Annals of Nuclear Energy 161 (2021) 108409

Contents lists available at ScienceDirect

Annals of Nuclear Energy

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

Machine learning application to single channel design of molten salt reactor

Mehmet Turkmen [a],*, Gwendolyn J.Y. Chee [a], Kathryn D. Huff [a]

[a] Department of Nuclear, Plasma, and Radiological Engineering, University of Illinois at Urbana-Champaign, MC-234 104 South Wright Street, Urbana 61801, IL, USA

ARTICLE INFO

Article history:
Received 19 December 2020
Received in revised form 1 April 2021
Accepted 11 May 2021
Available online 11 June 2021

Keywords:
Channel design
Molten salt reactor
Machine learning
Optimization
Simulation
Monte Carlo

ABSTRACT

This study proposes a robust approach to quickly design a nuclear reactor core and explores the best performing machine learning (ML) technique for predicting feature parameters of the core. We implemented the approach into a hypothetical channel of molten salt reactors to demonstrate the applicability of the method. We prepared a Python tool, named Plankton, which couples to a reactor physics code and an optimization tool, and imports ML methods. The tool performs three consecutive phases: reactor database generation, machine learning application, and design optimization. We identified the extra trees method as the best performing estimator. With the estimator, we found nine optimum designs in total, one for each fuel-salt pair, and estimated all the performance metrics of the designs with a $<5\%$ prediction error compared to their actual values. U-Pu-NaCl fuel-salt gave promising results with the highest conversion ratio, the most negative feedback coefficient, and the lowest fast flux.

1. Introduction

Current applications of learning-based techniques in nuclear engineering indicate that Artificial Intelligence (AI) will soon undertake the role of reactor designers in optimizing the designs of current and future reactors. AI methods, more specifically ML, have already become important to most sub-fields of nuclear science and engineering (e.g., in-core fuel management, reactor safety, flow regime prediction, and radiation protection) to achieve an optimal core loading map, to make nuclear reactors safer, to quickly comprehend flow data, and to accurately identify critical isotopes in nuclear materials. In this context, recently published two review papers (Gomez-Fernandez et al., 2020; Nissan, 2019) showed the potential, competence, and importance of these methods by exploring the recent studies on the use of learning-based methods in nuclear science and relevant fields. The articles also probed the suitability of the commonly used algorithms and algorithm selection criteria. However, AI/ML has not contemplated as an option to seek to find optimal designs of nuclear reactor cores despite being a proven technique for optimization. Instead, iterative approaches such as individual parameter searches and evolutionary algorithms have been widely employed for reactor core design optimization.

Among parametric studies recently reported, Wei et al. (2018) investigated several critical feature parameters of a Molten Salt Reactor (MSR) channel by parameter interpolation using KENO-VI in SCALE 6.1 (Wieselquist et al., 2020). In the study, critical feature parameters are the fuel type, ^{6}Li concentration, composition of fuel type, channel geometry, moderator and its density, and salt operation temperature. The research showed that ^{6}Li has a strong effect on criticality. In all different shapes of fuel channels, Wei et al. (2018) observed that the turning point from under-moderation to over-moderation for the multiplication factor (k) is at a pitch to channel ratio (P/D) between 2 and 4, and a fuel volume fraction between 0 and 0.1. Increasing the amount of moderator resulted in an increase in k, but then further increase caused a decrease. Meanwhile, the temperature coefficient changes from negative to positive for pitch >12 cm.

Anderson et al. (2019) assessed the importance of several parameters (fissile enrichment, lattice pitch, salt-moderator ratio, and the number of fuel channels) in a MSR core by looking into the change of fission and removal transition matrices. In that work, the results showed that some of these parameters have an increasing impact on the average relative difference of transition matrices while others have a decreasing effect.

The common deficiency of both studies is the neglect of strong interdependence of those investigated parameters, by carrying out only a single parametric study. The effect of a single parameter on performance metrics (multiplication factor, conversion ratio, reactivity feedback coefficient, etc.) can be either beneficial or detri-

* Corresponding author.
E-mail address: mturkmen@illinois.edu (M. Turkmen).

https://doi.org/10.1016/j.anucene.2021.108409
0306-4549/© 2021 Elsevier Ltd. All rights reserved.

mental. As for more than one parameter, correlations between parameters can be seen as superpositioning, canceling out another, or acting on each other up to some extent. Correspondingly, such multi-parameter optimizations would require more sophisticated searching techniques like genetic algorithm, particle swarm, and simulated annealing methods. Unlike parametric studies, these sophisticated optimization techniques are very useful in finding optimal parameters, particularly for multi-objective problems, and are a common and effective way of determining optimal core loading maps (Nissan, 2019).

In the case of using evolutionary algorithms in core designs, Kumar and Tsvetkov (2015) used a Genetic Algorithm (GA) method coupled to regression splines to optimize several performance metrics such as multiplication factor, fast fission factor, thermal efficiency, and burn up in the design of Gas Cooled Fast Breeder reactor. That method interpolates intermediate values of parameters and metrics. Regarding a fuel pin cell, optimal design values of radius, enrichment fraction, mass flow rate, and coolant temperature at the inlet were found to be 0.22 cm, 18.5 wt.%, 63.5 ^{o}C, and 35.9 kg/s, respectively. Zeng et al. (2020) performed a multi-objective optimization to the Advanced Burner Test Reactor (ABTR) core, a sodium-cooled fast test reactor (SFR), to get an ultimate core design. In that work, an integrated version of the US Department of Energy's Nuclear Energy Advanced Modeling and Simulation (NEAMS) Workbench (Lefebvre et al., 2019) along with GA was used for core modeling and optimization. Although they found about 3150 optimal core designs within Pareto front solutions by optimizing reactivity swing, Pu mass feed, core volume, core power, and peak fast flux, only six candidate designs passed their performance criteria based on their defined constraints.

Pereira et al. (1999) and Pereira and Lapa (2003) investigated a hypothetical reactor design in a cylindrical fuel rod geometry (i.e., composed of a moderator, clad, and fuel regions) to validate the GA applicability and to make a comparison with classical linear optimization methods. In the first study, only two continuous feature parameters were considered for optimization. Optimal designs were completed using 3500 reactor simulations (100 populations and 35 generations). Later, a more complex version of the previous problem was repeated with continuous and discrete variables, very identical to our problem from the aspects of types and number of feature parameters. Optimization results were obtained with 150000 reactor simulations (300 populations and 500 generations).

Recently published studies (Kim et al., 2018; Kim et al., 2020) reported the results of an assembly and core design at the Kyoto University Critical Assembly through an Artificial Neural Network (ANN) coupled to MCNP6 (Werner et al., 2017). These studies are similar to fuel loading pattern optimization studies in terms of the optimization approach. In these works, different core materials, the number of fuel regions, and fuel assembly types were defined as vital feature parameters whereas fast flux and multiplication factor were the target metrics to be optimized.

Regarding the use of ML methods in depletion prediction of reactor applications, Bae et al. (2020) successfully predicted Pressurized Water Reactor (PWR) Used Nuclear Fuel (UNF) composition by training the dense neural network in the Keras (Chollet et al., 2015). They showed that their proposed model outperforms the average recipe method, one of the transmutation methods which supplies neutronic calculations externally to the nuclear fuel cycle simulators, by yielding less than 1% error for the UNF inventory decay heat and activity and less than 5% error for the important isotopes.

As discussed previously, parametric studies which handle all feature variables independently ignore the complex interdependence of these variables and evaluate performance metrics by changing only one feature variable at a time, thus fixing others. Meanwhile, evolutionary algorithms suffer from the choices of optimizer hyperparameters such as population size, number of generations, and cross-over rate since it is unlikely to determine exactly what these parameters would be without making a preliminary calculation. Additionally, these optimization techniques do not provide any flexibility to designers, such as making optimization without running reactor simulator again when constraints are changed. This hence results in excessive usage of computational resources. Furthermore, optimization methods occasionally converge to local instead of global optima fail to achieve ideal designs that are eliminated by the optimizer's elimination and/or selection criteria. At this point, if sufficient data is provided, it seems rational to use learning-based methods to design a nuclear reactor. This is because they not only have an exceptional ability to accurately predict performance metrics but are also suitable for use with optimization tools as a reactor simulator.

The ML methods, which have not been considered for the sole design of a reactor core so far, will be the best option in this regard. This article compares various ML methods in terms of performance scores and recommends employing these methods in place of neutron transport codes for the prediction of performance metrics. This research demonstrates that the proposed method predicts performance metrics accurately and quickly.

2. Methodology

In this section, we described how to apply ML methods to design a reactor core. We followed a strict methodology consisting of three separate phases: (1) database generation, (2) machine learning, and (3) design optimization. For this, we developed a Python tool, called Plankton. The Plankton produces training and test datasets for given feature parameters, qualifies a best-performing method among the selected ML methods based on their performance scores, trains a predictive model for the method, predicts performance metrics of designs, and finds optimal designs. The tool calls a reactor physics code to compute the performance metrics in the database generation phase and applies an optimization technique to reveal optimal designs in the optimization phase. It uses utilities from the scikit-learn tool (Pedregosa et al., 2011) for regression/classification, preprocessing, postprocessing, and the quantification of predictions' quality. A computational flow chart showing the main computing steps of the Plankton code is displayed in Fig. 1. The following sections describe computational steps in detail.

