Research Article GU J Sci 38(2): x-x (2025) DOI: 10.35378/gujs.1521834

Gazi University

Journal of Science

http://dergipark.gov.tr/gujs

Deep Learning Application for Milne Problem with Linear Anisotropic Scattering

R. Gokhan TURECI*

Kırıkkale University, Kırıkkale Vocational High School, 71450 Kırıkkale, Türkiye

Highlights

- This paper focuses on a deep learning application to the extrapolation distance data.
- The ANN model is studied with two different independent variables and one dependent variable.
- Although the number of data is small, the predictions of the ANN model are quite good.

Article Info

Received: 24 July 2024
Accepted: 3 Dec 2024

Keywords

Deep learning
Milne problem
Linear anisotropic scattering
Artificial neural networks

Abstract

Using deep learning algorithms, which is a sub-branch of artificial intelligence, in this study, a deep learning model is developed according to both the number of secondary neutrons and the linear anisotropic scattering coefficient. These are independent variables that the dependent variable is the extrapolation distance. The training data set was calculated with H_N method. ANN algorithm was written by TensorFlow and Keras which are the modules in Python programming language. The performance of the deep learning model for this problem has high performance so that the predicted new data which doesn't be in the training data set is reliable according to the success of the model.

1. INTRODUCTION

From translation programmes to image processing, artificial intelligence has started to enter our daily lives effectively. In this study, an artificial intelligence model for the Milne problem has been created using deep learning algorithm and Python modules over an existing data set. The importance of the study can be expressed as the machine learning over the existing data and then predicting the data through this model. While solutions of the neutron transport equation have been obtained with many numerical or semi-analytical methods, sometimes with great effort, creating an artificial intelligence model over the existing data will provide great advantages especially in terms of reactor physics.

Milne's problem is a classical problem of astrophysics, the aim of which is to determine the extrapolation distance at which the neutral particle flux, in the original problem the photon flux, drops to zero. In this way, information about the atmospheric structure that causes the attenuation of the photon flux is obtained together with spectroscopic observations.

From the point of view of reactor physics, Milne's problem is important to determine where the neutron flux generated by neutrons emitted from a neutron source at infinite distance becomes zero, since the neutron is an uncharged particle.

Placzek and Seidel [1] investigated the problem for vacuum boundary conditions, and Noble [2] studied the problem by using the Wiener-Hopf method. The problem was solved for isotropic and linear anisotropic

*e-mail: tureci@gmail.com

scattering by Case and Zweifel [3], McCormick [4] and McCormick and Kuščer [5]. C_N method was applied to examine the problem by Kavenoky [6]. The problem studied with the singular integral equation by Sahni and Kumar [7]. Atalay [8, 9] solved the problem for a specular reflecting boundary with linear anisotropic and for specular and diffuse reflecting boundary with linear anisotropic scattering, respectively. Slama et al. [10] Hussein and Selim [11, 12] and were studied the problem in stochastic media by RVT technique.

Sahni et al. [13] recently developed the partial range orthogonality for the Case method solution and Ganapol [14, 15] also recently developed Lagrange interpolation techniques again for the Case method solution. Therefore, Case method for the one-speed, homogenous medium and plane geometry is a powerful method that both F_N method which was developed by Siewert [16] and H_N method which was developed by Tezcan et al. [17] basis on the Case method.

Gülderen et al. [18] recently investigated the problem for linear-quadratic anisotropic scattering with H_N method. Another recently method was examined by Türeci [19] for Anlı-Güngör scattering function. The training data set of this study is based on the calculations with H_N method for linear anisotropic scattering.

In the next section, a very summary of the calculation of the extrapolation distance using the HN method is given. For a more detailed review of the application of the H_N method to the Milne problem, we refer to the work of Gülderen and Türeci. In the next section, a brief preliminary information about deep learning is given. In the next section, necessary information about the deep learning algorithm used in this study and the results are given. The results obtained are quite successful despite working with two independent variables, c and f_1, at the same time and using a relatively small deep learning algorithm.

