Point Kernel Modification Including Support Vector Regression Neutron Buildup Factor Models

Paulina Dučkić, Krešimir Trontl, Davor Grgić, Mario Matijević

University of Zagreb, Faculty of Electrical Engineering and Computing

Unska 3, 10000 Zagreb, Croatia

paulina.duckic@fer.hr, kresimir.trontl@fer.hr, davor.grgic@fer.hr, mario.matijevic@fer.hr

ABSTRACT

This work presents the results of radiation shielding calculations using modified point kernel code QAD-CGGP. The modification includes a new approach to neutron buildup factor estimations based on machine learning technique called Support vector regression (SVR). SVR neutron buildup factor models for common shielding materials are developed and built into the QAD-CGGP. The development of the models consisted of acquiring the data to be used for learning the model, optimizing the SVR parameters, and application of active learning methods for improving the learning process. The modified code is tested, and the results are compared with the MCNP6 results.

Keywords: point kernel, neutron buildup factor, support vector regression, shielding

1   INTRODUCTION

Point kernel method is an approximate approach to radiation shielding analysis, involving the calculation of the direct component of radiation and a buildup factor to correct the direct component for the secondary radiation produced within the shield. In general, the direct component of radiation from an extended source at a detector point is obtained by assuming that the extended source consists of many point isotropic sources and that the resulting radiation is obtained by summing the contributions of the individual point sources. If the source is a plane emitting parallel beam of monoenergetic neutrons, the direct component is calculated using an exponential law:

$$\phi(x) = \phi_0 e^{-\Sigma x} \tag{1}$$

where ϕ is the total flux at a distance x from the source, ϕ_0 is the incident flux, and Σ is the macroscopic cross section. The point kernel method is widely used in gamma ray shielding analysis, but for neutrons somewhat different approach based on the removal cross section has been used [1]. This is due to variety of interactions neutrons may undergo with the shielding media, as well as a strongly non-linear dependence of the total macroscopic cross section on the incident neutron energy. It has been shown that the behaviour of the buildup factors is led by the macroscopic cross sections [2], thus the non-linearity of the macroscopic cross sections is transferred to the buildup factors making the use of the point kernel method in neutron shielding analysis impractical. This problem may be overcome because with the development of machine learning techniques, the determination of the neutron buildup factors can be simplified.

Support Vector Regression (SVR) is a machine learning technique that has been used in different fields of study, including medicine, pharmacy, engineering, etc. [3][4][5][6][7][8]. Its main characteristics are a) learning from data, b) structural risk minimization, c) convex optimization problem d) partial solution e) mapping in the features space. In the simplest form, the model developed by the SVR is a linear function, also called the target function (Eqn. 2), which is based only on the so called training set involving input vectors and corresponding output values:

$$D: \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_n, y_n)\} \subset R^n \times R.$$

$$f(\vec{x}) = \langle \vec{w}, \vec{x} \rangle + b \quad \vec{w} \in R^N, b \in R \tag{2}$$

The target function is obtained by minimizing structural risk composed of prediction error and model complexity (Eqn. 3)

$$\min \quad \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{l}(\xi_i + \xi_i^*)$$

$$\text{with respect to} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{3}$$

$$i = 1, 2, \dots, l$$

Real life problems are usually non-linear, therefore, SVR solves non-linear problems by mapping the input vector in a higher dimensional features space. There are many mapping functions, also called the kernel functions, that can serve this purpose (Eqn. 4 – Eqn. 7). The commonly used kernel function is Gaussian Radial Basis Function (Eqn. 4).

$$K(\vec{x}_i, \vec{x}) = e^{-\frac{\|\vec{x}_i - \vec{x}\|^2}{2\sigma^2}} \tag{4}$$

$$K(\vec{x}_i, \vec{x}) = ((\vec{x}_i \cdot \vec{x}) + \theta)^d \tag{5}$$

$$K(\vec{x}_i, \vec{x}) = \tanh(\kappa(\vec{x}_i \cdot \vec{x}) + \theta) \tag{6}$$

$$K(\vec{x}_i, \vec{x}) = \frac{1}{\sqrt{\|\vec{x}_i - \vec{x}\|^2 + c^2}} \tag{7}$$

Using a kernel function to transfer an input vector in a features space, an SVR model for non-linear problems is also a liner function, but in a features space (Eqn. 8).

$$f(\vec{x}) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)K(\vec{x}_i, \vec{x}) + b \tag{8}$$

Support vectors (SV) are those input vectors for which α_i and α_i^* are non-zero. That is why it is said that the solution is partial [9]. The number of SVs strongly depends on the selection of SVR parameters C, ε, and σ. Constant C determines the balance between the complexity of the function and prediction error. The larger the constant C, the model is more accurate in predicting the labels of the data that have been included in the learning process and the target function is much complex (more SVs form the solution). However, this may lead to an overfitting effect and poor prediction of the outputs of the data that have not been included in the learning process. Constant ε is a loss function parameter which represents the radius of the tube surrounding the target function. Lower ε value results in a lager number of SV forming the target function. On the other hand, lower C and higher ε values may result in an underfitting effect which also results in poor prediction. Parameter σ is the width of the Gaussian kernel function. Thus, the selection of SVR parameters affects the accuracy of the model and users must approach to this problem very carefully [10]. Optimization techniques are very popular for selecting SVR parameters due to their efficiency [11][12][13][14].

Active learning is a subfield of machine learning developed with the aim to speed up the learning process by selecting the training data carefully. This is very desirable in situations when obtaining the training data is time, resource and finance consuming. The goal of active learning is to develop an accurate SVR model with as low as possible training data. Different active learning algorithms have been developed and adopted for developing an SVR models [15] and it has been shown that such models achieve good accuracy in predicting labels as it would have been achieved with having so called full training (including all data).

In this work a point kernel code QAD-CGGP [16] is modified by including SVR models for total neutron buildup factors determination. The models are developed for materials usually encountered in radiation shielding problems. The training data for model development have been calculated using MCNP6 [17] and different optimization techniques have been investigated for selecting the SVR parameters, as well as different active learning algorithms have been considered to improve the learning process. The modified QAD-CGGP is tested and the obtained results have been compared with the ones obtained using MCNP6.

The rest of the paper is organized as follows. In Section 2 total neutron buildup factor modelling details are provided. Section 3 presents the obtained SVR models. In Section 4 QAD-CGGP is explained. In Section 5 final testing is made. Conclusions are given in Section 6.

