IOP Publishing Journal of Physics G: Nuclear and Particle Physics
J. Phys. G: Nucl. Part. Phys. **50** (2023) 055101 (21pp) https://doi.org/10.1088/1361-6471/acbaaf

Estimation of fission barrier heights for even–even superheavy nuclei using machine learning approaches

Cafer Mert Yesilkanat[1] **and Serkan Akkoyun**[2]

[1] Science Teaching Department, Artvin Çoruh University, Artvin, Turkey
[2] Department of Physics, Faculty of Science, Sivas Cumhuriyet University, 58140, Sivas, Turkey

E-mail: sakkoyun@cumhuriyet.edu.tr

Received 11 October 2022, revised 12 January 2023
Accepted for publication 9 February 2023
Published 21 March 2023

Abstract

With the fission barrier height information, the survival probabilities of super-heavy nuclei can also be reached. Therefore, it is important to have accurate knowledge of fission barriers, for example, the discovery of super-heavy nuclei in the stability island in the super-heavy nuclei region. In this study, five machine learning techniques, Cubist model, Random Forest, support vector regression, extreme gradient boosting and artificial neural network were used to accurately predict the fission barriers of 330 even–even super-heavy nuclei in the region $140 \leqslant N \leqslant 216$ with proton numbers between 92 and 120. The obtained results were compared both among themselves and with other theoretical model calculation estimates and experimental results. According to the results obtained, it was concluded that the Cubist model, support vector regression and extreme gradient boosting methods generally gave better results and could be a better tool for estimating fission barrier heights.

Supplementary material for this article is available online

Keywords: fission barrier, super-heavy nuclei, extreme gradient boosting, random forest, support vector regression, cubist, artificial neural network

(Some figures may appear in colour only in the online journal)

1. Introduction

The nuclear fission process, which was discovered as a result of the experimental observations of Hahn and Strassmann [1] on uranium nuclei, has been one of the most researched topics in

nuclear physics from the first theory announced by Meitner and Frisch [2] and Bohr and Wheeler [3]. For nuclear fission to occur, the nucleus must have a certain threshold energy defined as the fission barrier height which is the minimum amount of energy required to irreversibly break up the nucleus [4]. Fission barrier height is an important component in many aspects such as stellar nucleosynthesis [5], estimating the survival probabilities of the produced stable super-heavy nuclei [6], and calculating the competition between the fission process and neutron evaporation [7]. The height of the fission barrier cannot be directly observed [8], although there is little experimental information [9]. However, it is estimated theoretically with various models. The most common theoretical calculation models of fission barrier height in the literature can be listed as follows; the extended Thomas–Fermi plus Strutinsky integral (ETFSI) based on Skyrme SkSC4 functional[10], the finite-range liquid-drop model (FRLDM) [11], the heavy-nuclei (HN) model [7], and the Lublin–Strasbourg drop (LSD) approach [12, 13]. These methods, which are generally examined with macroscopic and microscopic approaches, are based on axially constrained quantum mechanical calculations [6, 14].

In recent years, there has been a limited number of innovative studies using machine learning (ML) algorithms (mainly artificial neural networks) to predict some parameters related to nuclear observables and the fission process [6, 15–21, 21]. Machine learning algorithms are used in many different application areas to describe, predict and classify the relationships between parametric/non-parametric, macroscopic or microscopic variables. Another advantage of machine learning algorithms is that intensive quantum phenomena such as calculating the height of the fission barrier can be determined by basic variables such as proton number (Z), neutron number (N) and mass number (A) of nucleus. In this study, the fission barrier height estimation performance of five ML algorithms, which have been quite up-to-date in recent years and have achieved very successful results in both classification and regression problems, has been comparatively evaluated. Machine learning algorithms used in this study are the Cubist model [22], Random Forest (RF) [23], Extreme Gradient Boosting (XGBoost) [24], Support Vector Regression (SVR) [25] and Artificial Neural Network [26].

Unlike traditional methods, only depending on the Z, N and A numbers of the nuclei, the aim of this study is to determine the inner fission barrier height of 775 exotic even–even super-heavy isotopes with proton numbers between $Z = 92$ and 120 ($132 \leqslant N \leqslant 216$) by using forecasting performance powerful and up-to-date machine learning methods. For this purpose, 75% of inner fission barrier height values of odd–odd, even–odd and odd–even super-heavy isotopes (987 neutron-rich exotic nuclei between $Z = 91$ and 120) calculated by ETFSI method which is reported by Mamdouh *et al* [10] were used as training data. For this purpose, inner fission barrier height values of 338 exotic neutron-rich odd–odd super-heavy nuclei ($Z = 91$ to 119) reported by Mamdouh *et al* [10] and calculated by ETFSI method were used as training data. The performance of the ML model predictions was separated from the main data set as testing data (the remaining 25%), and the cross-validation of the fission barrier height of even–even super-heavy nuclei calculated by ETFSI was determined. In addition to using the same training and test data sets in all of the machine learning models created, the same neutron number (N), mass number (A) and the amount of deviation between neutron and proton numbers (N–Z) were used as predictive variables. Furthermore, all prediction findings obtained from ML algorithms were compared both between themselves and with other predictions of the calculation of theoretical models [7, 11–13] and experimental results [9].

The paper organised as follows. In section 2, the methods for the artificial intelligence calculations in order to obtain fission barriers, methodological process steps and the performance indicators have been briefly summarized. In section 3, the results of the calculation

have been presented with the discussions of the results of each method in comparison. In the last section, conclusions of the study has been given.

2. Materials and methods

2.1. Machine learning algorithms

In this study, the fission barrier energies of the selected 12 super-heavy nuclei were predicted by using machine learning algorithms of the Cubist model, Support Vector Regression (SVR), Extreme Gradient Boosting (XGBoost) Random Forest (RF) and Artificial Neural Network (ANN).

2.2. Cubist model

The Cubist model developed by Quinlan [22, 27] is also called the M5 model. This model is a rule-based and an ensemble learning model in the boosting class. The main purpose of the Cubist model is to divide the data into sub-datasets with certain rules and create a non-parametric regression tree for each sub-dataset [28, 29]. The most important difference between the Cubist model and other regression tree methods is the development of a new regression model at each node for prediction in the Cubist model and the pruning of trees that produce high error predictions to prevent overfitting [29–31]. In addition, rule-based regression models called 'committees' are created from all sub-datasets by boosting, and with each new model, the errors of the previous model are tried to be reduced [32]. While the final Cubist estimation model is determined according to the rule that best fits the predictive variables [33], the significance level of the variables can be revealed by their contribution to the accuracy of the models created [29, 34]. Cubist models that do not require data reduction are successful in predicting complex relationships that are not linear [35].

2.3. Support vector regression

The version of support vector machines proposed by Cortes and Vapnik [36] for solving classification problems, adapted to regression problems, is called support vector regression (SVR) [25]. On the basis of SVR Structural Risk Minimization, which is a supervised learning model, it presents a prediction model in a high-dimensional space with the help of a kernel function to reduce the effects of outliers on the model [37, 38]. The basic function used for the SVR model can be defined as follows.

$$f(x) = w\phi(x) + b \pm \varepsilon, \tag{1}$$

where $\phi(x)$ is the kernel function used for nonlinear mapping, x is the model input vector, w and b are the regulation parameters of the function, and ε are the limit range values involved in the calculation on the SVR model curve. All points outside the ε tube range are outliers for the SVR model, and these outliers can be represented by a support vector [39]. This ε tube spacing can be calculated according to the constraints of equation (3) using equation (2) and data minimization

$$\min\left[\frac{1}{2}\|w\|^2 + c\sum_{i=1}^{N}(\xi_i, \xi_i^*)\right] \tag{2}$$

$$\text{subject to}\begin{cases} y_i - (w.\phi(x_i) + b) \leqslant & \varepsilon + \xi_i \\ (w.\phi(x_i) + b) - y_i \leqslant & \varepsilon + \xi_i^*, \\ \xi_i,\ \xi_i^* \geqslant 0,\ i = 1,\ldots,n \end{cases} \tag{3}$$

where ξ_i, ξ_i^* are slack variables, which is indicating the deviation of outliers, above and below an ε tube, and c is a positive constant penalty coefficient that determines the degree of penalized loss when a training error occurs. In equation (3), the left term represents the generalization of the model, while, the right term reflects the empirical risk, and the objective of SVR is to minimize these two values. A detailed description of the SVR algorithm can be found in the literature [39, 40].