2. MATERIAL METHOD

Python programming language is used in the deep learning application. Python has extremely powerful modules particularly in deep learning applications. Numpy [20], pandas [21], seaborn [22] and matplotlib [23] modules can be called as the basic modules in any numerical study with Python. Basic mathematical operations and linear algebra processes is studied with numpy module. Transferring any MS Excel file or csv file from the outside to the Python environment or sending these files to the outside or operations that can be done on these files in the Python environment can be provided with the pandas module. One of the most important stages of machine learning is highly dependent on the structure of the data set. Missing values should be carefully identified and replaced with new data, e.g. the average value of the data set, in a way that does not distort the data set. Otherwise, the machine learning process will become very difficult or even may not be realised. The pandas module and the seaborn module provide us with practical ways to perform such operations, so much so that the seaborn module is again a very powerful module for statistical operations.

However, the closer the data set is to a Gaussian distribution, the more successful the learning is. Therefore, if there are skewness conditions in the data set, these should be determined. If there are skewness conditions in the data set, that is, if the data set is distorted in a way that does not fit the Gaussian distribution, these skewness conditions should be removed by taking logarithm or applying Yeo-Johnson transformation [24]. This process is also an important condition that facilitates the learning process.

In another case, if there are outlier data in the data set, eliminating these outlier data by methods such as IRQ is also one of the important issues. IRQ technique is a statistical method applied on the candlestick chart of the data set. Because outlier data negatively affect the learning success. These operations can be done with the seaborn module, which can work with the matplotlib module used to draw graphics in Python language.

2.1. A Brief of Milne Problem with H_N Method

The one-speed, time independent, homogenous medium and slab geometry neutron transport equation is given as

$$\mu\frac{\partial\Psi(x,\mu)}{\partial x}+\Psi(x,\mu)=\frac{c}{2}\int_{-1}^{1}f(\mu,\mu')\Psi(x,\mu')d\mu'. \tag{1}$$

Here x and μ correspond to the spatial and the angular variables, respectively. $\Psi(x,\mu)$ is the neutron flux at x point and μ direction, and $f(\mu,\mu')$ defines the scattering properties of neutrons after interaction the medium material. It is assumed as linear anisotropic scattering:

$$f(\mu,\mu')=1+3f_1P_1(\mu)P_1(\mu'),\quad f_n=0,\quad n\geq 2. \tag{2}$$

The medium is placed $x>0$ region. The left side of the medium is vacuum region. The intersection surface, boundary of the medium, is placed at $x=0$, Neutrons diffuse this medium from a source at $x=+\infty$. The boundary condition is given as

$$\Psi(0,\mu)=0 \text{ for } \mu>0. \tag{3}$$

This condition implies that there is no incoming neutron from vacuum to medium. The asymptotic solution of Equation (1) is given for this problem as

$$\Psi(x,\mu)=\phi(-\nu_0,\mu)e^{x/\nu_0}+A(\nu_0)\phi_{lq}(\nu_0,\mu)e^{-x/\nu_0}+\int_0^1 A(\nu)\phi(\nu,\mu)e^{-x/\nu}d\nu. \tag{4}$$

The neutron density is given by

$$\rho_{as}=2\pi\int_{-1}^{1}\left[\phi(-\nu_0,\mu)e^{x/\nu_0}+A(\nu_0)\phi(\nu_0,\mu)e^{-x/\nu_0}\right]d\mu, \tag{5}$$

such that this expression must be zero at the point $x=d$ for the neutron density to be zero. ρ_{as} equal to the zero for $x=d$ which is called as the extrapolation distance. Therefore, the extrapolation distance is found as

$$d=-\left[\frac{\nu_0}{2}\ln A(\nu_0)\right]. \tag{6}$$

2.2. Application of Deep Learning and its Results

In this study, extrapolation distances calculated for different values of c using the H_N method for the linear anisotropic scattering case are used and this data set is our training data set. The linear anisotropic scattering coefficient for different c values are calculated in the range [-0.3, 0.3] with a step interval of 0.02. Secondary neutron number and linear anisotropic scattering coefficient were kept as independent variables, while extrapolation distance was taken as the dependent variable (which is called as target in machine learning notation).

