Superconductor Science and Technology

Artificial intelligence-based models for reconstructing the critical current and index-value surfaces of HTS tapes

OPEN ACCESS
IOP Publishing
Superconductor Science and Technology
Supercond. Sci. Technol. **35** (2022) 124002 (15pp)
https://doi.org/10.1088/1361-6668/ac95d6

Artificial intelligence-based models for reconstructing the critical current and index-value surfaces of HTS tapes

Giacomo Russo[1], Mohammad Yazdani-Asrami[2,*], Riccardo Scheda[1], Antonio Morandi[1] and Stefano Diciotti[1,3]

[1] Department of Electrical, Electronic and Information Engineering, University of Bologna, Bologna, Italy
[2] Propulsion, Electrification & Superconductivity group, James Watt School of Engineering, University of Glasgow, Glasgow, United Kingdom
[3] Alma Mater Research Institute for Human-Centered Artificial Intelligence, University of Bologna, Bologna 40136, Italy

E-mail: mohammad.yazdani-asrami@glasgow.ac.uk

Received 15 June 2022, revised 19 September 2022
Accepted for publication 28 September 2022
Published 21 October 2022

Abstract

For modelling superconductors, interpolation and analytical formulas are commonly used to consider the relationship between the critical current density and other electromagnetic and physical quantities. However, look-up tables are not available in all modelling and coding environments, and interpolation methods must be manually implemented. Moreover, analytical formulas only approximate real physics of superconductors and, in many cases, lack a high level of accuracy. In this paper, we propose a new approach for addressing this problem involving artificial intelligence (AI) techniques for reconstructing the critical surface of high temperature superconducting (HTS) tapes and predicting their index value known as n-value. Different AI models were proposed and implemented, relying on a public experimental database for electromagnetic specifications of HTS tapes, including artificial neural networks (ANN), eXtreme Gradient Boosting (XGBoost), and kernel ridge regressor (KRR). The ANN model was the most accurate in predicting the critical current of HTS materials, performing goodness of fit very close to 1 and extremely low root mean squared error. The XGBoost model proved to be the fastest method, with training computational times under 1 s; whilst KRR could be used as an alternative solution with intermediate performance.

Keywords: artificial neural network, critical current, HTS tapes, kernel ridge, regression, superconductors, XGBoost

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

1. Introduction

The critical current I_c, as well as the index value known as n-value, of practical high temperature superconducting (HTS) tapes, depends on the magnitude of the magnetic field B acting on tape, field orientation θ with respect to the tape surface, and the operating temperature T. Knowing the $I_c(B,\theta,T)$ and $n(B,\theta,T)$ functions, or more commonly the I_c per unit width of the tape (I_{cw}), is essential for any practical applications. Moreover, extracting the local $J_c(B,\theta,T)$ from the measured $I_c(B,\theta,T)$ curve and incorporating it into finite element (FE) models is a key feature for the design and optimization of HTS devices for large-scale power applications. FE models are usually computationally costly due to the many

* Author to whom any correspondence should be addressed.

degrees of freedom needed to cope with their high aspect ratio and the strong nonlinearity of the E–J characteristic. The dependence of the critical current density on the magnitude and the orientation of the magnetic field (and on the temperature if no isothermal assumption is made, and thermal equations are also included in the model) must be properly taken into account. In fact, neglecting this dependence would lead to unacceptable inaccuracy in the final results of practical interest (e.g. in terms of AC loss). Including this dependence in the modelling process further complicates the calculation due to the strong coupling introduced between the magnetostatic and the eddy current problems. The most common way to include this dependence in numerical models is through analytical formulas [1–5]. However, analytical approaches rely on fitting parameters to be preliminarily evaluated and consequently suffer from lack of enough accuracy in complicated problems. Look-up tables are an alternative, but they rely on the availability of dense experimental data even when the model they belong to is running. Furthermore, look-up tables do not effectively account for all interdependencies between inputs, they do not allow any extrapolation, and are known to cause dramatic computational slow down when dealing with large datasets. In this paper, we propose an alternative method to address the problem using artificial intelligence (AI) techniques. AI refers to the development of computational systems able to perform different tasks resembling human intelligence. Such techniques need to be trained with data in order to be able to act as an accurate estimator, but once the model is adequately trained, it becomes available to be used in other modelling purposes. AI techniques exploit statistical and optimization algorithms to analyse data, learn from them, and make decisions based on this process. Moreover, they may use biologically inspired systems called artificial neural networks (ANN) to find interdependencies between input and output data in huge amounts of data or make decisions according to available information. AI methods are well suited for clustering, optimization, classifications, and regression problems. The aim of this study is the determination of the $I_c(B,\theta,T)$ and $n(B,\theta,T)$ characteristics using AI models.

