Applications of different machine learning methods on nuclear charge radius estimations

Physica Scripta

PAPER

Applications of different machine learning methods on nuclear charge radius estimations

Tuncay Bayram[1], **Cafer Mert Yeşilkanat**[2] and **Serkan Akkoyun**[3]

[1] Department of Physics, Karadeniz Technical University, 61080, Trabzon, Turkey
[2] Department of Science Education, Artvin Çoruh University, 08000, Artvin, Turkey
[3] Department of Physics, Sivas Cumhuriyet University, 58140, Sivas, Turkey

E-mail: t.bayram@ymail.com

Keywords: nuclear charge radius, artificial intelligence, machine learning, nuclidic chart

OPEN ACCESS

RECEIVED
20 July 2023

REVISED
25 September 2023

ACCEPTED FOR PUBLICATION
17 October 2023

PUBLISHED
21 November 2023

Abstract

Theoretical models come into play when the radius of nuclear charge, one of the most fundamental properties of atomic nuclei, cannot be measured using different experimental techniques. As an alternative to these models, machine learning (ML) can be considered as a different approach. In this study, ML techniques were performed using the experimental charge radius of 933 atomic nuclei ($A \geqslant 40$ and $Z \geqslant 20$) available in the literature. In the calculations in which eight different approaches were discussed, the obtained outcomes were compared with the experimental data, and the success of each ML approach in estimating the charge radius was revealed. As a result of the study, it was seen that the Cubist model approach was more successful than the others. It has also been observed that ML methods do not miss the different behavior in the magic numbers region.

1. Introduction

Nuclear radii are fundamental properties of atomic nuclei that have been studied extensively in nuclear physics [1, 2]. The size and shape of a nucleus play an important role in determining its stability and interactions with other particles and nuclei. Direct information about the Coulomb energy of nuclei can be obtained by examining the radii of nuclear charges and, more generally, the distributions of charge density in atomic nuclei. For this reason, charge radii have long attracted attention for nuclear mass formulas [3]. It can be measured by various methods based on the electromagnetic interaction that takes place between atomic nuclei and electrons or muons. Commonly used methods are measurements of transition energies in muonic atoms, elastic electron scattering experiments, K_α x-ray, and optical isotope shifts. Details of these techniques for measuring root mean square (RMS) charge radii of nuclei can be found in [4, 5]. With the latest advances in experimental techniques, such as the use of radioactive ion beams, more nuclei away from the β-stability line have been reached, thereby gaining access to the experimental nuclear charge.

The measurement of nuclear charge radii which is related to exotic phenomena such as skin and halo has been among the most interesting topics [6]. Studying the nuclear charge radius is important for a better understanding of the proton distribution in nuclei and the skin and halo. For these reasons, accurate and reliable estimation of nuclear charge radii in the absence of experimental data is important for studies on exotic nuclei and effective nucleon-nucleon interactions. Recently updated experimental data for nuclear charge radii of over 1000 nuclei are already available in the literature [4, 5, 7]. In our study, these experimental data were used as a source for different artificial intelligence methods and thus machine learning was carried out.

Machine learning methods have been used in many fields in nuclear physics as in other fields, such as the development of nuclear mass systematics [8], the identification of impact parameters in heavy-ion collisions [9–11], estimating beta decay half-lives [12], estimating beta decay energies [13], adjustment of non-linear interaction parameters for relativistic mean field approach [14], predictions for α-decay half-life superheavy

nuclei [15], estimations of fission barrier heights [16, 17], studying ground-state energies of the nuclei [18], estimation of fusion reaction cross-sections [19] and shell-model calculations supported by artificial intelligence [20].

When the machine learning studies on nuclear charge radius are examined in the literature, we can summarize some examples. Our previous work used artificial neural networks to derive a charge radius formula for $A \geqslant 40$ and $Z \geqslant 20$ nuclei [21]. Utama *et al* [22] improved the nuclear charge radius estimation performance by combining the Bayesian neural network method with density functional theory. The main motivation in the work is to develop a model that will accurately predict the charge radius of isotopes whose charge radius has never been measured. In their work, they considered the atomic nuclei with $A \geqslant 40$ and $Z \geqslant 20$ and took the numbers A and Z as inputs to the network. Accordingly, by improving the results of the theoretical models by applying the neural network, they managed to improve the deviation between the results obtained for the radius from the models and the experimental values up to 3 times.

Wu *et al* [23] obtained nuclear charge radius by using feed-forward neural networks considering the Z, N, and electric quadrupole transition strength values of the nuclei as input parameters of the network. As a result of the training they performed by separating Ca, Sm, and Pb isotopes from the test data, they were able to observe the existence of magic numbers in the isotope chains of these isotopes. Therefore, besides estimating the nuclear charge radii, they were also able to observe the kink, which corresponds to the magic numbers in the Sn, Sm, and Pb isotope chains. In their study, which also emphasized the importance of $B(E2)$ in generating the kink, they proposed the existence of a new relationship between the symmetry energy and the charge radius by including the symmetry energy term in the inputs of the network. Recently, a relationship between nuclear quadrupole deformation and the nuclear size for $^{98-118}$Pd was pointed out in the study of Geldhof *et al* [24]. Furthermore, they showed that pairing correlations attribute to a more correct description of nuclear charge radii for density functional calculations.

Dong *et al* [25] successfully predicted the nuclear charge radii in the neural networks calculations they performed in the $A \geqslant 40$ and $Z \geqslant 20$ regions. In the study, a Bayesian neural network is used, in which A, Z, pairing term, and promiscuity factor values are considered as inputs. By combining the NP formula, which allows the calculation of the load radius, with the Bayesian neural network, the results of the radius calculations were improved by approximately 2.7 times. In the illustrations carried out on Ca and K isotope chains, it was shown that while the NP formula exhibits a linear behavior, it is in agreement with the experimental data if it is supported by a Bayesian neural network. Later, authors revisited their study [26] to reduce the rms deviation difference (%30) between the validation set and training set which can cause a possible over-fitting. For this purpose, they added new features containing physical information.

In the study of Ma *et al* [27] nuclear charge radii were systematically estimated by the naive Bayesian probability classifier and the estimations improved up to about 1.7 times. By combining the raw results of the theoretical models with the predicted residuals of the naive Bayesian probability method, the theoretical charge radii from the HFB model and Sheng's semi-empirical formula calculations were refined. The results were analyzed in Ca and Bi isotope chains and the success of NBP refinements was demonstrated.

The main purpose of this study is to use different machine learning algorithms on the nuclear charge radius through to nuclidic chart ($A \geqslant 40$ and $Z \geqslant 20$) as well as to obtain the best results with simple variables. The deformation effect on nuclear charge radii by taking experimental data of quadrupole transition strength values can be considered to be in the study of Wu *et al* [23] for improving the predictive power of machine learning methods. However, this is possible for even–even nuclei. In the present study, we have studied the global prediction of machine learning methods on nuclear charge radii covering even–even, odd–even, even–odd and odd–odd nuclei by using the same physical quantities in the study of Dong *et al* [25] such as proton number (Z), mass number (A), pairing effects, shell closure effects, isospin dependence, abnormal behavior of some Hg isotopes. We have used 8 different machine learning methods as an alternative approach for estimating a reliable model to obtain nuclear charge radius. Through detailed analysis of the results from the study, the success of machine learning in obtaining the nuclear charge radius was highlighted. After this extensive study with a large number of machine learning methods used in addition to previous literature studies, methods that improve the deviations between the experimental data and the estimation results have been determined. Thus, machine learning has been shown to be a suitable and reliable tool for estimating nuclear charge radii in the absence of experimental data.

The paper is organized as follows. In section 2, the materials used in the study are mentioned and the applied methods are explained in short summaries by supporting the relevant references. In section 3, the findings from the study are presented and discussed. In the last part section 4, there is the Conclusion section in which an evaluation of the study is made.