2.4. Extreme gradient boosting

The Extreme Gradient Boosting (XGBoost) algorithm, which is an optimized, scalable, and faster type of gradient boosting machine [41] algorithm, was created by Chen and Guestrin [24]. The XGBoost algorithm is clearly distinguished from GBM with its features such as regulation, tree and leaf pruning, optimization and penalty [42]. The objective function of XGBoost is shown in equation (4) [43]

$$L = l(.) + \Omega(.) = \sum_{i=1}^{N} l(y_i, \widehat{y_i}) + \sum_{k=0}^{K} \Omega(f_k), \tag{4}$$

where $l(\cdot)$ is the loss function, $\Omega(\cdot)$ is a regulation term, y_i is the ith actual value, $\widehat{y_i}$ is the estimated value for the ith sample, N is the number of samples, K is the number of decision tree and f_k is the model of the kth tree. Here, while the loss function enables the detection of error rates, the regulation term includes the penalty parameter (γ) and the regulation parameter (λ), which prevents the overfitting of the model. The editing parameter is defined as follows according to the number of leaves in the decision trees (T) and the gain score (w_j) in the jth leaf [24]

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \|w_j\|^2. \tag{5}$$

At the beginning of the XGBoost Learning process, gain levels are calculated from the branches of a tree structure created according to certain initial conditions. Tree branches with higher gain values contribute greater weight to the prediction model. If the winning score is less than the specified penalty parameter (γ), those branches are pruned and their contribution to the prediction result is prevented. This process continues as an iterative process by adding information from the previous tree until the determined number of trees is reached. Each new decision tree function series created by this process is created by optimizing the learning level errors obtained from the previous trees. Thus, by the tth iteration, the final objective function becomes as shown in equation (6)

$$L^{(t)} = \sum_{i=0}^{N} l(y_i, \widehat{y_i}^{(t-1)} + f_t(x_i)) + \gamma T \frac{1}{2}\lambda \sum_{j=1}^{T} \|w_j\|^2. \tag{6}$$

As a result, a second-order Taylor expansion to equation (6) should be applied in order to optimize the objective function to develop a high-performance prediction model [43–45].

2.5. The Random Forest

The Random Forest (RF) algorithm, which is based on many decision tree structures, was first created by Brieman [23] as a combination of bagging [46] and random subspace [47] approaches. This algorithm is a powerful algorithm that has been used successfully in both classification and regression problems in recent years [48–54]. RF model, which consists of random combinations of many decision trees, uses a supervised machine learning method to obtain prediction results. For this reason, all data to be analyzed and inferred are divided into two parts, the in-bag and the out-of-bag dataset. Afterward, the training data set is randomly divided into subdata sets and many randomly generated decision trees, using a method called 'bootstrap samples' among themselves. Ultimately, the RF final estimate is determined to produce an estimate by averaging all results from each individual tree. However, to increase the prediction success, trees with unsuccessful prediction results are pruned and the effect levels are reduced on the final prediction result, while trees with correct predictions increase their weight coefficients and contribute more to the correct prediction. Thus, it is aimed that each tree created will affect Random Forest estimation results at certain weight levels and increase estimation success.

RF algorithm has several prominent advantages over other machine learning algorithms [55, 56]. These are the need for no preprocessing (normalization or standardization) for the training data, it can be used with appropriate precision in complex or nonparametric systems, and the effect of randomness increases due to the random selection of variables at each tree node, thus maintaining the overfitting level as a result of keeping the prediction variance low.

2.6. Artificial neural network

Artificial neural network (ANN) mimics the brain functionality [26]. The processing units artificial neurons that perform the operations are connected to each other by synaptic weights and form the ANN. By the connections, the neurons in the different layers communicate each other and they allow data to flow. In our calculation, we have used feed-forward ANN, more concretely a multilayer perceptron. In the ANN structure, the first layer called input layer consists three neurons (corresponding to A, N and Z–N numbers of the isotopes), the intermediate layers are named as hidden layers including hidden neurons and the last one is the output layer with one neuron corresponding to the fission barrier height. There is no rule for predicting the hidden layer and the number of neurons in this layer, and the optimum numbers are found after trial and error. The input neurons collect data from the outside and the output neurons give the results. The data is transmitted to the next layer neurons by multiplying the weight values of the connections. All data entering one neuron are summed by an appropriate function and the resulted net data are activated by appropriate activated function. In the present study, the hidden neuron activation function is tangent hyperbolic ($\tanh = (e^{x} - e^{-x})/(e^{x} + e^{-x})$) which is sigmoid-like function. The main purpose of the method is to determine the final weight values for each neuron connection starting from random values. Having the best weights, ANN can give its outputs close to the desired values. Supervised ANN is a two-stage process, in the first, the ANN is trained to determine the final best weights with the given input and output data values. With appropriate modifications of the weights, the ANN changes its weights to an acceptable level of error between the ANN and the desired outputs. In the second step (test), another dataset of the problem is given to ANN and the results are predicted by using the final weights. If the predictions of the test data are good, the ANN is considered to have learned the relationship between input and output data.

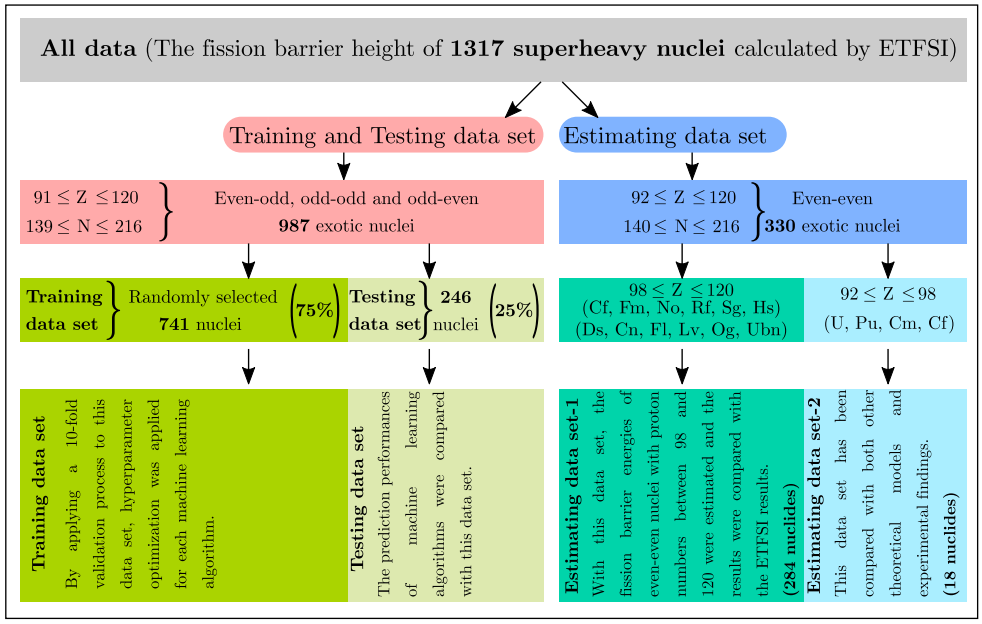

Figure 1. Fission barrier height estimations by machine learning methods for the isotopes.

For ANN with a single hidden layer, the desired output vector $\overrightarrow{y}$ is approximated by a network multi-output vector $\overrightarrow{f}$. The multi-output vector is defined by equation (7) as given below.

$$\overrightarrow{f} = R^p \longrightarrow R^r : \overrightarrow{f_k}(\overrightarrow{x}) = \Sigma \beta_j G(A_j(\overrightarrow{x})), \tag{7}$$

where $\overrightarrow{x} \epsilon R^p$, $\beta_j \epsilon R$, $A_j \epsilon A^p$, $k = 1, \ldots, r$, A^p is the set of all functions $R^p \rightarrow R^r$ defined by $A(\overrightarrow{x}) = \overrightarrow{w}.\overrightarrow{x} + b$, $\overrightarrow{w}$ is weight vector from the input layer to hidden layer, $\overrightarrow{x}$ is the input vector of ANN, b is the bias weight and $p(r)$ number corresponds to each input (output) variables.

2.7. The methodological process steps

The methodological steps used in this study are as follows.

Step 1: The first (inner) fission barrier height values of 1317 super-heavy nuclei with neutron-rich proton numbers ranging from 92 to 120 were calculated and reported by Mamdouh using the ETFSI method. These values are divided into two main data sets: Training and test data sets containing odd–odd nuclei, odd–even nuclei and even–odd nuclei (987 nuclei) and estimating data set containing even–even nuclei (330 nuclei) figure 1.

Step 2: A total of 987 odd–odd, odd–even and even–odd nuclei were randomly divided into two sub-datasets, 75% of which were training and the remaining 25% were testing. The training process was carried out with three predictive variables (N, A and N–Z) using the five machine learning algorithms (Cubist model, SVR, XGBoost, RF and ANN). The optimization process of each algorithm was carried out using grid search method with 10-fold cross-validation process. Thus, the model hyper-parameter values that will give the best prediction

result were determined. Detailed results obtained during the optimization process are presented in supplementary material-1.

