutrons and photons from a ^{252}Cf source. From this database, a set of features is extracted in Section 3. The use of neural networks (NN) and genetic programming to provide sample mass and enrichment values from the input sets of features is illustrated in Sections 4 and 5, respectively. Section 6 is a comparison of the results, while Section 7 is a brief summary of the work.

This page intentionally left blank.

2. ^{252}Cf-SOURCE-DRIVEN SIMULATIONS

In the ^{252}Cf-source-driven measurement the source undergoes spontaneous fission emitting neutrons and gamma rays. The timing of each spontaneous fission event is recorded in appropriate time bins. If fissile material is present inside the sample to be analyzed, the neutrons emitted by the source will initiate fission chains. Neutrons and gamma rays from the source as well as those eventually emitted by the fissile system are measured with two detectors. The detection times are also recorded, in time bins of 1 ns. The uranium sample to be analyzed is placed between the source and two fast plastic scintillation detectors. The source was located at 25.4 cm from the center of the uranium metal sample at a height of 10 cm. The detectors, 10.16 cm width and height and 5.08 cm thick, are placed one on top of the other at a distance of 25.4 cm from the center of the sample.

Simulations were performed with cylindrical and spherical samples of seven different masses (8 kg, 10 kg, 12 kg, 14 kg, 16 kg, 18 kg, and 20 kg). The different masses were obtained by increasing the sample radius, both in the case of the cylinders (in which case the height was kept constant at 20 cm) and in that of the spheres. For each mass, four different enrichments were tested ranging from depleted to highly enriched (0.2 wt% ^{235}U, 36.0 wt% ^{235}U, 50.0 wt% ^{235}U, and 93.15 wt% ^{235}U). Two additional simulations were run for both cylinders and spheres giving a total of 30 simulations for the cylindrical samples and 30 for the spherical ones. An additional simulation run with no sample between source and detectors will be referred to as the void simulation.

2.1 CROSS-CORRELATION FUNCTIONS

The source-detectors cross-correlation functions [$R_{12}(\tau)$] are generated by correlation of the source signal with the combined signal from the two detectors, and normalizing to the source count rate to remove the dependence on the source.

In Figure 2.1 the cross-correlation $R_{12}(\tau)$ as a function of the delay time between source fission and corresponding detection is shown for cylinders of varying enrichment and fixed mass (20 kg). The curve consists of two major components: a first peak due to directly transmitted gamma rays from the ^{252}Cf fission (the photo peak), and a second, broader peak due to directly transmitted and scattered neutrons from the source and secondary neutrons and gamma rays from fission induced inside the uranium sample. As it can be seen from the figure, the directly transmitted gamma rays are not very sensitive to the fissile mass since gamma ray attenuation is not related to fission. On the other hand, the second peak of the cross-correlation function depends strongly on enrichment.

In the first part of the second peak the curves show similar behavior for time lags below 20 ns, since the directly transmitted neutrons and secondary photons are not strongly dependent on enrichment. Above time lags of about 20 ns, the peak broadens: neutrons generated by secondary fission inside the fissile material increase and the number of neutron generations increases. The total path covered by the neutrons before a detection event occurs also increases.

In Figure 2.3 the cross-correlation function obtained with spherical samples is shown. The first peak is much higher than in the case of the cylinders. This can be explained in terms of the greater attenuation given by the geometry of the cylindrical samples whereas the spherical samples allow more gamma rays from the source to reach the detectors directly.

Figures 2.2 and 2.4 show the source-detectors cross-correlation function in the case of cylinders and spheres of varying mass and constant enrichment (36 wt% ^{235}U). In this case, both the gamma peak and the secondary peak height are inversely related to mass: as the sample mass increases, so does the attenuation of gamma rays and neutrons. A similar relationship was found with other values of enrichment.

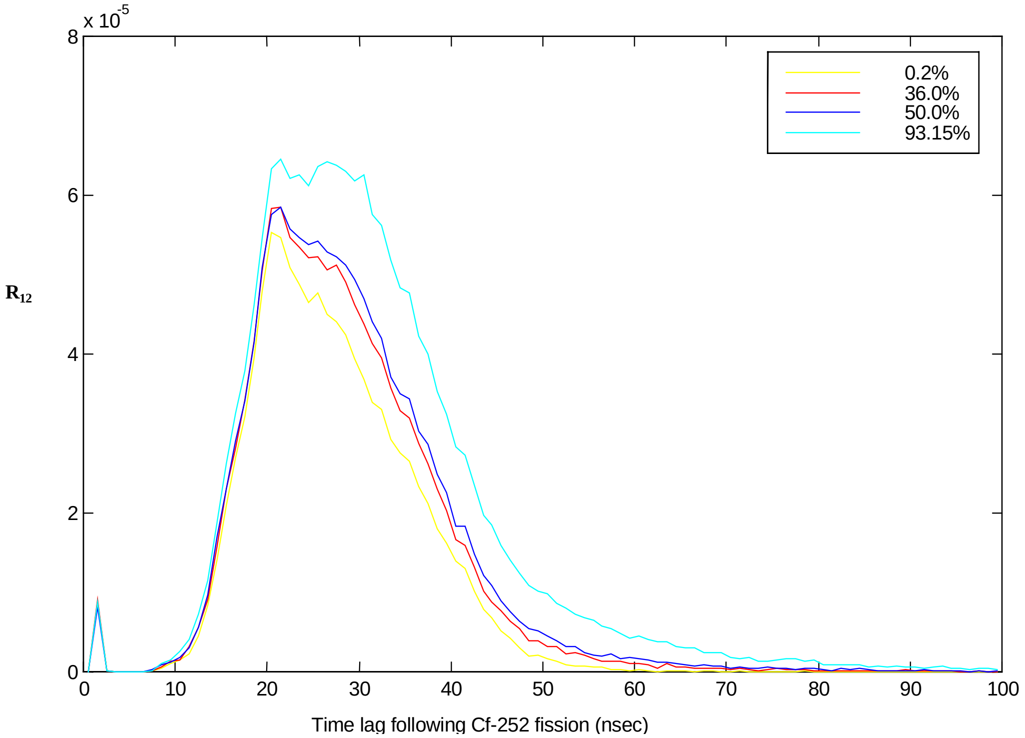


Fig. 2.1. Source-detectors cross-correlation functions for uranium cylinders of different enrichments and fixed mass (20 kg).

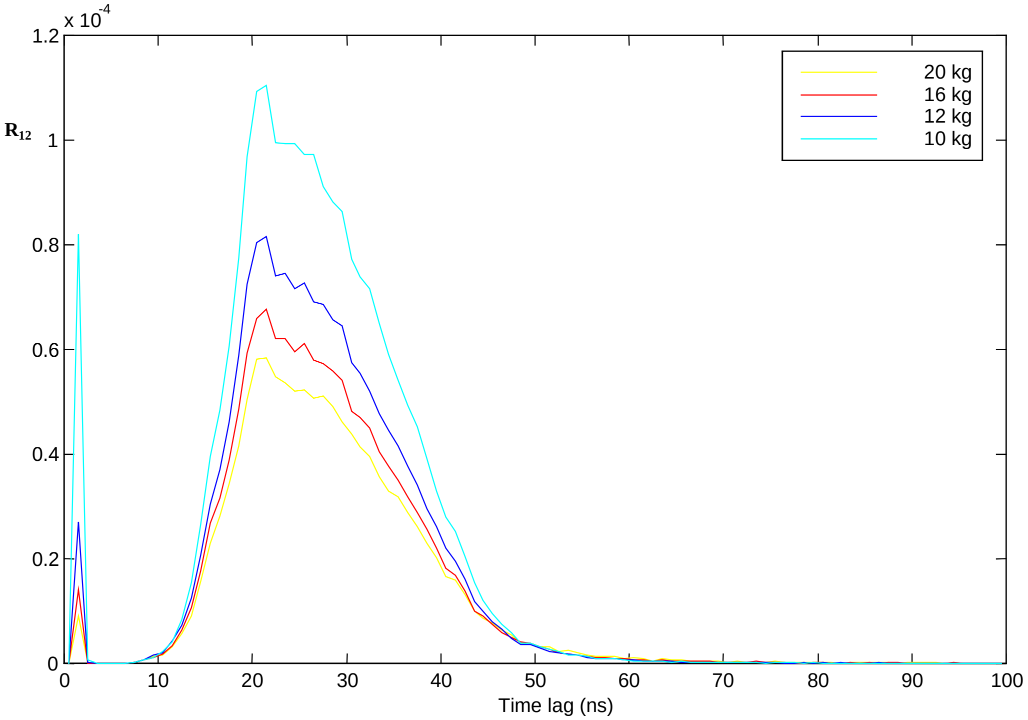


Fig. 2.2. Source-detectors cross-correlation function (R_{12}) for uranium cylinders of varying masses and fixed enrichment (36 %wt ^{235}U).

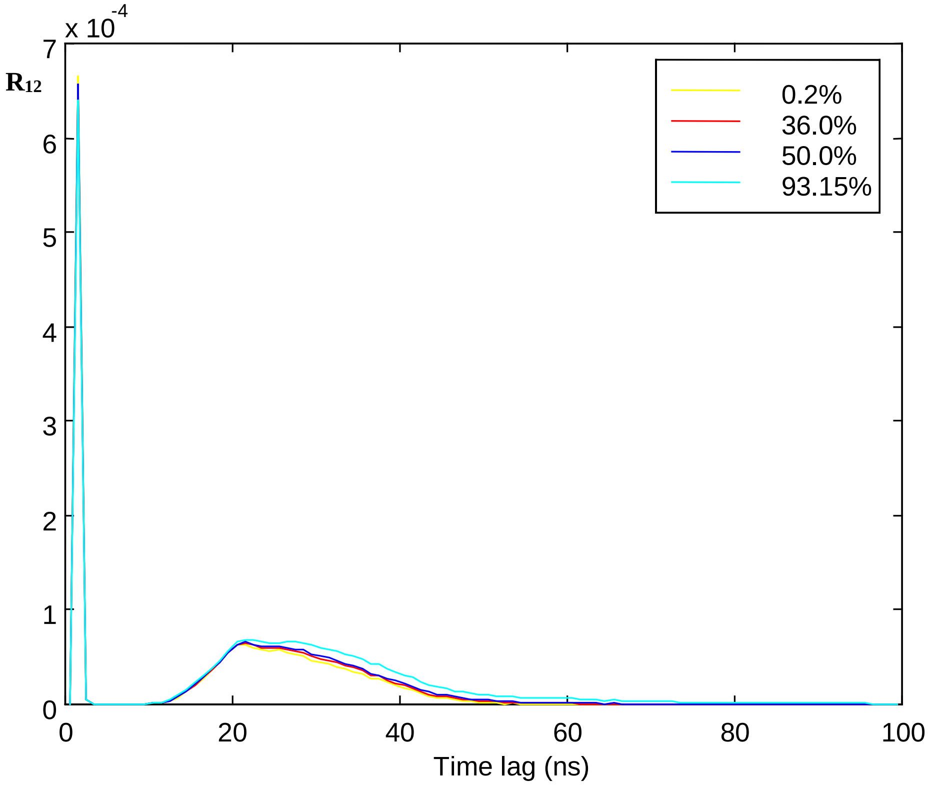


Fig. 2.3. Source-detectors cross-correlation functions for uranium spheres of different enrichments and fixed mass (20 kg).