2. Materials and methods

2.1. The data structure and software resources

In this research, experimental data reported in the literature [4, 5, 7] were used to estimate the nuclear charge radii with different machine learning algorithms. These data include the radius information of 933 nuclei ($A \geqslant 40$ and $Z \geqslant 20$). In this study, 699 randomly selected data were used as training data and the remaining 234 as testing data to evaluate the performance of the models. The predictive variables were taken as in the study of Dong *et al* [26]. These are mass number (A), proton number (Z), isospin dependence (I^2), pairing term (δ), the promiscuity factor (P) related to shell closure effects [28, 29], and a term related to abnormal charge radii behavior in ^{181,183,185}Hg (LI). The explicit form of the I^2, δ, P and LI are given by

$$I^2 = \left(\frac{N-Z}{A}\right)^2, \tag{1}$$

$$\delta = \frac{(-1)^Z + (-1)^N}{2}, \tag{2}$$

$$P = \frac{\nu_p \nu_n}{\nu_p + \nu_n}, \tag{3}$$

and

$$LI = \begin{cases} 1, & ^{181}\text{Hg},^{183}\text{ Hg},^{185}\text{ Hg} \\ 0, & \text{else} \end{cases} \tag{4}$$

The same input variables were used in all algorithms used in the study to estimate the atomic radius value in the machine-learning training process.

All analyses, calculations, and data visualizations in the research were prepared in the R programming environment [30]. The R library files used in this study are, `kernlab` for SVR and GPPK algorithms [31], `randomForest` for RF algorithm [32], `quantregForest` for QRF algorithms [33, 34], `xgboost` for XGBoost algorithm [35], `RSNNS` for Artificial neural network (ANN) algorithm [36], `cubist` for cubist model [37], `earth` for multivariate Adaptive Regression Spline (MARS) model [38], `caret` for data processing, data separation and optimization of machine learning algorithms [39], `openair` [40] and `ggplot2` [41] for cross-validation, scatter diagrams, and data visualization.

2.2. Machine learning algorithms

In this study, eight algorithms (artificial neural network, Cubist model, Gaussian process with polynomial kernel, multivariate adaptive regression splines, random forest, quantile random forest, support vector regression, and extreme gradient boosting) were used. These algorithms are summarized below.As will be seen in the next sections, the performance metrics of the Cubist model are higher than the others for nuclear charge radii predictions. Therefore more details for the Cubist model algorithm are given.

Artificial Neural Network (ANN): Artificial neural networks (ANNs) [42–44], which mathematically simulate biological nerve cell structure and functionality, have been developing in different structures in recent years to solve different problems [16, 45–49]. Artificial neural networks consist of small processing units called neurons, and each neuron is connected to each other through adaptive synaptic weights [50].

Gaussian Process with Polynomial Kernel (GPPK): Gaussian processes, developed to explain non-parametric relationships on a Bayesian basis, provide point estimates as well as confidence intervals for these estimates [51]. In this approach, the common distribution of training and test data is represented by the multidimensional Gaussian density function obtained on the basis of the Polynomial kernel, so the estimated distribution for each test data is determined by the distribution conditions of the training data [51, 52]. Detailed information for this algorithm can be found in the cited sources [53, 54].

Multivariate Adaptive Regression Splines (MARS): This method, which is a nonparametric modeling technique, was developed by Friedman [55]. In the MARS approach, the entire data set is significantly divided into sub-datasets and a separate linear model is established for each sub-dataset [56]. Thus, with the help of this piecewise linear model, which is automatically adapted to all data, the nonlinear connection between the independent variables and the target variable (splines) can be estimated [57, 58]. MARS is a very useful algorithm for problems where the relationships of the variables in the dataset may be different in each region [59, 60].

Random Forest (RF): This algorithm developed by Breiman [61] has been used in many different fields in recent years due to its high performance in classification and regression problems [62–67]. In the RF approach, the entire dataset is divided into subsets and more than one decision tree is created within each subset. Then Each tree is trained with randomly selected features. As a result, RF, an ensemble learning technique, is affected

by specific weights from each tree for the final model result [68]. Also, in the RF model, training each tree with randomly selected features and generating each tree dataset using a subset avoids the overfitting problem [63].

Quantile Random Forest (QRF): Unlike the traditional Random Forest algorithm, QRF uses only values from a certain percentile during splits in each tree [33]. Thus, it allows customizing each tree to include only values in a certain percentile. This method can show better results in some cases than a standard Random Forest algorithm in regression problems by reducing the effect of outliers in the data set and allowing the model to better explain the distribution of the target variable (predicted variable) in certain percentiles [69–71].

Support Vector Regression (SVR): This method is a regression method that utilizes support vectors and the Lagrange multiplier approach for analyzing and predicting data [72]. The SVR algorithm is based on the Support Vector Machine (SVM) algorithm, which provides effective solutions to classification problems [73]. SVR is particularly useful when dealing with outliers and non-linearities in data [74]. The SVR aims to obtain a regression estimate that accurately predicts the response values based on a subset of high-dimensional prediction variables [75]. It utilizes support vectors and the Lagrange multiplier approach to achieve this goal [76, 77].

Extreme Gradient Boosting (XGBoost): This algorithm, developed by Chen and Guestrin [35], has emerged by optimizing the Gradient Boosting Machine (GBM) [78] algorithm and expanding it to a scalable level. XGBoost is particularly successful in large datasets and high-dimensional feature spaces. XGBoost is a machine learning algorithm that models nonlinear relationships using multiple decision trees and is based on the principle of sequential error reduction [79, 80]. In this method, decision trees are created sequentially and each tree is reconstructed according to the predictions of the previous tree with a focus on error reduction. Thus, each tree created learns a new feature that increases the accuracy of the model and reduces the error rate [81]. Furthermore, in the XGBoost model, a weight value is assigned to each tree and the contribution of each tree is determined more clearly in order for the results to have higher performance. In addition, the XGBoost algorithm has many regularization parameters to prevent overfitting [17].

Cubist model: The cubist model is a rule-based model capable of handling both numerical and categorical variables [82, 83]. It uses a 'separate and conquer' methodology, to create a rule-based regression tree that identifies different paths by iteratively splitting predictor variables within the model [84, 85]. The Cubist model results contain several sets of rules that can be broken down into sub-datasets with similar characteristics to represent the entire dataset. A multivariate linear regression model is fitted to the subsets of data generated by these rule sets to reveal the pattern of association between the predictor variables and the target variable [86]. Unlike other rule-based tree models, Cubist uses a combination of models together with a smoothing coefficient to combine the linear models at each node of the tree, as expressed in equation (5) [85, 87].

$$\hat{y}_{par} = a \times \hat{y}_{(k)} + (1 - a) \times \hat{y}_{(p)} \tag{5}$$

where $\hat{y}_{(k)}$ is the prediction from the current model (child model), $\hat{y}_{(p)}$ is the prediction from the parent model located above it in the tree, and a is the smoothing coefficient. This coefficient can be determined as expressed in equation (6) [85]:

$$a = \frac{\mathrm{Var}[e_{(p)}] - \mathrm{Cov}[e_{(k)}, e_{(p)}]}{\mathrm{Var}[e_{(p)} - e_{(k)}]} \tag{6}$$

where $\mathrm{Var}[e_{(p)}]$ is the variance of the errors (i.e., $y - \hat{y}_{(k)}$) of the parent model, $\mathrm{Cov}[e_{(k)}, e_{(p)}]$ is the covariance of the residuals of the child and parent models, and $\mathrm{Var}[e_{(p)} - e_{(k)}]$ is the variance of the difference between the residuals. This process is based on the covariance of the residuals of the child and parent models. The covariance indicates that the errors of the two models are linearly related. If the variance of the errors of the parent model is greater than the covariance, the child model is weighted more than the parent model. In the opposite case, Cubist gives more weight to the parent model. Thus, the model with the lowest error value is more dominant in the adjusted model. If the error values of the two models are the same, both models will have equal weight [85]. Cubist combines the linear models at each node according to equation (5), creating a single linear model for each rule. This allows the models to be presented in a more organized and representative way [85]. With new regression models developed at each tree wedding, branches with a high error are pruned, thus preventing overfitting for the model [88, 89].

