NENE 2022
September 12-15
PORTOROŽ
SLOVENIA
31st International Conference Nuclear Energy for New Europe

APPLICATION OF MACHINE LEARNING TO INTERPOLATION AND EVALUATION OF DELAYED-NEUTRON MULTIPLICITY

Abdoulaye SAKHO
ENSIIE
1 Rue de la Résistance, 91000, Évry-Courcouronnes, France
abdoulaye.sakho@ensiie.fr

Ivan Kodeli
UKAEA, Culham Centre for Fusion Energy
Abingdon, OX14 3DB, United Kingdom
ivan.kodeli@ukaea.uk

Pierre Dossantos-Uzarralde
CEA, DAM
DIF 91297, Arpajon Cedex, France
pierre-jacques.dossantos-uzarralde@ensiie.fr

ABSTRACT

Traditionally, nuclear data evaluations are based on experimental data and nuclear model calculations using data assimilation methods and expert judgment. In this work an alternative approach using machine learning algorithms is tested on an example of computing the delayed-neutron multiplicity ($\overline{\nu}_d$) of neutron induced fission. Several machine learning models are used before being compared with each other. Then the correlations between different isotopes and energy ranges were deduced from the predictions of the best fitted algorithm and compared to reference results.

The GEF code is compared to experimental data from EXFOR database, then used to feed the machine learning models. GEF calculates the multiplicity of delayed-neutron as a function of input model parameters which are randomly varied according to the assigned standard deviation levels.

1 INTRODUCTION

In 2017 to 2019 studies ([1][2]) the Bayesian approach was used to derive the variance-covariance of the delayed fission yields of actinides based on theoretical model implemented in the GEF code. In this paper we present a novel approach using machine learning models to compute the average number of delayed neutrons per fission ($\overline{\nu}_d$).

The paper is divided in five chapters. In chapter 2, the process of the simulations performed with the GEneral Description of Fission Observables(GEF [3][4]) code is detailed. The simulations have been carried out on neutron-induced fission from 10^{-8} to 14 MeV by changing some parameters each time for ^{235}U, ^{238}U and ^{239}Pu. Chapter 3 presents the machine learning algorithms used in this study. In chapter 4, the performances of the different models are analyzed before calculating the correlations between the different models using the best model obtained. Chapter 5 concludes the outcomes.

2 GENERAL DESCRIPTION OF FISSION OBSERVABLES

2.1 GEF code

Delayed neutrons [3] are neutrons emitted after a nuclear fission event by one of the fission products (or fission product daughter after β^- decay), at times ranging from a few milliseconds to a few minutes after the fission event. Neutrons born within 10^{-14} seconds after the fission are referred to as prompt neutrons.

GEF code [3] calculates pre-neutron and post-neutron fission-fragment nuclide yields, angular momentum distributions, isometric yields, prompt-neutron yields and prompt-neutron spectra, prompt-gamma spectra, and several other quantities for a wide range of fissioning nuclei in spontaneous fission and neutron-induced fission. Output consists of fission observable on an event-by-event basis.

For the purpose of this work, GEF code's ability to compute elements on delayed neutrons processes is relevant. Furthermore a special version of GEF, enabling changes of some model parameters on the input, was kindly provided by the main author of the code solely for the purposes of this study. Fig.1 shows the comparison between the GEF results and experimental data extracted from the EXFOR data base. Experimental data include the associated uncertainties.