2 TOTAL NEUTRON BUILDUP FACTOR MODELLING

2.1 Total neutron buildup factor

Total neutron buildup factor is defined as the ratio of the total dose (neutron and secondary gamma rays) and the direct neutron dose (Eqn. 9) [2]:

$$B = \frac{D_{total}}{D_{n, direct}} = \frac{D_n + D_g}{D_0 e^{-\Sigma_T z}} \tag{9}$$

where D_n is the neutron dose, D_g is the secondary gamma ray dose, D_0 is the incident dose, Σ_T is total macroscopic cross section, and z is the shielding thickness. In this work total neutron buildup factors are defined in terms of ambient dose equivalent with the corresponding conversion coefficients taken from ICRP Publication 74 [18]. The process of calculating buildup factors is two-step. Firstly, neutron and secondary gamma ray ambient dose equivalent is calculated using MCNP6 and in the second step direct component is calculated manually using an exponential law with the macroscopic cross section calculated using MCNP6.

The modelled geometry includes a plane source emitting parallel beam of monoenergetic neutrons. The incident flux of 1 n/cm²/s is normally directed to the slab of finite thickness and infinite width and height. MCNP calculations are performed using cross section from ENDF/B-VII.1 library, as well as $S(\alpha, \beta)$ to account for thermal scattering. The number of particles run is one million and weight window generator is used to obtain results with acceptable statistics.

2.2 Training data

To learn an SVR model, a training set that describes the problem of interest well is required. In the case of total neutron buildup factor modelling, three features are selected to form an input vector, namely, shielding thickness z, incident neutron energy E, and total macroscopic cross section Σ_T (Eqn. 10), while output vector is the total ambient dose equivalent buildup factor (Eqn. 11).

$$x = \{z, E, \Sigma_T\} \tag{10}$$

$$y = \{B\} \tag{11}$$

Considered shielding thicknesses depend on the material. The calculations are run for seven commonly used shielding materials, namely, aluminum, carbon steel, iron, lead, Portland concrete, stainless steel and water. For each material 40 different shielding thicknesses are being observed. For all materials except concrete, shielding thicknesses vary up to 20 cm and for concrete up to 200 cm. From the incident neutron energy range [0.025 eV – 14 MeV] 30 different energies are considered. Thus, having 40 different shielding thicknesses and 30 different incident neutron energies, for each material, initial set is composed of 1200 samples. These samples are divided in several subsets:
- Training set (90 samples)
- Test set (240 samples)
- Validation set (120 samples)

- Unlabeled set (7500 samples)

Training set samples are selected in a manner to cover the edges and the middle of the shielding thickness domain at all energies. Test set includes uniformly distributed samples over the shielding thickness domain at all energies. Validation set is randomly selected, and the unlabeled set consists of the samples that are to be selected and added to the training set in active learning process. It is important to note that for the initial set of data output values are unknown. After dividing the samples into different subsets, MCNP calculation are run for training, testing and validation samples and in each iteration of active learning, samples are firstly selected from the unlabeled set, and then labeled.

2.3 SVR modelling

For SVR modelling LibSVM (Library Support Vector Machine) [19] is used. LibSVM is a computer code used for classification and regression model development. The two main functions of the code are svm-train used for learning the model, and svm-predict used for predicting the output values for new input vectors based on the developed model. The options of the svm-train and svm-predict functions are shown in Figure 1 and Figure 2, respectively. LibSVM requires train set in .train format and test set in .test format. To obtain a better correlation between the feature of the input vector, it is recommended to scale input vector. In this case, due to a large range of the incident neutron energies, as well as a large range of the buildup factors, the input vector and the output vector are scaled using natural logarithm (ln) function. Therefore, the input vector has the form $(\ln(x), \ln(y))$.

```
D:\libsvm-3.20\windows>svm-train
Usage: svm-train [options] training_set_file [model_file]
options:
-s svm_type : set type of SVM (default 0)
        0 -- C-SVC              (multi-class classification)
        1 -- nu-SVC            (multi-class classification)
        2 -- one-class SVM
        3 -- epsilon-SVR        (regression)
        4 -- nu-SVR            (regression)
-t kernel_type : set type of kernel function (default 2)
        0 -- linear: u'*v
        1 -- polynomial: (gamma*u'*v + coef0)^degree
        2 -- radial basis function: exp(-gamma*|u-v|^2)
        3 -- sigmoid: tanh(gamma*u'*v + coef0)
        4 -- precomputed kernel (kernel values in training_set_file)
-d degree : set degree in kernel function (default 3)
-g gamma : set gamma in kernel function (default 1/num_features)
-r coef0 : set coef0 in kernel function (default 0)
-c cost : set the parameter C of C-SVC, epsilon-SVR, and nu-SVR (default 1)
-n nu : set the parameter nu of nu-SVC, one-class SVM, and nu-SVR (default 0.5)
-p epsilon : set the epsilon in loss function of epsilon-SVR (default 0.1)
-m cachesize : set cache memory size in MB (default 100)
-e epsilon : set tolerance of termination criterion (default 0.001)
-h shrinking : whether to use the shrinking heuristics, 0 or 1 (default 1)
-b probability_estimates : whether to train a SVC or SVR model for probability estimates, 0 or 1 (default 0)
-wi weight : set the parameter C of class i to weight*C, for C-SVC (default 1)
-v n: n-fold cross validation mode
-q : quiet mode (no outputs)
```

Figure 1: Options of the svm-train function

```
D:\libsvm-3.20\windows>svm-predict
Usage: svm-predict [options] test_file model_file output_file
options:
-b probability_estimates: whether to predict probability estimates, 0 or 1 (default 0); for one-class SVM only 0 is supported
-q : quiet mode (no outputs)
```

Figure 2: Options of the svm-predict function

2.4 SVR parameters optimization

Selecting SVR parameters is not a simple task due to their mutual dependency, i.e. all three parameters must be selected at the same time. Lately, optimization algorithms play an important role in selecting SVR parameters due to their performance capabilities and efficiency.

In this work four different optimization techniques are considered and adopted to select SVR parameters for the problem of total neutron buildup factors determination, namely, Differential Evolution (DE), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Sine Cosine Algorithm (SCA). The optimization procedure of each technique is given in the references [13],[14],[20],[21]. The criteria for selecting an optimization technique are the accuracy of the developed models with the optimized SVR parameters and the run time of the optimization algorithm.