Step 3: After the training process and hyper-parameter optimization were completed, the performance comparisons of the models were defined with the test dataset. The predictive power performance levels of the created machine learning models were evaluated according to the model performance indicator explained in section 2.H.

Step 4: After the training and testing process had been completed, the fission barrier height values of even–even nuclei in the estimating dataset-1 ($98 \leqslant Z \leqslant 120$) were predicted based on the same input values for each algorithm model and compared with the values calculated by the ETFSI method. The results are summarized with cross-scatter diagrams. In addition, the prediction results of machine learning models of fission barrier heights of even–even nuclei in estimating dataset-2 ($92 \leqslant Z \leqslant 98$) are presented in comparison with both experimental and other theoretical findings.

Step 5: In this last step, the fission barrier heights of all even nuclei in the range of $92 \leqslant Z \leqslant 120$ and $132 \leqslant N \leqslant 216$ were estimated with machine learning models whose training processes and performance criteria were completed, and the results were mapped with a contour plot. In addition, the findings obtained from these maps were evaluated in comparison with similar studies in the literature.

2.8. Model performance indicators

The mean absolute error (MAE, equation (8)), root-mean square error (RMSE, equation (9)) Lin's correlation coefficient (ρ_c, equation (10)) and Coefficient of Determination (R^2, (11)) performance metrics were used to compare the fission barrier heights estimated using machine learning models with the theoretical results and to determine the performance levels. These performance metrics were calculated as follows

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=0}^{n}|A_i - P_i| \tag{8}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(A_i - P_i)^2} \tag{9}$$

$$\rho_c = \frac{2\rho\sigma_x\sigma_y}{(\mu_x - \mu_y)^2 + \sigma_x^2 + \sigma_y^2} \tag{10}$$

$$R^2 = 1 - \frac{\sum(A_i - P_i)^2}{\sum(A_i - \bar{A})^2}, \tag{11}$$

where n is the total number of data, A_i and P_i denotes actual data and predicted value of ith sample, $\bar{A}$ is average of actual values, ρ is the correlation coefficient between variables x and y, μ_x and μ_y the means for the two variables and σ_x and σ_x are the corresponding variances. The RMSE and MAE metrics should be close to zero for the model with high predictive performance, ρ_c and R squared values near $+1$ indicate strong concordance between x and y, values near -1 indicate strong discordance, and values near zero indicate no concordance.

2.9. Software resources

R programming environment [57] was used in all stages of data analysis, machine learning process and data visualization. The library files used can be listed as follows: for cubist model

`cubist` [58], for SVR algorithm `kernlab` [59], for XGBoost algorithm `xgboost` [60], for RF algorithm `randomForest` [61], for ANN algorithm `neuralnet` [62], for data preparation, separation and optimization caret [63] and for visualize the results `ggplot2`[64], `gstat` [65] and `openair` [66] packages are used.

3. Result and discussions

3.1. Training and testing

As mentioned in section 2.7 (Step 1, Step 2 and Step 3) and explained with figure 1, the data are divided into training and testing. During the training process, five different machine learning processes were carried out and the model hyper-parameters that gave the best results for each algorithm were determined as a result of the 10-fold validation process. All outputs of this process are given as supplementary material-1. The model hyper-parameters from which the highest performance metrics were obtained as a result of the training process and the performance values determined by the 10-fold validation process are summarized in table 1. To this table, the best-performing machine learning algorithm was the Cubist model ($R^2 = 0.976$, RMSE = 0.324 and MAE = 0.235), while the lowest performance was obtained by the ANN ($R^2 = 0.948$, RMSE = 0.511 and MAE = 0.365) algorithm.

After the machine learning process and optimization processes were completed, the fission barrier heights in the testing dataset were estimated with each of the models created. The estimation results and the performance metrics determined for the testing dataset of the models are shown in figure 2 with cross-validation diagrams. According to the results, $R^2 = 0.975$, RMSE = 0.332 and MAE = 0.234 for Cubist model, $R^2 = 0.970$, RMSE = 0.361 and MAE = 0.280 for SVR, $R^2 = 0.965$, RMSE = 0.394 and MAE = 0.265 for XGBoost, $R^2 = 0.964$, RMSE = 0.424 and MAE = 0.290 for RF and $R^2 = 0.958$, RMSE = 0.435 and MAE = 0.323 for ANN were determined.

According to these findings, similar and close results were obtained in the training process. It was determined that the Cubist model performed slightly better than the other models in terms of the performance of estimating the fission barrier height values in the testing dataset. In addition, when figure 2 is evaluated in terms of error distributions, it is remarkable that the SVR model has a low error distribution. The error distributions of the Cubist and XGBoost estimations were also found to be low, except for a few outliers. As a result, considering all the results obtained from the training and testing processes, it was determined that the performance of the SVR and XGBoost models was significantly higher, although the Cubist model was superior to the other models in terms of its performance in estimating the fission barrier height.

3.2. Estimating

In this part of the research, the results of the data separated as estimation dataset-1 were examined. In this dataset, there is fission barrier height information calculated by ETFSI of a total of 284 isotopes belonging to even–even 12 elements (Cf, Fm, No, Rf, Sg, Hs, Ds, Cn, Fl, Lv, Og and Ubn) with atomic numbers between 98 and 120. The fission barrier heights of 284 nuclei in the estimation dataset-1 were predicted with five machine learning algorithms whose training and testing processes were completed, the results of which were summarized in the previous section. All of the estimation results obtained are presented in detail in Supplementary Material-2, along with cross-validation diagrams and estimation performance metrics. The summary of these diagrams is shown in figure 3 according to the models with the

Table 1. Optimized hyper-parameter values of machine learning models and performance metrics determined as a result of 10-fold validation process.

Models	R^2	RMSE	MAE	Model hyper-parameters	Descriptions
Cubist	0.976	0.324	0.235	Committees: 92	committees: the number of iterative model trees
				neighbors: 2	neighbors: the number of nearest neighbors
SVR	0.968	0.370	0.278	Sigma: 2	Sigma: distribution parameter for Gaussian radial basis
				C: 150	C: the penalty coefficient
XGBoost	0.966	0.392	0.294	Nrounds: 100	nrounds: Max number of boosting iterations
				Alpha: 0.5	alpha: L1 regularization term on weights.
				Lambda: 0.05	lambda: L2 regularization term on weights.
Random Forest	0.958	0.431	0.317	Ntree: 700	ntree: number of tree
				Mtry: 4	mtry: splite number on the node
ANN	0.948	0.511	0.365	Layer1: 100	Layer1: the number of neurons in the first layer
				Layer2: 60	Layer2: the number of neurons in the second layer
				Layer3: 60	Layer3: the number of neurons in the third layer