There are clear specific differences between the Cubist model and other tree-based models, such as committee models (a boosting-like procedure for building iterative model trees), specific techniques used for model smoothing, rule generation, and pruning, and instance-based corrections (using nearby points from the training set data to adjust predictions) [85, 86]. In addition, the Cubist model is highly interpretable and can provide insights into the decision-making process without the need for techniques such as SHAP (SHapley Additive exPlanations) [90] and LIME (Local Interpretable Model-agnostic Explanations) [91]. Moreover, the ability to handle non-linear relationships between dependent and independent variables [17, 92] and to use both numerical and categorical variables as model inputs give the Cubist model significant advantages in terms of high forecasting performance [93].

2.3. Comparison of algorithms

Machine learning algorithms have their own strengths and weaknesses. The choice of algorithm depends on the specific problem and the characteristics of the dataset. It is important to carefully consider the advantages and disadvantages of each algorithm before selecting the most appropriate one for a given task. The algorithms used in this study have their advantages and disadvantages, which can be summarized as follows.

Artificial neural networks are known for their ability to model complex relationships and handle large amounts of data. They can learn from examples and generalize well to unseen data. However, they can be computationally expensive to train and require a large amount of data to achieve good performance [94]. They can also exhibit illogical behavior when not well trained [95]. Cubist models are rule-based models that can handle both numerical and categorical variables. They are interpretable and can provide insights into the decision-making process. However, they may not perform as well as other algorithms when the relationships between variables are non-linear [93]. Gaussian processes with polynomial kernels are flexible models that can capture complex relationships between variables. They can provide levels of uncertainty for predictions. However, Gaussian processes are computationally intensive and may not scale well to large datasets, and they offer a probabilistic interpretation [96, 97]. Multivariate adaptive regression splines are non-linear models that can capture complex relationships between variables and they are useful when the dataset is large and computation time is not an issue [98]. Random forests are ensemble models that combine multiple decision trees. They are robust to overfitting and can handle high-dimensional data. They can also provide estimates of variable importance. However, they may not perform well when there are strong interactions between variables [99]. Quantile random forests are a variation of random forests that can model the conditional distribution of the response variable. They can provide more information about the uncertainty of predictions. However, they may require more computational resources and may not perform as well as other algorithms when the conditional distribution is highly skewed or heavy-tailed [100]. Support vector regression can handle non-linear relationships between variables and can provide good generalization performance. However, it can be sensitive to the choice of hyperparameters and may not perform well when the dataset is noisy or contains outliers [93]. Extreme gradient boosting (XGBoost) can handle high-dimensional data and capture complex relationships between variables. XGBoost is known for its high performance and scalability, However, XGBoost can be sensitive to outliers and may not perform well on unstructured and sparse data [101]. Additionally, XGBoost may have a slower prediction speed compared to the random forest due to the generation of sequential decision trees [102].

2.4. Model performance metrics

In this study, three basic metrics were used to evaluate the performance of machine learning models. These are the mean absolute error (*MAE*) and root mean square error (*RMSE*). The mathematical expressions of these performance measures can be shown as follows.

$$MAE = \frac{1}{n}\sum_{i=0}^{n}|A_i - P_i| \tag{7}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(A_i - P_i)^2} \tag{8}$$

where n is the total number of data. A_i and P_i indicates actual data and predicted value of ith sample, respectively. The *RMSE* and *MAE* metrics should be close to zero for the model with high predictive performance.

3. Result and discussions

3.1. Training process

In this study, the results of the nuclear charge radius estimations for the atomic nuclei from eight different machine-learning methods were obtained. All of the data (933 nuclei) used in the machine learning process were separated into training (699 nuclei) and test (234 nuclei) data sets. The training process of machine learning algorithms was carried out with 10-fold cross-validation processes and the model hyperparameters that gave the best results for each algorithm were determined. Thus, it aims to increase each model's performance by ensuring that it is optimized within the training process. Table 1 shows the performance metrics and optimized model hyperparameter values determined as a result of the 10-fold cross-validation process. According to table 1, along with the learning performance of all models examined in the study being quite high, it can be determined that the best-performing machine learning algorithm is the Cubist model (*RMSE* = 0.01199 fm and *MAE* = 0.0077 fm).

3.2. Testing process

The performances of the models were evaluated by the test data and the results were shown with discrepancy distribution in figure 1. The one with the lowest *RMSE* value among these methods is the Cubist model, where

Table 1. Performance metrics and optimized model hyperparameter values determined as a result of the 10-fold cross-validation process of machine learning models.

Algorithm	*RMSE*	*MAE*	Optimized model hyperparameter	Descriptions
Cubist	0.01199	0.00770	committees: 97 neighbors: 3	committees: The number of iterative model trees neighbors: The number of nearest neighbors
XGBoost	0.01533	0.01114	nrounds: 80 alpha: 0.001 lambda: 0.01	nrounds: Max number of boosting iterations alpha: L1 regularization term on weights. lambda: L2 regularization term on weights.
RF	0.01746	0.01173	ntree: 700 mtry: 4	ntree: Number of tree mtry: Splite number on the node
QRF	0.01791	0.01232	ntree: 750 mtry: 4	ntree: Number of tree mtry: Splite number on the node
GPPK	0.03715	0.02901	scale: 0.1 degree: 3	scale: The scaling parameter of the polynomial and tangent kernel degree: The degree of the polynomial
MARS	0.03462	0.02745	degree: 5	degree: The degree of the polynomial
SVR	0.04286	0.03300	sigma: 0.1 C: 20	sigma: Distribution parameter for Gaussian radial basis C: The penalty coefficient
ANN	0.03992	0.03174	Layer 1: 200 Layer 2: 60 Layer 3: 40	Layer1: The number of neurons in the first layer Layer2: The number of neurons in the second layer Layer3: The number of neurons in the third layer