The performance of the SVR models combined with different optimization techniques for SVR parameters selection is given in Figure 3. The accuracy of the models is expressed in terms of MSE (Eqn. 12). In Table 1 times and the accuracy of the models combined with different optimization techniques in predicting output values of the validation and test set are presented. The accuracy is expressed in terms of Relative Average Deviation – RAD (Eqn. 13). From these results it can be seen that SVR model combined with PSO gives the smallest error in predicting output values of the validation set, as well as on the test set. Although less computational time is required for the GA method, an increase of about 65% time for PSO results in about two and a half times better accuracy.

$$MSE = \frac{\sum_{i=1}^{n}(y_i - f_i)^2}{n} \tag{12}$$

$$RAD = \frac{\sum_{i=1}^{n}\frac{|y_i - f_i|}{y_i} \cdot 100\%}{n} \tag{13}$$
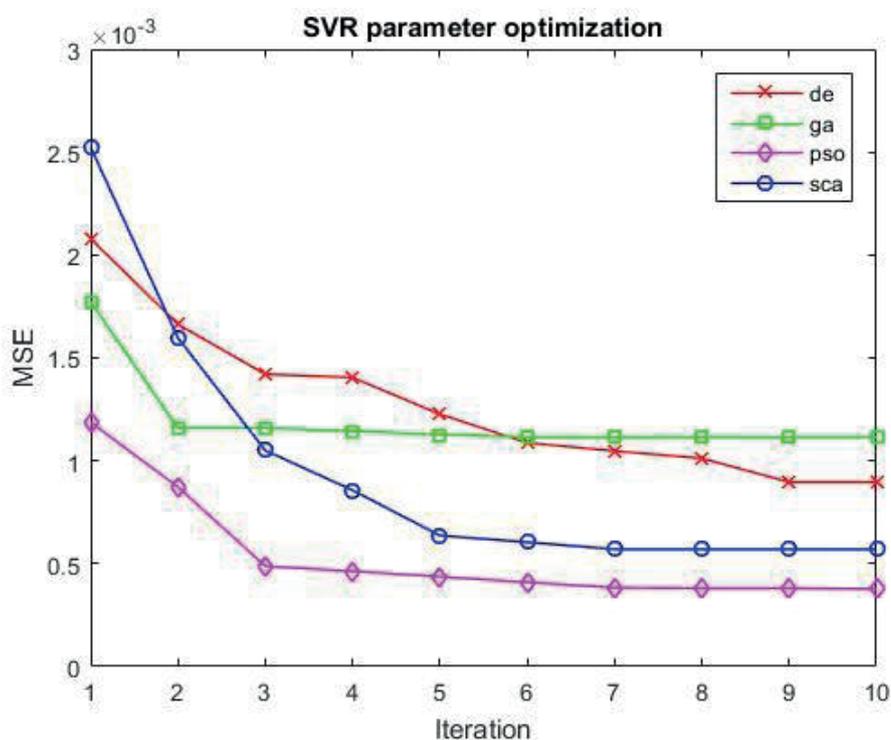

Figure 3: Comparison of the optimization techniques

Table 1: The average computational time of the optimization techniques, validation and testing of the developed models

Method	Run time (s)	RAD (%) validation	RAD (%) testing
DE	270±120	1.08±0.79	1.72±0.72
GA	119±48	1.35±0.81	2.8±1.7
PSO	196±41	0.54±0.20	1.15±0.17
SCA	336±69	0.83±0.42	1.51±0.51

2.5 Active learning algorithm selection

Active learning algorithms are developed with the aim to select the most valuable data that are to be included in the train set. In this way, the model is learned using as least as possible training samples, which in turns results in a reduction of an effort involved in obtaining the training samples (time, computational resources, finances, etc.).

In the case of buildup factor modelling, for each combination of incident neutron energy and shielding thickness, MCNP6 is engaged for obtaining the total ambient dose equivalent. Thicker shields require more computational time. Thus, it is presumed that involving active learning in the SVR model development could save computational time.

Three different active learning algorithms are investigated, namely, Differential Evolution Active Learning (DEAL), K-Nearest Neighbor (KNN), and Query By Committee (QBC). Active learning algorithm based on DE is derived from DE optimization algorithm [20] with a following modification. A new input sample is formed by combining two support vectors and then the nearest sample from the unlabeled set is selected for labelling. That input vector and the corresponding output value is added to the training set. In the KNN method [22], around the sample for which the model had the largest prediction error, K nearest samples are selected for labelling and those samples and the corresponding labels are added to the training set. QBC method [23] is based on developing two or more models. The models are then used for prediction of the labels of each sample in the unlabeled set. Those samples for which the models give the largest disagreement in predicted label are selected for labeling and adding to the training set.

The initial conditions are the same for every method. The size of the initial training set is 9 samples and in each iteration 10 most informative samples are selected and added to the training set. The stopping criterion is either maximal number of iteration (in this case 75) or the accuracy of the model prediction RAD=5%.

The results of the methods are given in Table 2 and compared with passive learning and full training results. From these results it can be seen that KNN method has the least computation time but has unacceptable prediction error on the test. DE method has the best prediction result but has the larger computation time. QBC method is a compromise between these two, with the reduction of time by a factor of about two, the prediction error increases about 23%, but it is below the allowed limit. Therefore, QBC is selected as an active learning algorithm suitable for total ambient dose equivalent buildup factor modelling. Compared to passive learning, a significant increase in computational time is observed, due to larger amount of training samples required. However, in this case less prediction error is obtained than with the active learning. The full training case is characterized by the lowest prediction error, but the computational time is about an order of magnitude higher compared to QBC.