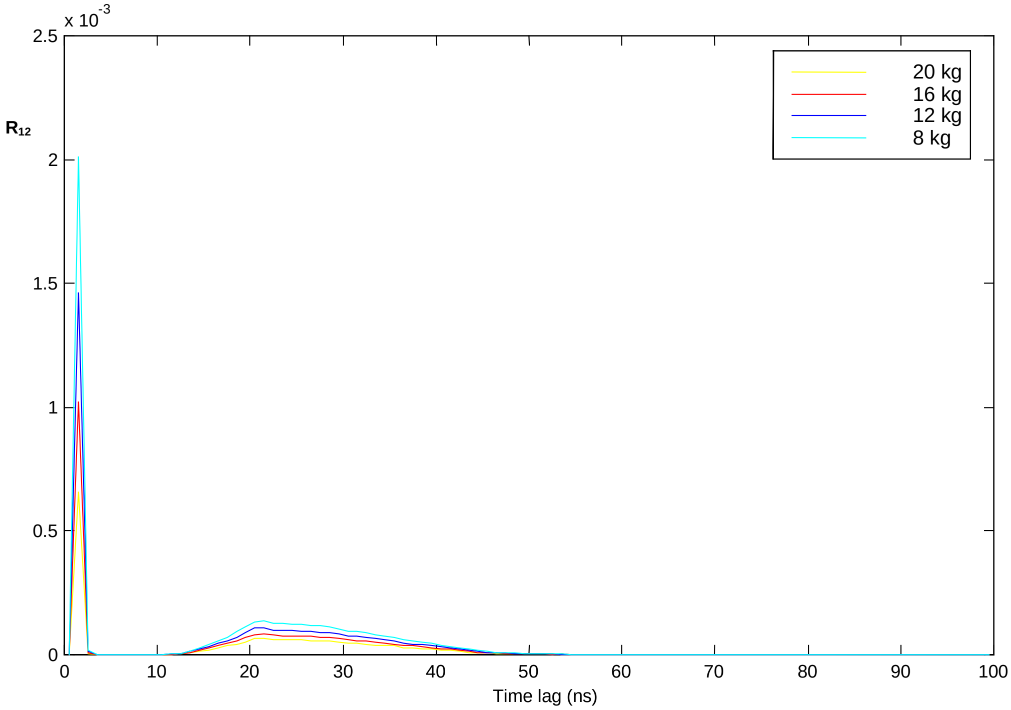


Fig. 2.4. Source-detectors cross-correlation function (R_{12}) for uranium spheres of varying masses and fixed enrichment (36 %wt ^{235}U).

3. SELECTION OF FEATURES FOR THE SAMPLE IDENTIFICATION ALGORITHM

The selection of features for the sample identification algorithm (SIA) was performed on the basis of their relationship to sample attributes and of their ability to discriminate between close numerical values within each attribute group.

The first feature (F_1) chosen is the integral of the cross-correlation function at time lags from 0 to 8 ns, normalized to the same integral of the void calculation. It essentially corresponds to the normalized area of the first peak of the cross-correlation function.

$$F_1 = \frac{\sum_{i=0}^{8} R_{12}(\tau_i)}{\sum_{i=0}^{8} R_{12Void}(\tau_i)} \tag{3.1}$$

A plot of F_1 as a function of the sample's total mass is given in Figures 3.1 and 3.2, for all values of enrichment. Figure 3.1 refers to cylinder simulations, while Figure 3.2 refers to sphere simulations. As expected, F_1 depends only on the sample mass.

The second feature chosen is the integral of the cross-correlation function at time lags from 0 to 100 ns, normalized to the same integral of the void simulation.

$$F_2 = \frac{\sum_{i=0}^{100} R_{12}(\tau_i)}{\sum_{i=0}^{100} R_{12Void}(\tau_i)} \tag{3.2}$$

Inspection of Figures 3.3 and 3.4, shows that F_2 is sensitive to both the sample's total mass and enrichment.

The moments of the cross correlation function were also examined i.e.:

$$\tau^{(n)} = \frac{\sum_{i=0}^{100} R_{12}(\tau_i)\,\tau_i^n}{\sum_{i=0}^{100} R_{12}(\tau_i)} \tag{3.3}$$

However, up to n=3, all the moments examined looked very much alike, with the n=1 moment giving the best resolution. Hence, the average delay time, $\bar{\tau}$, was selected as the feature F_3, shown in Figures 3.5 and 3.6 for cylindrical and spherical samples, respectively. This feature is essentially constant for the depleted samples, increases with sample enrichment and for high enrichments is very sensitive to sample mass.

Because the asymmetry of the second peak is generated by the neutron induced fission in the sample, the skewness of the cross correlation function was selected as feature, F_4, defined by the relation below.

$$F_4 = \frac{\mu_3}{\sigma^3} \tag{3.4}$$

where σ^3 is the cube of the standard deviation and μ_3 is the third moment about the mean value of the distribution. As shown in Figures 3.7 and 3.8, F_4 is especially sensitive to the lower values enrichment of the samples.

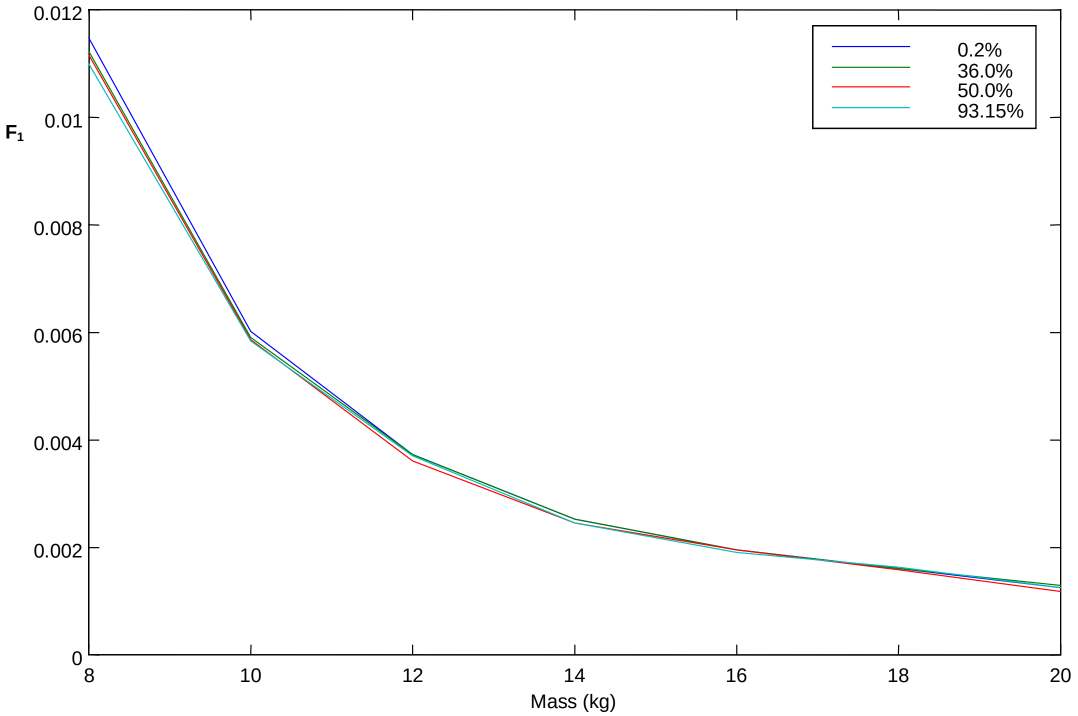


Fig. 3.1. Cylindrical samples: F_1 as a function of sample mass (kg) for the four different enrichments.

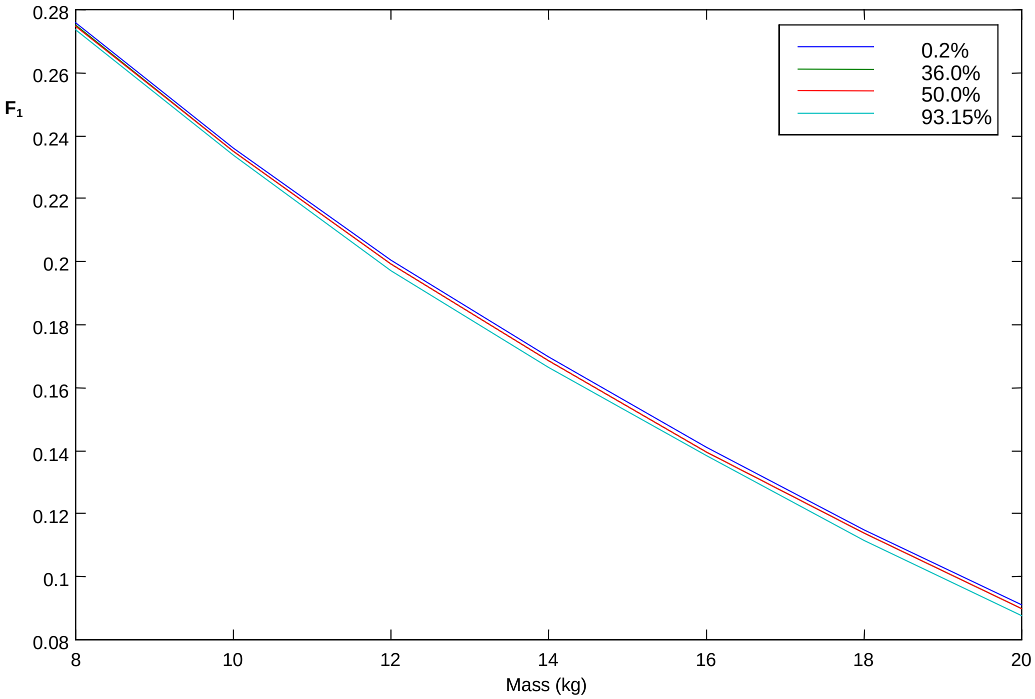


Fig. 3.2. Spherical samples: F_1 as a function of sample mass (kg) for the four different enrichments.