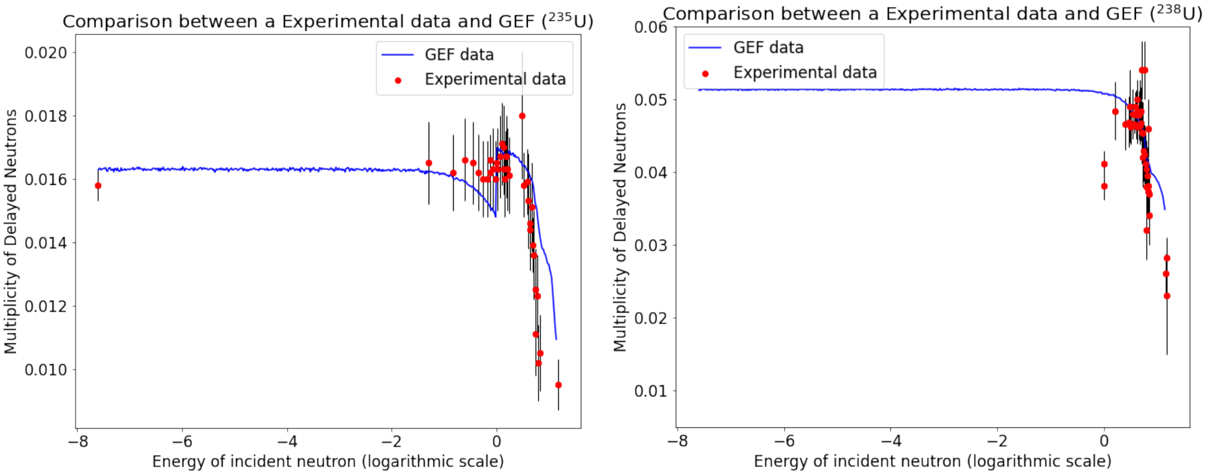

Figure 1: Comparison between GEF results and experimental data for ^{235}U (left) and ^{238}U (right).

2.2 Mathematical structures and simulations

Two mathematical structures are studied. To each of these structures, the same machine learning models were applied.

1. **Primary structure (PS)**: for a given isotope, we first use machine learning models to predict the multiplicity of delayed-neutron based on the energy of the incident neutron only:

$$\hat{Y} = f(E) \tag{1}$$

$$\begin{cases} E\ the\ energy\ of\ the\ incident\ neutron \\ \hat{Y}\ the\ estimator\ of\ the\ target\ \textit{variable}\ (\overline{\nu}_d) \\ f\ the\ model\ used. \end{cases}$$

For this purpose, these models were fed with experimental data from the EXFOR database as well as GEF-simulated data. GEF calculations were done for 500 energy points in the interval between $[2.53 \times 10^{-8}\,\mathrm{MeV}; 14\,\mathrm{MeV}]$ with a logarithmic step. Only default/reference parameters were used in these simulations. For a given energy, GEF computed 13 values of the multiplicity of delayed neutrons, making in total 6 500 simulations for each isotope. This data are referred to as Y_{ref}^{GEF}.

Furthermore, experimental data (Y_{exp}) and their uncertainties were taken as normally distributed around the experimental values.

2. **Advanced structure (AS)**: for a given isotope, we used machine learning models to predict the multiplicity of delayed neutrons as a function of incident neutron energy (E) and GEF code parameters (P_i):

$$\hat{Y} = f(E, P_1, ..., P_p) \tag{2}$$

As above we take 500 energy points for each isotope.For a given energy, 100 values of the multiplicity of delayed neutron were calculated with GEF with different parameter values for each simulation.

The parameters are chosen randomly, following a normal law (the mean value is the reference value of the parameter and the standard deviations were given by the author of the code). All of this data will be used to feed this type of machine learning models. We will have 50 000 simulations for each isotope. This data are are referred to as Y_{param}^{GEF}. Note that the experimental data are not taken into account here since the purpose is to train the model on parameters.

The primary structure is used to compare the accuracy of machine learning with regards to GEF predictions. The advanced structure is used to compute the correlations between energies and isotopes.

3 MACHINE LEARNING ALGORITHMS

Several machine learning algorithms were used [5]. For all of these models we predict a real-valued output Y as a function of an input vector $X^T = (X_1, X_2, ..., X_p)$.

- **The Linear regression** forecasting function has the form followed Eq. (3):

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j \tag{3}$$

Here β_j 's are unknown parameters or coefficients, and the variables X_j are the explanatory variables. Then we have a set of training data $(x_1, y_1)...(x_N, y_N)$ from which to estimate the parameters β. Each $x_i = (x_{i1}, x_{i2}, ..., x_{ip})^T$ is a vector of feature measurements for the i-th case. The common method is least squares, in which we pick the coefficients $\beta = (\beta_0, \beta_1, ..., \beta_p)^T$ to minimize the loss function named the residual sum of squares. The problem is shown on Eq. (4)

$$\hat{\beta} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} (y_i - f(x_i))^2 = \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \right\}. \tag{4}$$

- **The Lasso regression** adds a constraint to the linear regression optimization. Indeed we always build a model with the form of Eq. (3) but the optimization problem (in Lagrangian form) is now Eq. (5):

$$\hat{\beta}^{lasso} = \arg\min_{\beta}\left\{\frac{1}{2}\sum_{i=1}^{i=N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{i=p}|\beta_j|\right\}. \tag{5}$$

  λ is an hyperparameter of the model to optimize. Note that it is necessary to scale the training data before using this model.