Table 2: Comparison of the active learning algorithms

Method	Training set size	Time (s)	RAD validation (%)	RAD evaluation (%)
DEAL	273±19	780±210	2.30±0.81	2.64±0.97
KNN	178±22	87±51	2.98±0.62	6.3±1.0
QBC	220±21	360±160	2.90±0.58	3.26±0.66
PASS	319±50	1200±1100	2.04±0.93	2.33±0.95
FULL	960	3378	-	0.39

2.6 SVR model development procedure

SVR model development procedure is given in Figure 4. Based on the conducted investigation, the procedure of the SVR model development is as following. Firstly, it is necessary to obtain the training, testing, validation and unlabeled sets. PSO method is engaged for SVR parameters optimization. Having the training set and the corresponding SVR parameters, the SVR model is developed using LibSVM code. The accuracy of the model is tested in terms of RAD. If RAD>5%, QBC is engaged to select new samples from the unlabeled set. In the first iteration 20 samples are selected, in the second 10 and in the following iterations 5 samples are selected to be labeled and added to the training set. These steps are repeated until there are samples in the unlabeled set or stopping criterion is met.
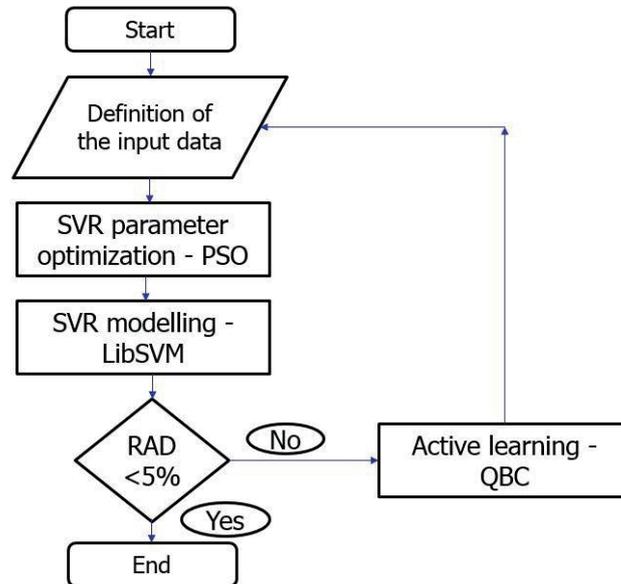
Figure 4: SVR model development procedure

3   SVR MODELS

In the following figures and tables, the results of the SVR model development for commonly used shielding materials are given. Figure 5 a) - Figure 11 a) show the learning process in terms of RAD of the validation set for each material, respectively, while in Figure 5 b) - Figure 11 b) the model testing in terms of Relative Error – RE (Eqn. 14) is given. Table 3 provides the details of the developed models: final training set size, the number of support vectors and the computational time. Exceptionally, for iron a comparison is made with the full training to check the assumption of computational time saving when active learning is involved in the process of SVR model development (Table 4). From these results it can be seen that all models have good generalization capabilities, i.e. the models can predict output values of the samples that have not been included in the model development process with generally low relative error. Exceptionally, somewhat higher relative errors for a few samples are observed in the prediction of SVR model for lead. This can be resolved by including more samples in the training set.
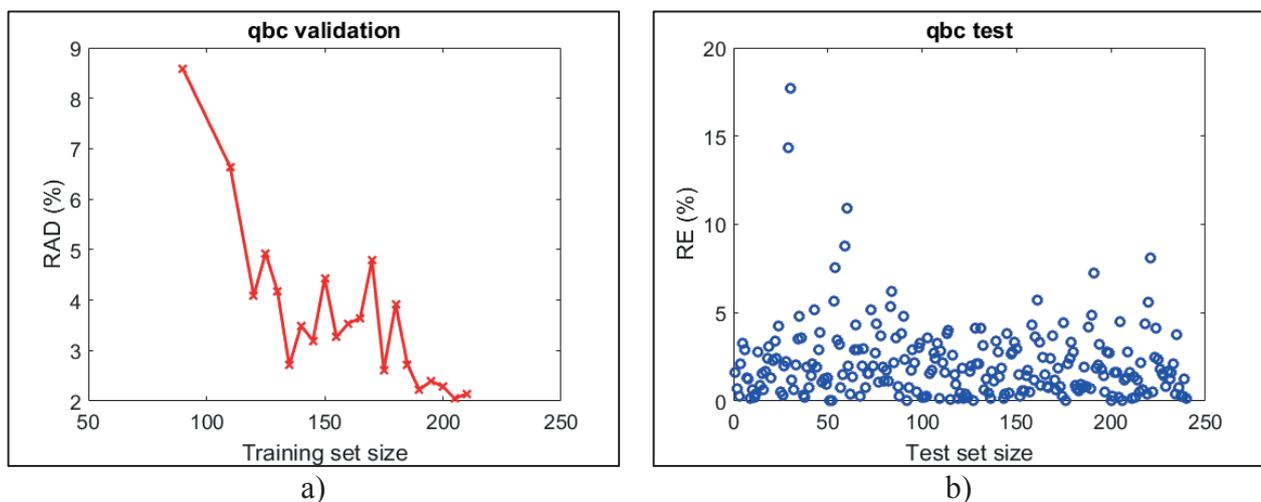
$$RE = \frac{|y_i - f_i|}{y_i} \cdot 100\% \tag{14}$$

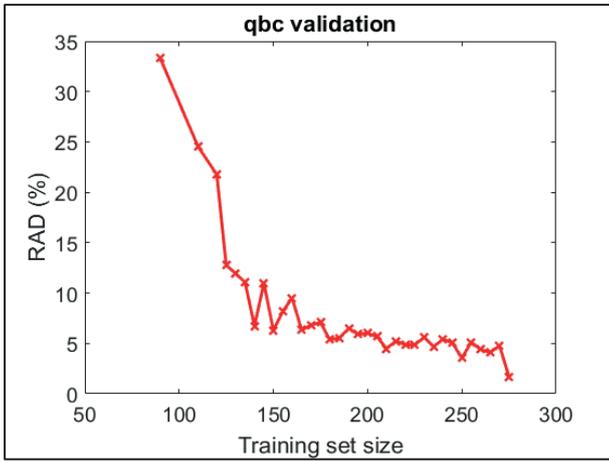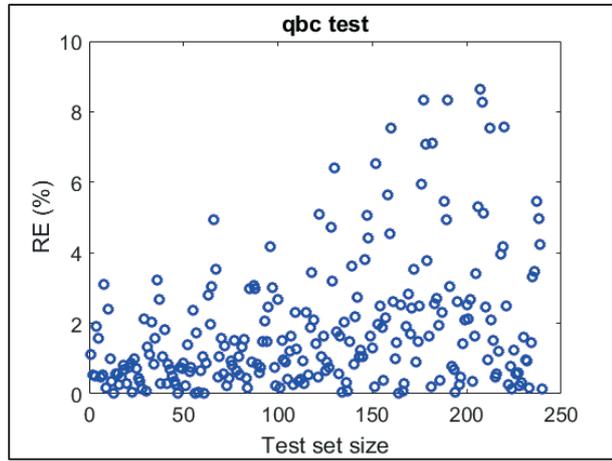a)   b)