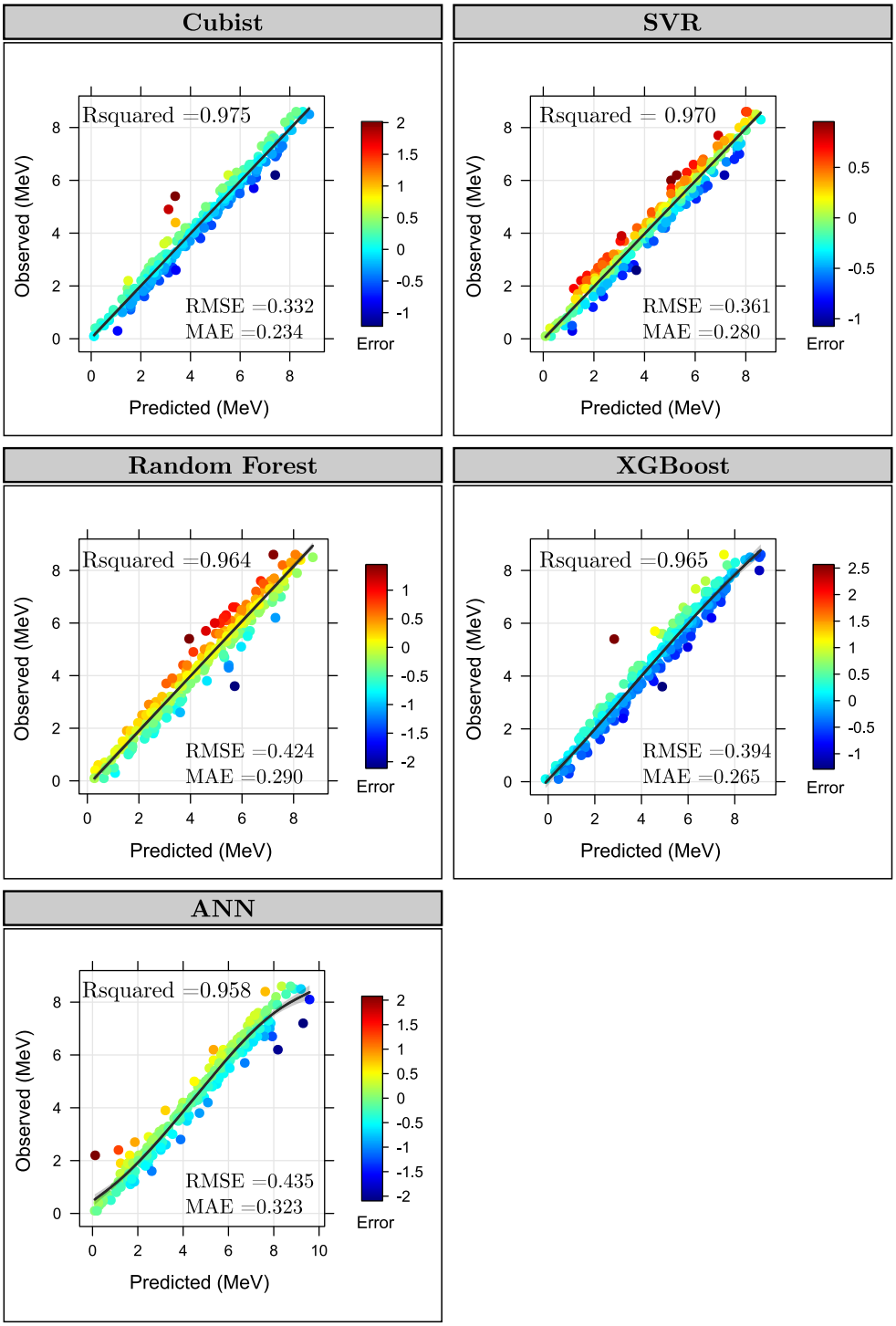

Figure 2. Performance of machine learning models in estimating fission barrier heights in the testing dataset and cross-validation diagrams.

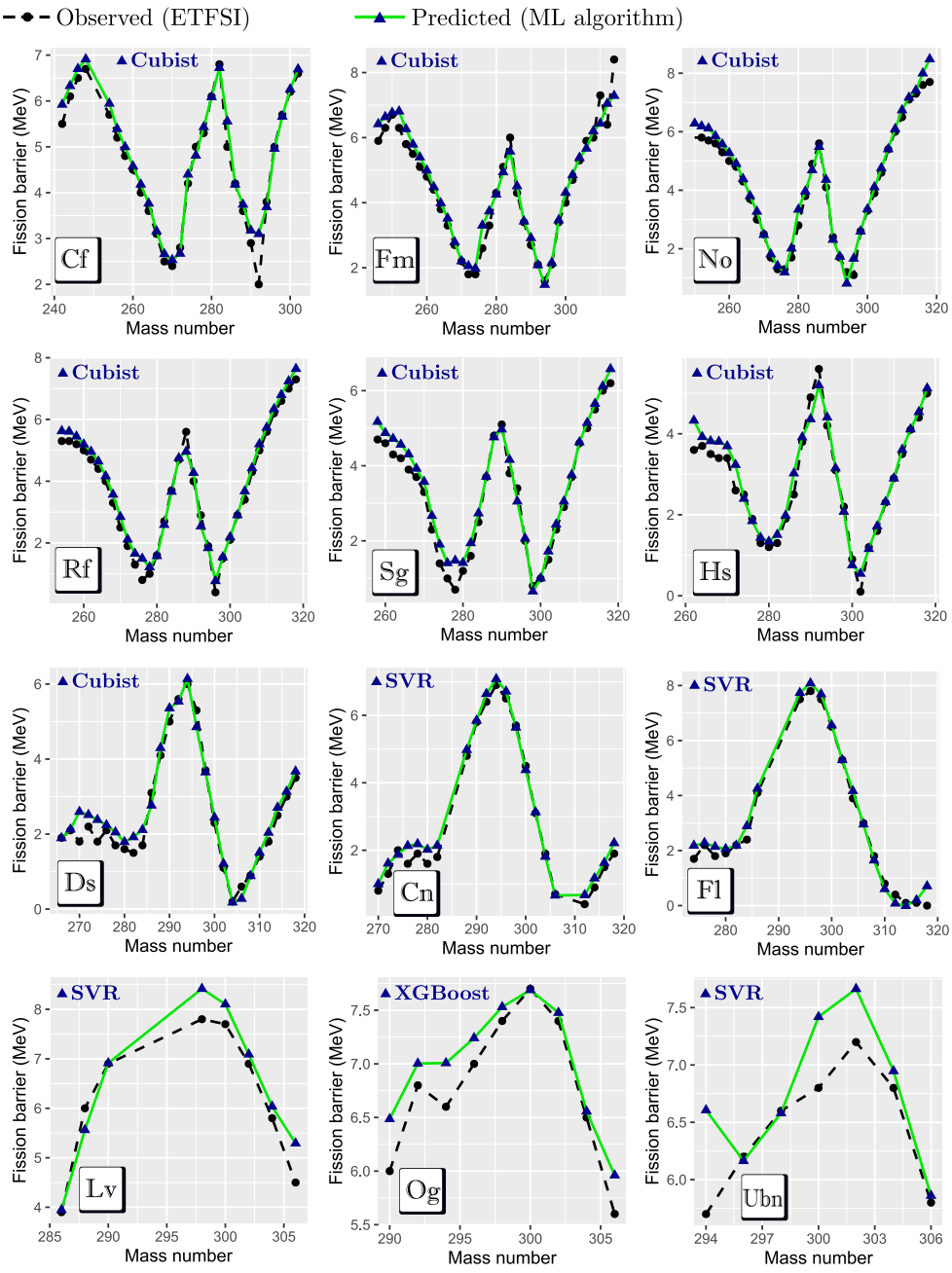

Figure 3. Fission barrier height estimations by machine learning methods for the isotopes.

best predictive performance. When figure 3 is examined, the model with the best fission barrier height prediction performance for Cf, Fm, No, Rf, Sg, Hs and Ds elements is the Cubist model, while the SVR for Cn, Fl, Lv and Ubn elements and XGBoost for Og elements.

Although the increase in the number of protons and the corresponding decrease in the number of isotopes decreased the estimation performance of the Cubist model a little, as can be seen from figure 3, it is seen that it makes very successful predictions especially for the elements up to $Z = 98$ to 110. The SVR model, on the other hand, showed a high performance especially in the elements after $Z = 110$. However, it is noteworthy that the prediction performance decreased significantly at the levels of Og and Ubn elements. The main reason for this situation is thought to be due to the insufficient number of training data at these levels.

Figure 4 shows comparative performance metrics calculated based on the estimated fission barrier height for each super-heavy nuclei. If the first two models with the best performance (Cubist model and SVR) are examined in detail, the results of the performance metrics can be summarized as follows (detailed results for all models are available in Supplementary Material-2). For Cubist model, the ρ_c values for Cf, Fm, No, Rf, Sg, Hs, Ds, Cn, Fl, Lv, Og and Ubn nuclei are 0.978, 0.976, 0.990, 0.989, 0.982, 0.975, 0.980, 0.990, 0.991, 0.953, 0.848 and 0.643, respectively. Among these, the highest ρ_c value is for the Fl isotope and the lowest value is for the Ubn isotope. The RMSE values for these isotopes were calculated as 0.282, 0.374, 0.286, 0.276, 0.304, 0.285, 0.298, 0.301, 0.350, 0.452, 0.360 and 0.485 MeV, respectively. The highest and lowest RMSE values are for Ubn and Rf isotopes. From the results of the SVR method, the ρ_c values for Cf, Fm, No, Rf, Sg, Hs, Ds, Cn, Fl, Lv, Og, and Ubn isotopes are calculated as 0.952, 0.973, 0.988, 0.982, 0.973, 0.971, 0.971, 0.993, 0.994, 0.952, 0.799 and 0.713, respectively. Among these, the highest ρ_c value is for the Fl isotope and the lowest value is for the Ubn isotope. The RMSE values for these nuclei were calculated as 0.418, 0.403, 0.309, 0.355, 0.375, 0.321, 0.360, 0.245, 0.282, 0.426, 0.442 and 0.455 MeV, respectively. The highest and lowest RMSE values are for Ubn and Cn isotopes. It is thought that the sudden changes in the performances of the elements observed in figure 4 are caused by the stochastic effects that occur in the machine learning process. Therefore, the models predicted some elements very well, while their predictive level of others was lower. In order to eliminate this negative situation and to reveal the performance of the model more clearly, the elements were evaluated as a whole (All data), not separately, and presented in figure 4. Accordingly, performance metrics and cross-validation graphs for all data are shown in figure 5.