2.1. Plankton

2.1.1. Database Generation

Database generation is a step in which samplings in datasets are created and, training and test datasets are produced based on two distinctive sampling strategies, Grid sampling and Monte Carlo.

The grid sampling strategy, the simplest exploration approach to explore an uncertain domain, was used to obtain the training dataset. It constructs an N-dimensional grid in which each dimension accounts for a single feature parameter. Also, each dimension is discretized with the step values (the rightmost column in Table 3) into equally-spaced intervals between its upper and lower bounds. A sample in a dataset is represented by a specific node in the grid system.

Unlike the grid strategy, the test dataset was generated by the Monte Carlo technique, where the value of each variable is randomly determined between its lower and upper limits.

As the size of the training dataset has a significant effect on performance scores, the criteria for the required size were discussed in detail in the learning curve section. For this, we examined estima-

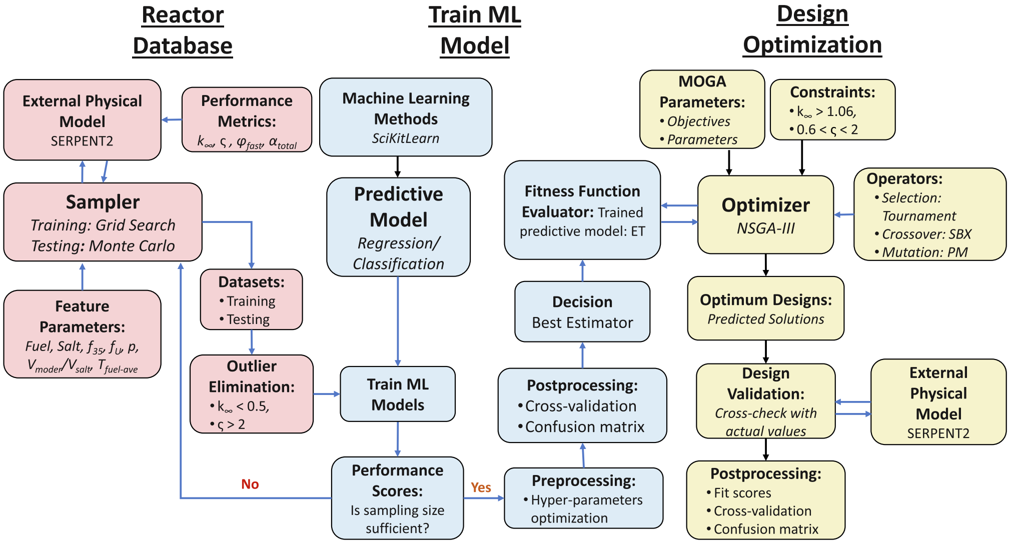

Reactor Database
Train ML Model
Design Optimization
External Physical Model SERPENT2
Performance Metrics: k_∞, ς, φ_{fast}, α_{total}
Machine Learning Methods SciKitLearn
Sampler
Training: Grid Search
Testing: Monte Carlo
Feature Parameters: Fuel, Salt, f_{35}, f_U, p, V_{moder}/V_{salt}, $T_{fuel\text{-}ave}$
Datasets:
• Training
• Testing
Outlier Elimination:
• k_∞ < 0.5,
• ς > 2
Predictive Model Regression/ Classification
Train ML Models
Performance Scores: Is sampling size sufficient?
No
Yes
Preprocessing:
• Hyper-parameters optimization
Postprocessing:
• Cross-validation
• Confusion matrix
Decision Best Estimator
Fitness Function Evaluator: Trained predictive model: ET
MOGA Parameters:
• Objectives
• Parameters
Constraints:
• k_∞ > 1.06,
• 0.6 < ς < 2
Optimizer NSGA-III
Operators:
• Selection: Tournament
• Crossover: SBX
• Mutation: PM
Optimum Designs: Predicted Solutions
Design Validation: Cross-check with actual values
External Physical Model SERPENT2
Postprocessing:
• Fit scores
• Cross-validation
• Confusion matrix

Fig. 1. Computational flowchart of the Plankton code.

tors' learning curves in conjunction with the cross-validation splitting strategy, which examines the change in the validation and training score of an estimator for which the number of training samples varies.

Computational steps followed in the database generation phase are as follows:

- Initialize feature parameters, properties of which are given in Table 3, for the grid search technique and Monte Carlo method.
- Prepare input files for the reactor physics code for neutron transport calculation and simulate each sample in the training and test datasets.
- Calculate performance metrics and extract them from the code output.
- Supply the prepared reactor database, a set of training/test samples that comprise the feature parameters with the corresponding performance metrics, to the machine learning phase.

2.1.2. Machine learning

In this part, we employed regression analysis for the estimation of performance metrics and considered classification analysis, which differs from the regression part in terms of feature parameters, to determine classes of a design. We intended to separately use both methods within the scope of this study, but we are also planning to integrate them for the determination of optimal designs for future work. We scoped out various regressors and classifiers from ML methods within the scikit-learn python package (Pedregosa et al., 2011) in order for the assessment of the best performing estimator(s). Table 1 lists the ML methods including semi-supervised and supervised methods such as neural network (NN), ensemble, linear, support vector machine (SVM), decision trees (DTs), naïve Bayes (NB), and nearest neighbours methods. Because there is more than one performance metric, estimators require a built-in or external multi-class/multi-label support. We, therefore, used the multi-target multi-output method in regression and multi-label multi-output method in classification when necessary. We performed a comprehensive performance assessment by examining various regression and classification metrics given in Table 2. Prior to regression/classification analyses, we standardized our datasets by scaling each feature parameter between zero and one.

We evaluated the performances of regressors by cross-validation plots and the performance of classifiers by confusion matrices. Cross-validation is for the visualization of prediction errors whereas confusion matrix is a table that contains all the defined classes in both the horizontal and vertical directions. In a confusion matrix, predicted outputs are listed along the top of a table whereas actual values are listed on the left-hand side column of the table. Anything on the primary diagonal is correctly predicted values. Values above and below the primary diagonal are incorrectly predicted labels called false negative and false possitive.

Table 1
Examined ML methods (Pedregosa et al., 2011).

Method	Estimator
ensemble	random forest (RF), bagging (B) AdaBoost (AB), extra trees (ET) gradient boosting (GB)
linear	elastic net (EN), Perceptron (P), ridge (R) stochastic gradient descent (SGD) lasso (L), lasso lars (LL), logistic (Log)
NB	Gaussian NB (GNB), Bernoulli NB (BNB)
nearest neighbors	k-nearest neighbors (KN)
SVM	SVM
NN	Multi-layer Perceptron (MLP)
DTs	decision tree (DT)
semi-supervised	label propagation (LP)

Table 2
Metrics for the performance assessment of estimators (Pedregosa et al., 2011).

Regression	Classification
fit score: ∈ [0, 1], best score: 1	fit score: ∈ [0, 1], best score: 1
mean square error (MSE), best score: 0	average precision (AP): ∈ [0, 1], best score: 1
mean absolute error (MAE), best score: 0	ranking-based average precision (LRAPS): ∈ [0, 1], best score: 1
explained variance regression score (EVR): ∈ [0, 1], best score: 1	label ranking loss (LRL): ∈ [0, 1], best score: 0
R^2: ∈ [0, 1], best score: 1	average Hamming loss (HL): ∈ [0, 1], best score: 0
	computed area under the receiver operating characteristic curve (ROCAUC): ∈ [0, 1], best score: 1

Table 3
Feature parameters for database generation.

Variable	Distribution	Material types		Steps
Fuel Type	Discrete	U, U-Th, U-Pu		3
Salt Type	Discrete	LiF–BeF$_2$, NaCl,	NaF-BeF$_2$	3
		Lower value	**Upper value**	
f_{35}	Continuous	1 wt.%	20 wt.%	10
f_U	Continuous	10 wt.%	90 wt.%	9
p	Continuous	1 cm	16 cm	10
ξ	Continuous	0.274	10.46	10
T_{ave}	Continuous	900 K	1200 K	5

For the optimization of hyperparameters of estimators, we performed an exhaustive search over specified parameter values to further improve the scores of estimators. The scoring parameters were set to f1 samples in classification and coefficient of determination (R^2) in regression.

Computational steps followed in the machine learning phase are as follows:

- As a starting point, use various ML methods listed in Table 1 from the scikit-learn tool.
- With the default settings of hyperparameters, assess potential estimators to be used in training just by looking into their fit scores and exclude the worst ones from the list of the potential estimators.
- Analyze learning curves of the selected estimators to understand the optimal size of a training dataset required for high-quality training.
- Make a hyperparameter optimization for the selected estimators to further improve the scores of the estimators.
- Eliminate outliers according to the performance metrics of the estimators seemingly impracticable from the viewpoint of designing a reactor and out of interest in order to further improve the estimators' scores and focus on only feasible solution domain.
- Select the best estimator(s) according to the fit scores of the estimators.
- To better understand the selected estimators' capability, get the final scores of the estimators, cross-validate the results of the regressors, and draw confusion matrices of the classifiers.
- Supply the best estimator(s) (along with the optimized hyperparameters) to the design optimization phase to predict the performance metrics of objective functions.

