Nuclear Engineering and Technology 54 (2022) 310–317

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original article

A comparative study on applicability and efficiency of machine learning algorithms for modeling gamma-ray shielding behaviors

Bayram Bilmez [a, b], Ozan Toker [a], Selçuk Alp [c], Ersoy Öz [d], Orhan İçelli [a, *]

[a] Department of Physics, Faculty of Science and Art, Yildiz Technical University, İstanbul, Turkey
[b] Department of Physics, Faculty of Science and Art, Ondokuz Mayıs University, Samsun, Turkey
[c] Department of Industrial Engineering, Faculty of Mechanical Engineering, Yildiz Technical University, İstanbul, Turkey
[d] Department of Statistics, Faculty of Art and Science, Yildiz Technical University, İstanbul, Turkey

ARTICLE INFO

Article history:
Received 30 September 2020
Received in revised form
18 June 2021
Accepted 18 July 2021
Available online 19 July 2021

Keywords:
Mass attenuation coefficient
Artificial neural network
Fuzzy logic
Non-linear regression analysis

ABSTRACT

The mass attenuation coefficient is the primary physical parameter to model narrow beam gamma-ray attenuation. A new machine learning based approach is proposed to model gamma-ray shielding behavior of composites alternative to theoretical calculations. Two fuzzy logic algorithms and a neural network algorithm were trained and tested with different mixture ratios of vanadium slag/epoxy resin/antimony in the 0.05 MeV–2 MeV energy range. Two of the algorithms showed excellent agreement with testing data after optimizing adjustable parameters, with root mean squared error (RMSE) values down to 0.0001. Those results are remarkable because mass attenuation coefficients are often presented with four significant figures. Different training data sizes were tried to determine the least number of data points required to train sufficient models. Data set size more than 1000 is seen to be required to model in above 0.05 MeV energy. Below this energy, more data points with finer energy resolution might be required. Neuro-fuzzy models were three times faster to train than neural network models, while neural network models depicted low RMSE. Fuzzy logic algorithms are overlooked in complex function approximation, yet grid partitioned fuzzy algorithms showed excellent calculation efficiency and good convergence in predicting mass attenuation coefficient.

1. Introduction

Radiation and matter interactions are complicated physical processes. High energy photons (X-rays and gamma rays) interact with matter via various mechanisms such as Rayleigh scattering, Compton scattering, photoelectric absorption, pair production, and nuclear field interactions. Mass attenuation coefficient (μ/ρ) is a widely used and essential photon atomic parameter for medical imaging departments, nuclear power plants, industrial irradiation rooms, and X-ray crystallography [1]. That parameter encompasses complicated physical mechanisms all at once. As the photon imaging and irradiating industry got sophisticated, more precise knowledge of the μ/ρ values is on demand. Since the significant increase in high energy photon radiation usage added to naturally occurring cosmic rays and gamma-emissions, there have been extensive tables of μ/ρ values, experimentally at first and theoretically later [2]. The experimental results are not satisfactory for every energy range, and different material types and the increase of materials at an unprecedented pace made this task futile. Monte Carlo simulations are also used in determining μ/ρ for specific material types, but they may require high processor hours [3,4]. When doing the theoretical calculations, although the underlying principles are known, it is generally hard to find an exact solution to real physical systems. Approximate methods are employed to overcome this obstacle. Most sophisticated approximate methods themselves can be of great complexity and may require intense calculations for valid results. For instance, the XCOM program of the National Institute of Standards and Technology [5] provides widely trusted and rather accurate cross-sections, calculated theoretically and semi-empirical corrections to some values. In the XCOM database, to determine the Compton process's contribution to μ/ρ, the famous Klein-Nishina formula and a non-relativistic scattering function S (Z,x) are used [6,7]. In this approach, photons are treated as particles instead of the *S*-matrix approach, where photons are an excitation in the electromagnetic field [8].

* Corresponding author.
E-mail addresses: oicelli@yildiz.edu.tr, oicelli@yildiz.edu.tr (O. İçelli).

https://doi.org/10.1016/j.net.2021.07.031

Another feature of the current methods (XCOM and Monte Carlo Simulations) is that they depend on the elemental content of the compounds and mixtures. However, materials in the industry often come as metal oxides and complex chemical compositions of organic materials. Moreover, their mixtures may have a complicated chemical profile. To determine the desired μ/ρ of those type of mixtures and composite materials require a continuum of known material mix ratios [7]. The machine learning (ML) approach was proposed for the optimization of μ/ρ values. A suitable composite material from our former study was chosen to illustrate our intended result [9]. Vanadium slag/epoxy resin/antimony oxide has already been evaluated as an excellent high-energy-photon shielding material, and varying ratios of constituents were evaluated. This time, we will try to achieve similar results with several machine learning algorithms.

Machine learning algorithms are currently being used in every field, from particle physics to analyzing and predicting human behavior to optimizing transfer routes, whenever modeling complex data is required [10–12]. They are quite useful in non-linear regression and can model any curve in theory. In this study, some of the well-known methods for being efficient are utilized. Which are: adaptive neuro-fuzzy fuzzy logic (NF) models [13], with two different input partitioning algorithms, first one is grid partitioning (GP) and the second one being subtractive clustering (SC) and feed-forward artificial neural network models (ANN) [14,15], with training algorithm Levenberg-Marquardt (LM) backpropagation. In the literature there have been numerous applications to model materials' different physical properties, and nuclear engineering materials are not exceptions [16–21].

Here, it would be appropriate to give a brief literature review.