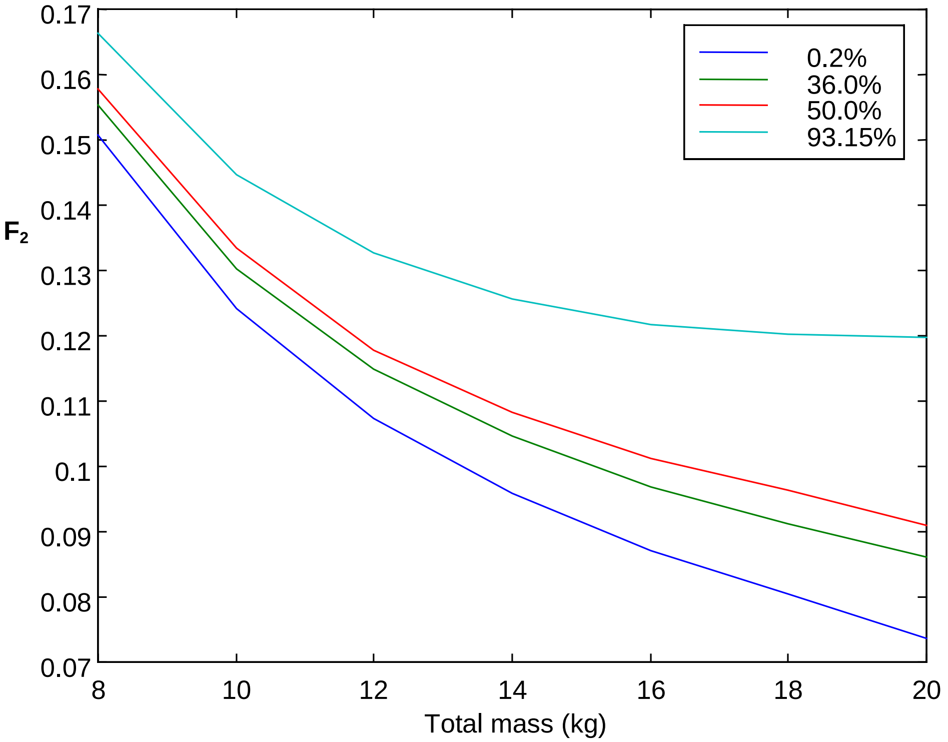


Fig. 3.3. Cylindrical samples: F_2 as a function of sample mass (kg) for the four different enrichments.

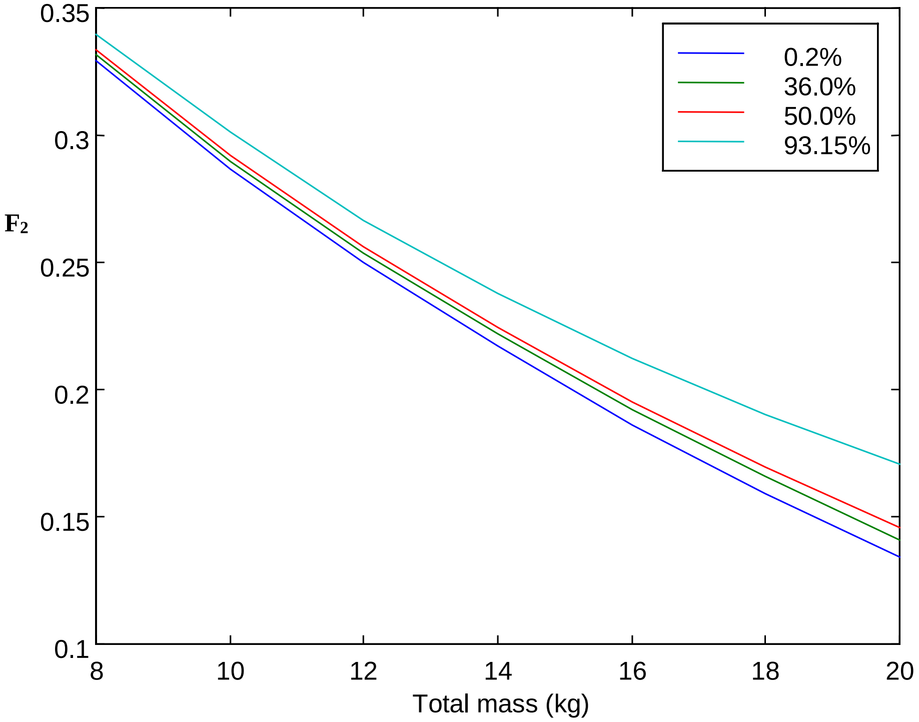


Fig. 3.4. Spherical samples: F_2 as a function of sample mass (kg) for the four different enrichments.

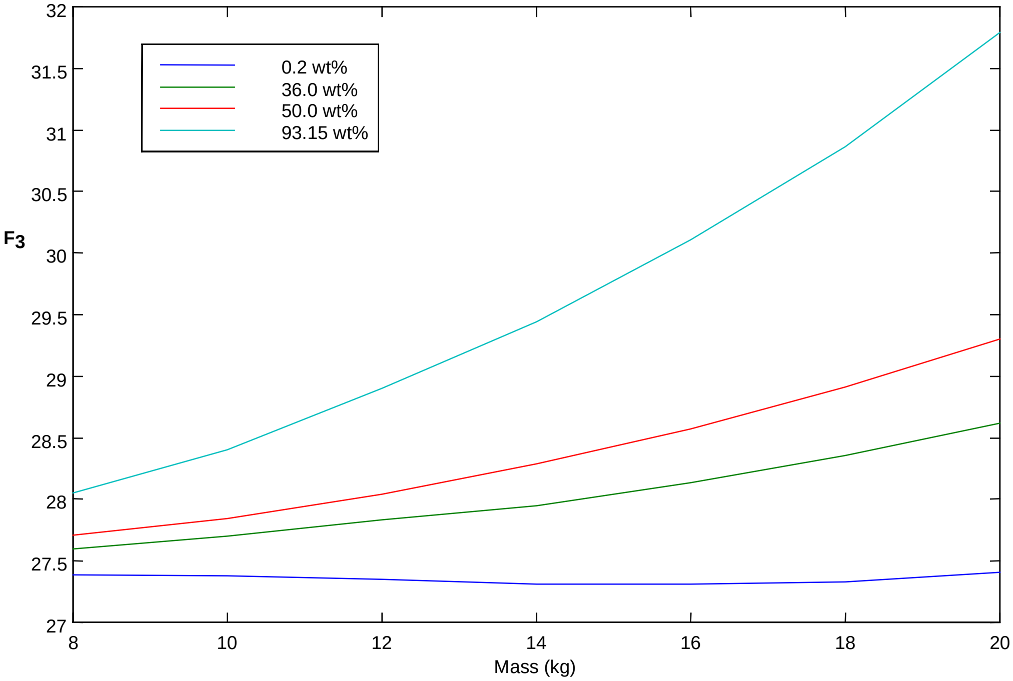


Fig. 3.5. Cylindrical samples: F_3 as a function of sample mass (kg) for the four different enrichments.

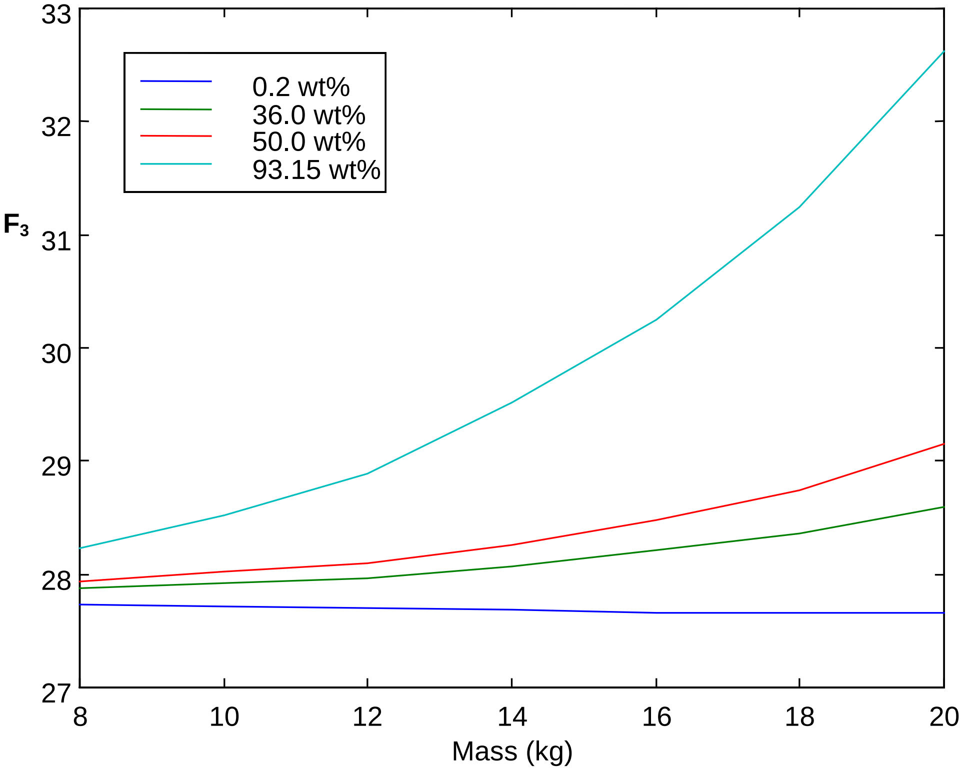


Fig. 3.6. Spherical samples: F_3 as a function of sample mass (kg) for the four different enrichments.

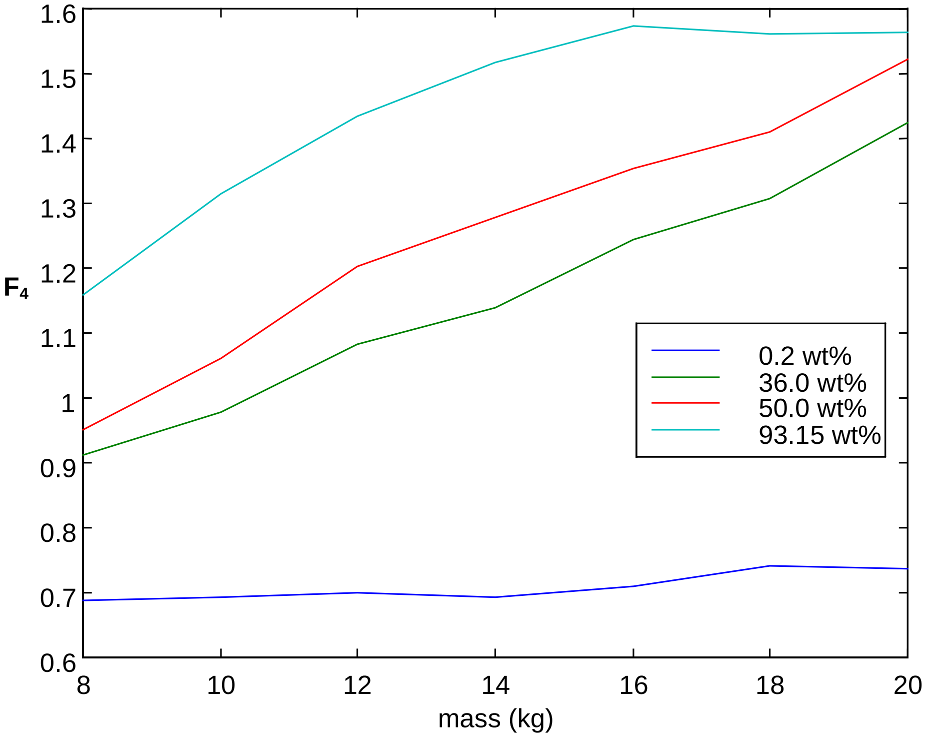


Fig. 3.7. Cylindrical samples: F_4 as a function of sample mass (kg) for the four different enrichments.