2.1.3. Design optimization

In the final stage of the channel design, we aimed at finding optimal designs. We implemented a multi-variable multi-objective evolutionary algorithm appropriate to design space's structure which manifests multimodal rather than unimodal. We selected this method because optimization methods such as gradient-based or non-linear least-square do not accept discrete and non-numeric variables, and do not work well in non-convex design spaces such as this problem.

We defined the number of generations as a convergence stopping criterion. After finding the optimal designs, we rerun Serpent tool (Leppanen et al., 2014) with their feature parameters to cross-check the solutions. The acceptance criterion for the accuracy of the optimal designs was selected as:

$$\frac{x_{i,a} - x_{i,p}}{x_{i,a}} \leqslant 5\% \tag{1}$$

where $x_{i,a}$ is the actual value of $i th$ metric and $x_{i,p}$ is the predicted value of $i th$ metric.

Computational steps followed in the design optimization phase are as follows:

- Specify an optimization tool (Multi-Objective Genetic Algorithm (MOGA) method) consistent with the structure of the feature parameters and the problem at hand.
- Determine tool's parameters such as decision variables, operators, number of generations, and initial population size.
- Define constraints for the performance metrics to eliminate undesired designs.
- Integrate the optimizer with the best performing estimator (in place of reactor physics code) to evaluate the performance metrics of objective functions.
- Iterate this ML-GA coupled system until either the constraints are met or the solutions are converged to one of the predefined criteria.
- Validate Pareto-front (predicted) solutions with actual values (with additional Serpent simulation).

2.2. Channel design with plankton

2.2.1. Design parameters

We applied the suggested method to a single fuel channel MSR geometry with square unit cell approximation to use in a basic geometry with the least number of feature parameters and performance metrics. An illustration of the channel geometry is shown in

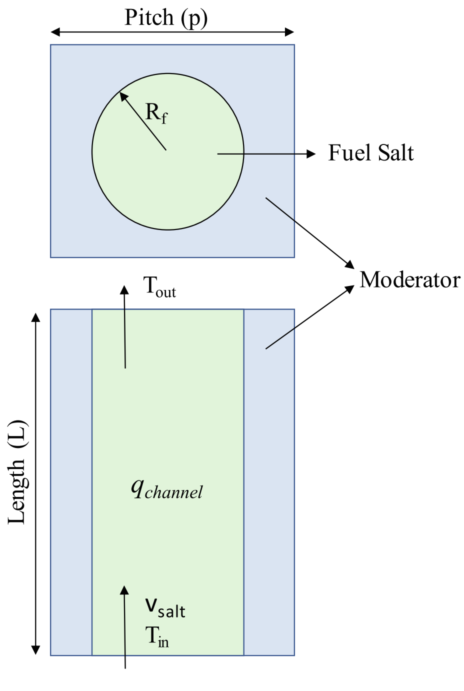


Fig. 2. Single fuel channel representation.

Fig. 2. In the figure, p is the pitch length, T_{in} and T_{out} are inlet and outlet temperatures, v_{salt} is the velocity of fuel-salt, L is the channel length, R_f is the radius of the fuel-salt channel and $q_{channel}$ is the total heat generation in the channel. Feature parameters describing material and geometry properties of the channel were selected as fuel type, salt type, ^{235}U fraction (f_{35}) in U, U fraction (f_U) in heavy metal, channel-to-channel pitch length (p), moderator-to-salt volume ratio (ξ), and average fuel-salt temperature (T_{ave}) along the channel. We varied these variables between upper and lower bounds described in Table 3. The ^{235}U fraction in U was limited to 20 wt.% which is the maximum limit for High Assay Low Enriched Uranium (HALEU). For the fuel types other than U fuel type, U content in heavy metal varies from 10 to 90 wt.%. The maximum p, based on Oak Ridge National Laboratory (ORNL)'s Molten Salt Reactor Experiment (MSRE) design (Robertson, 1971), was set to 16 cm. From the ratio of pitch length to fuel-salt channel radius (p/R_f), the ξ variable was calculated and found in the range of 0.274 (p/R_f = 0.5) and 10.46 (p/R_f = 1.5).

We studied three different fuel types, U, U-Th, and U-Pu in this work. Th in U-Th contains only ^{232}Th isotope whereas Pu in U-Pu comes from the weapons-grade Pu constituting the following isotopes: ^{238}Pu (0.05 at.%), ^{239}Pu (94.3 at.%), ^{240}Pu (5 at.%), ^{241}Pu (0.6 at.%) and ^{242}Pu (0.05 at.%). It is assumed that there is no ^{234}U in U as a parasitic absorber.

We examined three types of different salts widely used in prototype reactors by various companies (Terrestrial Energy, Terrapower, Thorcon, etc.): LiF–BeF$_2$, NaF-BeF$_2$, and NaCl. The upper value of the average fuel-salt temperature (T_{ave}) is set to 1200 K while its base temperature used to calculate the total feedback coefficient of reactivity is 900 K. As the temperature is the main driver for fuel-salt density and Doppler effect of feedback, salt density variation with the T_{ave} was taken into account based on the reports (Janz, 1988; Jerden, 2019).

In the regression analysis, the most critical performance metrics are selected as infinite multiplication factor (k_∞), conversion ratio (ς), fast flux (ϕ_{fast}) incident on the graphite moderator, and total feedback coefficient (α_{total}). But, we did not include the α_{total} as a performance metric since this metric caused huge errors in scores, probably due to zero values at 900 K. Yet, we promptly and accurately calculated it from Eq. 2 using k_∞ at the end of the design optimization phase.

In the classification analysis, the α_{total} metric did not cause any trouble so that we were able to use it. We labeled the single channel design with different labels according to its spectrum, type, criticality condition, and positive or the negative sign of the feedback coefficient. Accordingly, we addressed any channel design as breeder type when $\varsigma \geqslant 1.0$, as fast spectrum when energy corresponding to average lethargy of neutrons causing fission (EALF) $\geqslant 1eV$, as supercritical condition when $k_\infty \geqslant 1.0$, and as positive feedback when $\alpha_{total} \geqslant 0.0$. For the opposite of these conditions, the design is labeled as burner type, thermal spectrum, sub-critical condition, and negative feedback, respectively.

Associated with the channel design, we made the following approximations, simplifications, and approaches involved in the model/method development:

- The unit cell approximation in a square lattice geometry was chosen to represent the single channel geometry. It is an effective geometry structure for periodicity.
- The fuel-salt was assumed to be stationary, by neglecting thermal–hydraulic effects such as delayed neutron precursor movement.
- Calculations were performed for initial fuel loading assuming fresh fuel.
- The total heat generation rate along the channel was expressed in terms of an arithmetic average of the inlet and outlet temperatures, i.e., T_{ave} = (T_{in}+T_{out})/2, thereby making the design independent from the length of the channel.

Using these approximations, we eliminated four interdependent variables: inlet and outlet temperatures, heat generation rate, and channel length. We reduced the total number of feature parameters from eleven to seven parameters.

2.2.2. Optimization parameters

We used the non-dominated sorting genetic algorithm (NSGAIII) (Deb and Jain, 2014; Jain and Deb, 2014) from the pymoo tool (Blank and Deb, 2020) as an optimizer with the simulated binary crossover (SBX) and polynomial mutation (PM) operators. The population size and number of generations were set to 150 and 5000, respectively. The tournament selection, in which a set of chromosomes are selected randomly and then the fittest chromosomes are selected for further operation, was employed to select the fittest candidates from the current generation.

Several independent constraints imposed by the basic requisites of a MSR design were implemented to the solution objectives in line with the given references (Robertson, 1971; Rykhlevskii et al., 2019; Betzler et al., 2017; Ashraf et al., 2020) in the following way: We searched for k_∞ greater than 1.06 to make up for neutron leakage. Despite the MSRs typically have a conversion ratio of around 1.0, ς was chosen as higher than 0.6 to include as many designs as into the optimization process. As there is no reported constraints on ϕ_{fast}, we simply assumed ϕ_{fast} per source neutron on graphite to be lower than half (50) of what the maximum value ($\sim$ 100) of the fast flux in the training dataset is.

For optimization, feature parameters were assigned as decision variables, while performance metrics were assigned as solution objectives (or objective functions). The k_∞ and ς metrics were maximized while the ϕ_{fast} was minimized.