In their study, Ref [17] used fuzzy logic to predict μ/ρ values of concretes, where they specified membership functions manually and not algorithmically. Ref [18] studied neutron activation analysis with neural networks to determine the optimum mixture ratio in cement. They had limited experimental data, with 29 data points, in which the models were overfitted the data. Ref [19] studied gamma-ray build up factors for composites with similarly sized data to our study. Ref [20] used artificial neural networks to predict μ/ρ of minerals with varying thicknesses and wide energy. They reported remarkable agreement with ANN predictions and simulation results. Ref [21] used ANN to optimize the mixture ratio in terms of neutron shielding and compressive strength.

Former studies generally emphasized the physical mechanisms with limited data and generally used neural network structures. Fuzzy logic algorithms are lately overlooked in the mathematical analysis compared to more recent methods. We, however, concur that area expertise along with machine learning may produce good results. This study seeks to compare the training algorithms' efficiency to model physical phenomenon, i.e., μ/ρ. The main goal is to find the most efficient one among the studied algorithms to determine radiation shielding parameters.

The three algorithms were trained by changing the number of randomly chosen data and tested with the remaining data points. Each algorithm was first optimized by trying out different adjustable parameters. Afterward, the results of the three algorithms were evaluated and compared in error performance and computer time efficiency with each other. Each code was accounted for the computer elapsed time. Each code was repeated 100 times to reduce the statistical fluctuations within the algorithms' errors and times elapsed.

Former studies relied on limited data sets and a narrower energy range. An extensive data set is used to model radiation and matter interaction with different machine learning algorithms and compared with theoretical values. For the first time, a comparative study of different algorithms is presented, reducing errors down to 0.0001. Hence, this study shows ML methods' applicability to model μ/ρ for the composite family vanadium slag/antimony trioxide/epoxy resin. Fuzzy logic models trained with GP and SC and ANN trained with LM algorithm were first optimized to produce reliable models. Later each algorithm was compared with each other in terms of computing time and error rates. This study is a step into determining more accurate μ/ρ data with a machine learning based approach.

2. Materials and methods

2.1. Materials

Vanadium slag is a by-product of the steel industry and is composed of various metal oxides. Epoxy resin is a well-known and widespread organic compound that is generally used as an adhesive on composites. Mixing vanadium slag and epoxy aims to make a strong, durable, lightweight composite and good radiation shielding. Antimony trioxide is used as a flame retardant additive in industry and enhances the shielding behavior of materials due to the high atomic number of antimony. The addition is expected to increase flame retardant ability and shielding ability. The chemical composition of the vanadium slag, epoxy resin, and antimony trioxide is given in Table 1. The used mixture ratios of the three constituents determine the physical properties of the end product. The mixture ratios and more details can be found elsewhere [9].

2.2. Mass attenuation coefficient

The mass attenuation coefficient is an essential parameter to know about photon matter interactions. It is independent of the physical state and the density of the medium. The mass attenuation coefficient can be determined via two different approaches [22]. One approach stems from experimental setups in which gamma-ray photons are assumed to obey the Lambert-Beer law for homogenous medium given in equation (1).

$$I(t) = I_0 e^{-(\mu/\rho)t} \tag{1}$$

In equation (1), t represents the mass thickness of the material; I_0 is the incoming beam intensity, and I(t) is the reduced intensity of the incoming beam after interacting with t mass thickness. The Lambert-Beer law is valid for narrow beam attenuation and disregards scattered photons. Experiments use this model to determine the mass attenuation coefficient.

The second approach is originated from quantum mechanical calculations of several different cross-sections. They are assumed to be additive, which is an approximation. To clarify, the total mass attenuation coefficient can be written as a sum of total contributions from different physical interactions as given in equation:

Table 1
Chemical composition of constituent compounds.

Vanadium Slag		Epoxy Resin	
V	0.0572	C	0.7564
Fe	0.2863	H	0.1245
Si	0.0789	O	0.0865
Ti	0.0654	N	0.0326
Mn	0.0618	**Antimony Trioxide**	
Cr	0.0333	Sb	0.8354
Mg	0.0170	O	0.1646
Ca	0.0151		
Al	0.0150		
O	0.3700		

$$(\mu/\rho)_{Total} = C_{Rayleigh} + C_{Compton} + C_{Photoelectric} + C_{Pair\ production} \quad (2)$$

According to Thomson, the Rayleigh contribution is calculated for elastic scattering, and relativistic atomic form factors were used. The Compton contribution is determined with the Klein-Nishina formula, and a non-relativistic incoherent scattering function S(Z,x) is used. The photoelectric interaction is determined with the Hartree-Slater model, and a semi-empirical correction is applied above 1.5 MeV. The pair production mechanism is added with the Bethe-Heitler atomic differential cross-sections, and triplet production (near an orbital electron) cross-section is omitted in the calculations [5–7,23].

For elements, those values are tabulated. For different elemental mixtures and compounds, μ/ρ is calculated with weight fractions according to equation (3).

$$(\mu/\rho)_{mix} = \sum w_i(\mu/\rho)_i \quad (3)$$

As mentioned, there are various methods for the determination of mass attenuation coefficients of the materials. Some of these methods, such as experiments and detailed simulations, can be time-consuming and tedious for extensive energy ranges. We develop a new method for determining mass attenuation coefficient values of a composite family using machine learning algorithms. The μ/ρ values for various mixture ratios of vanadium slag/epoxy resin/antimony trioxide are calculated for the energy interval of 50 keV to 2 MeV.

2.3. Machine learning algorithms

First, a data set of 2400 data points with three inputs and one output for different mixture ratios was determined. The first input was vanadium slag/epoxy ratio, the second was the antimony weight percent, and the third is the photon's energy, whereby the output is the μ/ρ values.