Our first job is to transfer the dataset to the Python environment and determine the outlier data in the dataset and the skewness in the dataset. The next step is to look at the correlation of the variables. This process is one of the important steps in terms of determining the relationship between variables and the deep learning algorithm model to be used. In the analysis, it is understood that there is no outlier data in our data set. The skew values of data set are given in Table 1. These values are so small to handle them. Therefore, any transform process was not applied to our data set.

Table 1. *The skew values of the data*

Variable	Skew values
The secondary neutron number, c	0.00
The scattering coefficient, f_1	0.00
The extrapolation distance, z	0.18

The next step is to prepare the neural network. We use more than one neural network and therefore the algorithm we use is the deep learning algorithm. This term is closely related to the capacity of computer systems. Perhaps 5 or 10 years before today, a two or three -layer neural network is referred to as deep learning, while larger neural networks can be established with new generation of machines today.

However, there are three cases closely related to the data set to be subjected to deep learning process. If the deep learning algorithm is chosen very small or too large than the usual, the results are quite bad, and the learning process is extremely unsuccessful. This situation is called as underfitting in the machine learning notation. A second situation is the so -called fitting with machine learning notation. Fitting is required for quite general problems. This is average success on average data and is generally accepted for many situations in daily life. The latest situation is the memorization of artificial intelligence and is called overfitting. Since the result that can be achieved in a scientific study will always represent a single truth, we want to obtain a deep learning model that has memorized for a particular problem. One way to understand this will be the R-Squared value that is used with Sklearn [25] module. This module is a powerful module in terms of studying artificial intelligence for example min-max scaler transformation which converts data in [0,1] that this option increases the success of the deep learning, another useful sub-method is train test split operation which randomly splits data to train data set and test data set. Moreover, we can adjust the degree of this split process. Train-test split degree was selected as 20% in this study.

Deep learning algorithms include some hyperparameters. These parameters are called hyperparameter because there is no rule for the values or conditions of these hyperparameters and depends entirely on the condition of the data set. The hyperparameters used in this study are given in Table 2 with together with the success of our deep learning model.

A neural network consists of input layer, hidden layers and output layer. The number of these hidden layers determines the size of the neural network and consists of neurons. If a neuron meets the threshold value determined by the activation function, it triggers the next neuron. In this study, ReLu [26] is used as the activation function, and it is one of the hyperparameters. The connection of the neurons is chosen as the Sequential model of the Keras [27] module, so that all neurons forming the network are interconnected from layer to layer and the number of neurons is another hyperparameter. The sequential model is often used in numerical calculations.

In the process called forward propagation, weights are determined from the input layer to the output layer and the value produced by using these weights is compared with the test value and the difference between the produced value and the test is determined. At this stage, another hyperparameter called metric is used. For the optimization of the results, the optimization function is used, and the back propagation algorithm returns to the first hidden layer, the whole process is repeated with the number of iterations called epochs. The optimisation function is chosen as the most widely used adam function [28]. Another hyperparameter is called as learning rate that determines the performance of model. The learning rate is also important to prevent the model from over-fitting to training data or failing to learn sufficiently.

3. THE RESEARCH FINDINGS AND DISCUSSION

Machine learning applications with different algorithms and implementations have been recently studied by Zheng et al. [29], Xie et al. [30], Whewell and McCLarren [31], Zolfaghari et al. [32] and Türeci [33].

Table 2. *The used hyperparameters and the performance of deep learning for second application*

Hyperparameters	Neuron number	300
	Epoch	600
	Batch number	4
	Test size	%20
	Validation split	%20
	Number of hidden layers	9
	Keras model	Sequential
	Activation function	ReLu
	Optimizer	Adam
	Learning rate	0.001
	Loss function	Mse
Performance of the model		
	Training time (in sec.)	45.2307
	R2 value	0.9999554

Figure 1 is a diagram for the used neural network. c and f_1 correspond to the independent variables and the extrapolation distance, z, is the dependent variable. This neural network is run with the parameters in Table 2. Figure 2 shows the behaviour of the training data set including both two independent variables and the dependent variable. After this model executed, we have the artificial neural network model. Therefore, we can use this model to predict the new data. Figures 3a and 3b show the predicted data with together the training data by different points of view. Figure 4 represents the error or the loss in machine learning notation.