the deviations of the estimation results from the experimental values are given in figure 1(a). This model's *RMSE* and *MAE* values are 0.0102 fm and 0.0075 fm, respectively. As clearly seen from the figure, the deviations from the experimental values are concentrated around the zero line except for a few atomic nuclei. In figure 1(b), the graph of the deviations of the predictions from the XGBoost from the experimental data is presented. The distribution in the curve, where the deviations are clearly seen to be concentrated around the horizontal zero line, appears to be in the range of about −0.05 fm to +0.05 fm. The *RMSE* and *MAE* values of the results obtained from the XGBoost model are 0.0125 fm and 0.0125 fm, respectively, which is the method in which the best results are obtained after the Cubist model. After the XGBoost, the results of the RF model, the method with the better predictions, are given in figure 1(c). It is seen that the distribution here is concentrated in the range of −0.03 fm to +0.06 fm, around the horizontal zero line. The *RMSE* and *MAE* values of the results obtained from this model were 0.0138 fm and 0.0102 fm, respectively. Next better results are obtained with the QRF model (figure 1(d)). This model's *RMSE* and *MAE* values are 0.0140 fm and 0.0104 fm, respectively. As can be seen from the figure, the distribution of deviations from the experimental data of this model is concentrated in the horizontal zero line, in the range of −0.05 fm to +0.06 fm. When the results obtained from the GPPK model given in figure 1(e) are examined, it is seen that the *RMSE* and *MAE* values are 0.0346 fm and 0.0273 fm, respectively. The distribution here appears to be in the range of −0.08 fm to +0.11 fm. The distribution of results in this model is concentrated on the positive side of the zero line, indicating that the estimates are generally bigger than the experimental data. The results of the predictions of the MARS method are given in figure 1(f), showing their deviations from the experimental values. The *RMSE* and *MAE* values of this model are 0.0346 fm and 0.0277 fm, respectively, and it is seen that the concentration is in the range of −0.06 fm and +0.11 fm around the zero line. In figure 1(g), the distributions of the estimates obtained from the SVR model are given. In the model, in which *RMSE* and *MAE* values were obtained as 0.0439 fm and 0.0336 fm, respectively. It is seen that the deviations of the estimations from the experimental data fluctuate around zero. However, the distribution was found to be in the range of −0.11 fm to +0.11 fm. Finally, the results from the ANN model are presented in figure 1(h). In the graph showing the distribution, it is clearly seen that the zero line disappears and the distribution is shifted down. It can also be seen from the figure that the deviations from the experimental values spread around −0.05 fm to +0.12 fm. The *RMSE* and *MAE* values of the ANN model were found to be 0.0564 fm and 0.0494 fm, respectively. These results obtained by ANN show close results to the findings of our previous study [21]. It should be noted that the results of Cubist, XGBoost and RF model for *RMSE* and *MAE* are better than those of [23, 25, 27]. As seen in this study, the Cubist model gives the most successful results in nuclear charge radius estimations. Cubist model success is based on its algorithm, which correctly partitions the data and generates regression equations that best fit each subset. This is due to its ability to model complex relationships using a combination of both tree structure and regression equations.

This study used the bootstrap method to measure the uncertainty of machine learning predictions [103]. In this method, a large number of models (1000 for this study) are trained by resampling from the original training set and their predictions on the test data are recorded. The standard deviation of these predictions represents the uncertainty. For the Cubist model, which gave the best prediction results in our study, the minimum and maximum values of the uncertainty in the test data were determined as 0.000 85 fm (for ^{199}Hg) and 0.050 21 fm (for ^{77}Rb), respectively.

3.3. An application of machine learning models

In this subsection, we are searching for the feasibility of considering machine learning methods based on experimental data of Kr and Sr isotopic chains to check their prediction on shell closure at $N = 50$ because the shell structure of nuclei around $N = 50$ is clearly visible in the experimental nuclear charge radii data where the lowest values are located at $N = 50$ for Kr and Sr isotopic chains. (More details for N and Z dependence of nuclear charge radii can be found in [4].) We also compare our results with the prediction of conventional semi-empirical charge radii formulas and mean field models. As it is well known the radius of the nucleus is proportional to its mass number. However, the conventional A-dependent RMS charge radius formulation in equation (9) is not globally valid for all nuclei covered by nuclidic charts because some nuclei have different numbers of neutrons and protons even if they have the same mass numbers. On the other hand, the experimental nuclear charge radii values show that the $R/A^{1/3}$ ratio is not constant through the nuclidic chart [104, 105]. Therefore, Z or isospin-dependent formulas have been developed and it has been found that they describe nuclear charge radii of nuclei much better (Details and references can be found in [104, 106]). Some well-known semi-empirical charge radii formulas are given below.

$$R_c = r_A\, A^{1/3} \tag{9}$$

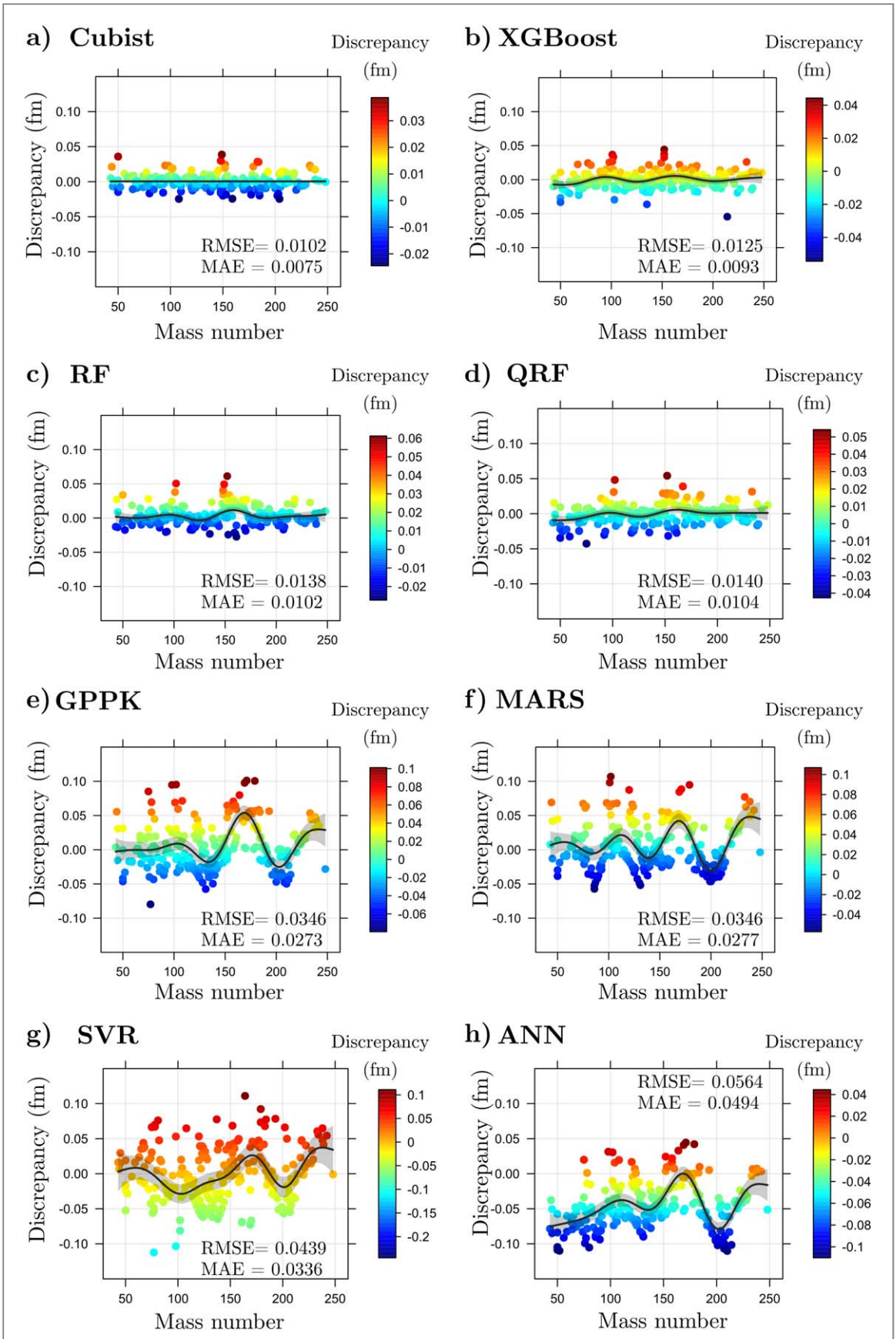

Figure 1. Discrepancy distribution for nuclear charge radius estimated from (a) Cubist, (b) XGBoost, (c) RF, (d)QRF, (e) GPPK, (f) MARS model, (g) SVR and (h) ANN. The units for *RMSE* and *MAE* are in fm.