Although to date, AI methods have been overlooked in applied superconductivity, ANN methods were successfully implemented for some purposes, including AC losses predictions [6–9], complex design optimization [10], and speeding up multiphysics simulations [11]. These studies demonstrate substantial potential for AI in applied superconductivity left to be discovered. In [12], a multi-width no-insulation magnet critical current at different operating temperatures was estimated using the critical current density distribution, which in turn was found using a double hidden layer Bayesian regularised neural network. Another extensive discussion about the advantages and drawbacks of AI methods with respect to conventional approaches, such as analytical formulas and look-up tables, which also investigates and reports a comparison regarding the different computational times, can be found in [13].

In this paper, we have applied several AI methods for developing intelligent models for reconstructing the critical current and n-value surfaces of HTS tapes in order to provide a comprehensive reference for the comparison of different AI methods for different large-scale applications in the future. Hyperparameters tuning and sensitivity analysis were also carried out to investigate, improve, and optimise the results of the understudied AI-based models. Experimental data are taken from a public database [14]. In this database, the critical current and n-value characteristics of most commercial HTS tapes, measured over a wide range of B, θ, and T, are reported. The test procedure, tape specification, and the apparatus used to produce the database are comprehensively described in [15]. In this study, different AI-based models were developed such as ANN, eXtreme Gradient Boosting (XGBoost), and kernel ridge regressor (KRR). The results of the AI techniques in addressing the $I_c(B,\theta,T)$ and $n(B,\theta,T)$ dependencies are discussed in detail in this paper. We show that AI models offer fast, accurate, and experimental-based approaches for predicting the critical current density characteristics of HTS tapes that could be embedded in both existing numerical modelling software/techniques and designing routines. Finally, it is worth mentioning that this study is the result of a collaboration [16], now in its second year, carried out in the framework and with the support of the COST Action CA19108, 'High-Temperature SuperConductivity for AcceLerating the Energy Transition', funded by EU commission [17].

2. Methodology: an overview of the developed AI models

Among many challenges related to superconductors, the problem of critical current dependence on electro-magneto-thermal quantities (temperature, magnetic field, etc) is well suited to the typical AI application as a regression task. In fact, it is well-known that in HTS materials, a certain combination of temperature, magnetic field magnitude, and orientation that act as the features of the problem determines one specific value of critical current density that represents the target of the problem. Like any other AI task, an adequately large and representative dataset is required to properly train a model. Therefore, the publicly open, accessible database of 'High-Temperature Superconductor critical current data' provided by the Robinson Research Institute (Victoria University of Wellington, New Zealand) [14, 18] played a key role in the investigations of this paper.

In [14], plenty of data relating the critical current and the n-value to temperature, external magnetic field magnitude and angle is provided for different HTS tape specimens. The available data ranges over large intervals of the features involved. Hence, it is suitable to be applied as a training dataset for AI models developed in this paper. For this study, the SuperOx GdBCO 2G HTS database was used because of the relatively large range of values it provides for both the temperature and the magnetic field magnitude. However, it is worth pointing out that any other tape specimen could have been chosen for applying the same approach to the corresponding data. The complete dataset that we used in this paper is composed of 14 747 combinations of critical current, critical current per unit width, n-value, operating temperature (ranging

from 15 to 90 K), external magnetic field magnitude (ranging from 0 to 7 T) and external magnetic field angle (ranging from 0° to 240°, where $\theta = 0°$ corresponds to the field applied perpendicular to the tape surface). The dataset was randomly decomposed into two sets, one dedicated to developing the model (both training and validating the quality of the training process) and the other for its testing. In this study, 70% of the complete dataset has been dedicated to training/validation, whereas the remaining 30% was for testing. Since an AI method that reaches the best performance for any problem *a priori* does not exist, different AI techniques were chosen and their performances were evaluated. ANN is widely being acknowledged as one of the most effective tools for regression and it is biologically inspired, XGBoost is a decision tree based model, and KRR relies on the minimization of the sum of the squared residuals. One more reason for this choice was that all the aforementioned models are among the most successful and commonly used for engineering problems solved by AI techniques.