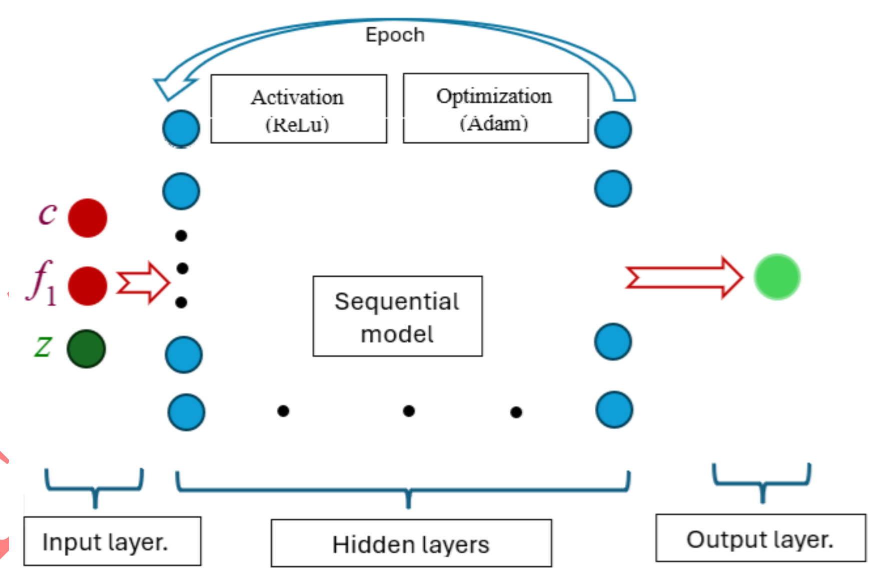

Figure 1. *A representative representation of the neural network used in this study*

Algorithm 1. *The steps of the deep learning application*

1. importing Python modules
2. importing data
3. data analysis
   - looking for NaN (not a number) values with pandas
   - search the outliers with seaborn

- search the skewness with seaborn
4. combine the variables (c and f_1) with numpy
5. applying deep learning
   - creating the neural network with hyperparameters
   - feeding the neural network
   - getting the results and predictions
   - determining the loss value and loss graph
6. predictions
   - R-squared value and the success of the model
   - Predictions of new data