$$R_c = r_Z\, Z^{1/3} \tag{10}$$

$$R_c = \sqrt{\frac{5}{3}}\left((r_p\, Z^{1/3})^2 + 0.64\right)^{1/2} \tag{11}$$

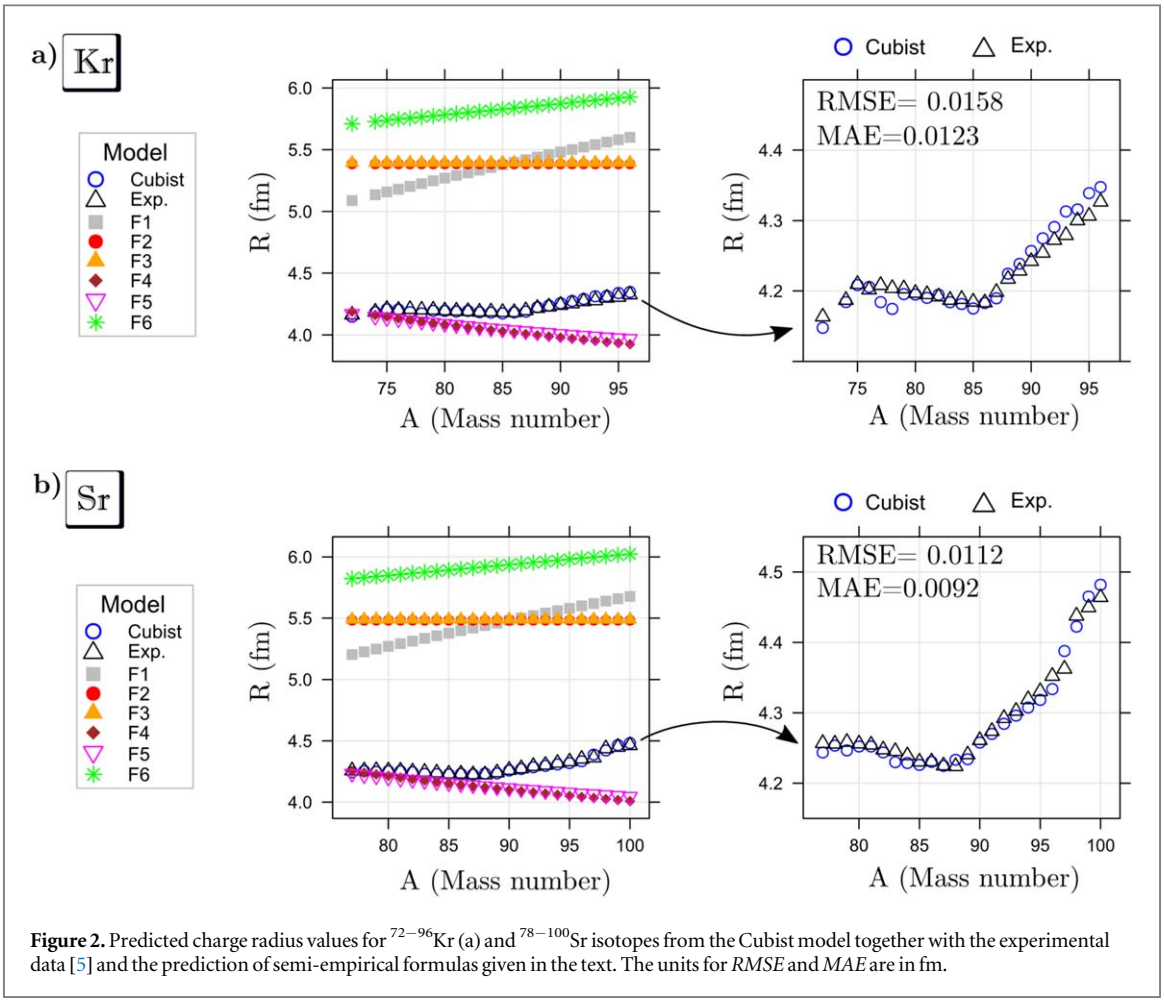

Figure 2. Predicted charge radius values for $^{72-96}$Kr (a) and $^{78-100}$Sr isotopes from the Cubist model together with the experimental data [5] and the prediction of semi-empirical formulas given in the text. The units for *RMSE* and *MAE* are in fm.

$$R_c = r_A\left(1 - b\,\frac{N-Z}{A}\right)A^{1/3} \tag{12}$$

$$R_c = r_A\left(1 - b\,\frac{N-Z}{A} + c\,\frac{1}{A}\right)A^{1/3} \tag{13}$$

$$R_c = r_Z\left(1 + b\,\frac{N-N/Z}{Z}\right)Z^{1/3} \tag{14}$$

In these formulas, R_c, Z, N, and A are charge radii, proton number, neutron number, and mass number of considered nuclei, respectively. r_A, r_Z, r_p, b, and c are fitted parameters by using experimental data. These parameters were fitted by [6] for a better description of nuclear charge radii in the case of $A > 40$ region. For easy follow, the formulas given in equations (9), (10), (11), (12), (13) and (14) will be named as to be F1, F2, F3, F4, F5 and F6 in the text and graphics, respectively. The prediction of the Cubist model for nuclear charge radii of $^{72-96}$Kr and $^{78-100}$Sr are shown in figures 2(a) and (b), respectively. The experimental data [5] and the predictions of semi-empirical formulas are shown for comparison. As can be seen in the figures, the F1, F2, F3, and F6 formulas give results far away from experimental data. The F4 and F5 formulas give close results to experimental data up to $N = 50$ neutron numbers then start to become far. On the right panels of figure 2(a) and (b) kinks around $N = 50$ are seen which means that shell closure is visible on experimental charge radii of *Kr* and *Sr* isotopes. The Cubist model predicts kinks around $N = 50$ for *Kr* and *Sr* isotopic chains very well. It should be noted that the Cubist model predicts the experimental data quite successfully.

Furthermore, phenomenological microscopic nuclear models based on the mean-field approach give close charge radii values to experimental data for nuclei cover nuclidic chart [107, 108]. The kink on nuclear charge radii around $N = 50$ for isotopic chains is produced very well by the relativistic mean field (RMF) model with $NL3^*$ interaction parameter [109]. Because of this reason, we have compared predictions of charge radius values for $^{72-96}$Kr and $^{78-100}$Sr isotopes obtained from either Cubist, MARS, SVR, GPPK, XGBoost, RF, QRF, ANN models together with the predictions of microscopic nuclear models and the experimental data. In figures 3 and 4, the predictions of considered machine learning methods for nuclear charge radii of $^{72-96}$Kr and $^{78-100}$Sr are shown, respectively. The experimental data [5] and calculated results of the HFB (Hartree-Fock-Bogoliubov)