Figure 5 shows the cross-validation diagrams and error diagrams of the estimated fission barrier height values for a total of 330 even–even nuclei ($92 \leqslant Z \leqslant 120$) in the estimating dataset for each model, according to the values calculated by ETFSI. Here, $\rho_c = 0.988$, $\mathrm{RMSE} = 0.318$ and $\mathrm{MAE} = 0.244$ for Cubist model, $\rho_c = 0.985$, $\mathrm{RMSE} = 0.350$ and $\mathrm{MAE} = 0.287$ for SVR, $\rho_c = 0.982$, $\mathrm{RMSE} = 0.386$ and $\mathrm{MAE} = 0.306$ for RF, $\rho_c = 0.982$, $\mathrm{RMSE} = 0.390$ and $\mathrm{MAE} = 0.308$ for XGBoost and $\rho_c = 0.961$, $\mathrm{RMSE} = 0.590$ and $\mathrm{MAE} = 0.448$ for ANN were obtained. According to these findings, it was determined that the Cubist model performed the best in terms of estimating the fission barrier height values in the estimating dataset. These results are consistent with the findings obtained from the training and testing datasets. In addition, when figure 5 is evaluated in terms of error distributions, it was determined that the error values increased due to the increase in mass, although it was partially less in the Cubist model. It is thought that the main reason for this is that the number of large-mass super-heavy nuclei in the training dataset is not sufficient and the models have not been adequately trained in terms of large masses.

3.3. Mapping and comparisons

If the fission barrier heights are mapped according to height values in the heavy and super-heavy mass region from 92 to 126 for Z and from 132 to 216 for N, contour map graphs are

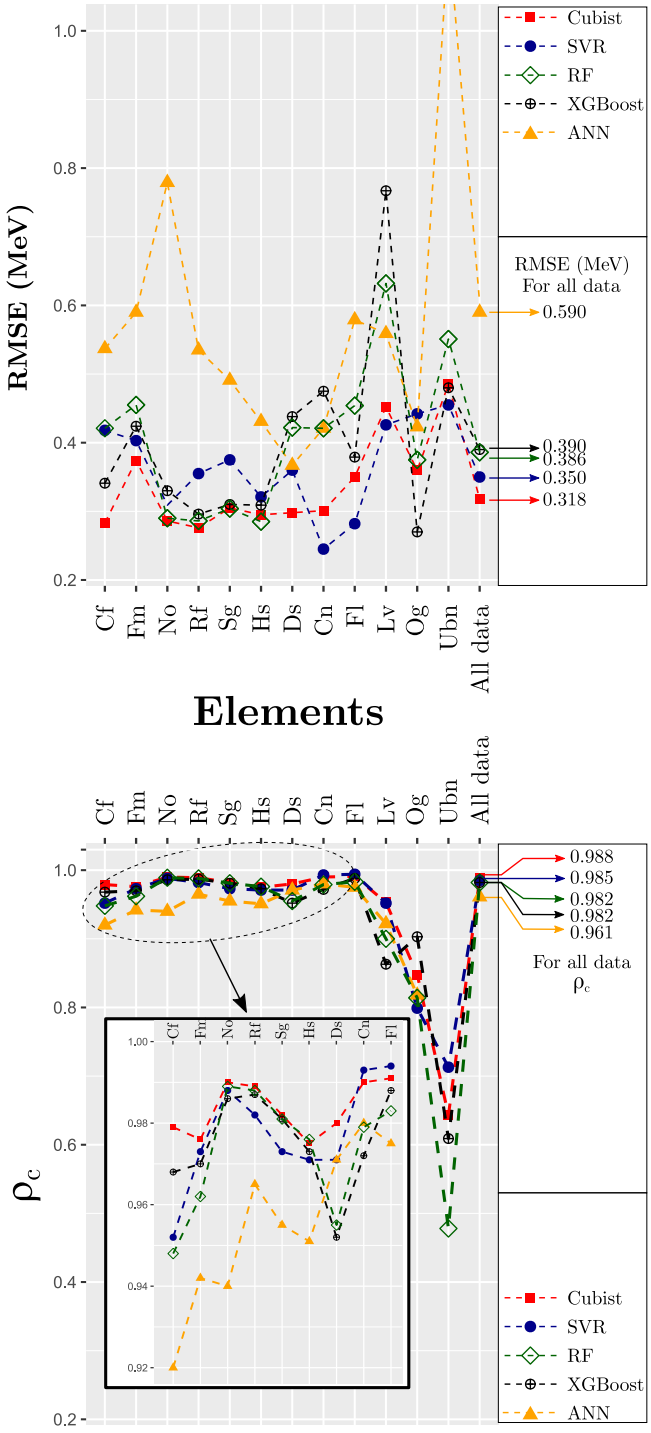

Figure 4. The RMSE (top) and ρ_c (bottom) values for the fission barrier height prediction for the nuclei from different machine learning methods.

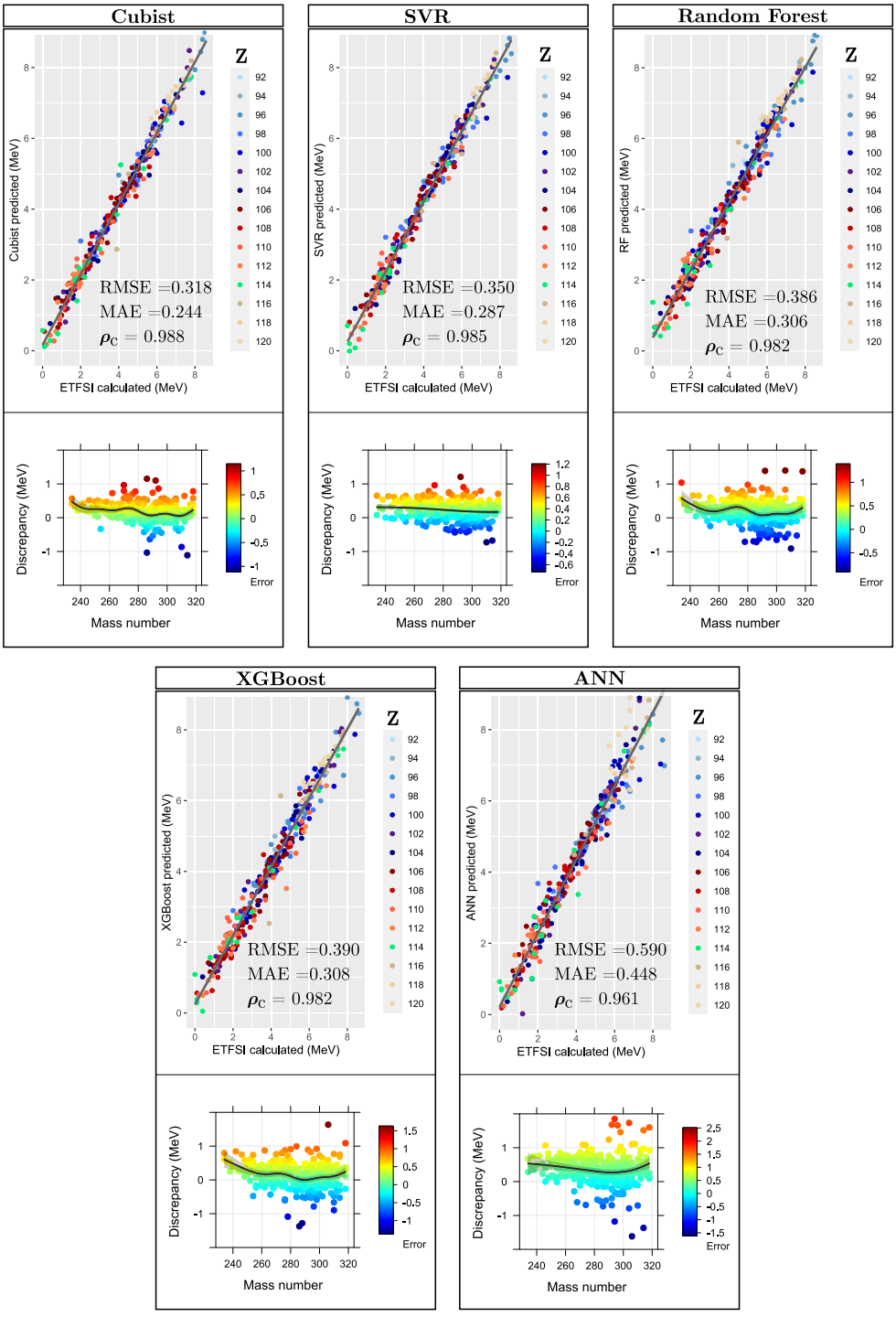

Figure 5. Comparisons of the predicted and calculated fission barrier heights for all isotopes considered according to different machine learning methods.