Machine learning algorithms perform better in better-processed data sets [14]. Some standard data transformation techniques were used to adjust to a more standardized format. Firstly, the natural logarithm of the third input data is taken, which is the energy variable. The logarithm operation tames the big difference between the minimum and the maximum values of the energies, from 50 keV to 2 MeV. To further clarify the data, all of the three inputs were normalized. The min-max normalization of the data is done according to the equation:

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (4)$$

here x_i is the initial value, and x_i' is the normalized value, x_{max} is the maximum valued input in the same kind of inputs, and x_{min} is the minimum valued input. After the normalization process, all the data points lie between 0 and 1. This process generally makes the training process faster.

After the normalization process, the data is fed to algorithms for training. MATLAB version 9.9 was used for the calculations [24].

2.3.1. Fuzzy logic

Fuzzy logic is a logical inference system that takes its roots from fuzzy set theory, conceptualized by Zadeh [25]. By assigning membership degrees to inherently vague objects, it enables mathematical reasoning. When newly introduced, fuzzy logic was used to mimic human intuition inputs to control machinery. Later, more sophisticated algorithms were developed to enable mathematical analysis and tackle complex optimization problems. Takagi-Sugeno-Kang or TSK for the short, system is a well-known fuzzy logic algorithm that is used to model single output data [26,27]. In the TSK model, the output membership function is linear primarily or constant. TSK system utilizes fuzzy logic to describe the system mathematically given a data set with multiple inputs and outputs. The description then allows one to deduce results for previously unknown data inputs, enabling new information about the system.

The procedure is relatively straightforward. Firstly we define inputs with membership functions. Those membership functions characterize the inputs for respective input types with relation to their possession of specific criteria. This criterion can be a linguistic variable, like being tall or short, which is essentially a subjective thing or an abstract concept, and the value of the membership goes from 0 to 1. Then a set of if-then rules are applied to infer results. In crisping the output data, we used a weighted average of rules. One of the rules can be formulated as follows: if x_1 is a, AND x_2 is b, AND x_3 is c, THEN output is y, where x_1, x_2, x_3 represent input data points, whereas a, b, c are membership functions. Afterward, each rule is given a weight. This weight is determined as follows.

$$w = a(x_1)b(x_2)c(x_3) \quad (5)$$

here $a(x_1)$ represents the membership degree of variable x_1 in the a membership function. Finally, the output is determined by weighted average of the rules as follows:

$$Output = \frac{\sum w_i y}{\sum w_i} \quad (6)$$

To model non-linear data, one must choose appropriate membership functions. To define a membership function, a set of parameters must be specified. The most common membership functions are triangular, sigmoidal, trapezoidal, and Gaussian membership functions. For instance, a Gaussian membership function can be described with equation (7).

$$G(x, \lambda, \sigma) = \exp\left(\frac{-(x-\lambda)^2}{2\sigma^2}\right) \quad (7)$$

whereas the Trapezoidal membership function can be formulated like equation (8).

$$T(x, p, q, r, s) = \max\left\{\min\left(\frac{x-p}{q-p}, 1, \frac{s-x}{s-q}\right), 0\right\} \quad (8)$$

here p is the left leg, and s is the right leg of the trapezoid, and q and r are the left and right shoulders, respectively.

The bell shaped membership function is described by equation (9).

$$B(x,\ a,\ b, c) = \frac{1}{1 + \left|\frac{x-a}{b}\right|^c} \quad (9)$$

here a variable is the center of the bell curve, and b and c determine the spread of the function.

Determining the correct kind, number, and place of the membership functions is crucial in fuzzy inference systems and affects the results dearly. It is generally too cumbersome to find ideal parameters for fuzzy inference systems. To make it easier to find the correct parameters of the membership function, neural network models are used along with fuzzy inference systems, hence named neuro-fuzzy (NF) systems [13].

To process the data for the NF algorithms, the input space must be classified into a more useful format. Two different partitioning methods are used to create the rule base from the inputs: grid

partitioning (GP) and subtractive clustering (SC).

In grid partitioning, each input variable type is parsed into grids by grid lines. Moreover, every possible combination of the input data is used to form the rule basis for inference. This operation gets exponentially more challenging as the data set grows. In our model, the number and the type of the membership functions were determined.

In subtractive clustering, the input data is first searched for the most probable cluster center. After the center is determined, a cluster bound is formed with a specified radius. All the data points within the region are subtracted from the data set, and the next cluster center is searched. This process continues until every input data point is placed into a cluster. In the SC algorithm, the data is pre-processed by clustering, and the rules are formed by the most relevant clusters of the input data set. SC algorithm enables the rule set to be more efficient in larger data sets, but it falls in efficiency for smaller data sets. When the SC method is chosen, the model gives good predictive results even before optimizing weights, which is due to the clustering process being more robust than the partitioning process. Consequently, this SC process can take an extensive amount of time for big data sets. For the same data set, the clustering process identifies clusters based on the radius number given, and as a result, there are n clusters. Each cluster can be thought of as a hyper-cube with the dimensions of input data types. The number of fuzzy rules produced is equal to the number of clusters. For the grid partitioned data, the number of the rules is the product of the grid numbers of each input data type.

In this work, the hybrid method was used to optimize the fuzzy rule weights after the data set is classified by both methods. Backpropagation of steepest descend to input related weights; least-squares estimation to determine output related membership functions were applied. Backpropagation will be discussed more thoroughly in the next section.

2.3.2. Artificial-neural networks

Artificial neural networks are a mathematical abstraction of the natural pattern recognition ability of conscious organisms. In a way, they resemble the biological brain by many interconnected neurons and layers. A single neuron can perform only an elementary task like multiplication and addition, yet their combination can be powerful tools for overcoming complicated tasks, like speech and image recognition, classifying complex data [11,12,15].