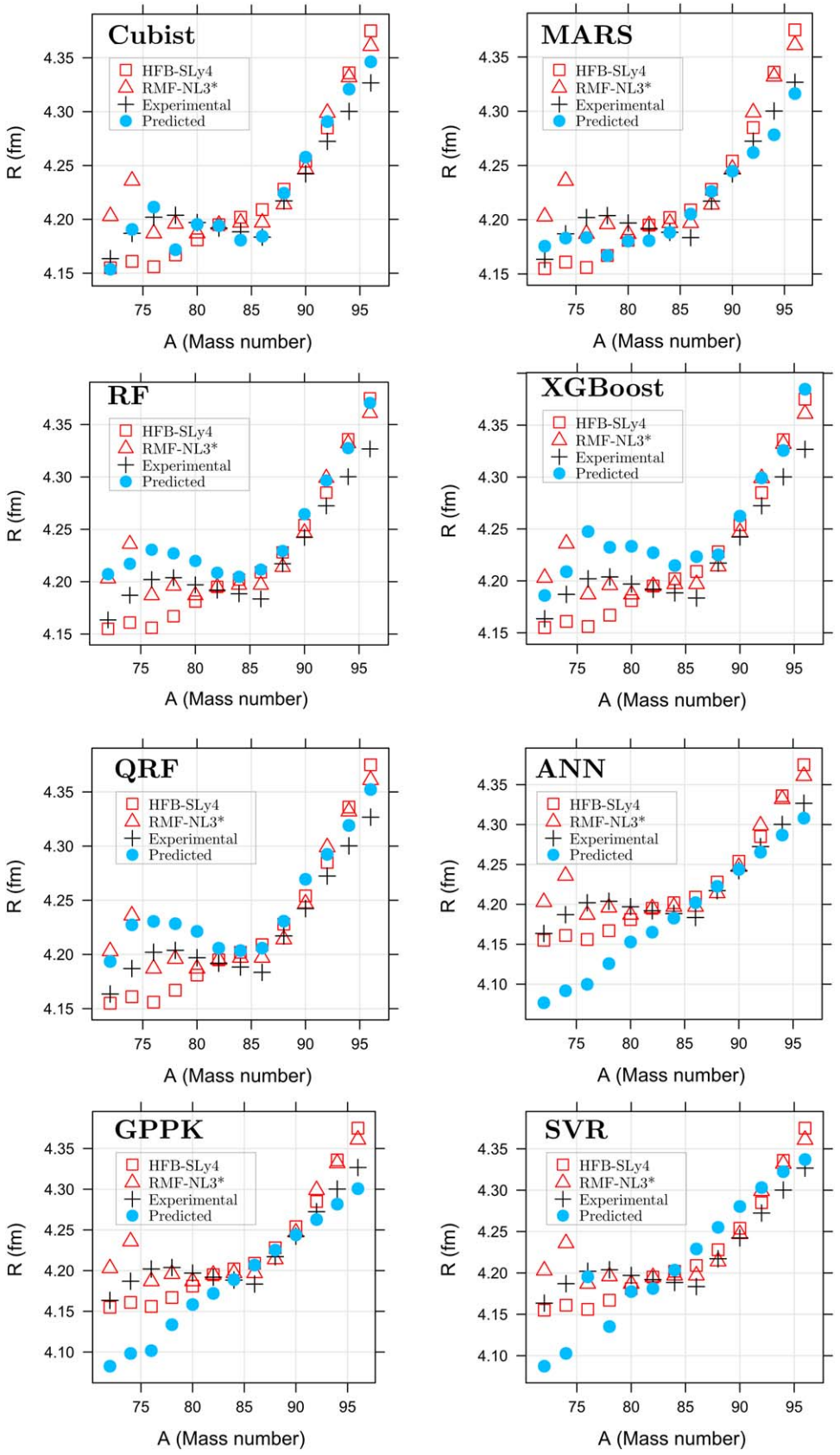

Figure 3. Predicted charge radius values for $^{72-96}$Kr isotopes from Cubist, MARS, RF, XGBoost, QRF, ANN, GPPK, SVR models. The experimental data [5] and the predictions of HFB-SLy4 [108] and RMF-NL3* [109] are shown for comparison.

method with SLy4 force [108] and RMF model with non-linear *NL3** interaction [109] are shown for comparison. As can be seen from the figures, both microscopic models and four machine learning models (Cubist, XGBoost, RF and QRF) are successful in the prediction of the tendency of nuclear charge radii as a

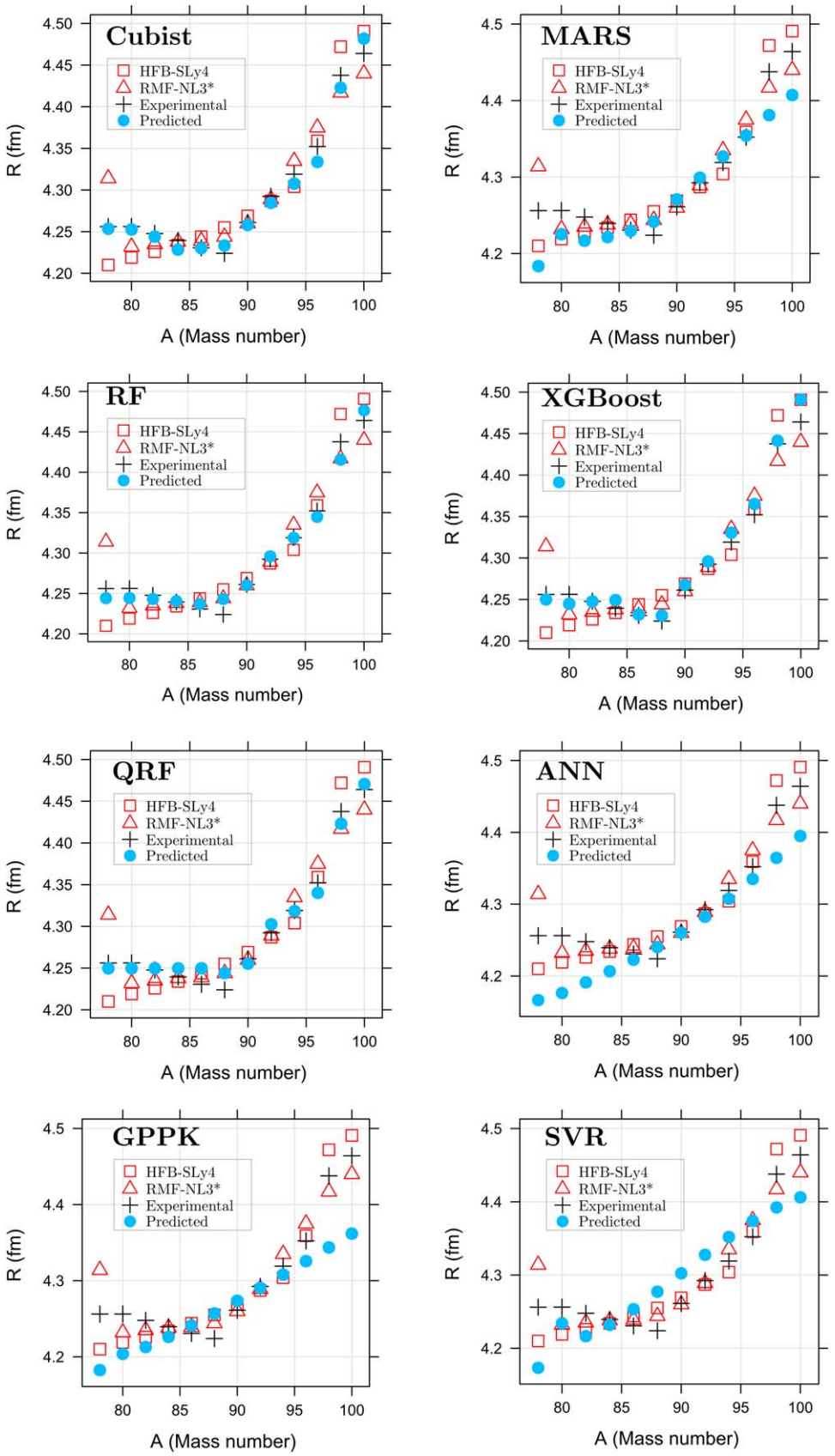

Figure 4. The same as figure 3 but for $^{78-100}$Sr isotopes.

function of mass numbers. However, it should be noted that the kinks on nuclear charge radii around $N = 50$ in figures 3 and 4 are produced well with the RMF model and our machine learning methods Cubist, XGBoost, RF and QRF.

4. Conclusions

The estimation of the nuclear charge radius for atomic nuclei in $A \geqslant 40$ and $Z \geqslant 20$ region was performed with eight different ML models, which is a strong alternative to theoretical models. In the training stage of ML methods, using the available experimental data, it has been shown that ML is a good tool for this purpose, and it is concluded that the Cubist model produces the most successful results. After a detailed analysis of the radius estimates, the success of the ML results in the Kr and Sr isotope chains was compared with the results of some available radius formulas. It has been seen that Cubist, XGBoost, RF and QRF ML models unambiguously detect kinks for isotopes corresponding to magic numbers. In the last stage, the radius values obtained from the microscopic models were examined in comparison with the ML results, again in the Kr and Sr isotope chains. It has been concluded that the four ML method is successful in estimating the charge radii of atomic nuclei with accuracy and even in capturing the unusual behavior in the magic number region. When compared with the formulas obtained from different models and the results produced by the microscopic models, it has been seen that ML is a really powerful alternative method to the theoretical models in estimating the radius, which is clearly successful.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

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.