- **Ridge regression** is similar to Lasso regression expect that we use a L_2 penalization in the optimization problem. So we find the value of β by solving Eq. (6):

$$\hat{\beta}^{Ridge} = \arg\min_{\beta}\left\{\frac{1}{2}\sum_{i=1}^{i=N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{i=p}(\beta_j)^2\right\}. \tag{6}$$

- **Support Vector Machine regressor** suppose the same shape as linear regression as well (Eq. (3)) but the optimization problem to estimate β is changed:

$$\begin{gathered}\hat{\beta}^{SVM} = \arg\min_{\beta}\left\{\sum_{i=1}^{i=N} V_\epsilon\Big((y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2\Big) + \frac{\lambda}{2}\sum_{j=1}^{i=p}\beta_j^2\right\} \\ \text{where } V_\epsilon = \begin{cases} 0 & \text{if } |r| < \epsilon, \\ |r| - \epsilon & \text{otherwise.}\end{cases}\end{gathered} \tag{7}$$

  λ and ϵ are hyperparameters of the model to optimize.

- **Random Forest regressor**: first several trees are build and then used for the forecasting function.The forecasting function used is:

$$\begin{gathered}\hat{f}_{RF}^{B} = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \\ \text{where B is the number of tree and } T_b, b = 1, .., B \text{ is a tree}\end{gathered} \tag{8}$$

  In fact, the mean of the output of the B trees is used for the forecasting [5].

- **Neural networks** A neural network is a two-stage regression model. The architecture with the best results for our study is the following: as many input $RElu$ neurons are used as the number of feature. Then three hidden layers with respectively 64, 256 and 64 neurons, all of them associated to a $RElu$ activation function. Due to the regression there is only one $linear$ neuron in the last layer [5].

3.1 Metrics used

Several metrics[1] were used to evaluate the performance of the models. We call Y the true value ($\overline{\nu}_d$) of a sample and $\hat{Y}$ the predicted value by a model. So Y can be Y_{exp} or Y_{ref}^{GEF} or Y_{param}^{GEF}, depending on what kind of data is used to train and test the model in consideration.

[1] Best models have the first metric close to 1 and the others close to 0 (minimization).

- Coefficient of determination (R^2) is defined by: $1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$ where $\bar{y}$ is the mean of the y_i.
- Root Mean Squared Error (RMSE): $RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$.
- Mean Absolute Percentage Error (MAPE): $MAPE = \frac{1}{N}\sum_{i=1}^{N}|\frac{y_i - \hat{y}_i}{y_i}|$
- Maximum error metric is just the value of the biggest forcasting error.

3.2 Correlations for the advanced structure

Correlations between GEF parameters and the multiplicity for the advanced structure are studied to determine the parameters for the forecasting models. This study is not necessary for the mathematical primary structure because of the fact that this structure does not take into account the GEF parameters.

The most correlated parameters have been selected for the forecasting of the target variable depending on the studied isotope. Here, we took only variables with a correlation superior to 0.1.

4 RESULTS

4.1 Models performance

Whether it is in the case of primary structure or advanced structure, the results between models have the same behavior. Here, only results concerning the primary structure are shown ($Y = Y_{ref}^{GEF}$). The Table 1 shows the results of the primary structure with ^{235}U.

Table 1: Metrics results (using cross validation) for ^{235}U (PS).

	RMSE	**R^2**	**MAPE**	**MAX_err**
Linear regression	0.00039	0.70	0.017	0.0014
Lasso regression	0.00039	0.70	0.017	0.0014
Ridge Regression	0.00039	0.70	0.017	0.0014
Random Forest	0.00011	0.97	0.006	0.0005
SVM	0.00023	0.89	0.010	0.0012
Neural Networks	0.00013	0.97	0.006	0.0011

Random Forest and Neural Networks are the best models for all metrices. Even if Random Forest has better metric results, Neural networks is the model used in the next step of our study because the metric difference with Random Forest is very low (10^{-5} in term of $RMSE$) and neural Networks is a more consistent model.

Fig. 2 illustrates the performance of several machine learning models depending on the percentage of data used to train them.