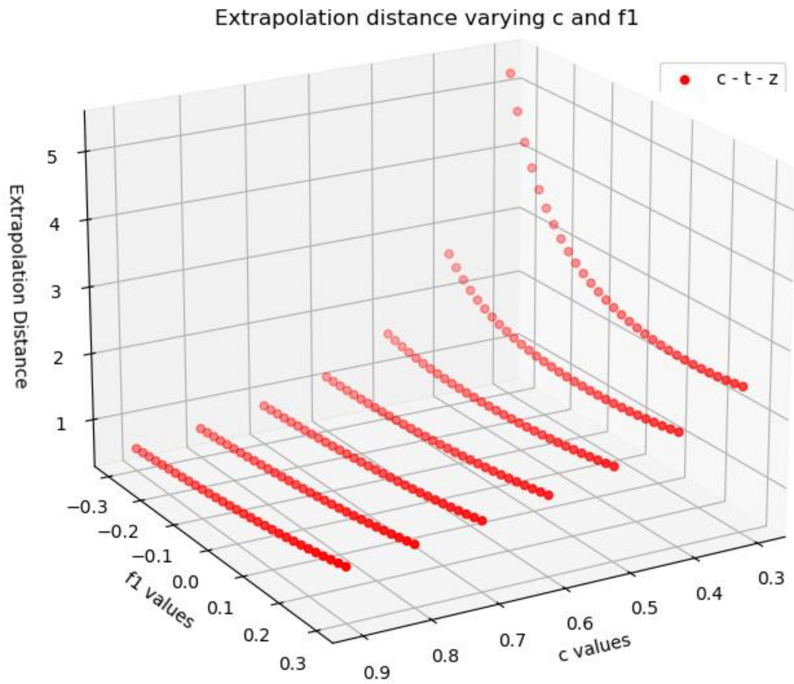

Figure 2. *The behaviour of training data set*

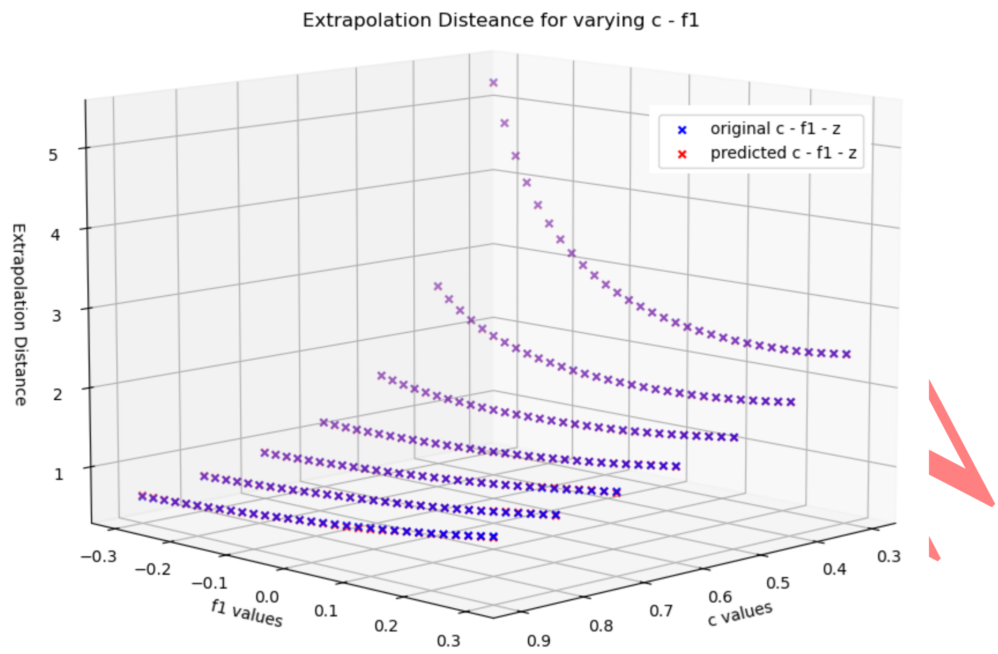

Figure 3a. *The training data set and the predicted results with deep learning model*