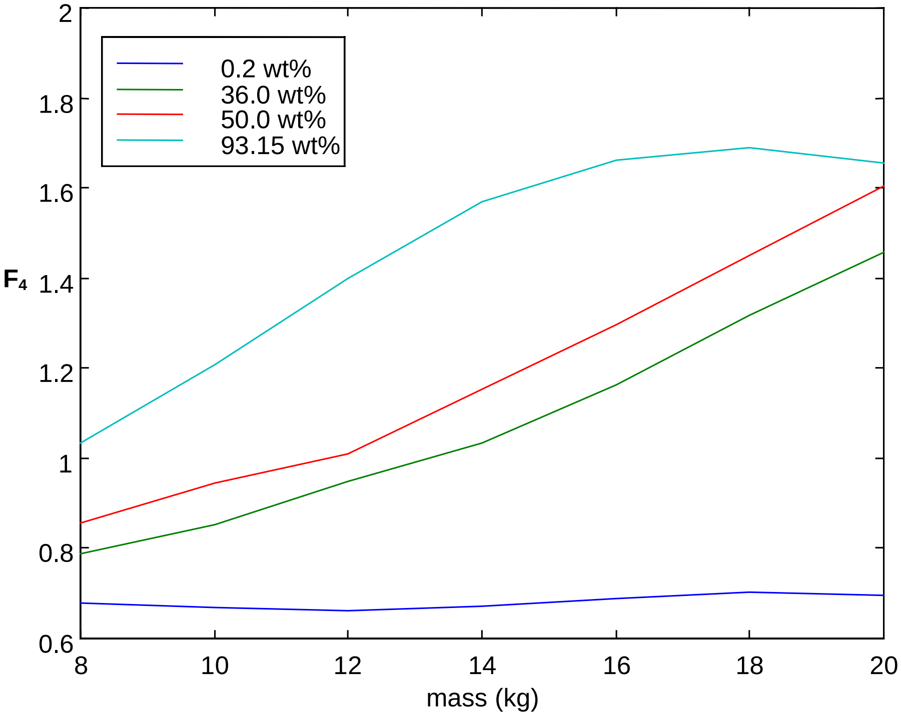
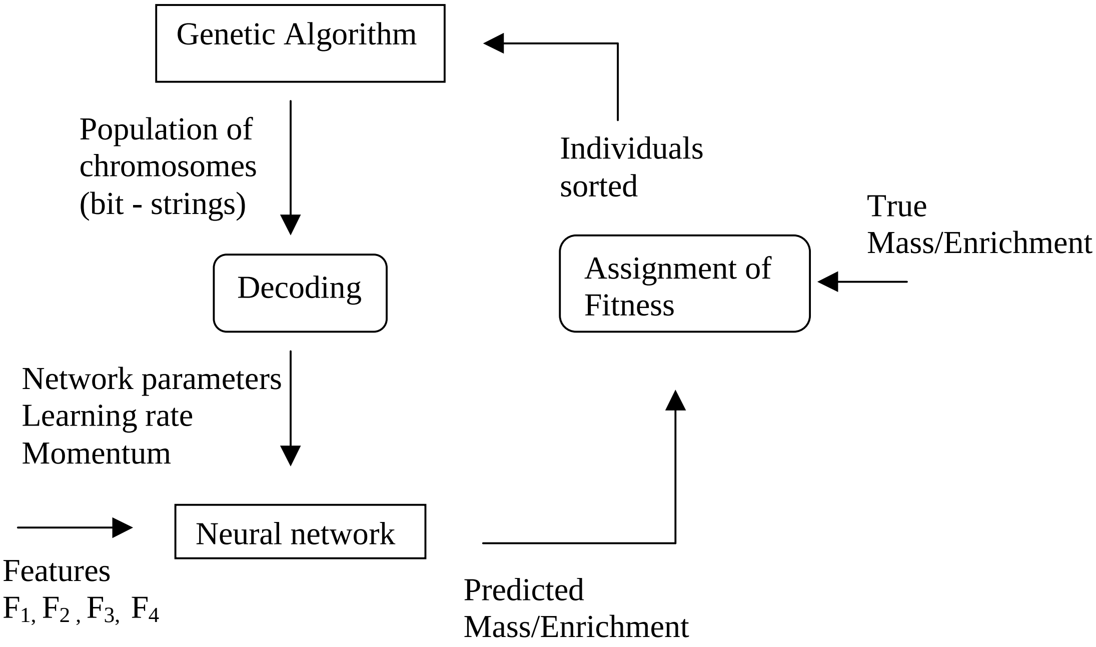


Fig. 3.8. Spherical samples: F_4 as a function of sample mass (kg) for the four different enrichments.

4. APPLICATION OF NEURAL NETWORKS TO NUCLEAR MATERIALS IDENTIFICATION SYSTEMS

Two three-layered artificial neural networks[4, 5, 6] were trained to generate a mapping from input (F_1, F_2, F_3, and F_4) to output (sample's mass and enrichment). The well known error back-propagation algorithm was used for the network's training. The activation functions were chosen to be sigmoidal from the input to the hidden layer and linear from the hidden layer to the output. The number of hidden nodes was set to two.

For each NN, the values of learning rate and momentum for the training were optimized by a genetic algorithm (GA)[7] (the choice of these parameters is usually made by a trial and error approach). In the GA an initial population of 50 chromosomes, each made up of two genes coding the quantities of interest, is allowed to evolve according to the rules of mating, crossover, and mutation, similarly to what occurs in biological systems. The objective is to maximize the fitness function, defined as the inverse of the network's training error. After a predetermined number of generations (100 in our case), the fittest chromosome is elected.



The values of learning rate and momentum selected as described above were then used in training the neural networks for the prediction of the total mass and enrichment of the samples on the basis of the four features F_1-F_4.

4.1 RESULTS

Having chosen a linear transfer function in the output nodes of the NN allows us to express the network mapping structure in terms of the simple analytical formula below:

$$Output = \frac{a_1}{1+e^{-\left(\sum_{i=1...4} b_i F_i + b_5\right)}} + \frac{a_2}{1+e^{-\left(\sum_{i=1...4} c_i F_i + c_5\right)}} + a_3 \qquad (4.1)$$

where a_i, (i=1,2,3), b_j, c_j, (j=1,…5) are coefficients which depend on the network's weights, given in table 4.1, and the output is the sample mass and enrichment.

Table 4.1: Coefficients of equation 4.1 for mass and enrichment prediction in both spherical and cylindrical samples

Cylinder mass	a_1	48.3	Sphere mass	a_1	24.9
	a_2	-28.1		a_2	58.9
	a_3	30.9		a_3	-1.81
	b_1	-156.1		b_1	1.09
	b_2	-33.1		b_2	2.85
	b_3	.0319		b_3	.34
	b_4	.344		b_4	1.0
	b_5	-.0559		b_5	-14.97
	c_1	15.1		c_1	-.118
	c_2	7.71		c_2	-5.70
	c_3	-.28		c_3	-.0237
	c_4	.131		c_4	.0255
	c_5	7.86		c_5	.855
Cylinder enrichment	a_1	-79.3	Sphere enrichment	a_1	-898.3
	a_2	-202.5		a_2	-499.4
	a_3	213.0		a_3	500.1
	b_1	196.8		b_1	-3.18
	b_2	-1.35		b_2	11.7
	b_3	-30.2		b_3	-1.33
	b_4	-2.93		b_4	-2.7
	b_5	8.96		b_5	33..2
	c_1	16.1		c_1	1.14
	c_2	-30.3		c_2	-10.0
	c_3	-.027		c_3	-0.15
	c_4	-1.82		c_4	-.958
	c_5	7.69		c_5	9.80

The sample mass and enrichment predictions obtained with the NN-GA approach are shown in the following Figures 4.1 – 4.4. Nineteen simulations, about two thirds of the data available, were selected for the NN training. Figures 4.1 and 4.2 refer to cylinder simulations and sphere simulations, respectively, used during the network training. The NN were tested with the remaining 11 cases. The results are shown in Figures 4.3 and 4.4. Inspection of these results shows that the present type of neural network can predict enrichment and mass values for uranium metallic samples to a very good approximation both in the case of the training patterns and in the test cases.

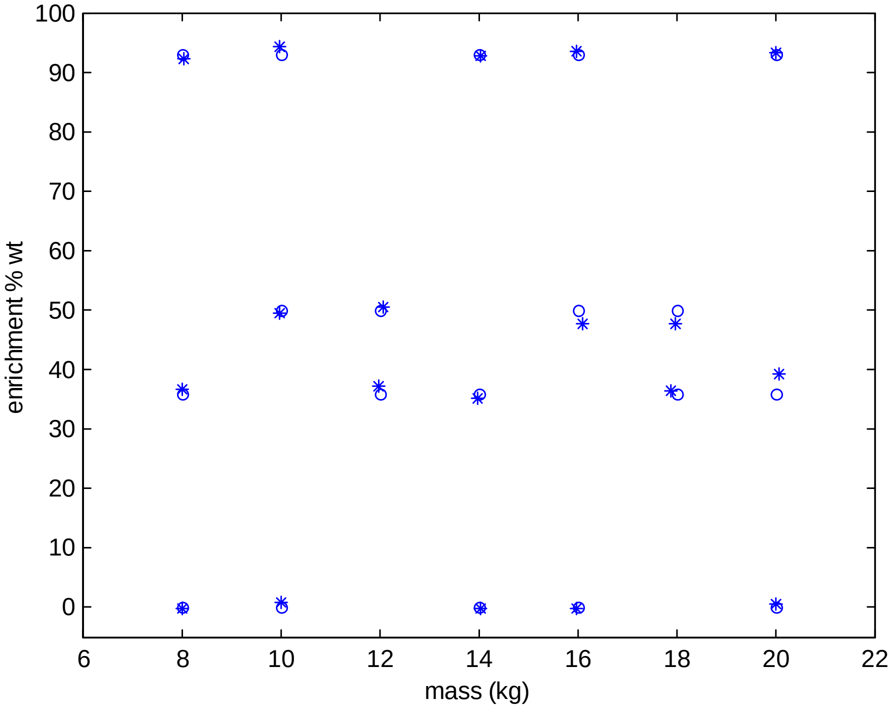


Fig. 4.1. Neural network prediction of mass and enrichment on the basis of features F_1, F_2, F_3, and F_4: training set of 19 cases relative to cylinder simulations. The true values are shown with the circles and the values predicted by the network with stars.

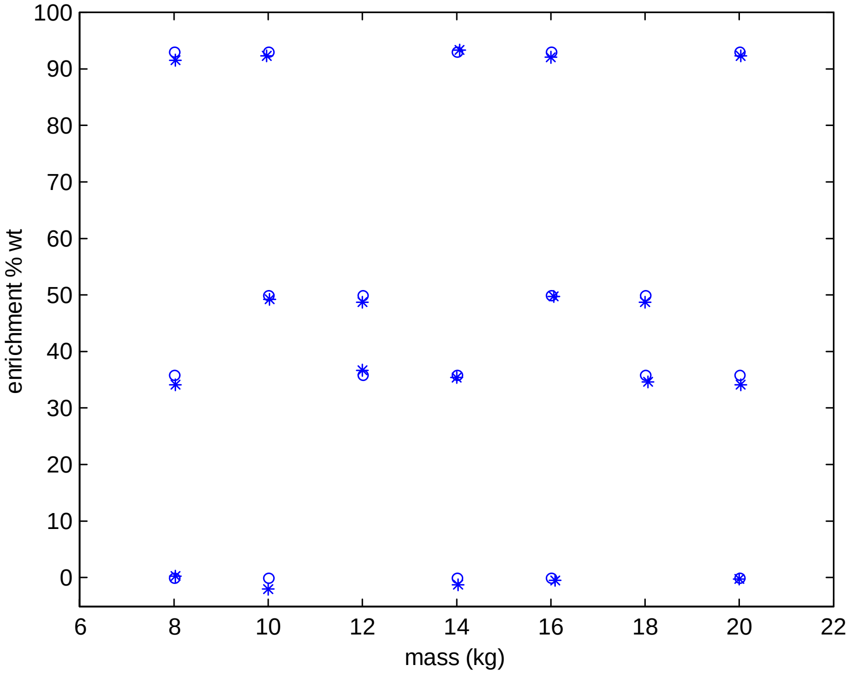


Fig. 4.2. Neutral network prediction of mass and enrichment on the basis of features F_1, F_2, F_3, and F_4: training set of 19 cases relative to sphere simulations. The true values are shown with the circles and the values predicted by the network with stars.