First, it is interesting to note that the $RMSE$ are small compared to the average value of the multiplicity of each isotope. For example, the mean value of the multiplicity of delayed neutron for ^{235}U is 0.0161 and 0.0506 for ^{238}U. After using more than 25% of the data for training, the RMSE values (except for linear, lasso and ridge regression) are respectively lower than 0.00025 and 0.00020. We make an average error of 1.5% for ^{235}U and 0.4% for ^{238}U.

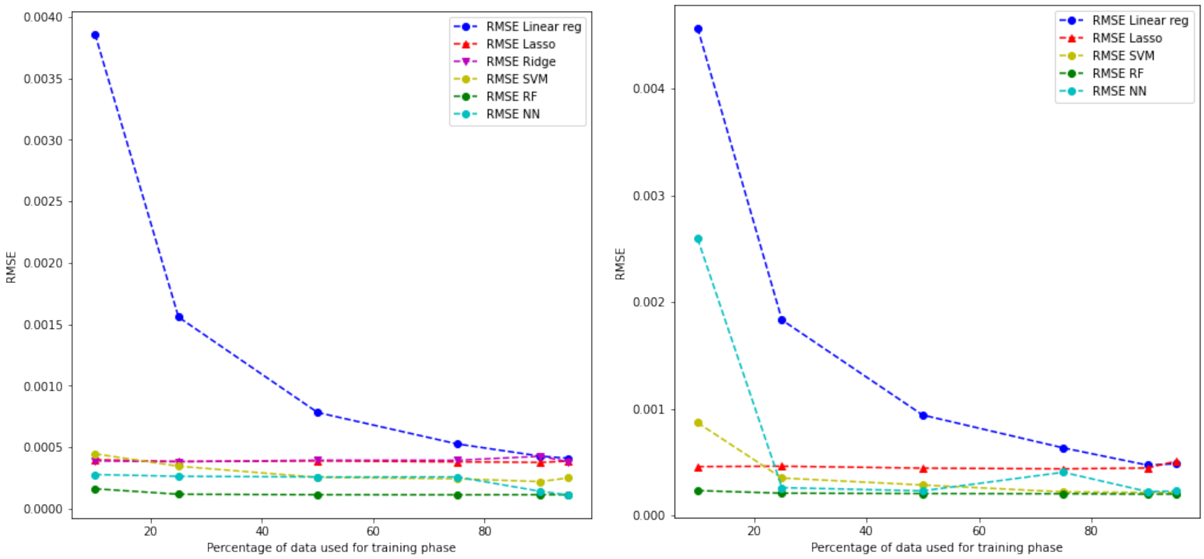

Figure 2: Comparison between machine models for the primary structure (^{235}U on left and ^{238}U on the right)

Then, as said before, using more than 25% of data allowed to obtain reasonable results for the best models. The best models are Random Forest, Support vector Machine and Neural Networks. Neural Networks shows the best performance for this case.

However, it is necessary to specify that increasing the amount of data does not necessarily improve the performance. It is the quality of the new data that matters. Therefore it is why it is possible to see jumps in the value of the $RMSE$ for some models in Fig. 2.

4.2 Primary structure: forecasting of multiplicity of delayed neutrons

After using several metrics to pick the optimal machine learning algorithm, the chosen algorithms are compared to GEF and experimental data in Fig. 3.

First of all, the models trained on Y_{exp} do not have the same behavior as the models trained on Y_{ref}^{GEF}. Both of them seems good but the models trained on experimental data have a better fitting on experimental data.