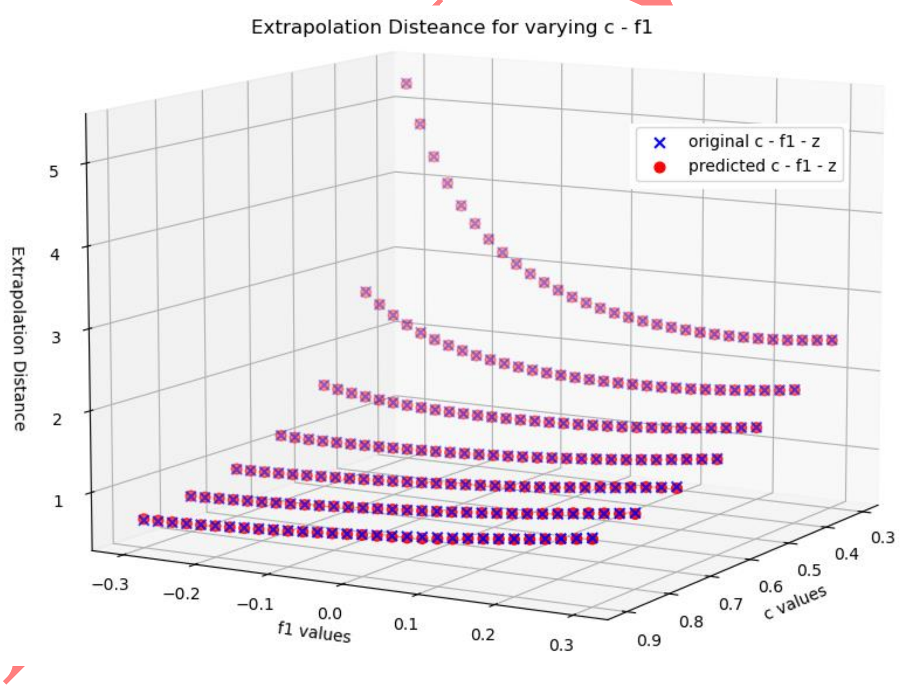

Figure 3b. *Another design of Figure 3a*

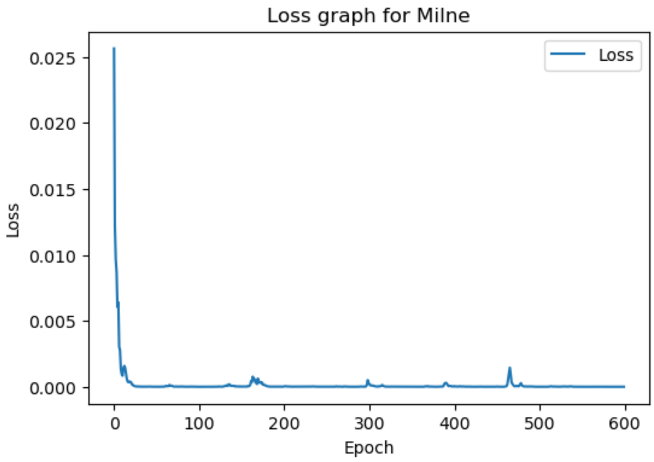

Figure 4. *The loss graph of deep learning algorithm*

4. RESULTS

In this study, a deep learning algorithm was run with secondary neutron number and linear anisotropic scattering coefficient as independent variables and extrapolation distance as the dependent variable. The values used as training data set are the values calculated by H_N method.

The hyperparameters used in the deep learning algorithm are given in Table 2 and their representation is given in Figure 2. In this study, artificial intelligence was applied to the data available in the literature and no analytical method was applied. The R-squared value indicating the success of the model is quite high and close to unity. Hyperparameters vary according to the data set and are determined by trial and error. Therefore, although it depends on the computer system used, the hyperparameters given in this study may be a clue for other researchers for different applications. But more importantly, the fact that we have run a neural network according to two independent variables in this study and obtained very successful results may be useful for real-time reactor physics applications.

CONFLICTS OF INTEREST

No conflict of interest was declared by the author.

REFERENCES

[1] Placzek, G., and W. Seidel., “Milne's Problem in Transport Theory”, Physical Review, 72: 550-555, (1947).

[2] Noble, B., The Wiener-Hopf Technique, Pergamon Press, Oxford, (1958).

[3] Case, K.M. and Zweifel, P.F., Linear Transport Theory, Massachusetts: Addition-Wesley, (1967).

[4] McCormick, N.J., “One-Speed Neutron Transport Problems in Plane Geometry”, PhD thesis, The University of Michigan, (1964).

[5] McCormick N. J., and Kuščer, I., “Half-Space Neutron Transport with Linearly Anisotropic Scattering”, Journal of Mathematical Physics, 6 (12): 1939-1945, (1964).

[6] Kavenoky, A., “The C_N Method of Solving the Transport Equation: Application to Plane Geometry”, Nuclear Science and Engineering, 65, 209, (1978).

[7] Sahni, D.C., and Kumar, V., “Numerical solution of singular integral equations of neutron transport problems”, Transport Theory and Statistical Physics, 16(7): 959-978, (1987).

[8] Atalay, M. A., “Milne problem for linearly anisotropic scattering and a specularly reflecting boundary”, Annals of Nuclear Energy, 27(16): 1483-1504, (2000).

[9] Atalay, M. A., “Milne problem for linearly anisotropic scattering and a specular and diffuse reflecting boundary”, Jornal of Quantative Spectroscopy and Radiative Transfer, 72(5): 589-606, (2002).

[10] Hussein, A., and Selim, M.M., “A general analytical solution for the stochastic Milne problem using Karhunen–Loeve (K–L) expansion”, Journal of Quantitative Spectroscopy and Radiative Transfer, 125: 84-92, (2013).