ORCID iDs

Tuncay Bayram https://orcid.org/0000-0003-3704-0818
Cafer Mert Yeşilkanat https://orcid.org/0000-0002-7508-7548
Serkan Akkoyun https://orcid.org/0000-0002-8996-3385

References

[1] Ring P and Schuck P 2004 *The Nuclear Many-body Problem* (Springer)
[2] Greiner W and Maruhn J A 1996 *Nuclear Models* (Springer)
[3] Buchinger F, Crawford J E, Dutta A K, Pearson J M and Tondeur F 1994 *Phys. Rev.* C **49** 1402–11
[4] Angeli I, Gangrsky Y P, Marinova K P, Boboshin I N, Komarov S Y, Ishkhanov B S and Varlamov V V 2009 *J. Phys. G: Nucl. Part. Phys.* **36** 085102
[5] Angeli I and Marinova K 2013 *At. Data Nucl. Data Tables* **99** 69–95
[6] Zhang S Q, Meng J, Zhou S G and Zeng J Y 2002 *Eur. Phys. J.* A **13** 285–9
[7] Li T, Luo Y and Wang N 2021 *At. Data Nucl. Data Tables* **140** 101440
[8] Athanassopoulos S, Mavrommatis E, Gernoth K and Clark J 2004 *Nucl. Phys.* A **743** 222–35
[9] Bass S A, Bischoff A, Hartnack C, Maruhn J A, Reinhardt J, Stocker H and Greiner W 1994 *J. Phys. G: Nucl. Part. Phys.* **20** L21
[10] David C, Freslier M and Aichelin J 1995 *Phys. Rev.* C **51** 1453–9
[11] Haddad F, Hagel K, Li J, Mdeiwayeh N, Natowitz J B, Wada R, Xiao B, David C, Freslier M and Aichelin J 1997 *Phys. Rev.* C **55** 1371–5
[12] Costiris N, Mavrommatis E, Gernoth K A, Clark J W, Mavrommatis E, Karabarbounis A, Stiliaris E and Tsapalis A 2020 A global model of β-decay half-lives using neural networks *Proceedings of XVI Hellenic Symposium on Nuclear Physics, 2006* ed E Mavrommatis *et al* (Hellenic Nuclear Physics Society) pp 210–7
[13] Akkoyun S, Bayram T and Turker T 2014 *Radiat. Phys. Chem.* **96** 186–9
[14] Bayram T, Akkoyun S and Sevki S 2018 *Phys. At. Nucl.* **81** 288–95
[15] Bayram T, Akkoyun S and Kara S O 2014 α–decay half-life calculations of superheavy nuclei using artificial neural networks *II International Conference on Mathematical Modeling in Physical Sciences 2013 (IC-MSQUARE 2013), Journal of Physics: Conference Series* vol 490 012105
[16] Akkoyun S and Bayram T 2014 *International Journal of Modern Physics E* **23** 1450064
[17] Yesilkanat C M and Akkoyun S 2023 *J. Phys. G: Nucl. Part. Phys.* **50** 055101
[18] Bayram T, Akkoyun S and Kara S O 2014 *Ann. Nucl. Energy* **63** 172–5
[19] Akkoyun S 2020 *Nucl. Instrum. Methods Phys. Res., Sect. B* **462** 51–4
[20] Akkoyun S and Yakhelef A 2022 *Phys. Rev.* C **105** 044309
[21] Akkoyun S, Bayram T, Kara S O and Sinan A 2013 *J. Phys. G: Nucl. Part. Phys.* **40** 055106
[22] Utama R, Chen W C and Piekarewicz J 2016 *J. Phys. G: Nucl. Part. Phys.* **43** 114002
[23] Wu D, Bai C L, Sagawa H and Zhang H Q 2020 *Phys. Rev.* C **102** 054323
[24] Geldhof S *et al* 2022 *Phys. Rev. Lett.* **128** 152501
[25] Dong X X, An R, Lu J X and Geng L S 2022 *Phys. Rev.* C **105** 014308