As performance metrics to report the error and quality of estimation, we used the the root mean squared error (RMSE), and the goodness of fit (R^2), as they are more practical [6, 19]:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n_s} (y_i - x_i)^2}{n_s}} \tag{1}$$

$$R^2 = \frac{\sum_{i=1}^{n_s} (\underline{x} - x_i)\left(\underline{y} - y_i\right)}{\sqrt{\sum_{i=1}^{n_s} (\underline{x} - x_i)^2 \sum_{i=1}^{n_s} \left(\underline{y} - y_i\right)^2}} \tag{2}$$

where, x_i is the actual value, y_i is the predicted value, $\underline{x}$ is the mean of actual (real experimental) values, $\underline{y}$ is the mean of predicted values, and n_s is the number of data samples.

In order to avoid overfitting and thus achieve fairly good estimations over the whole range of features, a visual comparison of the critical surfaces of the models was included among the evaluating criteria as well. Great effort has also been paid to hyperparameters tuning and sensitivity analysis to optimise the AI models and comprehensively investigate their performance dependence on their own controlling parameters. In fact, the capability of any AI model to extract specific trends from a dataset is strictly related to certain parameters, named hyperparameters. Established hyperparameter tuning techniques were applied in this study, such as the so-called grid-search method [20], which generates a user-defined grid of possible combinations of hyperparameters and trains the model for each of them and keeps the best case based on the chosen evaluation criteria. Grid-search was applied to the training set with a K-fold cross-validation procedure: the training set is divided into K splits in which, for each split, a different training of the model is carried out: the chosen split is used as a validation set for evaluating the performance of the training process, for which the remaining K-1 splits are used. This procedure is performed for each combination of the hyperparameters grid. In particular in this study, we used a five-fold cross-validation because it is reported to offer a favourable bias-variance trade-off [21].

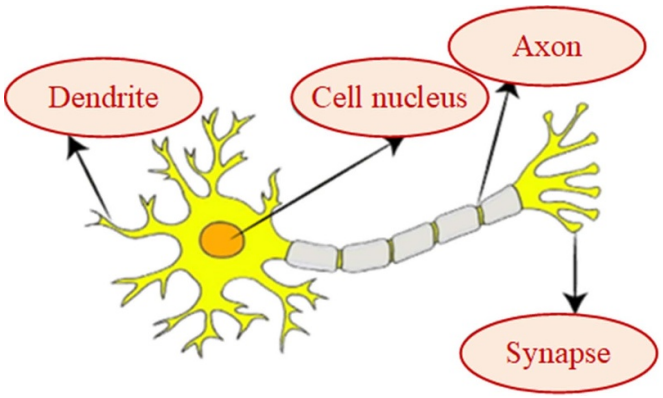


Figure 1. A simplified structural schematic of a neuron.

Both Python and MATLAB scripts were used for carrying out the calculations. The reason for that is to cover both popular coding environments of m-file and Python for any future inclusion of these methods into the FE simulation software.

The main features and characteristics of the three models that were identified are described briefly in the following subsections.

2.1. ANN

ANN [22] are computational systems that can model and predict sophisticated characteristics with a high level of nonlinearity. The ANN performance is inspired by how the human brain works. To make the performance of the ANN models clearer, figure 1 illustrates the structure of a brain neuron and how ANN resembles brain behaviour to model physics and engineering problems. In this system, dendrites serve as inputs that receive the data from other cells and neurons. Dendrites inject the data into the nucleus laid at the cell body and are considered a node of the system. The nucleus activity is modelled as a function applied to inputs to compute the output. Synapses are considered as the weights of the neural system that help the model predict or estimate the characteristic of dendrites. Finally, the axon provides the output of the process and connects the neuron to other neurons [23]. To create ANN models based on a real neural system, multiple layers are considered. The very first layer is the input layer which receives the input data. Then, the data are fed into a series of hidden layers, which calculate or estimate the characteristics of the input data. Each hidden layer consists of some neurons, weights vector, bias factor, and activation function.

At last, there is an output layer that offers the result of the estimation process. A simple structure of the model of a neuron in an ANN is shown in figure 2 [24]. To estimate the output, a simple neuron uses equation (3) [25]:

$$\hat{y} = f(\vec{W}^T \vec{x} + b) \tag{3}$$

where, $\hat{y}$ is the output, $\vec{x}$ is the input vector, f is the activation function, $\vec{W}$ is the weights vector, and b is the bias factor. The objective of the training stage is to reduce the error of the

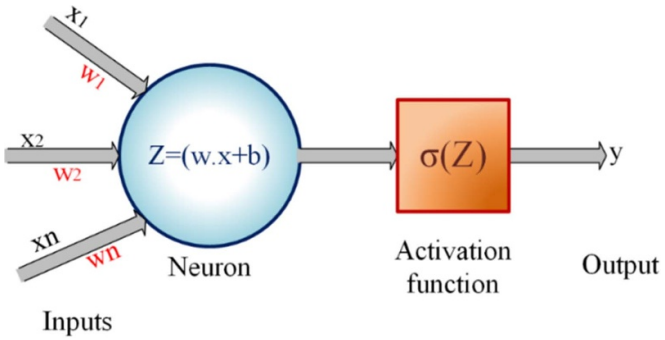


Figure 2. Structure of the model of a neuron used in an ANN.