Figure 5: SVR model for aluminum: a) model development b) final testing
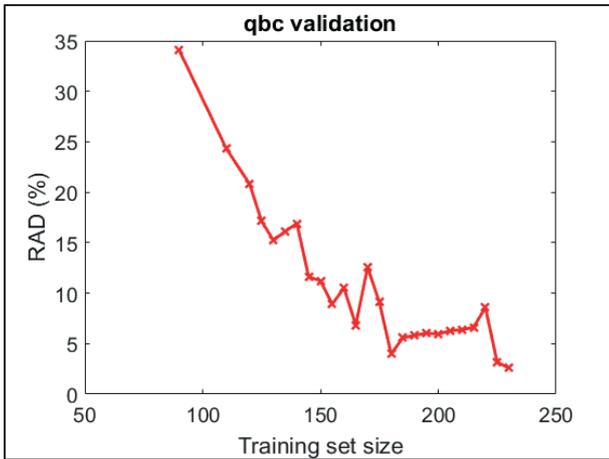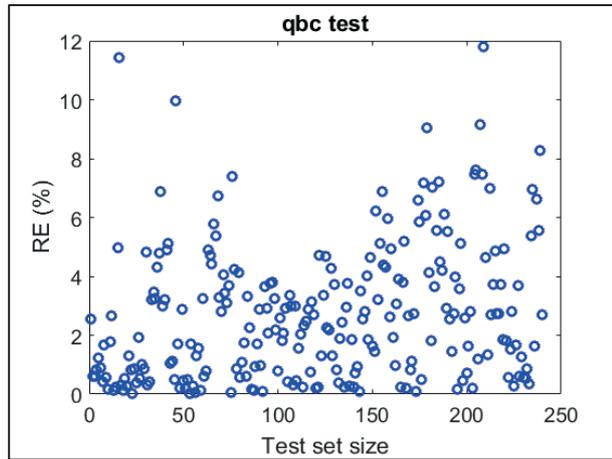
a)

b)

Figure 6: SVR model for carbon steel: a) model development b) final testing
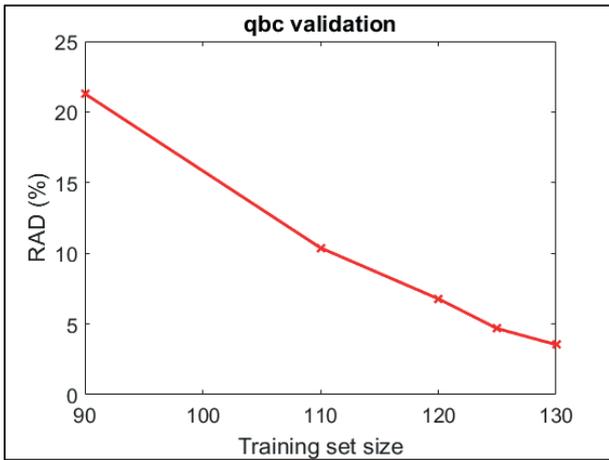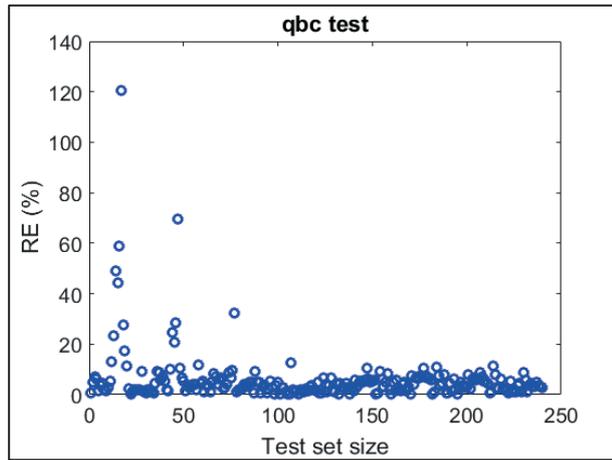

a)

b)

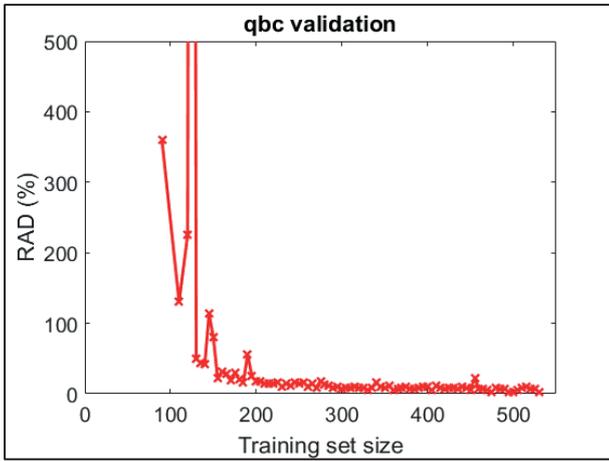Figure 7: SVR model for iron: a) model development b) final testing
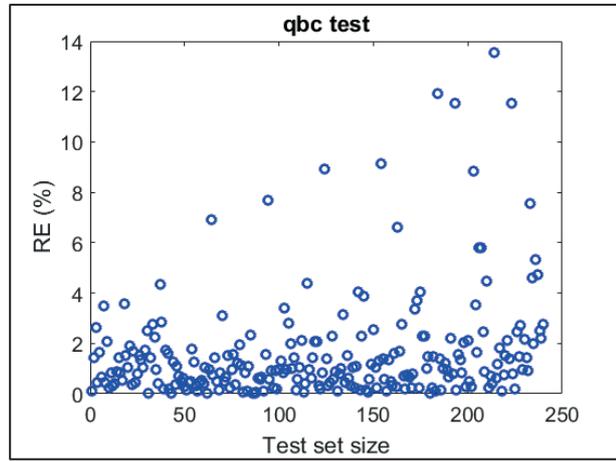

a)

b)