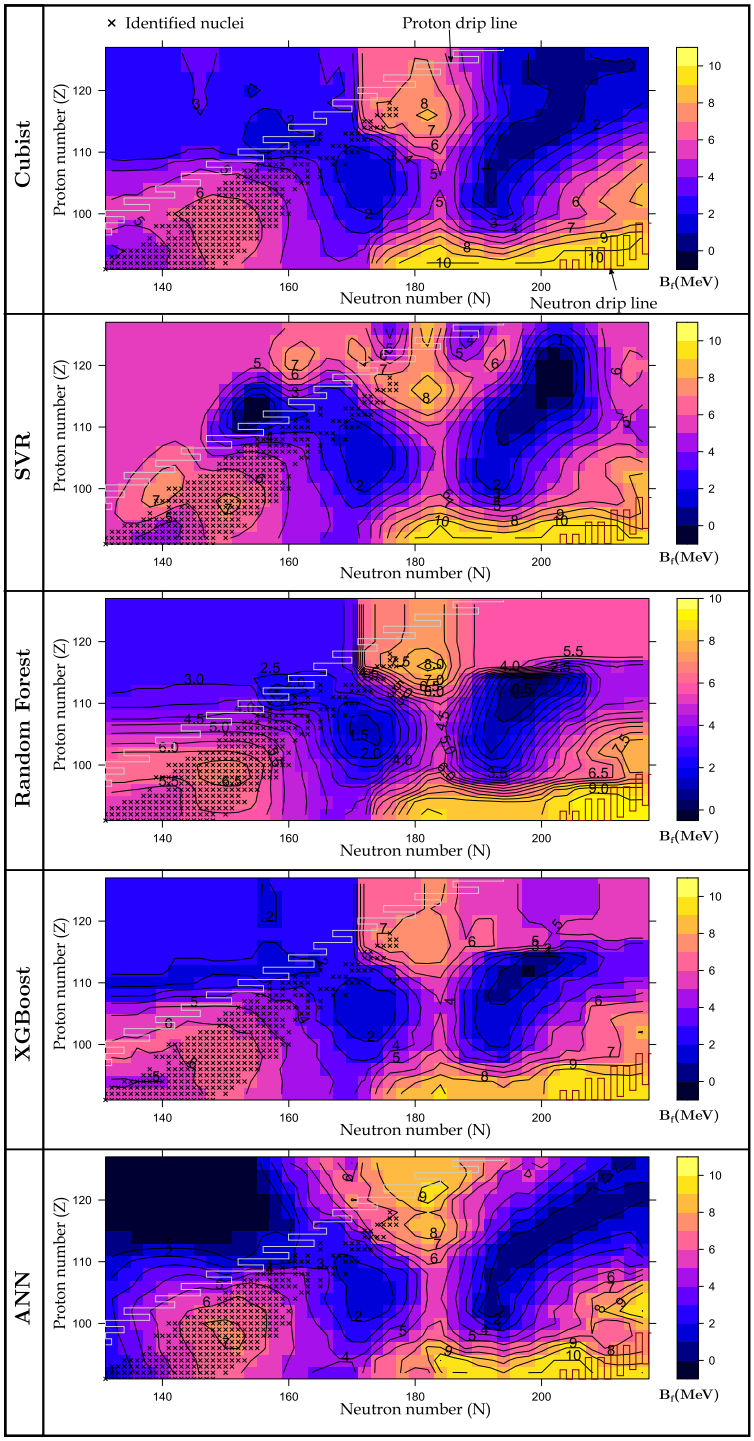

Figure 6. Contour maps of the fission barrier heights from different machine learning methods.

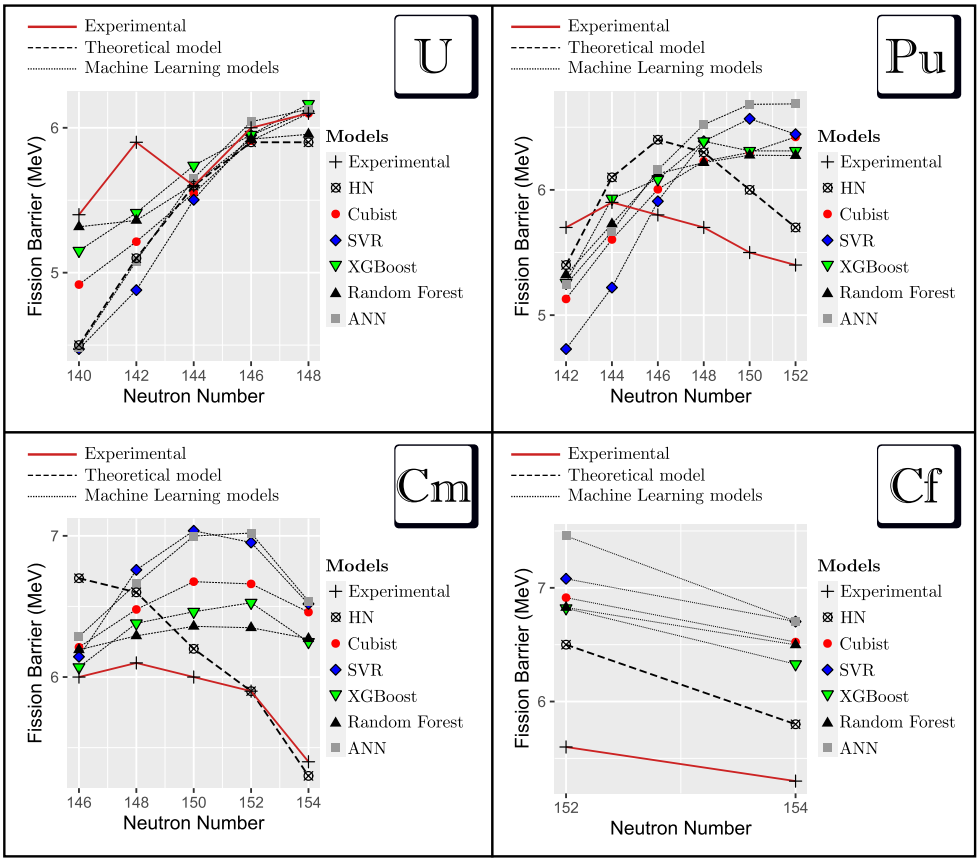

Figure 7. Fission barrier heights according to the neutron numbers for U, Pu, Cm and Cf isotopes from different methods and experimental data (HN model [7] and experimental results [9]).

obtained as given in figure 6 according to the different machine learning algorithms. Known nuclei are indicated by cross-symbols in the figure. In the figure, there is a hill around Z number 100 and N number 150 for these nuclei. Figure 6, it is observed that similar contour distributions are obtained in all machine learning models. Furthermore, it is seen that the hills where the barrier height is great are around $Z = 92$ and $N = 182$, $Z = 92$ and $N = 206$, $Z = 100$ and $N = 150$, $Z = 102$ and $N = 216$, $Z = 116$ and $N = 182$. In each five results, the island around $Z = 116$ and $N = 182$ is expected to be an island of stability in the region of super-heavy mass. The estimated barrier height is about 8 MeV.

In figure 7, the fission barrier heights obtained in this study and with other theoretical models for the isotopes of U, Pu, Cm and Cf are presented in comparison with the available experimental data and the other methods. It is seen that the values closest to the experimental data in the U isotope chain were obtained with ANN model in our study and with HN from the theoretical models. Among these, it can be said that Random Forest is more successful. In the Pu isotope chain, although all machine learning methods exhibit similar behavior, prediction values close to both the experimental results and the HN model were obtained around $N < 146$. When the results of Cm isotope are examined, machine learning models are better after $N < 150$, and HN is quite successful especially for $N = 152$ and $N = 154$. Finally, when

Table 2. Performance analysis of the different methods in determination of fission barrier heights. MaxD: the maximal discrepancy, MAD: mean absolute discrepancy, RMSD: root-mean squared discrepancy (units in MeV). ETFSI [10], FRLDM [11], HN model [7], LSD approach [12, 13].

Model name	MaxD	MAD	RMSD
FRLDM	2.20	0.97	1.09
LSD	1.80	0.90	1.01
ETFSI	1.20	0.60	0.70
HN	0.90	0.41	0.50
Cubist (this study)	1.31	0.57	0.69
SVR (this study)	1.47	0.75	0.88
RF (this study)	1.22	0.46	0.59
XGBoost (this study)	1.22	0.48	0.60
ANN (this study)	1.85	0.75	0.91

the Cf isotope chain is examined, it is seen that only two experimental data are available. Although the HN model is close to the experimental data, all of the machine learning models were successful in predicting the decrease in the fission barrier level as a result of the increase in the neutron number. When the findings in figure 7 are evaluated together, the results summarized in table 2 are reached. In table 2, maximal discrepancy (MaxD), mean absolute discrepancy (MAD) and root-mean-square discrepancy (RMSD) values are given in comparison with the theoretical model results for the five different machine learning methods used in this study.