2.2.3. Neutronic simulator

We employed a Monte Carlo neutron transport tool, Serpent 2 (2.1.31) (Leppanen et al., 2014) to compute performance metrics. It uses ENDF/B-VII.0 (Chadwick, 2006) as the neutron cross-section library alongside the S(α, β) thermal scattering libraries. In the simulator, neutron population per cycle, passive cycle, and active cycle were set to 2000, 50, and 100, respectively, to yield a computational standard error of less than 100 pcm in the k_∞. A two-group neutron energy structure for the calculation of ϕ_{fast} on the moderator was used, and lower and upper energy boundaries for the fast region were set to 0.625 eV and 20 MeV, respectively. The k_∞, ς, and ϕ_{fast} metrics were directly extracted from the output file whereas α_{total} was calculated by Eq. 2:

$$\alpha_{total} = \frac{\partial \rho}{\partial T} \approx \frac{\Delta \rho}{\Delta T} = \frac{\rho - \rho_b}{T - T_b} \tag{2}$$

where ρ is the reactivity, ρ_b is the reactivity at the base temperature, T is the temperature, and T_b is the base temperature. However, we purposefully avoided using $\partial\rho/\partial T$ in estimating the α_{total} since it goes to infinity when T_{ave} is very close to the base temperature. Instead, we used ∂k(mk) and converted to α_{total} at the end of the calculations. For the estimation of reactor spectrum in classification, we used the EALF value given in the code output.

2.2.4. Datasets

In this work, a total training sample size of 285000 and a total test sample size of 8000 were used.

3. Results

3.1. Preselection

For the estimators listed in Table 1, we presented performance score in Table 4 for regression analysis and in Table 5 for classification analysis. The results were obtained using the default hyperparameters. RF, ET, KN, SVM, and MLP estimators predicted the performance metrics with significantly higher accuracy than the others. Even without hyperparameter optimization, R^2 scores of these five estimators ranged from 90 to 98% for regression and from 92 to 96% for classification. This implies that the trained models by these estimators will have very low variance. The ET regressor most accurately predicted the performance metrics while the SVM classifier was superior to the others in classifying designs.

In general, the linear methods (i.e., SGD, R, EN, L, LL, P, and Log) yielded lower fit scores and higher errors with respect to the nonlinear methods such as ensemble, SVM, and ANN. These results clearly imply that linear methods are not quite enough to design a nuclear reactor and thus in predicting its performance metrics. Like linear methods, the NB classification methods such as GNB and BNB seem inappropriate due to the suboptimal performance scores. This is because these methods such as linear and NB are not sufficient to fit the training dataset (leading to underfitting) into a linear function due to the higher degree interdependencies of our feature parameters. In addition to this, the RF, DT, B, and AB estimators, gave similar results, except for fit score values, as they use the same base estimator (DT) in training. Likewise, the LP classifier produced the same results with the SVM classifier due to the use of the same kernel (*rbf*). In regression analysis, the ET regressor has the lowest MAE and MSE scores while the R regressor has the highest scores.

As to the classification, the SVM classifier has the lowest HL and LRL scores while the P classifier has the highest scores. Briefly, any design can be classified with high precision ($> 95\%$) and very few incorrectly predicted labels (false neagtive plus false positive: $<$ 5%), leading to an opportunity for an accurate prediction of reactor design even without making any regression analysis.

On the whole, five estimators (i.e., RF, ET, KN, SVM, and MLP) that satisfy a score of more than 90% in R^2 and fit scores appear to be suitable for the subsequent stage of calculations. Among the selected estimators, the SVM regressor interestingly shows the second-best performance right after the ET regressor despite its worst fit score. In fact, as the scores of the best performing estimators are very close to each other, at this point, it is not easy to distinguish which estimator is the best. Therefore, in the following section, the current scores of these estimators are further improved by tuning their hyperparameters and investigating their learning curves.

3.2. Hyperparameters optimization and learning curve

We applied two distinct methods to the estimators to improve their performance. The first method, expected to have an impact on the R^2 scores, prediction errors, and thus cross-validation results, aims to find an optimal size for the training dataset. For this, Fig. 3 demonstrates the estimator's learning curves for the (U-Th) F_4-NaF-BeF_2 fuel-salt pair from the viewpoint of R^2 score in regression and multi-label classification accuracy in classification.

At first look, the MLP and SVM predictive models do not require more data since the training and cross-validation scores converge together. On the contrary, the RF, ET and KN models require more data as their training scores are much higher than their cross-validation scores.

From all the learning curves, the RF, ET, MLP, and SVM models require a training size in the range of 32000 and 40000 in which the curves level off to a R^2 score of 0.99 whereas the KN model requires sample size more than 40000. A similar situation was observed in the classifiers' learning curves. As a result, we decided to use about forty thousand samples for each fuel-salt pair. In other words, for nine fuel-salt pairs, we generated around three hundred thousand samples in total. In the second method, we obtained optimal hyperparameters of the estimators tabulated in Table 6. Parameters not listed in the table are at their default values and do not need optimization.

3.3. Final estimators

After performing the hyperparameter optimization and learning curve assessment, we evaluated the final scores of the selected estimators and listed them in Table 7 and 8. At a glance, we saw that there is a slight enhancement in the performance scores of all regressors, except for the scores of the MLP regressor that exhibits a strong hyperparameter dependency. However, the hyperparameter optimization did not improve the scores of the classifiers. With the optimized hyperparameters and data size in the training dataset, the ET, SVM, and MLP regressors predicted performance metrics with a higher R^2 and lower MAE compared to their preselection scores.

On the other hand, the SVM and MLP classifiers labeled the design attributes with the highest AP and lowest LRL scores relative to the other estimators. Nonetheless, the others are also suitable as all the scores are very close to each other. As the scores did not give us sufficient information about determining the right estimator for the next phase, we investigated cross-validation plots and multi-label confusion matrices on the basis of each performance metric (i.e., k_∞, ς, ϕ_{fast}).

3.4. Cross-validation plot and confusion matrix

In this section, we compared the predicted test dataset with the actual test dataset. Fig. 4 visualizes the predicted performance metrics against their actual values for the chosen estimators in terms of cross-validation plots. These plots show the estimator's prediction quality separately for each performance metric. Results indicated that the ET, KN, and MLP estimators accurately predict performance metrics per the acceptance criterion (within $<$5% relative error).

According to these results, the ET estimator is the best option to employ in the optimization phase. In comparison to the actual values, it correctly estimated the k_∞ metric, slightly underestimated the ς metric, and predicted a few values of the ϕ_{fast} metric outside

Table 4
Performance scores of the studied regression methods.

	RF	ET	KN	SVM	MLP	DT	GB	B	AB	SGD	R	EN	L	LL
fit score	1.00	1.00	0.99	0.96	0.99	1.00	0.97	1.00	0.78	0.54	0.73	0.70	0.69	0.55
EVR	0.90	0.98	0.97	0.98	0.95	0.90	0.89	0.90	0.54	0.38	0.37	0.46	0.25	0.38
MAE	1.27	0.35	0.57	0.54	0.87	1.28	1.72	1.27	3.36	4.77	4.78	4.60	4.77	4.77
MSE	7.60	0.62	1.61	1.45	2.97	7.71	12.99	7.60	38.85	82.73	82.94	77.95	81.94	82.95
R^2	0.90	0.98	0.97	0.98	0.95	0.90	0.89	0.90	0.00	0.23	0.22	0.35	0.07	0.23

Table 5
Performance scores of the studied classification methods.

	RF	ET	KN	SVM	MLP	DT	B	LP	GB	AB	SGD	R	GNB	BNB	P	Log
fit score	1.00	1.00	0.91	0.90	0.90	1.00	0.98	0.90	0.88	0.84	0.61	0.65	0.64	0.55	0.45	0.68
AP	0.90	0.93	0.92	0.95	0.96	0.90	0.90	0.96	0.91	0.90	0.80	0.69	0.78	0.52	0.77	0.81
HL	0.05	0.04	0.04	0.03	0.04	0.05	0.05	0.04	0.05	0.06	0.17	0.18	0.13	0.18	0.22	0.17
LRAPS	0.94	0.95	0.95	0.96	0.96	0.94	0.94	0.95	0.94	0.93	0.81	0.81	0.86	0.82	0.77	0.82
LRL	0.09	0.07	0.07	0.06	0.06	0.09	0.09	0.06	0.08	0.09	0.27	0.28	0.21	0.27	0.34	0.27
ROCAUC	0.92	0.94	0.94	0.96	0.95	0.92	0.92	0.92	0.93	0.93	0.79	0.70	0.87	0.60	0.75	0.77