Figure 8: SVR model for lead: a) model development b) final testing
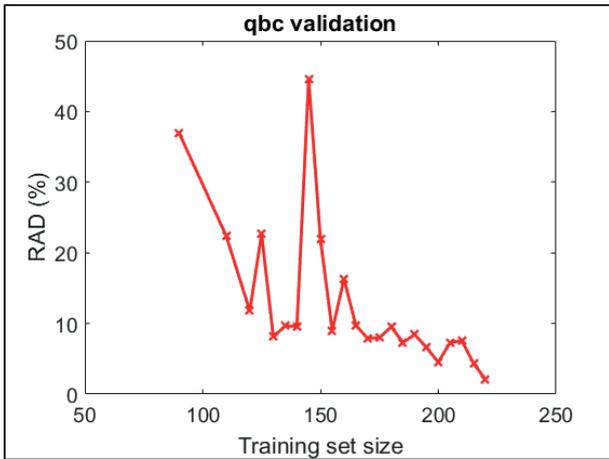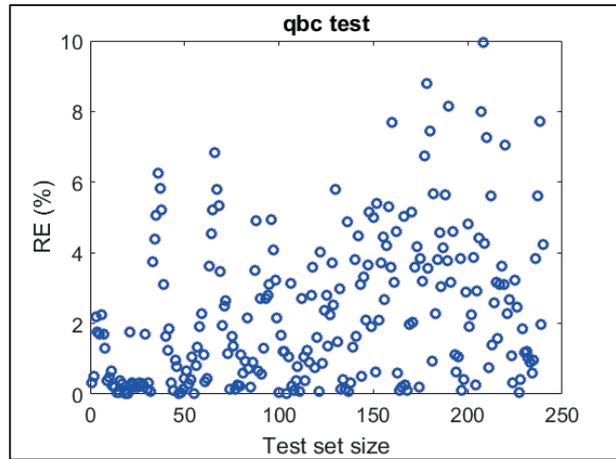
Figure 9: SVR model for concrete: a) model development b) final testing

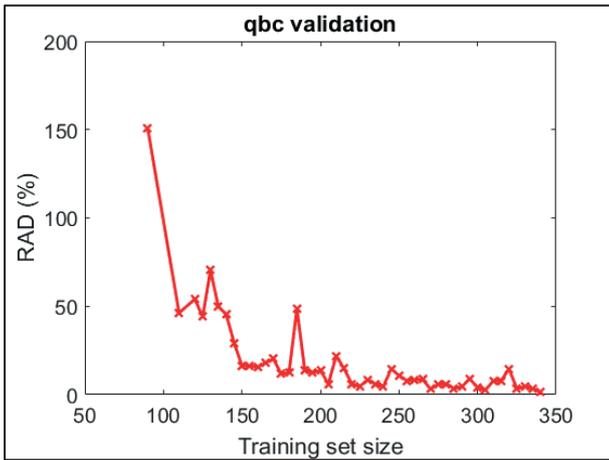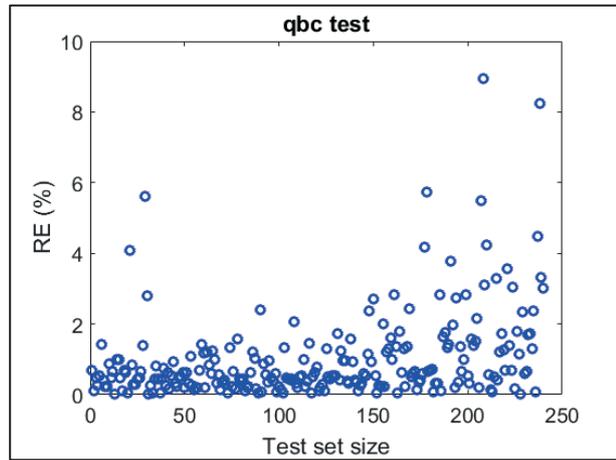Figure 10: SVR model for stainless steel: a) model development b) final testing

Figure 11: SVR model for water: a) model development b) final testing

Table 3: Training set size, number of support vectors, and computation time for all shielding materials

Material	Training set size	Number of SVs	Time (h)
Aluminum	210	97	7.36
Carbon steel	275	159	26.57
Iron	230	172	24.47
Lead	130	79	7.25
Concrete	530	416	160.21
Stainless steel	220	220	34.19
Water	340	304	35.83

Table 4: Comparison between active learning and full training for iron

Method	Training set size	Number of SVs	Time (h)	RAD (%) - test
QBC	230	172	24.47	2.72
FULL	960	410	49.40	0.58

4   QAD-CGGP MODIFICATION

QAD-CGGP is a point kernel code used for calculating fast neutron and gamma ray shielding problems. The code includes combinatorial geometry to make various shielding configurations. The preparation of an input file is simple, and the running time of the code is very fast, which make the use of the code in shielding analysis very practical.

Gamma ray shielding calculations require buildup factor data which in QAD-CGGP is resolved using Geometric Progression (GP) fitting function. Neutron shielding calculations are performed using Albert-Welton kernel or kernels obtained from the moments method solution of the Boltzmann equation.

QAD-CGGP was validated for shielding analyses of intermediate and low level waste drums by Šmuc et. al. [24]. Trontl et. al. modified QAD-CGGP to include a multi-layer option [25]. Baće et. al. introduced a multisource option in QAD-CGGP [26]. Baće et. al performed radiation shielding analyses for a dry storage facility using modified QAD-CGGP [27]. Trontl et. al used QAD-CGGP for a radiation dose evaluation of a hypothetical accident with transport package containing Iridium-192 [28].

At this stage of the research the primary goal was to develop the SVR neutron buildup factors models and test their applicability on realistic shielding problems. Therefore, the modifications performed on the QAD-CGGP code were as simple as possible without any code performance optimization being pursuit. Changes have been made to FORT10 library and gamma cross sections have been replaced with appropriate total neutron cross sections. In addition, minor modifications have been introduced into LENGTH and KERNEL subroutines in order to calculate neutron attenuation and appropriate neutron buildup factors.

5   TESTING

The testing of the modified QAD-CGGP code has been performed on the concrete spent fuel cask containing ten PWR spent fuel assemblies with the initial enrichment of 4.304 w/o U-235, burnup of 45,000 MWd/tU and a cooling period of 10 years. The total mass of the fuel in a fresh fuel assembly was presumed to be 412 kg of uranium (cca. 467 kg of UO2) with the density of 10.41 g/cm3 (95% of the theoretical density). The density of the moderator has been set to 0.709 g/cm3 as a result of the pressure of 150 bar (2,250 psi) and the temperature of 308°C (581 K, 586.2°F). The

rest of the input data regarding fuel assembly, used for isotopic inventory calculation and neutron source spectrum calculation by ORIGEN-S as a part of the SCALE SAS2H sequence, are based on a typical 16×16 Westinghouse fuel assembly.