As seen in table 2, the smallest value of MaxD among the theoretical models belongs to the HN method as 0.90 MeV. With our estimations in the current study, we were able to obtain this value as 1.22 MeV by the XGBoost and RF methods. The MaxD values of RF and XGBoost methods are also smaller than the values obtained from the FRLDM and LSD methods. Among the theoretical physical models, the method with the smallest MAD value is the HN method, which is 0.41 MeV. In our study, we were able to obtain this value as 0.46 MeV by RF model. When the results of Cubist, RF and XGBoost are examined, it is seen that the MAD value is smaller than FRLDM, LSD and ETFSI methods. Finally, when the RMSD values are examined, it is seen that the 0.50 MeV value obtained with HN. In the present calculations by machine learning models, RMSD value was obtained as 0.59 by XGBoost. Again, the results of Cubist, RF and XGBoost seem to be better for RMSD than FRLDM, LSD and ETFSI methods. According to these findings, the HN model among the theoretical methods and the RF and XGBoost algorithm from the machine learning model stand out as good predictive models. In figure 8, the contour plot graph of the results from five machine learning models compared with the HN is given. As can be seen, there are fission barriers around $Z = 100$, $N = 150$ and $Z = 116$, $N = 182$ regions in all the models. In the second hill, the barrier heights from Cubist, SVR, RF, XGBoost and ANN are 1.5, 2.0, 2.0, 1.5 and 2.5 MeV higher than HN result whereas in the first hill, heights are almost same in all models. According to the machine learning models, the fission pits were more common in the middle region of the graph. However, according to the HN method, while the upper end of the map is the fission pit, there is a small barrier according to machine learning models. The main reason for this is thought to be due to the fact that machine learning models are trained to predict the ETFSI fission barrier distribution well.

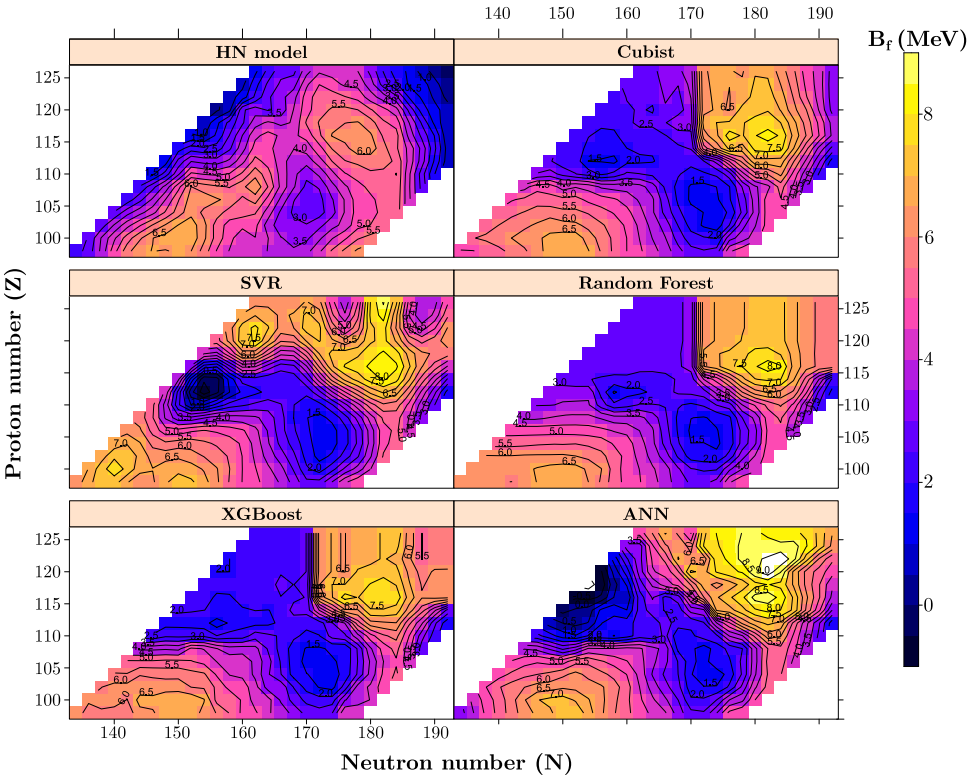

Figure 8. Contour map of the fission barrier heights from machine learning models in comparison with HN method.

4. Conclusion

In this study, five different machine learning models, namely Cubist model, support vector regression, random forest, extreme gradient boosting and artificial neural network, were used to estimate the fission barrier heights of even–even super-heavy nuclei. The obtained results were compared with each other and with existing literature data. In these estimates, it was concluded that the RF and XGBoost models are generally superior to others. The RMSE values in the estimations obtained from the Cubist model, SVR, RF, XGBoost and ANN were found to be 0.69, 0.88, 0.59, 0.60 and 0.91 MeV, respectively, for all isotopes examined. In addition, when the MaxD, MAE and RMSE values were examined, it was seen that the HN method had the lowest values whereas the values from FRLDM, LSD and ETFSI methods were larger than the results from RF and XGBoost models. The RF and XGBoost models allowed us to obtain comparable MaxD, MAE and RMSE values to the HN method. When the contour plot graphs of the fission barrier heights were examined, it was seen that the regions indicated by all machine learning methods for the barrier hills were almost the same. These regions are also compatible with the theoretically obtained HN method. As a result, we hope that on the basis of the findings obtained from this study, machine learning approaches will form a basis for further studies and contribute to the discovery of new clusters of atomic nuclei by presenting a different perspective to the theoretical studies on super-heavy nuclei.

Data availability statement

The data generated and/or analysed during the current study are not publicly available for legal/ethical reasons but are available from the corresponding author on reasonable request.

ORCID iDs

Cafer Mert Yesilkanat https://orcid.org/0000-0002-7508-7548
Serkan Akkoyun https://orcid.org/0000-0002-8996-3385

References

[1] Hahn O and Strassman F 1939 Concerning the existence of alkaline earth metals resulting from neutron irradiation of Uranium *Naturwissenschaften* **27** 11–5
[2] Meitner L and Frisch O R 1939 Disintegration of Uranium by neutrons: a new type of nuclear reaction *Nature* **143** 239–40
[3] Bohr N and Wheeler J A 1939 The mechanism of nuclear fission *Phys. Rev.* **56** 426–50
[4] Möller P and Nilsson S G 1970 The fission barrier and odd-multipole shape distortions *Phys. Lett.* B **31** 283–6
[5] Mahata K, Kailas S and Kapoor S S 2007 Fission barriers and shell corrections at the saddle point for mass ∼200 *Prog. Part. Nucl. Phys.* **59** 305–7
[6] Akkoyun S and Bayram T 2014 Estimations of fission barrier heights for Ra, Ac, Rf and Db nuclei by neural networks *Int. J. Mod. Phys.* E **23** 1–9
[7] Kowal M, Jachimowicz P and Sobiczewski A 2010 Fission barriers for even-even superheavy nuclei *Phys. Rev. C—Nucl. Phys.* **82** 1–10
[8] Bender M *et al* 2020 Future of nuclear fission theory *J. Phys. G Nucl. Part. Phys.* **47** 113002
[9] Itkis M G, Oganessian Y T and Zagrebaev V I 2002 Fission barriers of superheavy nuclei *Phys. Rev. C—Nucl. Phys.* **65** 7
[10] Mamdouh A, Pearson J M, Rayet M and Tondeur F 2001 Fission barriers of neutron-rich and superheavy nuclei calculated with the ETFSI method *Nucl. Phys.* A **679** 337–58
[11] Möller P, Sierk A J, Ichikawa T, Iwamoto A, Bengtsson R, Uhrenholt H and Åberg S 2009 Heavy-element fission barriers *Phys. Rev. C—Nucl. Phys.* 79
[12] Dobrowolski A, Pomorski K and Bartel J 2007 Fission barriers in a macroscopic-microscopic model *Phys. Rev. C—Nucl. Phys.* **75** 1–17
[13] Pomorski K and Dudek J 2003 Nuclear liquid-drop model and surface-curvature effects *Phys. Rev. C—Nucl. Phys.* **67** 13
[14] Hofmann H 1974 Quantum mechanical treatment of the penetration through a two-dimensional fission barrier *Nucl. Physics, Sect. A* **224** 116–39
[15] Neufcourt L, Cao Y, Nazarewicz W and Viens F 2018 Bayesian approach to model-based extrapolation of nuclear observables *Phys. Rev.* C **98** 1–17
[16] Akkoyun S 2020 Estimation of fusion reaction cross-sections by artificial neural networks *Nucl. Instrum. Methods Phys. Res. B* **462** 51–4
[17] Akkoyun S, Bayram T and Turker T 2014 Estimations of beta-decay energies through the nuclidic chart by using neural network *Radiat. Phys. Chem.* **96** 186–9
[18] Athanassopoulos S, Mavrommatis E, Gernoth K A and Clark J W 2004 Nuclear mass systematics using neural networks *Nucl. Phys.* A **743** 222–35
[19] Andrews W S, Lewis B J and Cox D S 1999 Artificial neural network models for volatile fission product release during severe accident conditions *J. Nucl. Mater.* **270** 74–86
[20] Koo Y H, Oh J Y, Lee B H, Tahk Y W and Song K W 2010 Artificial neural network modeling for fission gas release in LWR UO 2 fuel under RIA conditions *J. Nucl. Mater.* **405** 33–43
[21] Neudecker D, Grosskopf M, Herman M, Haeck W, Grechanuk P, Vander Wiel S, Rising M E, Kahler A C, Sly N and Talou P 2020 Enhancing nuclear data validation analysis by using machine learning *Nucl. Data Sheets* **167** 36–60
[22] Quinlan J R 1992 Learning with continuous classes *Proc. Australian Joint Conf. on Artificial Intelligence* pp 343–8