Feedforward neural networks (FFNN) are a type of ANN used in classification and fitting models. In the FFNN, the layers are arranged linearly and do not form loops. A simple model illustrating the basic mechanism of the FFNN's is given in Fig. 1. The main objective of the ANN models comes down to finding the minima of a cost function to specify the weights and biases of the network structure. The most common algorithm of feed-forward neural networks is Levenberg-Marquardt (LM) backpropagation algorithm [28]. LM allows fitting highly non-linear models in an efficient way. It is an iterative process to minimize the sum of the squares of deviations, similar to the Gauss-Newton (GN) method. In the GN method, to find the minimum of the cost function.

$$x_{i+1} = x_i + \left(J^T J\right)^{-1} J^T e \tag{10}$$

iteration formula is used, where e J is the Jacobian matrix, and $J^T e$ term represents the gradient of the cost function. In the LM algorithm, a damping term is added to this. LM algorithm iteration is formulated like;

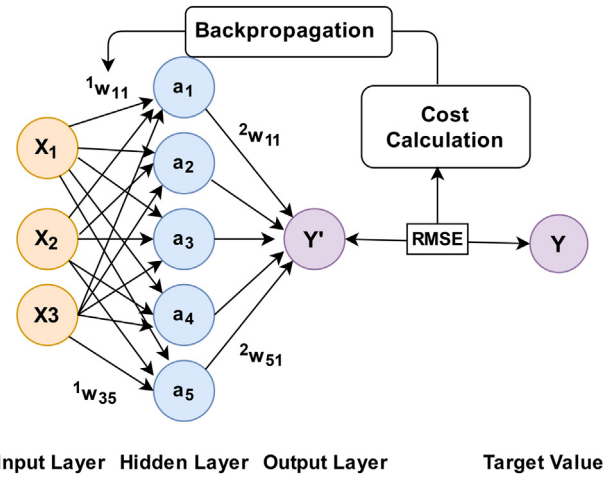

Fig. 1. Schematic representation of FFNN with a single hidden layer composed of 5 neurons. Three attribute types of input and one type of output are present.

$$x_{i+1} = x_i + \left(J^T J + \lambda I\right)^{-1} J^T e \tag{11}$$

where λI term is the damping term added to GN algorithm. If λ is zero, this is the same as the GN, and when λ is very big the method converges to gradient-descend. The choice of λ parameter determines how fast the convergence will happen. LM algorithm is generally a high-speed and robust algorithm, but it has flaws. For instance, it is known to be prone to overfitting. Overfitting occurs when the network training goes too far, and the model fails to address non-trained data. Several approaches can be taken to prevent overtraining. One of them is to reduce the training intensity and to increase the hidden layer size. Another approach is to modify the algorithm by adding validation data. The validation data is used to measure the representative ability of the model in the training process. While the training occurs, the model is tested with the validation data. RMSE is kept track of after each epoch. If the RMSE of the validation error keeps growing a certain number of times, then the algorithm stops to prevent overfitting.

The algorithm continues training until one of the several conditions are met:

a) When the pre-determined epoch number is reached.
b) When a pre-determined time limit is reached.
c) When the gradient drops below a particular value.
d) When the minimization goal is achieved.
e) When the maximum number of validations are reached.

2.3.3. Performance evaluation

For the evaluation and comparison of the studied algorithms, we used root mean squared error (RMSE), mean absolute percent error (MAPE), and maximum percent error deviation MPED values [29].

The formulas for RMSE, MAPE, and MPED are as follow:

$$RMSE = \sqrt{\frac{\sum_i^n \left(x_i' - x_i\right)^2}{n}} \tag{12}$$

$$MAPE = \left(\frac{1}{n}\sum_{i}^{n}\left|\frac{x_i' - x_i}{x_i}\right|\right) \times 100 \quad (13)$$

$$MPED = \max\left(\left|\frac{x_1' - x_1}{x_1}\right|, \cdots, \left|\frac{x_n' - x_n}{x_n}\right|\right) \times 100 \quad (14)$$

where x_i' is the value predicted by the model, x_i is the value known, n is the number of the data points.

Neural network models start from random initial weights and biases and fix them according to a cost function, and the cost functions generally have multiple local minima. Hence, the outcome is determined by the starting point and can be quite different. To reduce the statistical fluctuation of the models, we run each code 100 times and present the 100 runs combined results. Because scaling of the data does affect the error values, MAPE and MPED values do not always accurately represent the errors in the models. This lack of representativeness is especially relevant in processed data sets, as in our case.

Nevertheless, they can still be utilized to compare the results within the same model [29,30]. We represent and compare the best 20 of the MAPE and MPED values and their standard deviations in our results to get the most meaningful comparisons. This is justified as the models not performing well enough would not be chosen as the final model.

2.3.4. Data division

When working with an adequate data set in machine learning algorithms, it is generally beneficial to divide data into training sets, validating and testing data.

Training data number is the main factor that determines the learning outcome. The predictive power of the trained algorithm directly depends on training data representing the data set. Generally, a more extensive training set ensures training to be less erroneous and more robust.

Validation data is required to keep the training data set from overfitting. As a rule of thumb, we used a training/validation data ratio to be 4/1.

Table 2
Training, validation and testing data division percentages.

training%	validation%	testing%
8	2	90
16	4	80
24	6	70
32	8	60
40	10	50
48	12	40
56	14	30
64	16	20
72	18	10

In the optimization process, the total data set of 2400 data points were divided as follows:

a) 576 training data points (24%)
b) 144 validating data points (6%)
c) 1680 testing data points (70%)

Training set size is chosen to ensure enough data point is represented for each run to model correctly and still enable the extensive number of try-outs, as bigger training data makes the training process prolonged. The 24/6/70 ratio for train/validation/test data was both practical and adequate for all our purposes.

After each algorithm is optimized with those ratios, we divided the data set with varying ratios to compare each other. This way, we could keep track of each algorithm's predictive power by changing the number of data points. The different divided data numbers are given in Table 2.