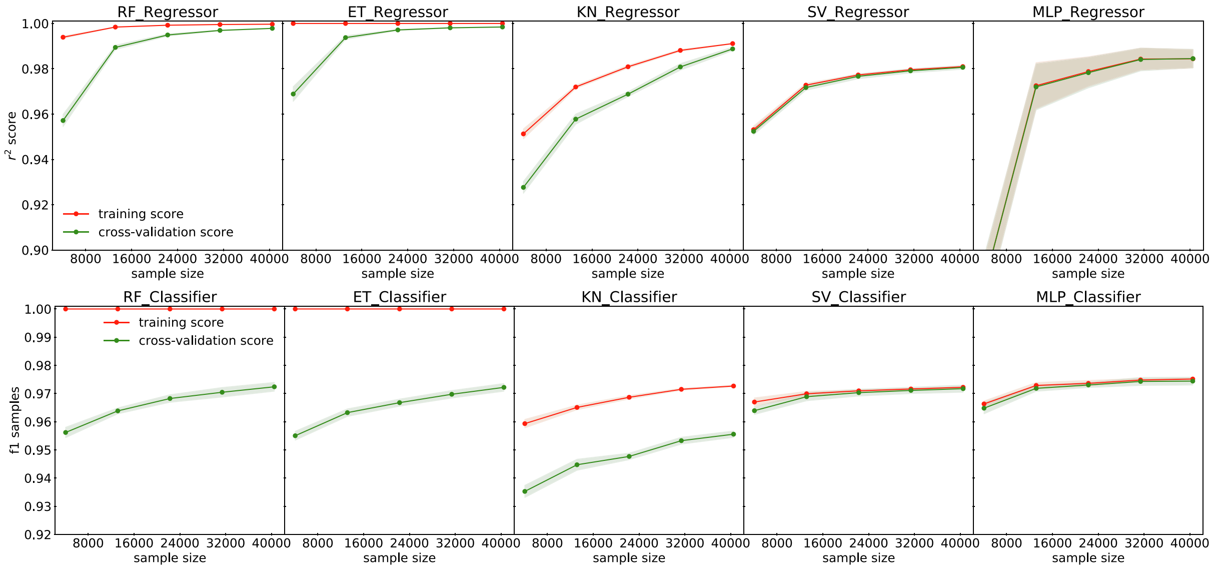


Fig. 3. Learning curves of the selected estimators. Solid lines represent mean scores, while the shaded areas around lines represent the standard deviation of the mean due to error.

of the region. The KN and MLP estimators have good results comparable to the ET and, therefore, be considered as alternative estimators for regression analysis.

For classification analysis, we presented confusion matrices, normalized over the total test sample size, in Fig. 5 to compare the predicted labels with the actual labels in terms of true/false positive and true/false negative scores. From the prediction capability standpoint, these estimators have very close scores in each label and predict all labels with high precision (<95%). Related to the scores of the labels, the only label that causes the highest misclassification score by 10% is the feedback coefficient. This mislabeling originates from the feedback coefficients very close to zero as they are not accurately estimated. Other labels were, however, correctly estimated with an error of less than 1%. As a result, any of these classifiers can easily handle the classification of channel designs without causing a notable loss in labeling. In this context, as in the regression analysis, we decided to use the ET classifier in the design optimization phase to estimate the labels of optimum designs.

Table 6
Optimal hyperparameters of the selected estimators.

Method	Parameters (Regressor/Classifier)
RF	maximum features: None/None, number of trees: 400/400, minimum number of samples: 7/7
ET	maximum features: None/None, number of trees: 800/800, minimum number of samples: 1/7, quality criterion: -/gini
KN	weight function: distance/distance, number of neighbors: 7/44, distance metric: manhattan/manhattan, leaf size: 136/200
SVM	kernel type: rbf/poly, polynomial degree: -/3, kernel coefficient: auto/scale, epsilon: 0.1/-, regularization parameter: 3.4/1.2, stopping criterion: 1e-5
MLP	activation function: relu/tanh, hidden layers: 1000/250, maximum iterations: 1e4/1e4, tol: 1e-5, solver: adam/sgd, learning rate: adaptive/constant

3.5. Optimum designs

Based on the constraints applied to performance metrics, we found a set of optimum solutions for each fuel-salt pair. As an illustration, we presented the Pareto-front solutions for the (U-Th) F_4-

Table 7
Final performance scores of the selected regressors.

	RF	ET	KN	SVM	MLP
fit score	1.00	1.00	1.00	0.97	1.00
EVR	0.94	0.99	0.99	0.96	1.00
MAE	1.07	0.28	0.41	0.37	0.23
MSE	5.52	0.41	0.83	0.44	0.24
R^2	0.94	0.99	0.98	0.96	1.00

Table 8
Final performance scores of the selected classifiers.

	RF	ET	KN	SVM	MLP
fit score	0.93	1.00	1.00	0.89	0.90
AP	0.92	0.94	0.95	0.96	0.97
HL	0.04	0.04	0.04	0.03	0.03
LRAPS	0.95	0.96	0.96	0.96	0.97
LRL	0.07	0.06	0.06	0.05	0.05
ROCAUC	0.94	0.95	0.96	0.95	0.96

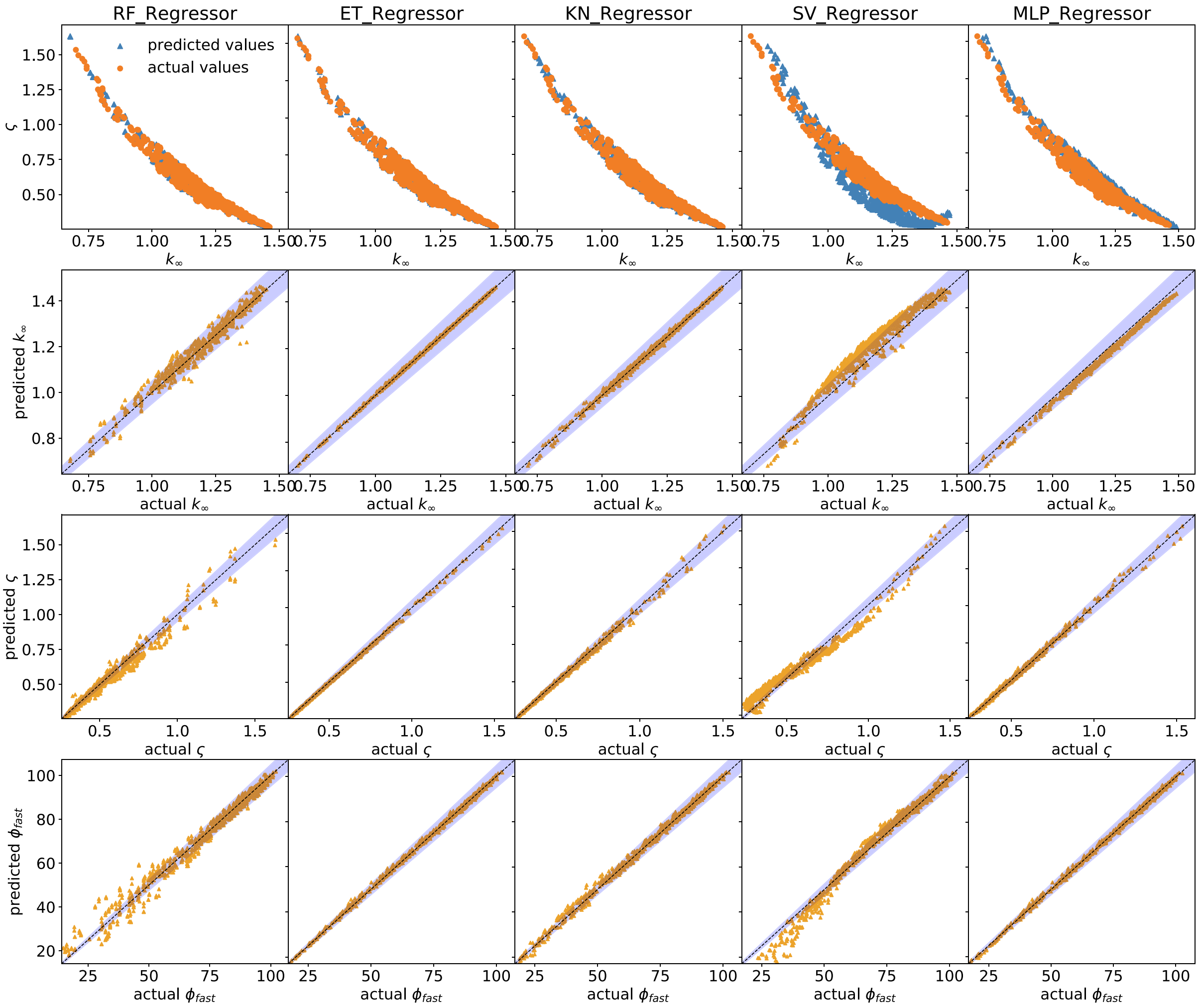
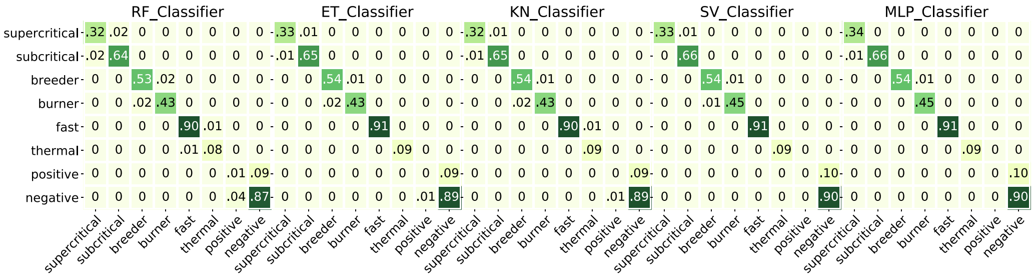


Fig. 4. Cross-validation plots of the selected regressors, top to bottom: (i)k_∞ versus ς (ii) k_∞ (iii) ς and (iv) ϕ_{fast}.