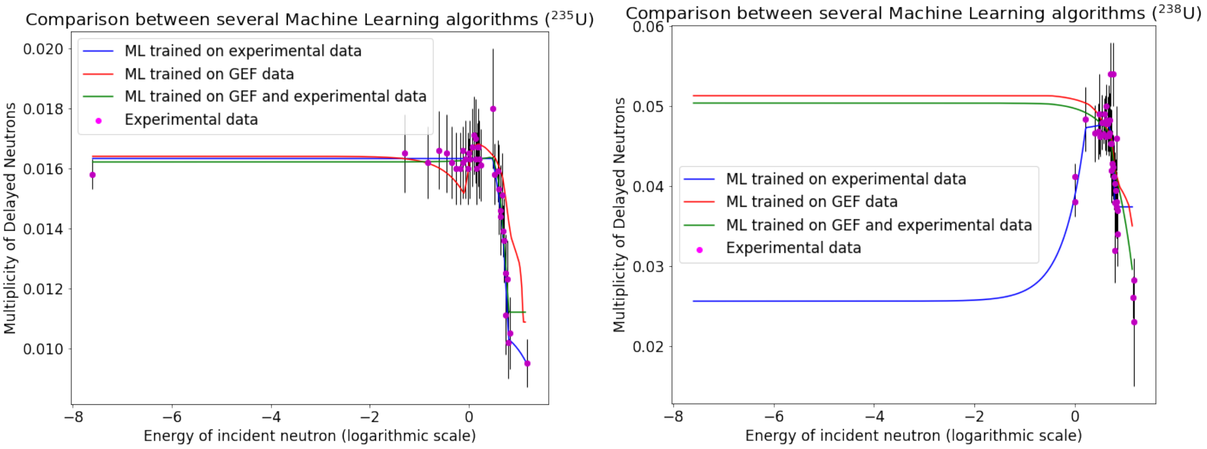

Figure 3: Results for models trained on GEF data and models trained on experimental data

Differences among them are indicators of consistency between the model predictions and experimental measurements. Different behavior at low energies for ^{238}U can be explained by the lack of measured data.

4.3 Advanced structure: Correlations between energies and isotopes

The energies correlations are studied and compared with previous studies [2]. Neural network allowed us to calculate correlation matrices (see fig 4 the energy correlation matrix for 235U). We remind that these matrices contribute to the evaluations made available to users in the libraries of evaluated data Fig. 4

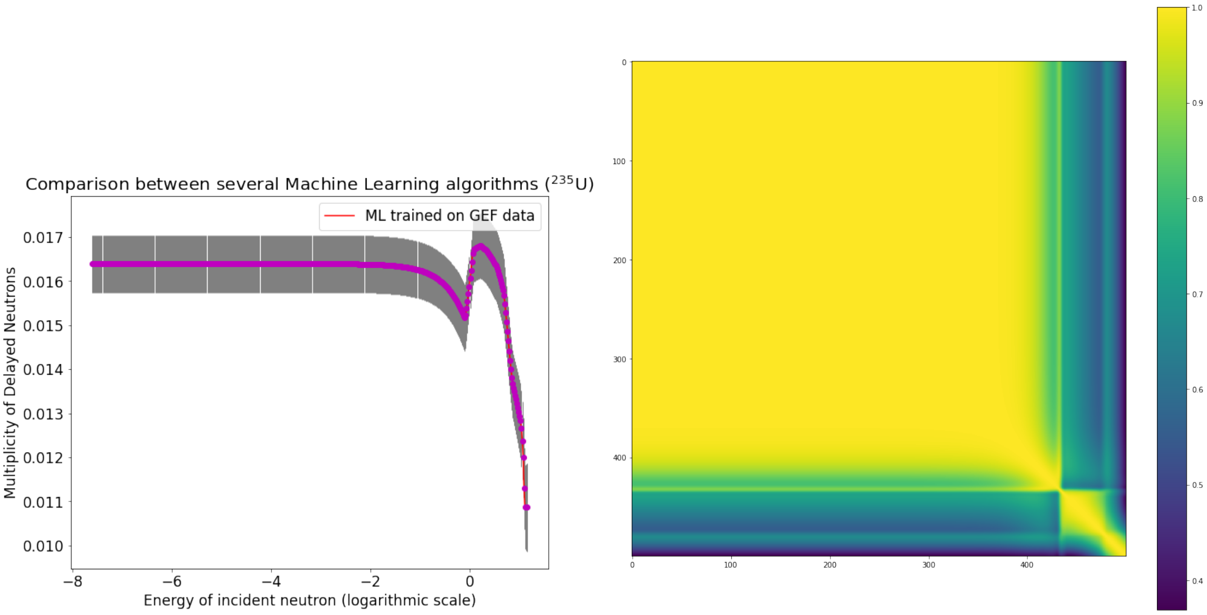

Figure 4: ^{235}U-Energy correlation matrix and uncertainties calculated using neural network training

The correlations between ^{235}U/^{238}U, ^{235}U/^{239}Pu and ^{238}U/^{239}Pu were computed for several energy values thanks to the advanced structure. Our results were compared with previous studies [1].