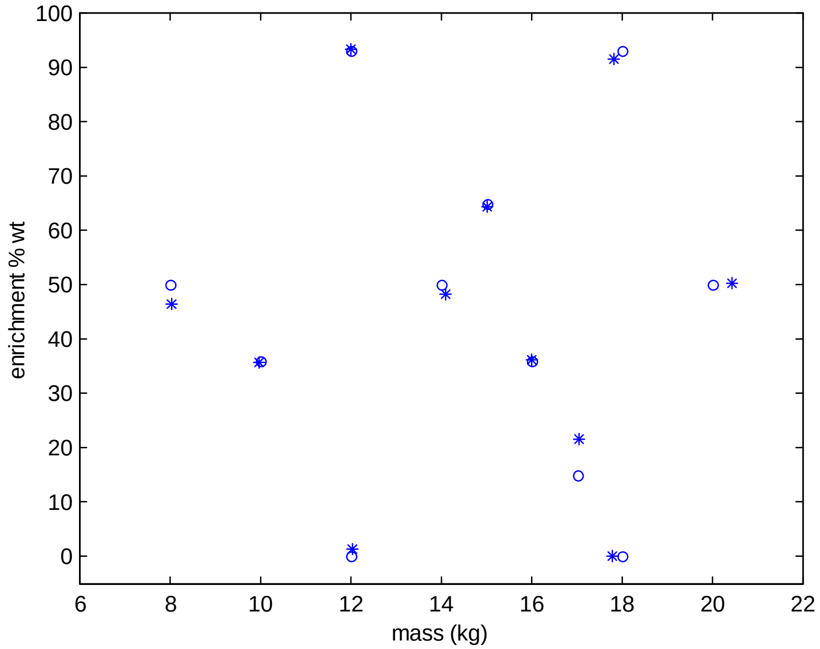


Fig. 4.3. Neural network prediction of mass and enrichment on the basis of features F_1, F_2, F_3, and F_4: test set of 11 cases relative to cylinder simulations. The true values are shown with the circles and the values predicted by the network with stars.

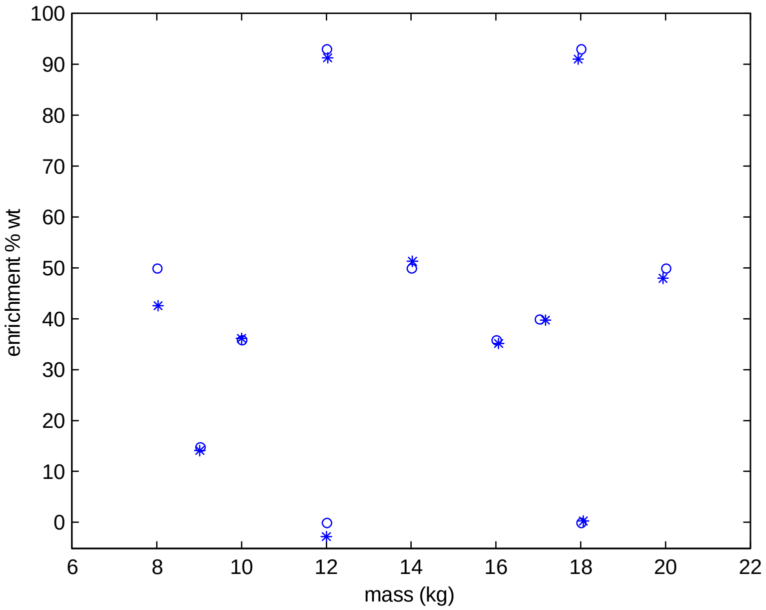


Fig. 4.4. Neural network prediction of mass and enrichment on the basis of the features F_1, F_2, F_3, and F_4: test set of 11 cases relative to sphere simulations. The true values are shown with the circles and the values predicted by the network with stars.

5. APPLICATION OF GENETIC PROGRAMMING TO NUCLEAR MATERIALS IDENTIFICATION SYSTEMS

Genetic Programming (GP) is an evolutionary computation technique that is able to evolve Lisp programs[8, 9] to perform various tasks. A mathematical expression (just a Lisp sentence) can be easily expressed in the form of a parse tree. In Figure 5.1 the mathematical expression, (x/3.5)*(-y) is shown as a parse tree.

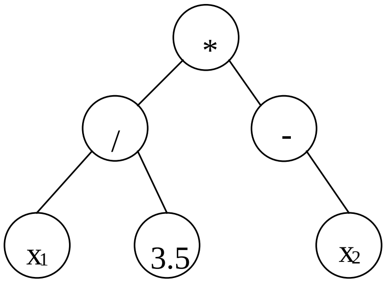


Fig. 5.1. A parse tree

GP technique practicians use it to solve regression problems, i.e., problems in which a data set in the form of pairs

$$\left\{ X_i , Y_i \right\}$$

describe a sampling of the mathematical expression

$$Y = f\ (X)\,.$$

GP is used to obtain an explicit mathematical expression for the function f with the unique information given by the data set.

In the following section we will describe a simple version of the GP algorithm similar to the one used in the present simulations.

5.1 A SIMPLE GENETIC PROGRAMMING ALGORITHM FOR REGRESSION PROBLEMS

The algorithm works with a population of possible solutions described in parse trees as the one shown in Figure 5.1. The algorithm performs a set of operations on the initial population producing a new population which hopefully will include new members which, in average, will outperform the previous population when solving the problem. The process should be repeated many times, and usually a new member is created that totally solves the problem. Before starting with a more precise description of the algorithm, a few concepts should be stated:

Fitness value: it is the degree of performance of a mathematical expression. In this paper a simple error for each parse tree is used over the data set:

$$Error = - \ _{N} \ (y_{i} - parse\_tree\_evaluation(X_{i})), \tag{5.1}$$

where N ranges for all members of the data set. The error should be maximized to solve the problem.

Terminal set: is the name for the set of possible values for the leafs of the parse tree. They are usually the variables of the problem. In Figure 5.1 X_1 and X_2 belong to the terminal set and when evaluating the expression, they are substituted from values taken from the data set. Constants are also terminals, as 3.5 in Figure 5.1. They are created randomly and just once (see algorithm below).

Function set: is the name of all possible functions and operators that may be used in the internal nodes of the trees. In Figure 5.1 *,/, and (-), were used.

Protected division: a division by 0 may crash the algorithm when evaluating an expression. Instead of using a standard division, GP practitioners use what is called a protected division, a division operator that when dealing with 0 denominators returns a constant. In our experiments a return value of 1 was used.

Closure of the function and terminal sets: a generalization of the concept of protected division is the concept of closure, which means that each function of the Function set should be able to handle gracefully all values it might receive as input.

Genetic operations: are operations made on the trees in order to create new trees. Typical operations are Reproduction, Mutation and Crossover described below.

Reproduction operator: a new tree is created just making a copy of an existing tree. The selection of the parent tree is made at random proportionally to the fitness values of the existing trees, giving more opportunities to reproduce to the trees with best performances. The important fact here is that it does not matter how bad a tree is performing because a chance to reproduce is given to it (this is how natural evolution works, isn't it?).

Mutation: a new tree is created just selecting random nodes and substituting their values for new ones. Mutation operators may have different versions. They may imply the substitution of a node with a node, a leaf with a leaf, a leaf with a branch, or a constant by a new constant generated around the original one. This last version is used to fine tune solutions where not other operator gets better results, and it is needed just a small modification in the constants of the mathematical expression. In Figure 5.2 it is shown an example of the four versions having as the parent tree the one shown in Figure 5.1.

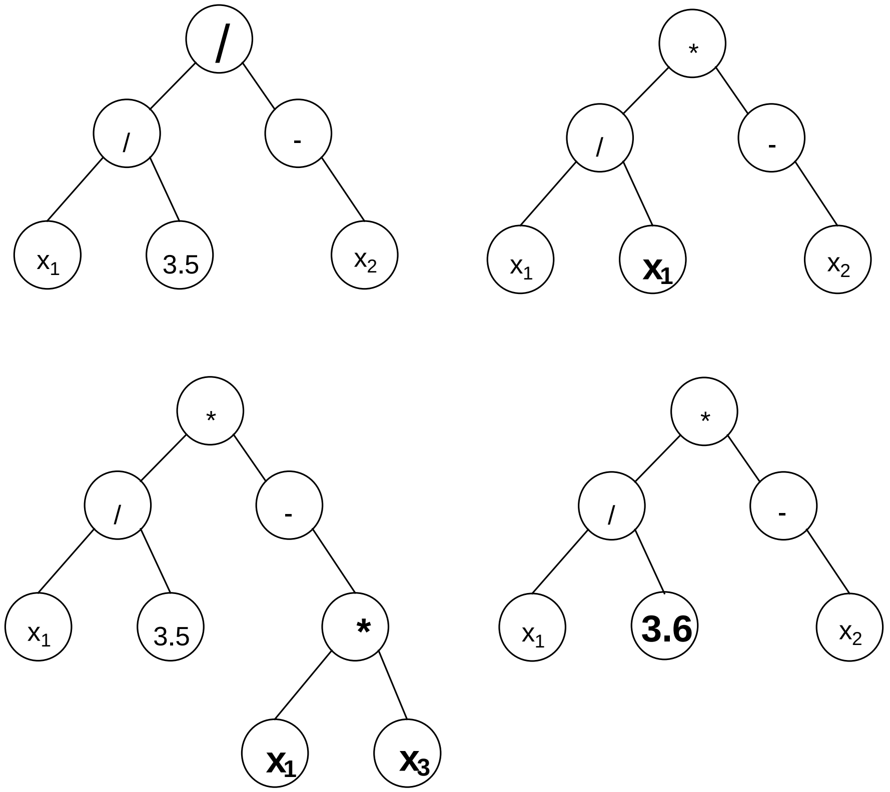


Fig. 5.2. Four mutations over the tree of Figure 5.1.

Crossover: it takes two trees and choosing a random crossover point in each tree and exchanging the subtrees beneath those points produces two offspring. Figure 5.3 illustrates the operator.