RF_Classifier

	supercritical	subcritical	breeder	burner	fast	thermal	positive	negative
supercritical	.32	.02	0	0	0	0	0	0
subcritical	.02	.64	0	0	0	0	0	0
breeder	0	0	.53	.02	0	0	0	0
burner	0	0	.02	.43	0	0	0	0
fast	0	0	0	0	.90	.01	0	0
thermal	0	0	0	0	.01	.08	0	0
positive	0	0	0	0	0	0	.01	.09
negative	0	0	0	0	0	0	.04	.87

ET_Classifier

	supercritical	subcritical	breeder	burner	fast	thermal	positive	negative
supercritical	.33	.01	0	0	0	0	0	0
subcritical	.01	.65	0	0	0	0	0	0
breeder	0	0	.54	.01	0	0	0	0
burner	0	0	.02	.43	0	0	0	0
fast	0	0	0	0	.91	0	0	0
thermal	0	0	0	0	0	.09	0	0
positive	0	0	0	0	0	0	0	.09
negative	0	0	0	0	0	0	.01	.89

KN_Classifier

	supercritical	subcritical	breeder	burner	fast	thermal	positive	negative
supercritical	.32	.01	0	0	0	0	0	0
subcritical	.01	.65	0	0	0	0	0	0
breeder	0	0	.54	.01	0	0	0	0
burner	0	0	.02	.43	0	0	0	0
fast	0	0	0	0	.90	.01	0	0
thermal	0	0	0	0	0	.09	0	0
positive	0	0	0	0	0	0	0	.09
negative	0	0	0	0	0	0	.01	.89

SV_Classifier

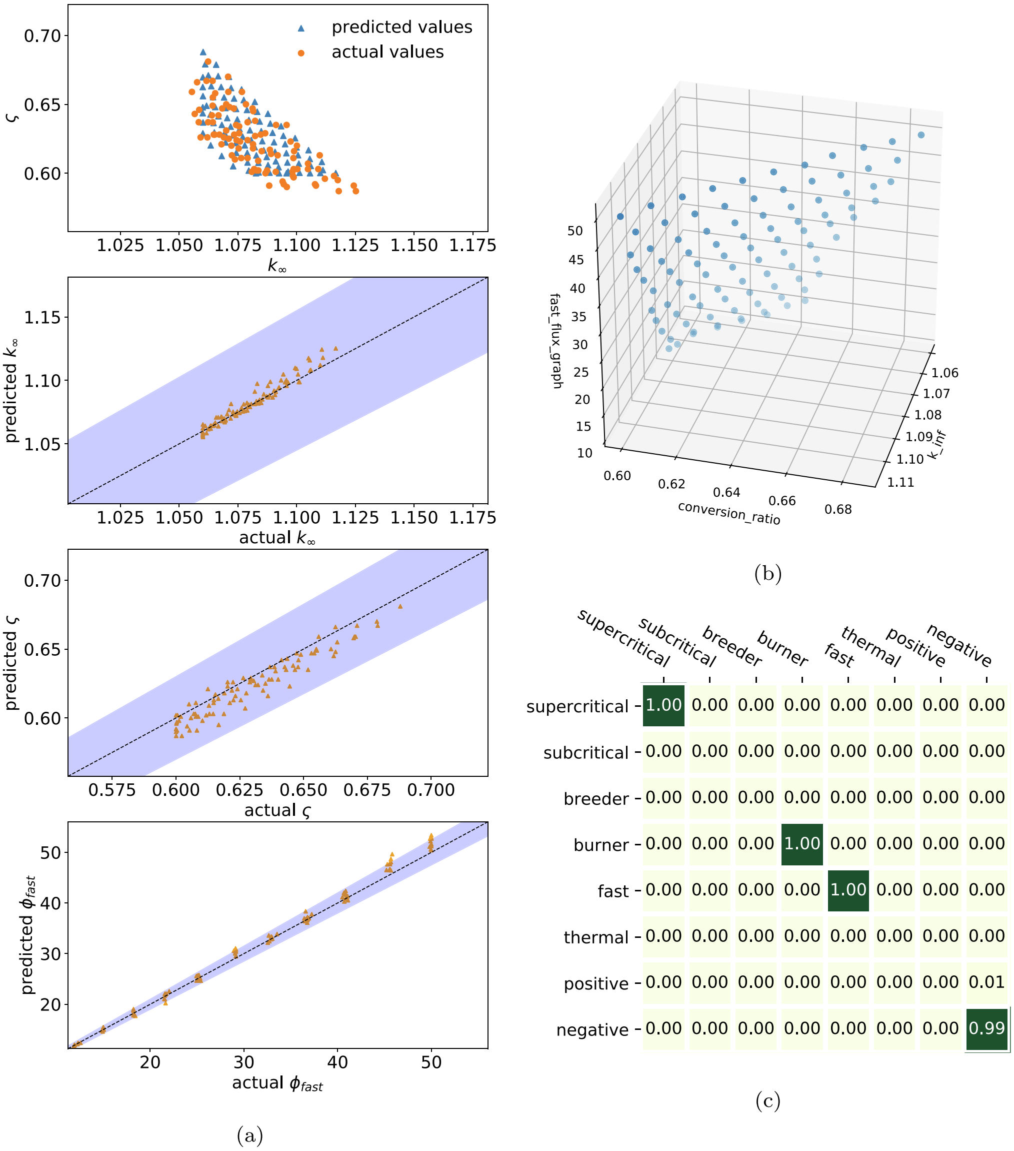
	supercritical	subcritical	breeder	burner	fast	thermal	positive	negative
supercritical	.33	.01	0	0	0	0	0	0
subcritical	0	.66	0	0	0	0	0	0
breeder	0	0	.54	.01	0	0	0	0
burner	0	0	.01	.45	0	0	0	0
fast	0	0	0	0	.91	0	0	0
thermal	0	0	0	0	0	.09	0	0
positive	0	0	0	0	0	0	0	.10
negative	0	0	0	0	0	0	0	.90

MLP_Classifier

	supercritical	subcritical	breeder	burner	fast	thermal	positive	negative
supercritical	.34	0	0	0	0	0	0	0
subcritical	.01	.66	0	0	0	0	0	0
breeder	0	0	.54	.01	0	0	0	0
burner	0	0	0	.45	0	0	0	0
fast	0	0	0	0	.91	0	0	0
thermal	0	0	0	0	0	.09	0	0
positive	0	0	0	0	0	0	0	.10
negative	0	0	0	0	0	0	0	.90

Fig. 5. Normalized confusion matrices of the selected classifiers.

NaF-BeF$_2$, which is very similar to the results of other fuel-salt pairs, in Fig. 6. This figure includes (a) cross-validation plots for each performance metric, (b) the distribution of the Pareto-front solutions in 3-D performance metric space, and (c) the normalized confusion matrix. As seen from the subplot (a), the Pareto-optimal solutions (blue triangle) overlapped very well with the actual val-



(a)

(b)

(c)

Fig. 6. Optimum solutions for (U-Th) F_4-NaF-BeF_2: (a) cross-validation plots of performance metrics, (b) the distribution of the Pareto-front solutions in 3-D performance metric space, and (c) the normalized confusion matrix.

ues (orange circles). The other subplots also confirm this situation by showing good distributions in the very proximity of the intersection line of each performance metric.

The cross-validation subplots also indicated that all the predicted metrics totally stay within the 5% relative error (shaded area) and very close to the intersection line. We found even better results for the k_∞ metric with no more than 1% relative error.

As in the regression analysis results, the classification results of the optimal solutions in subplot (c) agreed very well with the actual labels. The only noteworthy complication is the misclassification of a tiny fraction of the feedback indicators. This mislabeling certainly arose from the prediction of the α_{total} values as discussed before and were estimated as positive instead of negative. In conformity with the constraints applied to performance metrics in the optimization phase, all the optimum designs we found are of characteristics of supercritical eigenvalue, fast spectrum, burner type, and negative feedback.

Of the entire Pareto front solutions (around 650) which account for each combination of the fuel-salt pairs, only the nine best solutions are tabulated in Table 9. We simply opted for the solutions that grant the highest breeding ratio (ς) and the lowest fast flux (ϕ_{fast}) at the k_∞ = 1.06. The only exception for the multiplication factor criterion is for the NaCl salt in U-Pu fuel as there are no solutions with a multiplication factor less than 1.21 for this fuel-salt pair.

As provided in the table, the suggested solutions can be grouped more easily by fuel type as the solutions show a certain relationship with the fuel type rather than the salt type. When the fuel types are compared to each other, U-Th fuels require the highest ^{235}U content (18–20%) due to the existence of the ^{232}Th isotope which has an absorption cross-section approximately four times higher than ^{238}U. U fuels reach their top performance when a moderate amount of ^{235}U (14–15%) is used. U-Pu fuels need the lowest amount of ^{235}U (5%) as not only does Pu in U-Pu already contain a certain fraction of fissile isotopes like ^{239}Pu and ^{241}Pu, but also Pu fissile isotopes emit greater average fission neutrons per absorption. Similarly, we noticed that U-Th fuels demand 13–14% Th in U-Th while U-Pu fuels favor the highest grade of U content (90%).