3. Results and discussion

The algorithms were first optimized before they were compared. Different types of membership functions, several influence radii, and hidden layer sizes were tried while everything else is held fixed to demonstrate their performance, different membership functions, several influence radii, and hidden layer sizes. As there were many possible parameters to choose from, only those that stood out are presented in the results. Firstly, in each algorithm's optimization processes, the 30% training plus validation set and 70% test set are used.

3.1. Subtractive clustering optimization

The fuzzy models trained with the SC algorithm had influence radius parameter to decide. Influence radius versus the RMSE of the training set and test set is given in Fig. 2. It can be seen from Fig. 2 that giving too high and too low values of influence radius results in unsatisfying models. This outcome was expected due to the influence radius being the main parameter determining the input membership function number. The number and the type of the membership functions have an immense effect on fuzzy systems. The best results in terms of the influence radius are seen to be around r = 0.4. To further hone the models' predicting power, visual feedback from the previously obtained models' membership functions was helpful to determine the behaviors of the input and output data. As expected after a series of trials, the variation within each feature of the data dictates the optimal influence radius. An illustration of one of the trained models with membership functions, rules, and the output surfaces generated from two input functions is given in Fig. 3. The fuzzy surfaces presented can be deceiving, as the first surface seems to fluctuate less than the second surface. It is the opposite, as the μ/ρ varies faster in lower

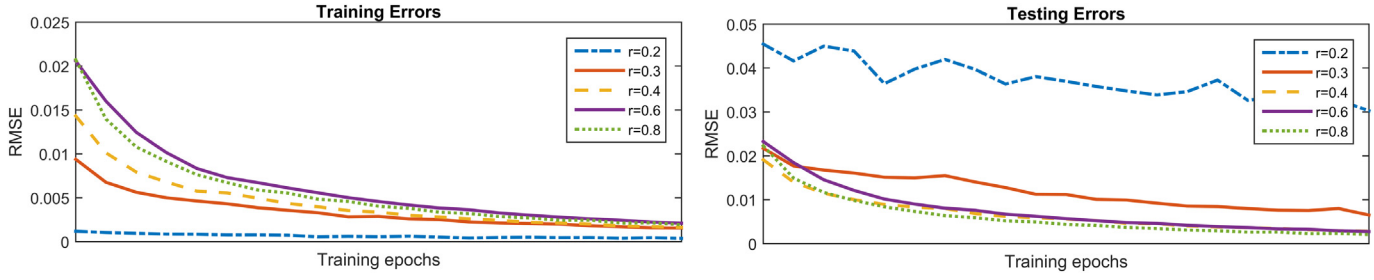

Fig. 2. RMSE values of SC algorithm for different influence radius (upper part Train RMSE, lower part Test RMSE).

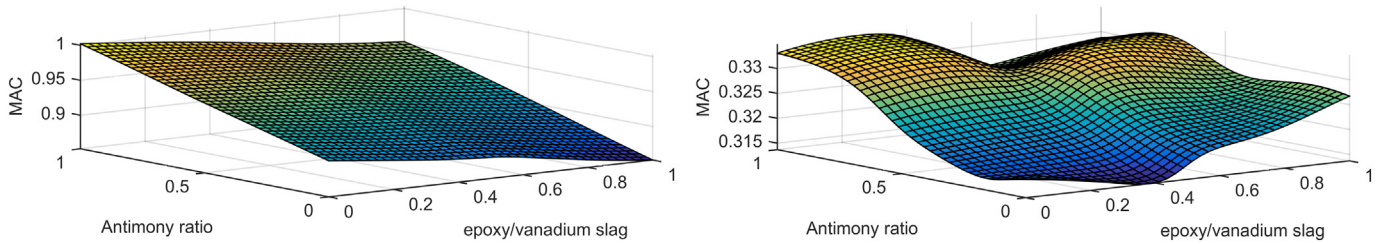

Fig. 3. The fuzzy surfaces generated with the SC algorithm. (Two inputs versus mass attenuation coefficients are given with each figure. In the upper fig, the third input is fixed at 0.1. In the lower fig, the third input is fixed at 0.5).

energies, but the y axis representing the μ/ρ values is given in different grid intervals.

3.2. Grid partitioning optimization

For the fuzzy models with GP, initially, a number and membership function must be stated. The membership function types and numbers were determined similarly to the previous model. First of all, the membership functions were held fixed at four, and the membership functions were varied. The model's errors and training epochs versus train RMSE and test RMSE for different types of membership functions were illustrated in Fig. 4. The tried functions were trapezoidal, triangular, Gaussian, two Gaussian functions combined, and bell-shaped. Fig. 4 indicated that using the Gaussian membership function fits the data set better, and it was generally the fastest among them.

3.3. Levenberg marquardt optimization

For the feed-forward neural network with an LM training algorithm, to evaluate the optimum parameters, each option was run 100 times, and the best value for hidden layer size was chosen. The models evaluated with their RMSE for different hidden layer sizes were illustrated in Fig. 5. Up around 30 to 35, there is a sweet spot for hidden layer neuron number, in which the best predictive power is reached, and further increasing the hidden layer size promises hardly any benefit. Further optimization with multiple hidden layers is also possible.