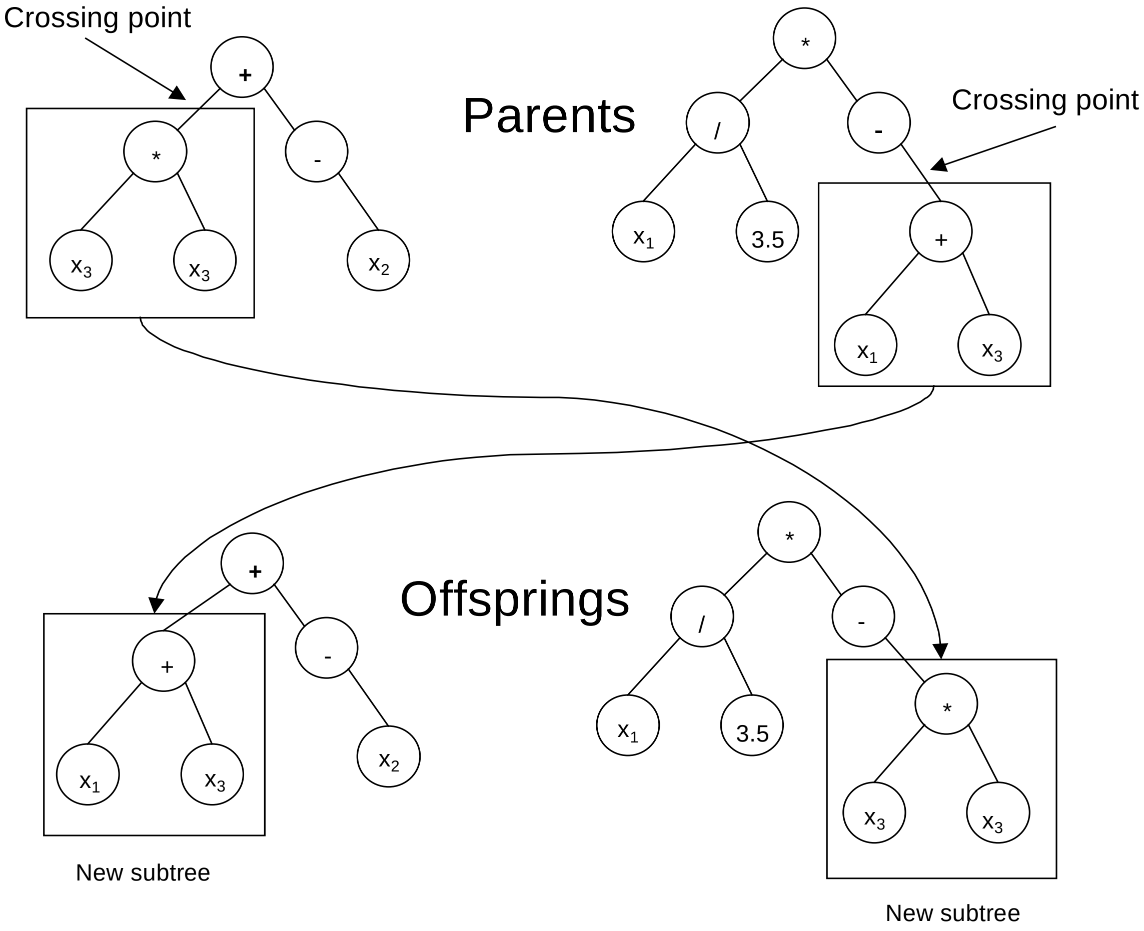


Fig. 5.3. Crossover operation.

Training and test sets: the data from the experiments should be split in two sets. One, the training set, is used to generate solutions evaluating the fitness values with it, and the second, the test set, is used to evaluate the performance of the final solution.

The algorithm comprises eight steps (this version is called steady state[11]).

1. Initialize the population. A population of mathematical expressions, parse trees, is created from scratch at random. Several important decisions should now be made:

   a. Choose size of operation.
   b. Choose maximum depth of the trees (maximum size of the mathematical expressions)
   c. Choose values for the function and terminal sets. In general, one does not know in advance what are the functions and variables needed to form the tree solution, so the technician can just make a guess of them. Also, different probabilities can be

used when selecting a function or terminal, giving more opportunities to those elements that are expected to be needed more frequently in the solution.

When creating new branches from a node in a tree, the number of branches must equal the number of arguments taken by the function or operator in it. For example, if the node contains the operator +, the number of branches must be two, while if the node contains the function sine, the number of branches must be one. In the case that the node is a leaf, no more branches can be added from that point.

Random constants are placed in leafs at random. They are created at this point of the algorithm, and not in any later points.

2. Randomly choose a subset of the population to take part in a tournament. A size of seven trees for a subset is typical.

3. Evaluate the fitness value of each member of the tournament.

4. Select one or two winners, i.e., ones with the best fitness values.

5. Apply genetic operators to the winner or winners of the tournament. Operators are selected at random accordingly to some predefined probabilities.

6. Replace the losers in the tournament with the results of the application of the genetic operators to the winners of the tournament. Alternatively the replacement can be made with bad trees selected in the population, or with old trees, trees that have been being reproduced without changes during many iterations of the algorithm.

7. Repeat steps 2-7 until a termination criterion is met. The desired termination criterion is a very low error in the trees of the population, close to 0, but sometimes if the population is stuck in a local minimum, new iterations can be performed and no improvement is achieved in a long time. Then a restart is needed (go back to 1).

8. Choose the best tree in the population as the solution of the problem.

5.2 GP APPLIED TO THE PREDICTION OF THE SAMPLE TOTAL MASS AND ^{235}U ENRICHMENT

The previous GP algorithm was applied to generate a function mapping the features, F_1, F_2, F_3, and F_4, to the sample mass and enrichment. The parameters of the algorithm were as follows:

Terminal set: F_1, F_2, F_3, and F_4, normalized in [-1, +1], and random constants in (-01., +0.1). A normalization of the output was also done. It is important to note that the normalization was not necessary for the enrichment data, but it was required for the sphere’s to get good results. Similar normalization are done when using other techniques like neural networks. Random constants were selected with a probability of 28%, while each F_i with 9%.

Function set: operators +,-,*,/ (protected), selected with equal probabilities of 9%. The accumulative probability of selection between terminals and functions in a genetic operation is 100%.

Population size: 10,000.

Maximum initial size for trees (step 1 of the algorithm): 4 nodes.

Maximum size for trees after any genetic operator: 50 nodes.

Termination criterion: -0.001 error or 100,000 iterations. It is important to note that the criterion based on the number of iterations, which is the criterion that says that the population of trees cannot be improved in its present configuration and a restart should be done, it is unlikely reached in these experiments.

Tournament size: 7. The replacement of bad elements is done within the entire population.

Probabilities for selecting genetic operators: reproduction 7%, mutation 20%, crossover 73%.

Size of training set: 19 cases.

Size of test set: 11.

The following equations show one example of results obtained for each sample (note that the features and desired outputs were normalized). We got many of them with an acceptable error and this selection was not made under any objective criteria.

CYLINDER:

$$\boxed{\begin{array}{l}\text{Mass} = (((((F1 - -0{,}514) * ((F1 - ((((F1 * F1) - F1) + -0{,}876) / (-0{,}502 + -0{,}584))) * -0{,}58)) * \\ -0{,}582) * -0{,}574) - (((-0{,}876 - F1) + ((F1 * ((((F1 * F1) - F1) + -0{,}882) / (-0{,}49 + -0{,}544))) * \\ (F1 * F1))) / (-0{,}498 + -0{,}59)))\end{array}} \tag{5.2}$$

$$\boxed{\begin{array}{l}\text{Enrichment} = (((F4 - (-0{,}014 * (F4 - (-0{,}014 / F4)))) - (((F2 * 0{,}748) * ((F3 * 0{,}366) + (F4 - \\ (-0{,}014 / ((F1 + ((F4 - (-0{,}014 / (F2 - (F4 * F3)))) - (-0{,}014 * F4))) - (F4 * (F3 / 0{,}748))))))) * \\ F3)) + F2)\end{array}} \tag{5.3}$$

SPHERE:

$$\boxed{\begin{array}{l}\text{Mass} = ((-0{,}204 * ((F2 * (((-0{,}49 * ((F4 - -0{,}722) - (F2 * ((-0{,}49 / -0{,}88) + F2)))) * (-0{,}45 / \\ ((((F3 * F2) * ((F2 * F2) - F2)) * ((F3 * -0{,}722) - F2)) - F2))) - F2)) - F3)) - F2)\end{array}} \tag{5.4}$$

$$\boxed{\begin{array}{l}\text{Enrichment} = ((((-0,5 + (F4 + (F2 * (F4 - F3)))) * F2) * (F4 * -0,742)) - ((F4 * -0,49) + \\ ((((F4 + F2) - (-0,496 + ((F4 - ((((F4 - F3) - -0,81) * ((F4 - F3) * (F4 + F2))) * F4)) * -0,49))) \\ + F2) * -0,54)))\end{array}} \quad (5.5)$$

The constants that are found in each equation were created at the initial population and passed from trees to trees in each iteration of the algorithm through the application of the genetic operators. Some of the major achievements of the GP when solving a regression problem is how it manages to combine constants to create new ones[8].

The equations may be simplified, but we didn't do it. When simplifying an equation, it is easy to find introns which is the name for parts of the trees that do not affect the performance of the tree[11]. Examples of introns could be (F_1/1) or (F_2+0).

After inspection of the above equations, it can be found that:

a). Mass for the cylinder is calculated using just F1. This simplification can be found in most of the solutions. The explanation of this result is that the cylinder intercepts most of the source photons, thus, the area under the photo peak, normalized to the same area for the void run, is a measure of the photon attenuation that depends on the sample mass. Because for cylinders, the F_1 feature depends so strongly on the mass of the sample, the GP program selected just F_1 among the four inputted features.

b). Mass for the sphere is calculated using just F_2 and F_3. We have found this kind of simple dependence in many other solutions. The spherical sample intercepts less photons than in the case of the cylindrical sample, thus, F_1 does not contain too much sample mass information. The F_2 feature, on the contrary, contains the mass information from the photo peak (F_1), plus the sensitivity to sample mass imprinted in the left side of the second peak. The third feature, F_3, is also selected by the program for its high sensitivity to sample mass at high sample enrichments.

For both spherical and cylindrical samples, the equations constructed by the GP algorithm to determine the sample enrichment depend only on the F_2, F_3, and F_4 features because of the F_1 feature exclusive dependence on sample mass.

Figs. 5.4 – 5.7 show the results from the application of equations (5.2) up to (5.5) to cylindrical and spherical samples, respectively.

For both samples the mass prediction is very accurate in and out of the training set. The sample enrichment prediction is within a 4% band for enrichments above 15%. The error increases as the enrichment decreases towards the depleted case. A significant property of the (GP) algorithms is its capability to select and then combine in an explicit fashion a given set of features for optimal attribute determination.