[26] Dong X X, An R, Lu J X and Geng L S 2023 *Phys. Lett.* B **838** 137726
[27] Ma Y, Su C, Liu J, Ren Z, Xu C and Gao Y 2020 *Phys. Rev.* C **101** 014304
[28] Casten R, Brenner D and Haustein P 1987 *Phys. Rev. Lett.* **58** 658–61
[29] Casten R and Zamfir N 1996 *J. Phys. G: Nucl. Part. Phys.* **22** 1521–52
[30] RCoreTeam 2020 R: A language and environment for statistical computing
[31] Karatzoglou A, Smola A, Hornik K and Zeileis A 2004 *Journal of Statistical Software* **11** 1–20
[32] Liaw A and Wiener M 2002 *R News* **2** 18–22
[33] Meinshausen N 2006 *J. Mach. Learn. Res.* **7** 983–99
[34] Meinshausen N (2017) quantregForest: Quantile Regression Forests *R package version 1.3-7* (https://cran.r-project.org/web/packages/quantregForest/index.html)
[35] Chen T and Guestrin C 2016 Xgboost: A scalable tree boosting system *Proceedings of the XXII ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* pp 785–94
[36] Bergmeir C and Benítez J 2012 *Inf. Sci.* **191** 192–213
[37] Kuhn M, Quinlan R and Weston S (2021) Fit a Cubist model *Cubist version 0.4.3* (https://topepo.github.io/Cubist/reference/cubist.default.html)
[38] Milborrow S (2023) earth: Multivariate Adaptive Regression Splines *R package version 5.3.2* (https://cran.r-project.org/web/packages/earth/index.html)
[39] Kuhn M *et al* (2021) caret: Classification and Regression Training *Caret version 6.0-94* (http://cran.r-project.org/web/packages/caret/index.html)
[40] Carslaw D and Ropkins K 2012 *Environ. Modell. Softw.* **27-28** 52–61
[41] Wickham H 2016 *ggplot2: Elegant Graphics for Data Analysis* (Springer-Verlag)
[42] Rumelhart D E and McClelland J L 1986 *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations* (The MIT Press)
[43] Haykin S 1999 *Neural Networks: A Comprehensive Foundation* (Prentice Hall)
[44] Chen K, Kvasnicka V, Kanen P and Haykin S 2001 *IEEE Trans. Neural Networks* **12** 644–7
[45] Manavi F, Sharma A, Sharma R, Tsunoda T, Shatabda S and Dehzangi I 2023 *Gene* **853** 147045
[46] Hemmat M, Toghraie D and Amoozad F 2023 *Eng. Appl. Artif. Intell.* **121** 105948
[47] Mohammed A and Kora R 2023 *Journal of King Saud University—Computer and Information Sciences* **35** 757–74
[48] Megha O A, Bulusu S S and Banerjee A 2023 *Computational and Theoretical Chemistry* **1220** 113985
[49] Martinez-Hernandez E, Zenteno C, Valencia D and Aburto J 2023 *Fluid Phase Equilib.* **565** 113648
[50] Yeşilkanat C M, Kobya Y, Taşkın H and Çevik U 2017 *J. Environ. Radioact.* **175-176** 78–93
[51] Olmos P M, Murillo-Fuentes J J and Perez-Cruz F 2010 Gaussian processes and its application to the design of digital communication receivers *Application of Machine Learning* ed Y Zhang (IntechOpen)
[52] Perez-Cruz F, Van Vaerenbergh S, Murillo-Fuentes J J, Lazaro-Gredilla M and Santamaria I 2013 *IEEE Signal Process Mag.* **30** 40–50
[53] Rasmussen C and Williams C 2005 *Gaussian Processes for Machine Learning* (The MIT Press)
[54] MacKay D J *et al* 1998 *NATO ASI Series F Computer and Systems Sciences* **168** 133–66
[55] Friedman J H 1991 *The Annals of Statistics* **19** 1–67
[56] Naser A H, Badr A H, Henedy S N, Ostrowski K A and Imran H 2022 *Case Studies in Construction Materials* **17** e01262
[57] Lewis P A W and Stevens J G 1991 *J. Am. Stat. Assoc.* **86** 864–77
[58] Chen W H, Lo H J, Aniza R, Lin B J, Park Y K, Kwon E E, Sheen H K and Grafilo L A D R 2022 *Appl. Energy* **324** 119775
[59] Majeed F, Ziggah Y Y, Kusi-Manu C, Ibrahim B and Ahenkorah I 2022 *Geosystems and Geoenvironment* **1** 100038
[60] Pramanik R, Mukherjee S and Sivakumar Babu G 2022 *Transportation Geotechnics* **36** 100816
[61] Breiman L 2001 *Mach. Learn.* **45** 5–32
[62] Xia Z and Stewart K 2023 *Health & Place* **80** 102986
[63] Yeşilkanat C M 2020 *Chaos, Solitons Fractals* **140** 110210
[64] Kirkwood C, Cave M, Beamish D, Grebby S and Ferreira A 2016 *J. Geochem. Explor.* **167** 49–61
[65] Mandal I and Pal S 2020 *Adv. Space Res.* **66** 1351–71
[66] Stevens F R, Gaughan A E, Linard C and Tatem A J 2015 *PLoS One* **10** e0107042
[67] Mao X, Peng L and Wang Z 2022 *Comput. Stat. & Data Analysis* **170** 107436
[68] Parsa M 2021 *J. Geochem. Explor.* **228** 106811
[69] Žížala D, Minařík R, Skála J, Beitlerová H, Juřicová A, Reyes Rojas J, Penížek V and Zádorová T 2022 *Catena* **212** 106024
[70] Haliduola H N, Bretz F and Mansmann U 2022 *Comput. Methods Programs Biomed.* **226** 107172
[71] Rohmer J 2020 *Stochastic Environmental Research and Risk Assessment* **34** 867–90
[72] Drucker H, Burges C J C, Kaufman L, Smola A and Vapnik V 1996 Support vector regression machines *Advances in Neural Information Processing Systems* ed M Mozer, M Jordan and T Petsche vol 9 (MIT Press)
[73] Cortes C 1995 *Mach. Learn* **20** 273–97
[74] Brereton R G and Lloyd G R 2010 *Analyst* **135** 230–67
[75] Schwieder M, Leitão P, Suess S, Senf C and Hostert P 2014 *Remote Sensing* **6** 3427–45
[76] Ibrahem Ahmed Osman A, Najah Ahmed A, Chow M, Feng Huang Y and El-Shafie A 2021 *Ain Shams Engineering Journal* **12** 1545–56
[77] Wu J, Liu H, Wei G, Song T, Zhang C and Zhou H 2019 *Water* **11** 1327
[78] Friedman J H 2001 *The Annals of Statistics* **29** 1189–232
[79] Zhu X, Chu J, Wang K, Wu S, Yan W and Chiam K 2021 *Journal of Rock Mechanics and Geotechnical Engineering* **13** 1231–45
[80] Ma M, Zhao G, He B, Li Q, Dong H, Wang S and Wang Z 2021 *J. Hydrol.* **598** 126382
[81] Rajliwall N S, Davey R and Chetty G 2018 *Cardiovascular risk prediction based on xgboost 2018 V Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)* pp 246–52
[82] Quinlan J 1992 *Learning with continuous classes Proceedings of the V Australian Joint Conference on Artificial Intelligence (World Scientific)* pp 343–8
[83] Quinlan J 1993 Combining instance-based and model-based learning *ICML'93: Proceedings of the Tenth International Conference, University of Massachusetts (University of Massachusetts)* pp 236–43
[84] Ma Z, Shi Z, Zhou Y, Xu J, Yu W and Yang Y 2017 *Remote Sens. Environ.* **200** 378–95
[85] Kuhn M and Johnson K 2013 *Regression Trees and Rule-Based Models* (Springer) pp 173–220
[86] Chen Q, Yang X, Ouyang Z, Zhao N and Jiang Q 2020 *Environ. Pollut.* **266** 115183
[87] Zhou J, Li E, Wei H, Li C, Qiao Q and Armaghani D J 2019 *Applied Sciences* **9** 1621

[88] Pouladi N, Møller A B, Tabatabai S and Greve M H 2019 *Geoderma.* **342** 85–92
[89] Nguyen H, Bui X N, Tran Q H and Mai N L 2019 *Appl. Soft Comput.* **77** 376–86
[90] Lundberg S M and Lee S I 2017 *Advances in Neural Information Processing Systems* ed I Guyon *et al* (Curran Associates, Inc.)
[91] Ribeiro M T, Singh S and Guestrin C 2016 Why should i trust you? *Proceedings of the XXII ACM SIGKDD International Conference on Knowledge Discovery and Data Mining∀Why Should I Trust You?∀: Explaining the Predictions of Any Classifier* pp 1133–1144
[92] Lamichhane S, Adhikari K and Kumar L 2022 *Geoderma Regional* **30** e00568
[93] Morellos A, Pantazi X E, Moshou D, Alexandridis T, Whetton R L, Tziotzios G and Mouazen A M 2016 *Biosystems Eng.* **152** 104–16
[94] Otero J, Ochoa E, Tanarro E, Morgado P, Lantada A, Munoz-Guijosa J and Sanz J 2013 *Lubr. Sci.* **26** 141–62
[95] Horák J, Šuleř P and Vrbka J 2021 *Communications* **23** A32–43
[96] Xing L, Goulet R J and Johnson K 2011 *J. Chem. Inf. Model.* **51** 1582–92
[97] Willemsen F, van Nieuwpoort R and van Werkhoven B 2021 Bayesian optimization for auto-tuning GPU kernels arXiv:2111.14991
[98] Khaledian Y and Miller B S 2020 *Appl. Math. Modell.* **81** 401–18
[99] Xu R, Nettleton D and Nordman D J 2016 *Journal of Computational and Graphical Statistics* **25** 49–65
[100] Gupta R, Pierdzioch C, Vivian A and Tiwari A K 2019 *Finance Research Letters* **29** 315–22
[101] Afsaneh E and Ghobadi M 2022 *Diabetol. Metab. Syndr.* **14** 196
[102] Gupta A and Singh A 2023 *Wireless Pers. Commun.* **131** 1013–31
[103] Hu Y M, Liang Z M, Liu Y W, Wang J, Yao L and Ning Y 2015 *Int. J. Climatol.* **35** 1847–57
[104] Royer G 2008 *Nucl. Phys.* A **807** 105–18
[105] Bayram T, Akkoyun S, Kara S O and Sinan A 2013 *Acta Phys. Pol.* B **44** 1791–9
[106] Royer G 2009 *Eur. Phys. J.* A **42** 541–5
[107] Lalazissis G, Raman S and Ring P 1999 *At. Data Nucl. Data Tables* **71** 1–40
[108] Stoitsov M V, Dobaczewski J, Nazarewicz W, Pittel S and Dean D J 2003 *Phys. Rev.* C **68** 054312
[109] Bayram T and Yılmaz A H 2013 *Mod. Phys. Lett.* A **28** 1350068