3.4. Comparison between algorithms

After the optimization processes, RMSE, MAPE, and MPED values for the three algorithms are given in Table 3. While determining the most suitable algorithm, it is necessary to consider the computer times and the error values. Also, each algorithm's test errors of RMSE are given in Fig. 6. The importance of data divisions becomes apparent here. Appropriate percentiles should be determined for the lowest RMSE values. In order to see the success of the models concretely, the mass attenuation coefficients of the selected

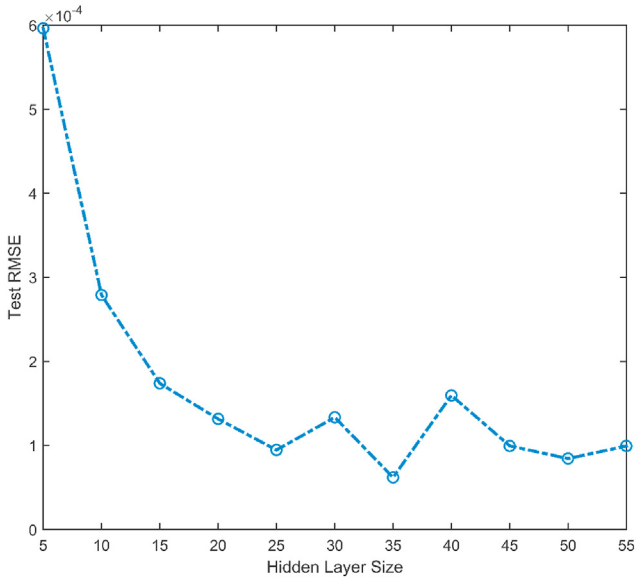

Fig. 5. Hidden layer size dependency of RMSE values for LM algorithm.

Table 3
Error values and computer times for studied algorithms. (MAPE and MPED are given as mean ± SD).

	Sub. Clust. NF	Grid Part. NF	Lev. Marq. NN
RMSE	1.83e-3	8.13e-5	9.08e-5
MAPE	0.227 ± 0.025	0.156 ± 0.011	0.065 ± 0.007
MPED	26.7 ± 6.1	17.7 ± 3.6	14.9 ± 2.9
CTM (s)	1093.4	1222.1	3040.9

material were predicted. The percentage deviations from XCOM calculations of μ/ρ values for the mixture in standard gamma-ray energies are given in Table 4. The models used to get Table 4 results were trained after the optimization. Algorithms are accurate for three digits. The GP algorithm makes a better prediction in the

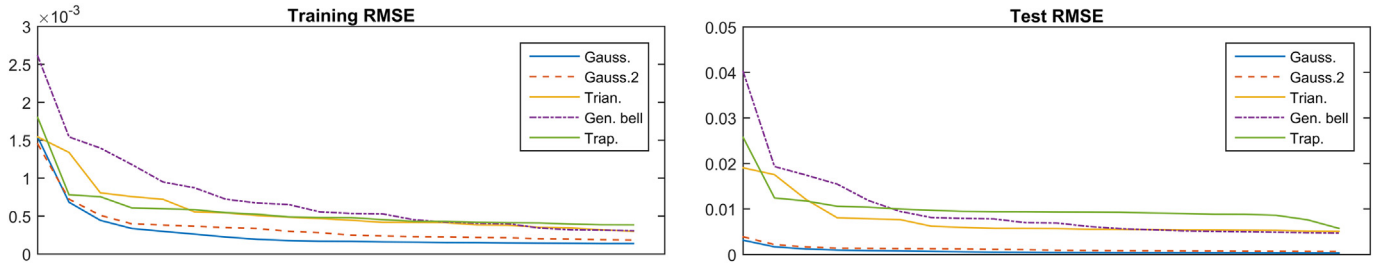

Fig. 4. RMSE values of different membership function types in GP algorithm (upper part Train RMSE, lower part Test RMSE).

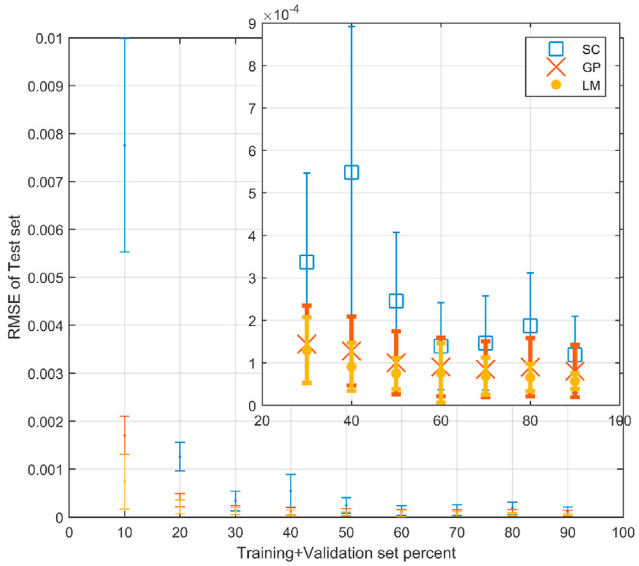

Fig. 6. Comparison of three models' RMSE values for different number of training, validation, and testing data numbers.

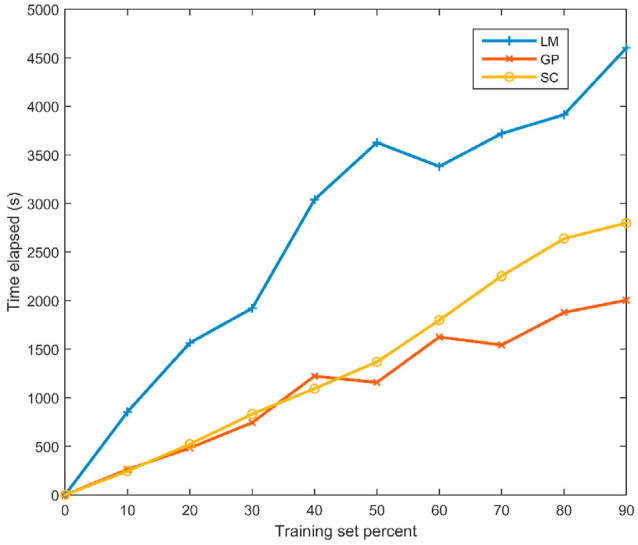

Fig. 7. Comparison of three models' training time depending on training set size.