To simplify the modelling only the construction materials and most important uranium and plutonium isotopes have been used in MCNP and modified QAD-CGGP calculations. The entire content of the cask basket containing the spent fuel assemblies has been homogenized. The dimensions and materials of the cask are the following. The inner radius is 48 cm and the outer radius is 104 cm. The first layer of the cask is 2 cm of stainless steel, the second layer is 52 cm of Portland concrete and the third layer is 2 cm of carbon steel. A point detector is placed at the surface of the cask radially and in the middle of the cask axially. The final MCNP model of the problem is depicted in Figure 12.

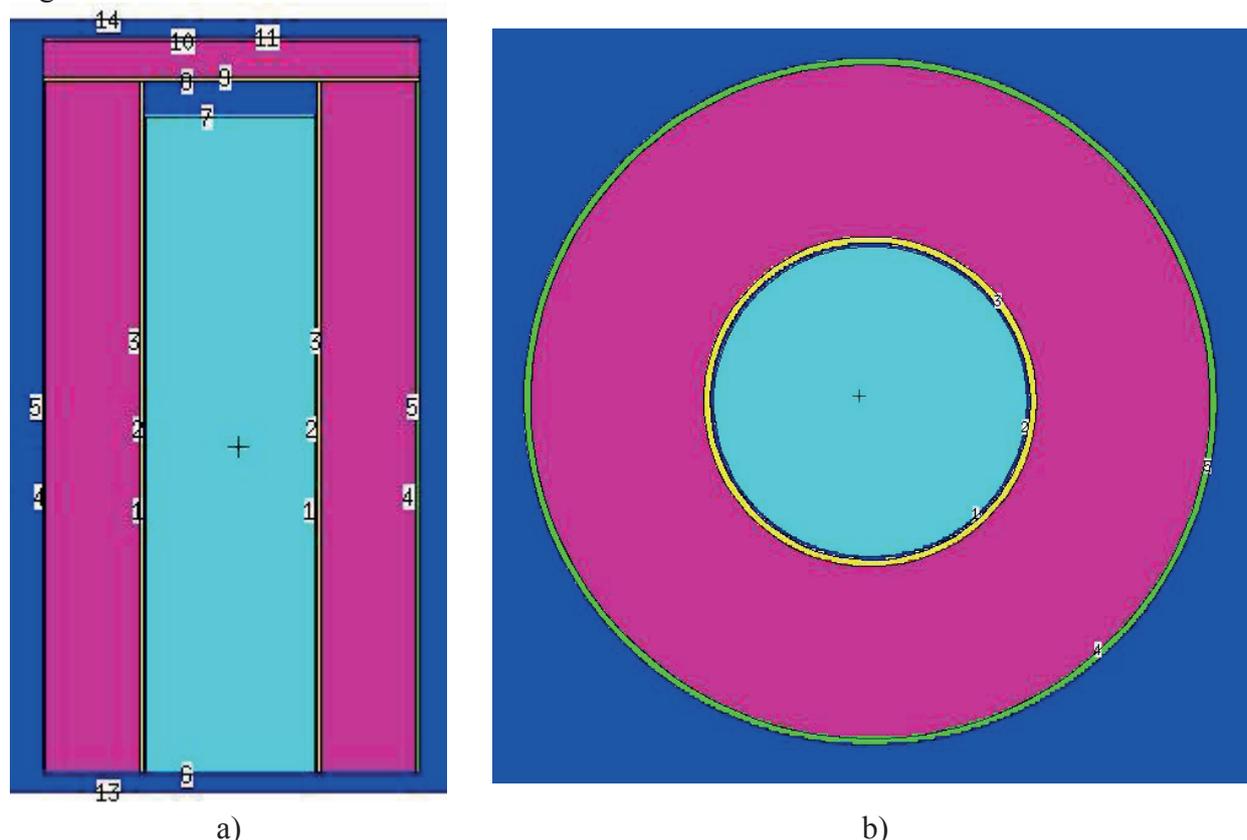a)   b)

Figure 12: MCNP model of the cask a) axially b) radially

The results of the test case calculated using the modified QAD-CGGP and MCNP are given in Table 5. From these results it can be seen that the modified QAD-CGGP and MCNP agree well at 1.2 MeV. It is interesting to note that at energies higher than 1.2 MeV QAD-CGGP results are higher than MCNP results, while at energies lower than 1.2 MeV QAD-CGGP produces lower results than MCNP. This can be attributed to the complexity of the test case geometry, which includes multiple layer shielding. The SVR models used for buildup factors calculation are developed for a simple geometry including a planar source and a monolayer slab shield. Since in multilayer configuration neutrons pass through layers of different materials having different cross sections, neutron spectrum is affected. Therefore, applying the buildup factors for a layer in the middle of the sandwich type shield (steel-concrete-steel) is inappropriate.

Table 5: Test case results

Energy (MeV)	MCNP NEUTRON REL. ERR. (%)	MCNP GAMMA REL. ERR. (%)	MCNP TOTAL REL. ERR. (%)	QAD-CGGP
14	3.73E-09 3.39	5.31E-10 9.48	4.26E-09 10.07	1.25E-06
4.5	2.17E-08 2.48	3.70E-09 5.59	2.54E-08 6.12	1.56E-06
2.5	1.36E-08 4.07	3.49E-09 3.97	1.71E-08 5.69	1.61E-07
1.5	1.89E-09 6.06	1.56E-09 6.91	3.45E-09 9.19	1.20E-07
1.2	1.48E-09 4.78	1.75E-09 6.68	3.23E-09 8.21	5.75E-09
0.5	1.51E-09 5.13	2.32E-09 12.35	3.83E-09 13.37	1.22E-11

6 CONCLUSION

In this work, total ambient dose equivalent buildup factors are calculated using SVR method. The steps of the SVR model development procedure are determined, including optimization techniques for SVR parameters selection and active learning. The SVR models are developed for commonly used shielding materials and embedded in the point kernel code QAD-CGGP. At this stage of research, the modifications of the code are as simple as possible.

The ambient doses obtained by modified QAD-CGGP code with integrated SVR neutron buildup factor model agree well at 1.2 MeV. At energies higher than 1.2 MeV, the modified QAD-CGGP results are conservative compared to MCNP results. However, at energies lower than 1.2 MeV, the modified QAD-CGGP results are lower than the MCNP results. The differences between the modified QAD-CGGP and MCNP results reach about two orders of magnitude. Although a certain level of conservatism was expected and is welcomed when an engineering approach is evoked, the difference of approximately two orders of magnitude, is a reason for concern which requires additional research. Prior testing as well as model development were performed on a pure single layer shield configuration with a planar source and a slab shield in monolayer configuration. The cask model is a multi-layer shield problem with neutrons passing through fuel assembly material and the sandwich type shield (steel-concrete-steel) which influences the neutron spectrum. The obtained results indicate that additional effort is required in developing an SVR model capable of handling multi-layer shields. To treat such a shielding configuration, redesign of an input vector is required accompanied by multi target SVR modelling.