A comprehensive look at the common properties of the optimal designs suggests that T_{ave} should be in the range of 1100–1200 K

Table 9
Optimal designs for each fuel-salt pair*.

Fuel Type	Salt Type	f_{35}	f_U	p	T_{ave}	ξ	k_∞ †	ς ‡	ϕ_{fast} ⊥	α_{total} ^
U	LiF–BeF$_2$	14.47	100.0	86.7	1200	0.274	1.0600(+60)	0.661(-6)	18.6(-0.5)	−3.9
U	NaF-BeF$_2$	14.88	100.0	7.18	1125	0.274	1.0600(-120)	0.638(+6)	19.0(-0.9)	−4.0
U	NaCl	14.28	100.0	1.00	1176	0.274	1.0604(-20)	0.692(0)	12.6(0)	−3.0
U-Th	LiF–BeF$_2$	19.10	86.7	5.50	1123	0.276	1.0601(-70)	0.665(-7)	17.1(-0.2)	−4.3
U-Th	NaF-BeF$_2$	20.00	86.8	5.98	1125	0.275	1.0600(+30)	0.643(-11)	16.0(-0.2)	−2.8
U-Th	NaCl	18.48	86.2	1.00	1199	0.274	1.0600(+140)	0.696(-1)	14.4(0)	−2.8
U-Pu	LiF–BeF$_2$	5.00	90.0	4.40	1198	0.274	1.0600(-20)	0.732(+3)	16.0(0.2)	−4.5
U-Pu	NaF-BeF$_2$	5.00	89.3	4.37	1125	0.275	1.0600(+460)	0.723(+15)	13.6(0.3)	−3.3
U-Pu	NaCl	5.00	90.0	1.00	1193	0.274	1.2192(+10)	0.753(+1)	13.0(0)	−3.4

* Performance metrics given in the table are the predicted values. The values inside the brackets are the actual values in terms of difference from the predicted values.
† The computational error of the infinite multiplication factor is ±0.0009. The values within the brackets are in the units of pcm.
‡ The computational error of the conversion ratio is ±0.005. The values within the brackets are in the units of milli.
⊥ The computational error of the fast flux is ±0.007.
^ The propagated computational error of the feedback coefficient of reactivity is ±0.4.

and ξ needs to be at its minimum value (0.274). Furthermore, as expected, Th necessitates higher fissile material than any other fuel types whereas the fissile material required by the U-Pu fuel is the lowest as the weapons-grade Pu comprises a gread deal of fissile isotopes by almost 50% of its content. On the other hand, unlike the other feature parameters that display a direct characteristic relationship with the fuel type, there is no noticeable evidence to associate the channel-to-channel pitch length (p) with the fuel types. The p parameter is, however, likely to vary with the salt type just as the NaCl salt calls for a constant pitch value of 1.0 cm in all fuel types.

In summary, among the optimal designs, it appears that all the fuels mixed with the NaCl salt are the most promising designs. In particular, the F$_4$-PuF$_3$-NaCl fuel-salt yields the highest ς and k_∞. The most unfavorable side of this fuel-salt is to have a lower α_{total} than other salt types but the coefficient is still quite sufficient compared to the exiting designs of MSRs at the initial state with a total fuel-salt feedback coefficient of around −3.4 pcm/K (Rykhlevskii et al., 2019; Robertson, 1971). From the reactor safety point of view, the LiF–BeF$_2$ salt with the highest α_{total} is the most favorable.

4. Conclusion

This study explored the promise of AI/ML techniques to identify optimal designs for a single channel of a MSR. The main focus of this study was to compare the prediction accuracy of different ML methods and to decide the best estimator suitable for single channel design. We created a reactor database to train and test ML methods, trained the best performing estimator, and used the genetic algorithm technique (NSGAIII) coupled to the best performing estimator (ET) to seek the optimal channel designs. Major findings throughout this study and general remarks on the results are as follows:

As a result of the assessment of the scores of the estimators, we found RF, ET, KN, MLP, and SVM as the best performing estimators for the design optimization phase. Estimators of ensemble, SVM, NN, and DTs methods performed well, while the linear methods (L, R, SGD etc.) performed poorly as compared to the non-linear methods (DT, GB, and RF etc.).

For sufficiently accurate and reliable results, we determined from the learning curves of the estimators that the optimum size of the training dataset should be at least 12,000 for each fuel-salt pair and 300,000 in total.

The cross-validation plots and confusion matrices of the selected estimators shed light on the best estimator for the design optimization phase by pinpointing the individual performance metric. We predicted all the performance metrics with less than 5% error and classified with less than 1% error except for the feedback label which has an error of 10%. We also understood that the MLP handles performance metrics as good as ET, but with considerably higher computational run-time. The ET estimator was the best performing estimator among the explored ML methods.

We estimated all the performance metrics of the optimum designs within a prediction error of at most 5% in comparison to their actual values. A comprehensive overview of the common properties of the optimal designs pointed to a particular enrichment level and U content associated with fuel type (but not with salt type), T_{ave} higher than 1100 K, and a moderator-to-salt ratio of 0.274. The relation of the channel-to-channel pitch length (p) with either fuel type or salt type was not as clear as the other feature variables.

All the optimum designs were accurately labeled as a supercritical reactor, burner type, fast spectrum, and negative feedback. As indicated by confusion matrices, only a few designs, most of which belong to the feedback coefficient, were mislabeled.

Based on the constraints in this study, the optimal fuel-salt option was a combination of U-Pu fuel and NaCl salt. It achieved the longest graphite lifetime (i.e., the lowest ϕ_{fast}), the highest ς, and a sufficient negative α_{total}.

Using the very basic sampling method, we achieved optimal designs, which have seven feature parameters, with a total of 280,000 reactor simulations. We used up to 10-discretization points for each continuous parameter. When compared to the outcomes of the papers discussed in the Introduction (Pereira et al., 1999; Pereira and Lapa, 2003) from the perspective of computing speed and computational resource utilization, recalling that 3500 against 100 reactor simulations for two feature parameters, each with 10-discretization points, the proposed method in this study outperforms the traditional methods for less than five feature parameters. In the case of more feature parameters, recalling that 150,000 against 280,000 for seven feature parameters, this method will be as fast as others at a single optimization run if advanced sampling methods like Latin hypercube are employed. Meanwhile, it, contrary to other methods, provides utmost flexibility to the designers by allowing any optimization tool to run numerous times with different hyperparameters and constraints at no additional costs. This is the main reason why this method provides faster convergence and better solutions than many traditional techniques.

In addition to the above numerical findings, the most important success of this study is the integration of the ML-enabled technique coupled to an optimization tool into the reactor core design process. Another important success achieved by this proposed method is the examination of optimal designs in a very short time (within minutes), regardless of the value and number of performance metrics or feature parameters, once a reactor database is generated. Without using any reactor physics code, this method makes available evaluating performance metrics of any design with very high accuracy by using a training dataset specific to the feature parameters of interest to be loaded from a reactor database generated initially. Due to the very fast computing speed and high accuracy in estimating performance metrics, we recommend using this method for neutronic calculations (also for thermal–hydraulics) in nuclear engineering along with nuclear computer codes such as deterministic and Monte Carlo that require significant user expertise.

Besides these achievements, we encountered various difficulties that have a direct and indirect impact on the outcomes of this study throughout the work. First, we understood that the number of performance metrics, data size, and assigning different estimators (composite estimators) to an individual performance metric significantly affect performance scores. For instance, the larger the sample size in a training dataset, the higher the fit scores, thus better fitting results. Second, there were several known drawbacks to getting better estimators' scores: hyperparameter dependency, data distribution on the search domain, and quality of the training dataset (outlier elimination), such as very low k_∞ and very high ς. Although the effects of hyperparameters are limited, outliers have a profound effect on the fit scores, therefore on the predicted performance metrics and optimal designs.

We also figured out that the primary difficulty concerning the applicability of ML methods for reactor design is to generate a reactor database that consumes the entire computer's resources, requires a powerful supercomputer, and takes a long time. In fact, this is not a critical issue for the applicability of the proposed method since the database generation is a quite easy process in comparison to other tasks and it is enough to do it once at the very beginning of the study. Third, it is likely to find different solutions when the population size and the number of generations are further increased, or a different optimization method, such as bayesian optimization, is used. But, we anticipate that all the solutions to be obtained using different parameters would be close to the optimal solutions presented in this work as we already verified our solutions with different optimizers and their varying hyperparameters. The only concern with the optimization process is the defined constraints on performance metrics as an optimizer directs offspring according to these constraints.