Table 4
The percentage deviation of three algorithms' prediction with XCOM results at standard gamma-ray energies.

Energy (keV)	SC	GP	LM
59.54	1.04%	0.11%	0.14%
81	0.94%	0.02%	0.02%
356	0.27%	0.01%	0.04%
511	0.10%	0.01%	0.02%
662	0.07%	0.04%	0.00%
1130	0.17%	0.03%	0.00%
1330	0.20%	0.07%	0.04%

lower end of the spectrum, whereas the deviations in the LM algorithm drop to zero in the middle energies.

It can be seen from Table 3 that test RMSE down to 9×10^{-5} for LM algorithm, and 8×10^{-5} with GP. This test RMSE values are quite remarkable; previously train RMSE values as low as 10^{-10} were reported [18], but such low values of test RMSE are new. The best algorithm with 30% training according to test RMSE values is the GP algorithm, but only slightly. The other two markers, MAPE, and MPED values, show LM's superiority in these regards. This can be explained by the models produced by GP being more stable, as the first 20 models from GP are worse than the best 20 of LM, but in the RMSE aspect, GP is slightly better.

The trained models showed a determination coefficient (R-squared) of 1, which supports the claim that μ/ρ can be modeled via ML algorithms outside the edge and resonance regions, consistent with Ref [20].

As it can be seen from Table 4, the models illustrate strong predictive power, suggesting that μ/ρ values can easily be determined with sufficient data. The training data did not include energy grid variations below 10 keV, making the models' success quite remarkable. As an extensive data set was used to train models, and it is not always possible to get this sort of a complete training set, it is better to develop insight and compare with previous reports of Ref [31]. To further analyze how large a data set is required to get adequate models, training, validating, and testing data sets were varied. Increasing the number of the training set affects the accuracy, but only slightly after a saturation region of around 40%. Moreover, some of the increase can be attributed to shrinking the test set as the training set gets large. Again, from Fig. 6, it can be concluded that for GP and LM algorithms, RMSE gets as low as 10^{-4}, which is near the generally presented precision of μ/ρ.

According to Fig. 6, the GP algorithm performed well in the smallest training set, i.e., 10% with little standard deviation, and the LM model evened up at 20% training set. From there on, the LM algorithm indicated better results. The LM models reached a learning saturation around 50% training set with 1000 data points. Furthermore, GP was still improving even at 2000 training data points. The deviation in the LM was also lower after from 30% data set. Under these observations, The LM model indicates better predictive power for moderate data sets when the data is proposed as we did, but not by a considerable margin. The GP algorithm shines when we evaluate the computer times to complete 100 runs for each code. Fig. 7 depicts the elapsed computer time of the training against train data size for three algorithms. On smaller data sets, GP and SC algorithms perform identically, and LM is three times slower. After a 40% training set, i.e., more than 1000 training plus validation points, the GP algorithm begins to shine. The GP algorithm requires less computer time than both LM and SC, still performs comparable to LM, and outclasses SC in performance. The LM is significantly slower when compared to fuzzy logic algorithms but is ahead of the other two algorithms in terms of lowering errors.

4. Conclusion

In this study, three well-known machine learning algorithms were introduced to model mass attenuation coefficients as a function of photon energy and mixture ratios of the vanadium slag/epoxy resin/antimony composite family. All three of the models showed good agreement with XCOM data and simulation results. The SC neuro-fuzzy model performed poorly compared to the other two and needed finer tuning. The GP neuro-fuzzy models showed low errors (RMSE = 0.0001) and low training times, less than 10 s of training time with around a thousand training data. Thus, GP can be used for large data sets training to model μ/ρ of composites. The LM algorithm showed great predictive power with MAPE down to 0.05%, yet it was three times slower than GP. LM can be used to model μ/ρ data, bearing the computer time in mind.

It is shown that machine learning based approaches are successful in modeling μ/ρ given enough training data, predicting the mass attenuation data down a 1 part in 10 thousand. It should be

kept in mind that in lower energies, especially in the vicinity of absorption edges, finer energy resolution of training data may be needed. Machine learning algorithms also offer flexible input types, supporting non-conventional inputs, like ratios of compounds in a mixture, rather than a complete chemical composition. They can also be used to optimize composites with different physical properties.

Future works are encouraged with other types of machine learning methods and large preferably experimental data sets to model shielding sufficiency, with μ/ρ, and some other parameter, for composites formed with arbitrary elements, such as C, H, O, and N as well as various metal oxides. It is also possible to get multiple outputs, some other parameters like individual contributions from different mechanisms. Once trained adequately, the model can determine the photon-matter interaction of composites in a comprehensive energy range alternative to other time-consuming and tricky methods.

All in all, the developed GP model and LM models accurately estimated mass attenuation values of a composite family. Therefore, they can be considered as candidates for machine learning models for radiation protection.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J.K. Shultis, R.E. Faw, Radiation Shielding and Radiological Protection in Handbook of Nuclear Engineering, Springer, 2010.

[2] J.H. Hubbell, Review and history of photon cross section calculations, 13, Phys. Med. Biol. 51 (2006) 245, https://doi.org/10.1088/0031-9155/51/13/R15.

[3] H.O. Tekin, P.S. Vishwanath, T. Manici, E.E. Altunsoy, Validation of MCNPX with experimental results of mass attenuation coefficients for cement, gypsum and mixture, Journal of Radiation Protection and Research 42 (3) (2017) 154–157. https://doi.org/10.14407/jrpr.2017.42.3.154.

[4] V.P. Singh, S.P. Shirmardi, M.E. Medhat, N.M. Badiger, Determination of mass attenuation coefficient for some polymers using Monte Carlo simulation, Vacuum 119 (2015) 284–288, https://doi.org/10.1016/j.vacuum.2015.06.006.