[23] Breiman L 2001 Random forests *Mach. Learn.* **45** 5–32
[24] Chen T and Guestrin C 2016 XGBoost: a scalable tree boosting system *22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* pp 785–94
[25] Drucker H, Surges C J C, Kaufman L, Smola A and Vapnik V 1997 Support vector regression machines *Adv. Neural Inf. Process. Syst.* **9** 155–61 (http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.5909)
[26] Haykin S 1998 *Neural Networks: a Comprehensive Foundation* (Englewood Cliffs, NJ: Prentice-Hall)
[27] Quinlan J R 1993 *Combining Instance-based and Model-based Learning* (San Mateo, CA: Morgan Kaufmann Publishers), vol 236–243
[28] Lee G G, Kim M C and Lee B S 2021 Machine learning modeling of irradiation embrittlement in low alloy steel of nuclear power plants *Nucl. Eng. Technol.* **53** 4022–32
[29] Pouladi N, Møller A B, Tabatabai S and Greve M H 2019 Mapping soil organic matter contents at field level with Cubist, random forest and kriging *Geoderma* **342** 85–92
[30] Nguyen H, Bui X N, Tran Q H and Mai N L 2019 A new soft computing model for estimating and controlling blast-produced ground vibration based on Hierarchical K-means clustering and Cubist algorithms *Appl. Soft Comput.* **77** 376–86
[31] Malone B P, Styc Q, Minasny B and McBratney A B 2017 Digital soil mapping of soil carbon at the farm scale: a spatial downscaling approach in consideration of measured and uncertain data *Geoderma* **290** 91–9
[32] Walton J T 2008 Subpixel urban land cover estimation: comparing Cubist, random forests, and support vector regression *Photogramm. Eng. Remote Sensing* **74** 1213–22
[33] Appelhans T, Mwangomo E, Hardy D R, Hemp A and Nauss T 2015 Evaluating machine learning approaches for the interpolation of monthly air temperature at Mt. Kilimanjaro, Tanzania *Spat. Stat.* **14** 91–113
[34] Lacoste M, Minasny B, McBratney A, Michot D, Viaud V and Walter C 2014 High resolution 3D mapping of soil organic carbon in a heterogeneous agricultural landscape *Geoderma* **213** 296–311
[35] Lamichhane S, Adhikari K and Kumar L 2022 National soil organic carbon map of agricultural lands in Nepal *Geoderma Reg.* **30** e00568
[36] Cortes C and Vapnik V 1995 Support-vector networks *Mach. Learn.* **20** 273–97
[37] Kecman V 2001 *Learning and Soft Computing : Support Vector Machines, Neural Networks, and Fuzzy Logic Models* vol 541 (Cambridge, MA: The MIT Press)
[38] Ibrahem Ahmed Osman A, Najah Ahmed A, Chow M F, Feng Huang Y and El-Shafie A 2021 Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia *Ain Shams Eng. J.* **12** 1545–56
[39] Wu C L, Chau K W and Li Y S 2008 River stage prediction based on a distributed support vector regression *J. Hydrol.* **358** 96–111
[40] Yu P S, Chen S T and Chang I F 2006 Support vector regression for real-time flood stage forecasting *J. Hydrol.* **328** 704–16
[41] Friedman J 2001 Greedy function approximation: a gradient boosting machine *Ann. Stat.* **29** 1189–232
[42] Zhu X, Chu J, Wang K, Wu S, Yan W and Chiam K 2021 Prediction of Rockhead using a hybrid N-XGBoost machine learning framework *J. Rock Mech. Geotech. Eng.* **13** 1231–45
[43] Ma M, Zhao G, He B, Li Q, Dong H, Wang S and Wang Z 2021 XGBoost-based method for flash flood risk assessment *J. Hydrol.* **598** 126382
[44] Hu L, Wang C, Ye Z and Wang S 2021 Estimating gaseous pollutants from bus emissions: a hybrid model based on GRU and XGBoost *Sci. Total Environ.* **783** 146870
[45] Wang J, He L, Lu X, Zhou L, Tang H, Yan Y and Ma W 2022 A full-coverage estimation of PM2.5 concentrations using a hybrid XGBoost-WD model and WRF-simulated meteorological fields in the Yangtze River Delta Urban Agglomeration, China *Environ. Res.* **203** 111799
[46] Breiman L 1996 Bagging predictors *Mach. Learn.* **24** 123–40
[47] Ho T K 1998 The random subspace method for constructing decision forests *IEEE Trans. Pattern Anal. Mach. Intell.* **20** 832–44
[48] Antoniadis A, Lambert-Lacroix S and Poggi J M 2021 Random forests for global sensitivity analysis: a selective review *Reliab. Eng. Syst. Saf.* **206** 107312
[49] Kurban H 2021 Atom classification with Machine Learning and correlations among physical properties of ZnO nanoparticle *Chem. Phys.* **545** 111143

[50] Zheng C, Chen C, Chen Y and Ong S P 2019 Random forest models for accurate identification of coordination environments from x-ray absorption near-edge structure *Patterns* **1** 100013
[51] Kalaiselvi B and Thangamani M 2020 An efficient Pearson correlation based improved random forest classification for protein structure prediction techniques *Meas. J. Int. Meas. Confed.* **162** 107885
[52] Xu R and Luo F 2021 Risk prediction and early warning for air traffic controllers' unsafe acts using association rule mining and random forest *Saf. Sci.* **135** 105125
[53] Hariharan R 2021 Random forest regression analysis on combined role of meteorological indicators in disease dissemination in an Indian city: a case study of New Delhi *Urban Clim.* **36** 100780
[54] Yesilkanat C M 2020 Spatio-temporal estimation of the daily cases of COVID-19 in worldwide using random forest machine learning algorithm Chaos *Solitons & Fractals* **140** 110210
[55] Strobl C, Boulesteix A L, Kneib T, Augustin T and Zeileis A 2008 Conditional variable importance for random forests *BMC Bioinf.* **9** 1–11
[56] Horning N 2013 Introduction to decision trees and random forests *Am. Museum Nat. Hist.* **2** 1–27
[57] (2022) A language and environment for statistical computing R Development Core Team
[58] Kuhn M and Quinlan R 2021 *Cubist: Rule- And Instance-Based Regression Modeling, R package version (3)*
[59] Karatzoglou A, Smola A, Hornik K and Zeileis A 2004 Kernlab—An S4 Package for Kernel Methods in R *J. Stat. Softw.* **11** 1–20
[60] Chen T and Guestrin C 2016 *Extreme Gradient Boosting* R package xgboost version 1.5.0.2
[61] Liaw A and Wiener M 2002 Classification and Regression by random forest *R News* **2** 18–22
[62] Gunther F and Fritsch S 2010 Neuralnet, training of neural networks *R J.* **2** 30–8
[63] Kuhn M *et al* 2020 caret: Classification and Regression Training R Packag. version 6.0-86
[64] Wickham H 2016 *ggplot2: Elegant Graphics for Data Analysis* (New York: Springer)
[65] Pebesma E J and Wesseling C G 1998 Gstat: a program for geostatistical modelling, prediction and simulation *Comput. Geosci.* **24** 17–31
[66] Carslaw D C and Ropkins K 2012 Openair—an R package for air quality data analysis *Environ. Model. Softw.* **27-28** 52–61