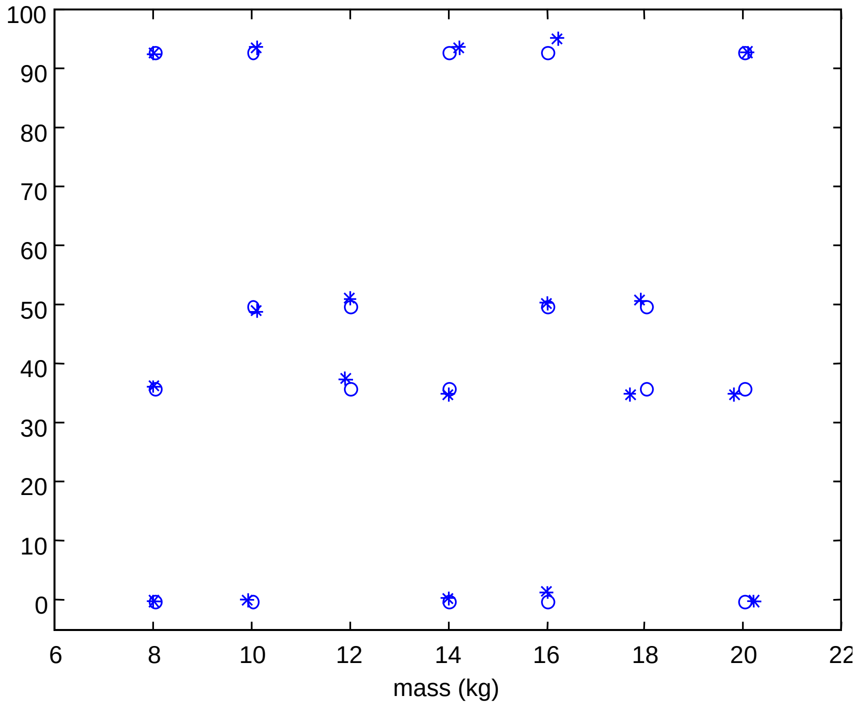


Fig. 5.4. Genetic Programming: prediction of mass and enrichment on the basis of features F_1, F_2, F_3 and F_4: training set of 19 cases relative to cylinder simulations. The true values are shown with the circles and the values predicted by the algorithm with stars.

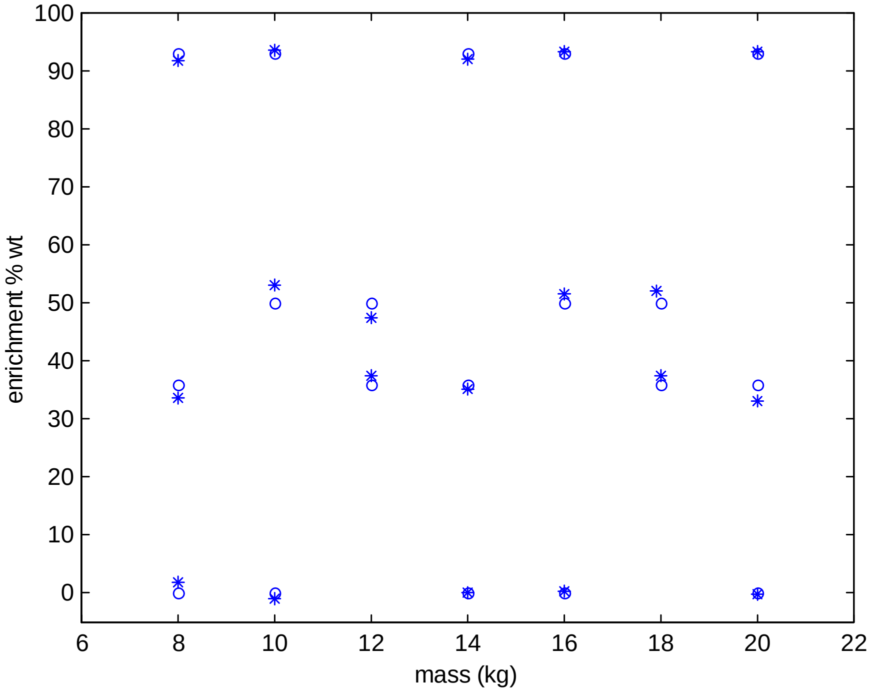


Fig. 5.5. Genetic Programming: prediction of mass and enrichment on the basis of features F_1, F_2, F_3 and F_4: training set of 19 cases relative to sphere simulations. The true values are shown with the circles and the values predicted by the algorithm with stars.

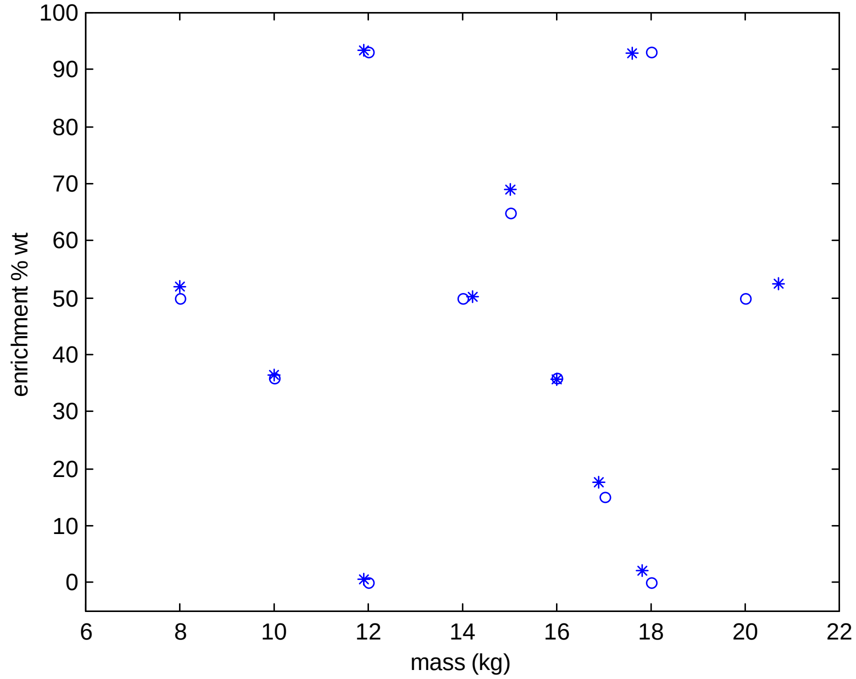


Fig. 5.6. Genetic Programming: prediction of mass and enrichment on the basis of features F_1, F_2, F_3 and F_4: set of 11 cases used for testing relative to cylinder simulations. The true values are shown with the circles and the values predicted by the algorithm with stars.

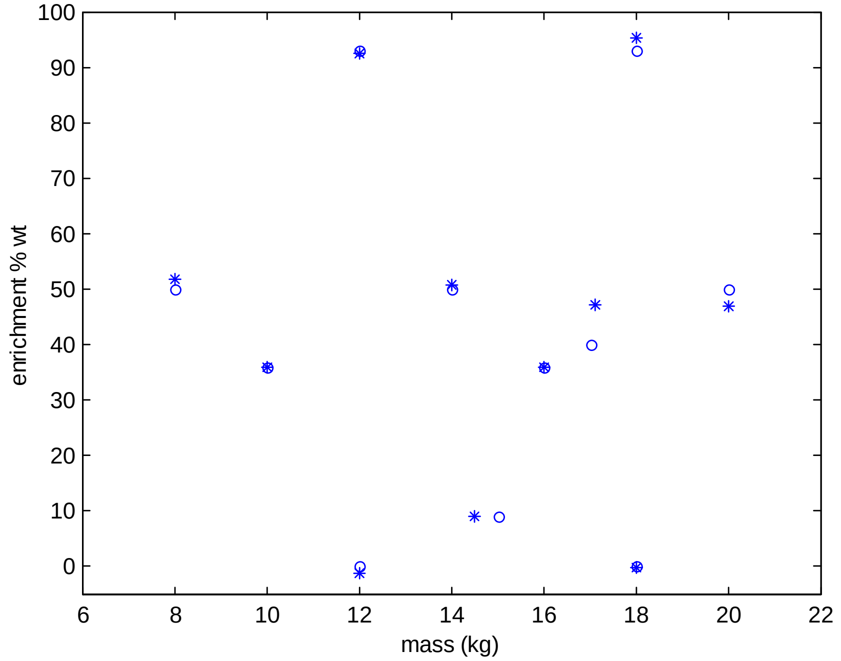


Fig.5.7. Genetic Programming: prediction of mass and enrichment on the basis of features F_1, F_2, F_3 and F_4: set of 11 cases used for testing relative to sphere simulations. The true values are shown with the circles and the values predicted by the network with stars.

This page intentionally left blank.

6. COMPARISON OF RESULTS

In order to make a more meaningful comparison of the results we applied a standard regression to predict the mass and enrichment of the cylindrical and spherical samples. Tables 6.1 and 6.2 summarize the error for the three techniques for both training and test cases. The error measure used is:

$$Error = \frac{\sum_i \left| real_i - predicted_i \right|}{\sum_i real_i}$$

Table 6.1: Error results for the cylindrical samples

Cylinder	PREDICTED BY GP		PREDICTED BY NN		PREDICTED BY Regression	
	MASS	ENRICH	MASS	ENRICH	MASS	ENRICH
Training	0.71%	1.67%	0.22%	2.07%	3.05%	14.81%
Test	1.34%	2.16%	0.81%	2.14%	2.69%	12.68%
Extra	0.45%	8.18%	0.13%	9.05%	1.87%	10.52%

Table 6.2: Error results for the spherical samples

Sphere	PREDICTED BY GP		PREDICTED BY NN		PREDICTED BY Regressio	
	MASS	ENRICH	MASS	ENRICH	MASS	ENRICH
Training	0.17%	2.95%	0.27%	2.20%	1.95%	21.37%
Test	0.15%	2.38%	0.27%	4.59%	2.18%	14.71%
Extra	0.38%	14.00%	0.72%	13.33%	1.08%	43.40%

The tables show that NN and GP are comparable and more effective than regression in solving the prediction problem. NN and GP are capable of dealing with non-linear problems and this is demonstrated in the case of the enrichment for both configurations, cylinder and sphere, in which the linear solution, the regression, performs very poorly, indicating that the problem is strongly non-linear.

We have found non remarkable differences in the performance between the artificial intelligence techniques. Two cases (rows labeled as ‘Extra’ samples in the Tables 6.1 and 6.2, and bottom two rows in Tables 6.3 and 6.4) of the test sets had enrichment values selected outside of the training set range. These can be used to test the overfitting of the models. The error in predicting these enrichment values range from 8% to 14%, indicating that some overfit has taken place. This can be explained by considering that there were only four values of enrichment in the training set, covering a wide range of enrichment: from depleted to highly enriched uranium. Better results can be obtained by adding more cases to the training set.

Tables 6.3 and 6.4 show the prediction results and error for both AI techniques (GP and NN) compared to the linear regression for cylinders and spheres, respectively.

Table 6.3: Results for uranium cylinders: NN, GP, and regression predictions. Training cases are shown in gray, test cases in white.