In our follow-up studies, we will take the following steps to do away with some of the assumptions and simplifications we have made in this research, to complement the shortcomings in the study, and to further improve the existing method. Initially, we have a plan to use the suggested method for a full-core MSR design as a second phase of the ongoing study. Different from the single channel design, designing a full reactor core will bring several additional variables (e.g., the number of fuel channels, reactor diameter) to the existing feature parameters and a few additional performance metrics (e.g., power peaking factor) to the existing performance metrics. The rest is expected to be the same as the procedure suggested in this study.

Concurrently, we will replace the grid-based search technique used in generating the reactor database with an advanced sampling strategy (e.g., adaptive, Latin hypercube, or hybrid) as generating a high-quality training dataset is a critical step in building a robust predictive model. This is because, despite being the simplest and capable of evaluating all the possible scenarios among the different parameters of feature space, the grid technique is computationally expensive and consumes computing resources by a considerable amount. An advanced method can reduce the number of samples in datasets while improving the quality of the data and thus use computational resources more efficiently by effective space-filling.

On the other hand, in order to simplify the optimization problem and make a significant gain in the simulation run-time, we had deliberately disregarded the thermal–hydraulics effects such as delayed precursor movement, temperature distribution along the channel, and reactivity feedback on the neutronic calculations. Related to this, we will integrate the Moltres MOOSE application that solves the thermal–hydraulics and neutronics equations for liquid-fueled MSRs with the Plankton python tool having been developed in this work and increase the accuracy of neutronic calculations.

To sum up, the research will make a significant contribution to the further advancement of optimization studies by eliminating the deficiencies of the existing methods in reaching ideal designs such as the tournament approach of the optimizers, convergence criteria, strict constraints on performance metrics, and concerns about computer resource utilization. The results of this study could benefit researchers who are interested in a multi-purpose reactor design that provides more flexibility for in-core fuel management, improved neutron economy for fuel utilization, more safety margin against core melt-down, prolonged reactor lifetime, and less spent nuclear fuel.

CRediT authorship contribution statement

Mehmet Turkmen: Conceptualization, Methodology, Software, Writing - original draft, Data curation. **Gwendolyn J. Y. Chee:** Validation, Writing - review & editing, Software. **Kathryn D. Huff:** Supervision, Project administration, Resources, Funding acquisition, Writing - review & editing.

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.

Acknowledgments

This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications. The authors would also like to acknowledge financial support from the Scientific and Technological Research Council of Turkey (TUBITAK) BIDEB-2219 Postdoctoral Research Program. Prof. Huff is supported by the Nuclear Regulatory Commission Faculty Development Program, the Blue Waters sustained-petascale computing project supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois, the NNSA Office of Defense Nuclear Nonproliferation R&D through the Consortium for Verification Technologies and the Consortium for Nonproliferation Enabling Capabilities (awards DE-NA0002576 and DE-NA0002534), the DOE ARPA-E MEITNER Program (award DE-AR0000983), and the International Institute for Carbon Neutral Energy Research (WPI-I2CNER), sponsored by the Japanese Ministry of Education, Culture, Sports, Science and Technology.

References

Anderson, K., Skutnik, S., Wheeler, A., Chvala, O., 2019. Parameter Interpolation for MSR Core Physics Modules, in: The 4th annual Technical Workshop on Fuel Cycle Simulation, ARFC, UIUC, Urbana, IL, US..

Ashraf, O., Rykhlevskii, A., Tikhomirov, G., Huff, K.D., 2020. Whole core analysis of the single-fluid double-zone thorium molten salt reactor (SD-TMSR). Ann. Nucl.

Energy 137,. https://doi.org/10.1016/j.anucene.2019.107115. URL:http://www.sciencedirect.com/science/article/pii/S0306454919306255 107115.

Bae, J.W., Rykhlevskii, A., Chee, G., Huff, K.D., 2020. Deep learning approach to nuclear fuel transmutation in a fuel cycle simulator. Ann. Nucl. Energy 139, 107230.

Betzler, B.R., Powers, J.J., Worrall, A., 2017. Molten salt reactor neutronics and fuel cycle modeling and simulation with SCALE, Annals of Nuclear Energy 101 (2017) 489–503. URL:http://linkinghub.elsevier.com/retrieve/pii/S0306454916309185. doi: 10.1016/j.anucene.2016.11.040..

Blank, J., Deb, K., 2020. pymoo: Multi-objective Optimization in Python. URL: https://pymoo.org/index.html..

Chadwick, M.B. et al., 2006. ENDF/B-VII.0: Next generation evaluated nuclear data library for nuclear science and technology. Nuclear Data Sheets 107, 2931–3060. https://doi.org/10.1016/j.nds.2006.11.001.

Chollet, F., others, Keras, 2015. URL:https://keras.io..

Deb, K., Jain, H., 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. IEEE Trans. Evol. Comput. 18, 577–601. https://doi.org/10.1109/TEVC.2013.2281535. URL:https://ieeexplore.ieee.org/document/6600851.

Gomez-Fernandez, M., Higley, K., Tokuhiro, A., Welter, K., Wong, W.-K., Yang, H., 2020. Status of research and development of learning-based approaches in nuclear science and engineering: A review. Nucl. Eng. Des. 359.

Jain, H., Deb, K., 2014. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: handling constraints and extending to an adaptive approach. EEE Trans. Evol. Comput. 18, 602–622.

Janz, G.J., 1988. Thermodynamic and Transport Properties for Molten Salts: Correlation Equations for Critically Evaluated Density, Surface Tension, Electrical Conductance, and Viscosity Data. Am. Chem. Soc. Am. Institute Phys.

Jerden, J., 2019. Molten Salt Thermophysical Properties Database Development: 2019 Update, Technical Report ANL/CFCT-19/6, Chemical and Fuel Cycle Technologies Division. Argonne National Laboratory.

Kim, S.H., Pyeon, C.H., Um, W., Kim, M.H., 2018. Design of LEU Fuel Assembly Using Artificial Neural Network at Kyoto University Critical Assembly, in: Transactions of the Korean Nuclear Society Spring Meeting, Korean Nuclear Society, Jeju, Korea..

Kim, S.H., Shin, S.G., Han, S., Kim, M.H., Pyeon, C.H., 2020. Feasibility study on application of an artificial neural network for automatic design of a reactor core at the Kyoto University Critical Assembly. Prog. Nucl. Energy 119.

Kumar, A., Tsvetkov, P., 2015. A new approach to nuclear reactor design optimization using genetic algorithms and regression analysis. Ann. Nucl. Energy 85, 27–35.

Lefebvre, R.A., Langley, B.R., Miller, L.P., Delchini, M.-O.G., Baird, M., Lefebvre, J.P., 2019. NEAMS Workbench Status and Capabilities, Technical Report ORNL/TM-2019/1314, Oak Ridge National Lab. (ORNL), Oak Ridge, TN (United States)..

Leppanen, J., Pusa, M., Viitanen, T., Valtavirta, V., Kaltiaisenaho, T., 2014. The Serpent Monte Carlo code: Status, development and applications in 2013. Ann. Nucl. Energy 82, 142–150.

Nissan, E., 2019. An Overview of AI Methods for in-Core Fuel Management: Tools for the Automatic Design of Nuclear Reactor Core Configurations for Fuel Reload, (Re)arranging New and Partly Spent Fuel. Designs 3, 1–45.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Pereira, C.M., Lapa, C.M., 2003. Coarse-grained parallel genetic algorithm applied to a nuclear reactor core design optimization problem, Annals of Nuclear Energy 30 (2003) 555–565. ISBN: 0306-4549 Publisher: Elsevier..

Pereira, C.M. d. N.A., Schirru, R., Martinez, A.S., 1999. Basic investigations related to genetic algorithms in core designs, Annals of Nuclear Energy 26 (1999) 173–193..

Robertson, R.C., 1971. Conceptual Design Study of a Single-Fluid Molten-Salt Breeder Reactor., Technical Report ORNL–4541. ORNL.

Rykhlevskii, A., Bae, J.W., Huff, K.D., 2019. Modeling and simulation of online reprocessing in the thorium-fueled molten salt breeder reactor. Ann. Nuclear Energy 128, 366–379. https://doi.org/10.1016/j.anucene.2019.01.030. URL: http://www.sciencedirect.com/science/article/pii/S0306454919300350.

Wei, H., Chen, Y.-T., Lan, K.-C., Cheng, J., 2018. Parametric study of thermal molten salt reactor neutronics criticality behavior. Prog. Nucl. Energy 108, 409–418.

C.J. Werner, MCNP USER'S MANUAL Code Version 6.2, User Manual la-ur-17-29981, Los Alamos National Lab. (LANL), Los Alamos, NM (United States), Los Alamos, NM, United States, 2017..

Wieselquist, W., Lefebvre, R.A., Jessee, M.A., 2020. SCALE Code System, Code Manual ORNL/TM-2005/39. Oak Ridge National Laboratory, Oak Ridge, TN.

Zeng, K., Stauff, N.E., Hou, J., Kim, T.K., 2020. Development of multi-objective core optimization framework and application to sodium-cooled fast test reactors. Prog. Nucl. Energy 120, 103184.