[11] Slama, H., El-Bedwhey, N.A., El-Depsy, A. and Selim, M. M., “Solution of the finite Milne problem in stochastic media with RVT Technique”, The European Physical Journal Plus, 132: 505, (2017).

[12] Hussein. A., and Selim, M. M., “A complete probabilistic solution for a stochastic Milne problem of radiative transfer using KLE-RVT technique”, J Jornal of Quantative Spectroscopy and Radiative Transfer, 232: 54-65, (2019).

[13] Sahni, D. C., Türeci, R. G., and Bozkir, A. Z. “Partial Range Completeness of Case Eigenfunctions and Numerical Solution of Singular Integral Equations of Particle Transport Problems”, Journal of Computational and Theoretical Transport. 49(7): 349–367, (2020).

[14] Ganapol, B. D., “The Ln/Ln0 Method for 1D Neutron Transport in a Slab Medium”, Journal of Computational and Theoretical Transport, 51(5): 239-264, (2022).

[15] Ganapol , B. D., “Numerical Caseology by Lagrange Interpolation for the 1D Neutron Transport Equation in a Slab”, Nuclear Science and Engineering, 197(1): 1-13, (2023).

[16] Siewert, C. E., “The FN method for solving radiative-transfer problems in plane geometry”, Astrophysical Space Science. 58 (1): 131–7, (1978).

[17] Tezcan, C., Kaşkaş, A., and Güleçyüz, M. Ç., 2003. “The HN method for solving linear transport equation: Theory and applications”, Jornal of Quantative Spectroscopy and Radiative Transfer, 78 (2): 243-254, (2003).

[18] Gülderen, D., Sahni, D. C., Türeci, R. G., and Aydın, A.,“The Milne Problem for Linear-Triplet Anisotropic Scattering with HN Method”, Journal of Computational and Theoretical Transport, 51(6): 329-353, (2022).

[19] Türeci, R. G., “The Milne Problem with the Anlı-Güngör Scattering”, Journal of Computational and Theoretical Transport, 51(6): 354-371, (2022).

[20] Numpy, https://numpy.org/. Access date: 22.07.2024

[21] Pandas, https://pandas.pydata.org/. Access date: 22.07.2024

[22] Seaborn, https://seaborn.pydata.org/. Access date: 22.07.2024

[23] Matplotlib, https://matplotlib.org/. Access date: 22.07.2024

[24] Yeo, I. K., and Richard A. J., “A New Family of Power Transformations to Improve Normality or Symmetry”, Biometrika, 87 (4): 954–959, (2000).

[25] Sklearn, https://scikit-learn.org/stable/. Access date: 22.07.2024

[26] Nair, V., and Hinton, G. E., “Rectified Linear Units Improve Restricted Boltzmann Machines”, Conference: Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, Haifa, Israel, (2010).

[27] Keras, https://keras.io/. Access date: 22.07.2024

[28] Diederik P. K., Jimmy B.,“Adam: A Method for Stochastic Optimization”, Computer Science > Machine Learning, arXiv:1412.6980, (2014).

[29] Zheng C., Liub L., and Muc L., “Solving the linear transport equation by a deep neural network approach”, Journal of Discrete and Continuous Dynamical System-S, 15(4): 669-686, (2021).

[30] Xie Y., Wang Y., Ma Y., and Wu Z., “Neural Network Based Deep Learning Method for Multi-Dimensional Neutron Diffusion Problems with Novel Treatment to Boundary”, Journal of Nuclear Engineering, 2: 533-552, (2021).

[31] Whewell, B., and McClarren, R. G., “Data reduction in deterministic neutron transport calculations using machine learning”, Annals of Nuclear Energy, 176, 109276, (2022).

[32] Zolfaghari M., Masoudi S. F., Rahmani F., Fathi A., “Thermal neutron beam optimization for PGNAA applications using Q-learning algorithm and neural network”, Scientific Reports, 12, 8635, (2022).

[33] Türeci R. G., “Machine Learning Applications to the One-speed Neutron Transport Problems”, Cumhuriyet Science Journal, 43(4): 726-738. (2022).