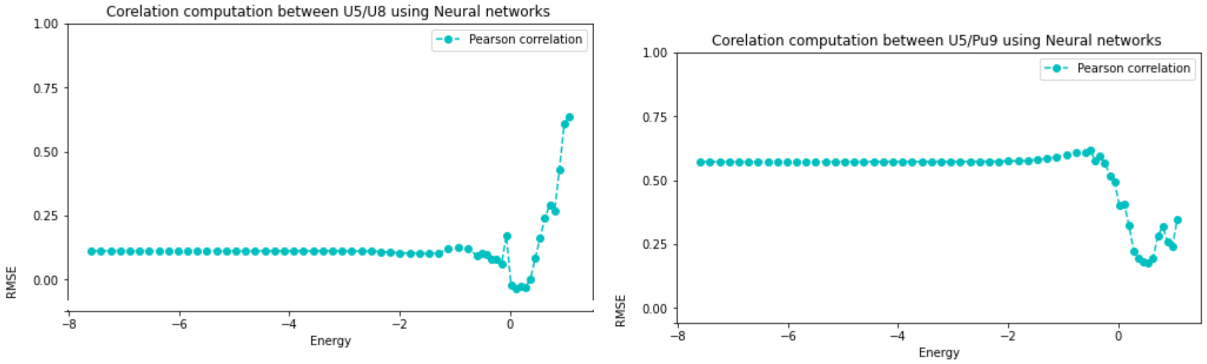

Figure 5: Correlations between ^{235}U/^{238}U (left) and ^{235}U/^{239}Pu (right)

5 CONCLUSIONS

The GEF code and experimental data from the EXFOR database were used to feed machine learning algorithms. The aim was to use these algorithms to forecast the multiplicity of delayed neutrons, compute the covariance matrix for each isotope and compute the correlations of $\overline{\nu}_d$ between isotopes for several energies.

It was proven that machine learning trained on GEF data is able to reproduce with high precision the GEF results. However, we got better fitting results on experimental data with algorithms trained on experimental data.

Comparison between GEF and EXFOR based machine learning predictions provides the information on the consistency between the model predictions and experimental measurements. It can serve to guide the improvement of nuclear models. Most complete predictions are obtained using all the available data, based on experiments and theoretical model, in the machine learning algorithms.

The computation of the correlations between isotopes shown that machine learning is able to replace the existing methods in this field too, even if improvements are possible. For the energy matrix correlations, the results using machine learning are the same as previous studies.

ACKNOWLEDGMENTS

The authors are grateful to Karl-Heinz Schmidt for providing a special version of the GEF code and for his valuable comments and suggestions regarding the use and interpretation of the fission models. We would like to thank Gašper Žerovnik and Klemen Ambrožič for their help. The work was financially supported by the EU SANDA project.

This work has been partly funded by the RCUK Energy Programme [grant number EP/T012250/1] and by several projects within EC fusion programme (EUROfusion 2021 -2022). To obtain further information on the data and models underlying this paper please contact Publications-Manager@ukaea.uk. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

REFERENCES

[1] Solene Tarride, Ivan Kodeli, Pierre Dossantos-Uzarralde, Karl-Heinz Schmidt; ”Mathematical Aspects of Nuclear Data Covariance Matrix Preparation - An Example of Delayed Neutron Data”, Proc. Int. Conf. Nuclear Energy in Central Europe 2017, Portorož, Slovenia, September 11–14, Nuclear Society of Slovenia, 2017, pp. 1-11.

[2] Thibault Cordier, Ivan Kodeli, Pierre Dossantos-Uzarralde, Karl-Heinz Schmidt; ”Estimation of Delayed Fission Yield Covariance Information and Corresponding Uncertainty to Improve Nuclear Data of ^{235}U, ^{238}U, ^{23}Pu and ^{241}Am”, Proc. Int. Conf. Nuclear Energy in Central Europe 2018, Portorož, Slovenia, Sept. 10–13, Nuclear Society of Slovenia, 2018.

[3] Karl-Heinz Schmidt, Beatriz Jurado and Charlotte Amouroux, ”General Description of Fission Observables”, Nuclear Energy Agency, 2014, pp.15-61.

[4] Karl-Heinz Schmidt, Beatriz Jurado, Review on the progress in nuclear fission - experimental methods and theoretical descriptions, Rep. Prog. Phys. 81 106301, 2018,pp. 18-44.

[5] Trevor Hastie, Robert Tibshirani, Jerome Friedman; The Elements of Statistical Learning, Simo Puntanen, 2009, pp. 49-300.