REAL		PREDICTED BY GP		ERROR		PREDICTED BY NN		ERROR		PREDICTED BY Regre		ERROR	
MASS	ENRICH	MASS	ENRICH	MASS	ENRICH	MASS	ENRICH	MASS	ENRICH	MASS	ENRICH	MASS	ENRICH
8.00	0.20	8.0	-0.03	0.02	0.23	8.0	-0.09	0.00	0.29	8.6	14.23	0.6	14.03
10.00	0.20	9.9	0.23	0.08	0.03	10.0	0.88	0.01	0.68	10.5	14.36	0.5	14.16
12.00	0.20	11.9	0.63	0.12	0.43	12.0	1.30	0.04	1.10	12.5	7.23	0.5	7.03
14.00	0.20	14.0	0.47	0.01	0.27	14.0	-0.25	0.03	0.45	14.1	-0.96	0.1	1.16
16.00	0.20	16.0	1.48	0.04	1.28	16.0	-0.06	0.02	0.26	15.8	-8.05	0.2	8.25
18.00	0.20	17.8	2.18	0.24	1.98	17.8	0.20	0.20	0.00	17.1	-12.70	0.9	12.90
20.00	0.20	20.2	-0.18	0.17	0.38	20.0	0.57	0.00	0.37	18.6	-19.80	1.4	20.00
8.00	36.00	8.0	36.20	0.01	0.20	8.0	36.79	0.01	0.79	8.2	37.40	0.2	1.40
10.00	36.00	10.0	36.56	0.02	0.56	10.0	35.82	0.03	0.18	10.2	42.57	0.2	6.57
12.00	36.00	11.9	37.54	0.13	1.54	12.0	37.25	0.02	1.25	12.6	42.76	0.6	6.76
14.00	36.00	14.0	34.85	0.02	1.15	14.0	35.12	0.04	0.88	14.4	40.42	0.4	4.42
16.00	36.00	16.0	35.58	0.01	0.42	16.0	36.28	0.00	0.28	16.4	40.25	0.4	4.25
18.00	36.00	17.7	34.90	0.34	1.10	17.9	36.43	0.13	0.43	18.1	38.77	0.1	2.77
20.00	36.00	19.8	35.01	0.17	0.99	20.1	39.43	0.06	3.43	19.8	41.67	0.2	5.67
8.00	50.00	8.0	51.94	0.00	1.94	8.0	46.45	0.02	3.55	7.9	43.54	0.1	6.46
10.00	50.00	10.1	49.06	0.05	0.94	10.0	49.50	0.04	0.50	10.0	52.34	0.0	2.34
12.00	50.00	12.0	51.12	0.01	1.12	12.0	50.52	0.01	0.52	12.5	55.34	0.5	5.34
14.00	50.00	14.2	50.28	0.18	0.28	14.1	48.35	0.11	1.65	14.5	54.75	0.5	4.75
16.00	50.00	16.0	50.23	0.02	0.23	16.1	47.82	0.08	2.18	16.6	53.02	0.6	3.02
18.00	50.00	17.9	50.89	0.10	0.89	18.0	47.63	0.03	2.37	18.4	52.20	0.4	2.20
20.00	50.00	20.7	52.47	0.67	2.47	20.4	50.22	0.43	0.22	20.4	54.75	0.4	4.75
8.00	93.15	8.0	92.45	0.02	0.70	8.0	92.30	0.02	0.85	7.0	69.83	1.0	23.32
10.00	93.15	10.1	93.53	0.08	0.38	10.0	94.32	0.03	1.17	9.1	84.38	0.9	8.77
12.00	93.15	11.9	93.51	0.10	0.36	12.0	93.23	0.01	0.07	11.6	89.51	0.4	3.64
14.00	93.15	14.2	93.71	0.22	0.56	14.0	92.71	0.02	0.45	13.8	92.79	0.2	0.36
16.00	93.15	16.2	95.09	0.25	1.94	16.0	93.61	0.03	0.46	16.0	93.74	0.0	0.59
18.00	93.15	17.6	92.77	0.38	0.38	17.8	91.47	0.20	1.68	17.9	91.69	0.1	1.46
20.00	93.15	20.1	92.94	0.15	0.21	20.0	93.33	0.00	0.18	20.0	92.32	0.0	0.83
15.00	65.00	15.0	68.93	0.00	3.93	15.0	64.51	0.01	0.49	15.6	69.32	0.6	4.32
17.00	15.00	16.9	17.61	0.14	2.61	17.0	21.75	0.03	6.75	17.0	19.09	0.0	4.09

Table 6.4: Results for uranium spheres: NN, GP, and regression predictions. Training cases are shown in gray, test cases in white.

REAL		PREDICTED BY GP		ERROR		PREDICTED BY NN		ERROR		PREDICTED BY Regre		ERROR	
MASS	ENRICH	MASS	ENRICH	MASS	ENRICH	MASS	ENRICH	MASS	ENRICH	MASS	ENRICH	MASS	ENRICH
8,00	0,20	8,0	2,04	0,00	1,84	8,0	0,49	0,02	0,29	7,8	31,08	0,25	30,88
10,00	0,20	10,0	-0,85	0,03	1,05	10,0	-1,90	0,01	2,10	10,3	18,43	0,28	18,23
12,00	0,20	12,0	-1,09	0,04	1,29	12,0	-2,76	0,00	2,96	12,5	7,86	0,47	7,66
14,00	0,20	14,0	0,05	0,04	0,15	14,0	-1,24	0,02	1,44	14,4	-0,31	0,43	0,51
16,00	0,20	16,0	0,40	0,00	0,20	16,1	-0,32	0,10	0,52	16,2	-7,30	0,25	7,50
18,00	0,20	18,0	-0,01	0,02	0,21	18,1	0,50	0,07	0,30	17,9	-13,73	0,14	13,93
20,00	0,20	20,0	-0,19	0,04	0,39	20,0	-0,10	0,01	0,30	19,4	-21,25	0,65	21,45
8,00	36,00	8,0	33,77	0,02	2,23	8,0	34,14	0,04	1,86	7,7	41,67	0,29	5,67
10,00	36,00	10,0	35,95	0,02	0,05	10,0	36,14	0,00	0,14	10,2	35,97	0,24	0,03
12,00	36,00	12,0	37,54	0,03	1,54	12,0	36,82	0,02	0,82	12,4	34,22	0,45	1,78
14,00	36,00	14,0	35,15	0,01	0,85	14,0	35,38	0,04	0,62	14,4	33,19	0,41	2,81
16,00	36,00	16,0	35,99	0,04	0,01	16,1	35,10	0,07	0,90	16,3	36,76	0,30	0,76
18,00	36,00	18,0	37,44	0,02	1,44	17,9	34,82	0,13	1,19	18,0	43,12	0,03	7,12
20,00	36,00	20,0	33,15	0,02	2,85	20,1	34,19	0,06	1,82	19,6	49,50	0,40	13,50
8,00	50,00	8,0	51,73	0,01	1,73	8,0	42,75	0,04	7,25	7,3	53,82	0,68	3,82
10,00	50,00	10,0	53,13	0,00	3,13	10,0	49,36	0,04	0,64	9,9	49,56	0,10	0,44
12,00	50,00	12,0	47,52	0,02	2,48	12	48,79	0,00	1,21	12	45,09	0,14	4,91
14,00	50,00	14,0	50,71	0,00	0,71	14,0	51,36	0,02	1,36	14,2	48,61	0,18	1,39
16,00	50,00	16,0	51,70	0,01	1,70	16	49,85	0,05	0,15	16	52,98	0,10	2,98
18,00	50,00	17,9	52,04	0,05	2,04	18	48,71	0,01	1,29	18	59,77	0,17	9,77
20,00	50,00	20,0	47,02	0,03	2,98	19,9	48,14	0,07	1,87	19,5	67,54	0,48	17,54
8,00	93,15	8,0	91,80	0,01	1,35	8	91,65	0,04	1,50	7	66,48	0,54	26,67
10,00	93,15	10,0	93,57	0,05	0,42	10	92,28	0,03	0,87	10	71,82	0,04	21,33
12,00	93,15	12,0	92,58	0,01	0,57	12,0	91,28	0,02	1,87	12,2	80,42	0,25	12,73
14,00	93,15	14,0	92,14	0,03	1,01	14	93,23	0,05	0,08	14	90,07	0,30	3,08
16,00	93,15	16,0	93,33	0,02	0,18	16	91,97	0,01	1,18	16	94,54	0,22	1,39
18,00	93,15	18,0	95,31	0,02	2,16	17,9	91,02	0,06	2,14	18,1	95,40	0,05	2,25
20,00	93,15	20,0	93,27	0,03	0,12	20	92,40	0,03	0,75	20	94,09	0,12	0,94
17,0	40,00	17,1	47,20	0,10	7,20	17,2	46,64	0,18	6,64	17,2	48,94	0,23	8,94
9,0	15,00	9,0	14,50	0,00	0,50	9,0	14,31	0,01	0,69	9,0	29,93	0,05	14,93

This page intentionally left blank.

7. SUMMARY AND CONCLUSIONS

Monte Carlo simulations of the source-detector cross correlation function for various sample shapes, mass, and enrichment values have been performed to serve as a training set for two artificial intelligence algorithms (AI): neural networks (NN) and genetic programming (GP). The input presented to the AI algorithm has been in the form of features extracted from the physical properties of the cross-correlation functions related to mass (beam attenuation) and to enrichment (fission induced pulse broadening). Both the NN and GP algorithms have shown good capabilities and robustness for mass and enrichment predictions of uranium metal samples.

These results serve as a proof of principle for the application of combined stochastic and AI methods to safeguards procedures.

This page intentionally left blank.

8. REFERENCES

1. J. T. Mihalczo, T. E. Valentine, J. A. Mullens, and J. K. Mattingly, *Physical and Mathematical Description of Nuclear Weapons Identification System (NWIS) Signatures,* Report Number Y/LB-15,946, R3, September 26, 1997.

2. M. S. Krick, D. G. Langner, et. al., *Nuclear Material Management* (Proc. Issue) XXI, 779 (1992).

3. T. E. Valentine, "MCNP-DSP Users Manual," ORNL/TM-13334, Oak Ridge National Laboratory, January 1997.

4. D. E. Rumelhard and J. L. McClelland, Parallel Distributed Processing, Vol. 1, MIT Press, Cambridge, MA (1986).

5. M. Marseguerra, S. Minoggio, A. Rossi, and E. Zio, "Artificial Neural Networks Applied to Multiple Signals in Nuclear Technology," *Progress in Nuclear Energy,* Vol. 27, **4** (1992).

6. R. E. Urig, "Potential Application of Neural Networks to the Operation of Nuclear Power Plants," *Nuclear Safety,* Vol. 32, **1** (1991).

7. L. Chambers, *Practical Handbook of Genetic Algorithms – Applications,* CRC Press (1995).

8. J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection,* MIT Press (1992).

9. J. R. Koza, *Genetic Programming II: Automatic Discovery of Reusable Programs,* MIT Press (1994).

10. P. J. Angeline, *Genetic Programming and Emergent Intelligence,* K. E. Kinnear, Jr., Editor, Advances in Genetic Programming, MIT Press (1994).

11. W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, *Genetic Programming: An Introduction* (1998).

This page intentionally left blank.

DISTRIBUTION

Internal
L. G. Chiang
J. A. March-Leuba
J. K. Mattingly
J. T. Mihalczo
J. A. Mullens
R. B. Oberer
R. B. Perez
T. E. Valentine
J. D. White
M. C. Wright

External
J. L. Munoz-Cobos, Polytechnic University of Valencia