[5] M. Berger, XCOM: Photon Cross Sections Database, 2010, https://doi.org/10.2172/6016002.

[6] O. Klein, Y. Nishina, Uber die Streuung von Strahlung durch freielektronen nach der neuen relativistischen quantendynamik von Dirac, Z. Phys. 52 (1928) 853–868.

[7] J. Hamilton, I. Øverbö, I, B. Tromborg, Coulomb corrections in non-relativistic scattering, Nucl. Phys. B 60 (1973) 443–477.

[8] R.H. Pratt, P.M. Bergstrom Jr., L. Kissel, New Relativistic S-Matrix Results for Scattering beyond the Usual Anomalous Factors/beyond Impulse Approximation. No. UCRL-JC-114583, Lawrence Livermore National Lab., CA (United States), 1993. CONF-9208186-5.

[9] H.B. Kavanoz, Ö. Akçalı, O. Toker, B. Bilmez, M. Çağlar, O. İçelli, O, A novel comprehensive utilization of vanadium slag/epoxy resin/antimony trioxide ternary composite as gamma ray shielding material by MCNP 6.2 and BXCOM, Radiat. Phys. Chem. 165 (2019) 108446, https://doi.org/10.1016/j.radphyschem.2019.108446.

[10] S. Alp, T. Ozkan, Modelling of multi-objective transshipment problem with fuzzy goal programming, International Journal of Transportation 6 (2018) 9–20, https://doi.org/10.14257/ijt.2018.6.2.02.

[11] F. Rosenblatt, Principles of Neurodynamics. Perceptrons and the Theory of Brain Mechanisms, Cornell Aeronautical Lab Inc. Buffalo, NY, 1961.

[12] A.S. Lapedes, R.M. Farber, How Neural Nets Work. Neural Information Processing Systems, 1988, pp. 442–456.

[13] A.G. Bakirtzis, J.B. Theocharis, S.J. Kiartzis, K.J. Satsios, Short term load forecasting using fuzzy neural networks, IEEE Trans. Power Syst. 10 (3) (1995) 1518–1524.

[14] G. Zhang, B.E. Patuwo, M.Y. Hu, Forecasting with artificial neural networks: the state of the art, Int. J. Forecast. 14 (1) (1998) 35–62.

[15] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford university press, 1995.

[16] J.C.F. Pujol, J.M.A. Pinto, A neural network approach to fatigue life prediction, Int. J. Fatig. 33 (3) (2011) 313–322.

[17] I. Akkurt, C. Başyigit, S. Kilincarslan, A. Beycioglu, Prediction of photon attenuation coefficients of heavy concrete by fuzzy logic, J. Franklin Inst. 347 (9) (2010) 1589–1597.

[18] E.E. Zadeh, S.A.H. Feghhi, G.H. Roshani, A. Rezaei, Application of artificial neural network in precise prediction of cement elements percentages based on the neutron activation analysis, The European Physical Journal Plus 131 (5) (2016) 167.

[19] N. Kucuk, Modeling of gamma ray energy-absorption buildup factors for thermoluminescent dosimetric materials using multilayer perceptron neural network: a comparative study, Radiat. Phys. Chem. 86 (2013) 10–22, https://doi.org/10.1016/j.radphyschem.2013.01.021.

[20] O. Gencel, The application of artificial neural networks technique to estimate mass attenuation coefficient of shielding barrier, 12, Int. J. Phys. Sci. 4 (2009) 743–751.

[21] A. Yadollahi, et al., Application of artificial neural network for predicting the optimal mixture of radiation shielding concrete, Prog. Nucl. Energy 89 (2016) 69–77, https://doi.org/10.1016/j.pnucene.2016.02.010.

[22] F.H. Attix, Introduction to Radiological Physics and Radiation Dosimetry, John Wiley & Sons, 2008.

[23] H. Bethe, W. Heitler, On the stopping of fast particles and on the creation of positive electrons, Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character 146 (856) (1934) 83–112, https://doi.org/10.1098/rspa.1934.0140.

[24] The Math Works, Inc. MATLAB. Version 2020b, The Math Works, Inc., 2020 computer software, https://www.mathworks.com.

[25] L.A. Zadeh, 3, Fuzzy sets, Information and control 8 (1965) 338–353, https://doi.org/10.1016/S0019-9958(65)90241-X.

[26] M. Sugeno, G.T. Kang, Structure identification of fuzzy model, Fuzzy Set Syst. 28 (1) (1988) 15–33, https://doi.org/10.1016/0165-0114(88)90113-3.

[27] L.J. Herrera, et al., Clustering-Based TSK neuro-fuzzy model for function approximation with interpretable sub-models, in: International Work-Conference on Artificial Neural Networks, Springer, Berlin, Heidelberg, 2005, https://doi.org/10.1007/11494669_49.

[28] J.J. Moré, The Levenberg-Marquardt algorithm: implementation and theory, in: Numerical Analysis, Springer, Berlin, Heidelberg, 1978, pp. 105–116.

[29] A. Davydenko, R. Fildes, Forecast error measures: critical review and practical recommendations, in: Business Forecasting: Practical Problems and Solutions, 34, Wiley, 2016.

[30] P. Goyal, P. Dollar, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, K. He, Accurate, Large Mini Batch Sgd: Training Image Net in 1 Hour, 2017 arXiv preprint arXiv:1706.02677, https://arxiv.org/abs/1706.02677v2.

[31] M.E. Medhat, Application of neural network for predicting photon attenuation through materials, 3-4, Radiat. Eff. Defect Solid 174 (2019) 171–181, https://doi.org/10.1080/10420150.2018.1547903.