REFERENCES

[1]   J. Wood, Computational Methods in Reactor Shielding, Pergamon, London, 1982.

[2]   P. Dučkić, R. B. Hayes, „Total ambient dose equivalent buildup factors determination for NBS04 concrete", *Health Physics*, Vol. 114(6), pp. 569-581, 2018.

[3]   J. D. Sousa de Almeida, A. C. Silva, J. A. Meireles Teixeira, A. Cardoso Paiva, M. Gattass, "Surgical planning for horizontal strabismus using Support Vector Regression," *Computers in Biology and Medicine*, Vol. 63, pp. 178-186, 2015.

[4] R. Sałat, K. Sałat, "Modeling analgesic drug interactions using support vector regression: A new approach to isobolographic analysis," *Journal of Pharmacological and Toxicological Method,* Vol. 71, pp. 95-102, 2015.

[5] N. Marković, S. Milinković, K. S. Tikhonov, P. Schonfeld, "Analyzing passenger train arrival delays with support vector regression," *Transportation Research Part C*, Vol. 56, pp. 251-262, 2015.

[6] L. Zhu, M. S. Li, Q. H. Wu, L. Jiang, "Short-term natural gas demand prediction based on support vector regression with false neighbours filtered," *Energy*, Vol. 80, pp. 428-436, 2015.

[7] G. Santamaría-Bonfil, A. Reyes-Ballesteros, C. Gershenson. "Wind speed forecasting for wind farms: A method based on support vector regression," *Renewable Energy*, Vol. 85, pp. 790-809, 2016, doi:10.1016/j.renene.2015.07.004.

[8] P. Vrablecova, A. B. Ezzeddine, V. Rozinajova, S. Šarik, A. K. Sangaiah, "Smart grid load forecasting using online support vector regression" *Computers and Electrical Engineering*, Vol. 65, pp. 102-117, 2018.

[9] A. Smola, B. Schölkopf: „A Tutorial on Support Vector Regression", *Statistics and Computing*, Vol. 14, pp. 199-222, 2004.

[10] V. Cherkassky and Y. Ma, "Selection of Meta-parameters for Support Vector Regression," *ICANN '02 Proceedings of the International Conference on Artificial Neural Networks*, pp. 687-693, 2002.

[11] C.L. Huang, C. J. Wang, "A GA-based feature selection and parameters optimization for support vector machines," *Expert Systems with Applications*, Vol. 31, pp. 231-240, 2006.

[12] Q. Yu, Y. Liu, F. Rao, "Parameter Selection of Support Vector Regression Machine Based on Differential Evolution Algorithm", Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009.

[13] G. Ren, S. Wen, Z.Yan, R. Hu, Z. Zeng, Y. Cao, "Power Load Forecasting Based on Support Vector Machine and Particle Swarm Optimization", 12th World Congress on Intelligent Control and Automation (WCICA), June 12-15,2016, Guilin, China.

[14] S. Li, H. Fang, X. Liu, "Parameter optimization of support vector regression based sine cosine algorithm", Expert Systems With Applications, Vol. 91, pp. 63-77, 2018.

[15] B. Settles, "Active Learning Literature Survey", Computer Sciences, Technical Report 1648, University of Wisconsin–Madison. 2009.

[16] Oak Ridge National Laboratory, RSICC Computer code collection: QAD-CGGP-A Point Kernel Code System for Neutron and Gamma-Ray Shielding Calculations Using the GP Buildup Factor, 1995.

[17] J. Goorley, M. James, T. Booth, F. Brown, J. Bull, L. Cox, J. Durkee, J. Elson, M. Fensin, R. Forster, MCNP^{TM} user's manual. D. B. Peplowitz, 2013.

[18] ICRP, "Conversion coefficients to Use in Radiological Protection against External Radiation. ICRP Publikacija 74", Annals of ICRP, Vol. 26, No. 3/4, 1996.

[19] C. C. Chang, C. J. Lin, LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, Vol. 2, pp. 27:1-27:27, 2011.

[20] R. Storn, K. Price, "Differential Evolution-A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, Vol. 11, pp. 341-359,1997.

[21] D. Whitley, "A Genetic Algorithm Tutorial," Statistics and Computing, Vol. 4, Issue 2, pp. 65-85, 1994.

[22] L. E. Peterson, "K-nearest neighbor," Scholarpedia, Vol.4(2):1883.

[23] R. Burbidge, J. J. Rowland, R. D. King, "Active Learning for Regression Based on Query by Committee", U: Yin H., Tino P., Corchado E., Byrne W., Yao X. (eds) Intelligent Data Engineering and Automated Learning - IDEAL 2007. IDEAL 2007. Lecture Notes in Computer Science, vol 4881. Springer, Berlin, Heidelberg.

[24] T. Šmuc, R. Ječmenica, K. Trontl, "Validation of QAD-CGGP for Shielding Analyses of Intermediate and Low Level Waste Drums", Croatian Nuclear Society, pp. 456-463, 1998.

[25] K. Trontl, M. Baće, T. Šmuc, "Incorporation of multi-layer option into QAD-CGGP code", International Conference Nuclear Energy in Central Europe '98, Terme Čatež, September 7-10, 1998.

169

Paulina Dučkić, Krešimir Trontl, Davor Grgić, Mario Matijević, Point Kernel Code Modification Including Support Vector Regression Neutron Buildup Factor Model, Journal of Energy, vol. 68 Number 2–3, Special Issue (2019), p. 156–170

[26] M. Baće, K. Trontl, D. Grgić, "Evaluation of a multisource option introduced into QAAD-CGGP code", International Conference Nuclear Energy in Central Europe 2001, Portorož, Slovenia, September 10-13, 2001.

[27] M. Baće, K. Trontl, D. Pevec, "Model of a dry storage facility for a medium nuclear power plant", Radioactive Waste Management and Environmental Remediation – ASME 2001, Bruges: ICEM, 2001. 185.1-185.6

[28] K. Trontl, M. Baće, D. Pevec, "Radiation dose evaluation for hypothetical accident with transport package containing Iridium-192 source", International Conference Nuclear Energy for New Europe 2002, Kranjska Gora, Slovenia, September 9-12, 2002.