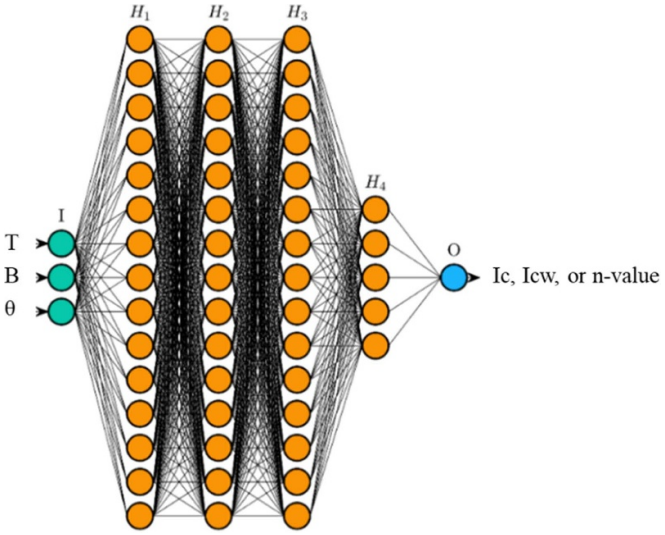


Figure 3. The structure of the proposed ANN model for critical surface predictions.

predicted values and the real ones, known as the loss function. When the loss function is minimised, the ANN model uses the rest of the data for validation and test phases. It should be mentioned that the loss function minimization during the training phase is usually conducted by an optimization procedure of weights vector and bias factor based on a method known as backpropagation (BP). In BP, initially, the training is conducted by considering weights as some small random numbers; after that, and for the first stage, the so-called loss function is calculated. To minimise and update the values of weights, the gradient descent method is used as an optimizer that can be handled using the Levenberg–Marquardt method [26].

The proposed structure of the ANN model used in this paper consists of four hidden layers, in which hidden layers 1, 2, and 3 consist of 15 neurons each, while the last one, i.e. hidden layer 4, consists of 5 neurons. The sigmoid activation function is used as an activation function in this study. In the training stage, the Levenberg–Marquardt method is used to train the model based on 70% of the total input data. To minimise the loss function, a maximum of 1000 epochs is considered for stopping the training phase.

Figure 3 shows the structure of the proposed ANN for the modelling purpose. In this figure, the three inputs are temperature, magnetic field magnitude and orientation angle. The H1–H4 are hidden layers, and their neurons are shown with orange circles. Finally, there is the output layer, which could be I_c, or I_c per cm width, or n-value.

2.2. XGBoost

XGBoost is a decision-tree-based ensemble machine learning algorithm that uses a gradient boosting framework [27]. Briefly, many decision trees are created in an additive manner as the model is trained with the input data. Since it is not possible to evaluate all possible tree structures, the algorithm starts from a single node and iteratively adds branches in the form of split candidates. The formula used for evaluating the instance split candidate nodes I_L (left) and I_R (right) from the starting node I is reported in equation (4) [28]:

$$L_{\text{split}} = \frac{1}{2}\left(\frac{G_{I_L}^2}{H_{I_L}+\lambda} + \frac{G_{I_R}^2}{H_{I_R}+\lambda} - \frac{G_I^2}{H_I+\lambda}\right) - \gamma \qquad (4)$$

where G is the sum of the residuals (namely the difference between the real value and the effective prediction, which is set to a default value at the first iteration) that refer to the corresponding node, H is the number of residuals that refer to the corresponding node, and λ and γ are regularisation parameters [28]. The former regularisation parameter is intended to reduce the prediction sensitivity to individual observations, thus avoiding overfitting. In contrast, the latter determines whether a further partition is to be made (the higher γ the more conservative the model will be by pruning branches). It is worth mentioning that in this paper, the appropriate parameters were searched and optimised by operating hyperparameters tuning. The slip candidate from node I, which returns the higher value of L_{split}, is chosen, since it is the best one at splitting the residuals into the cluster of similar values. Once a tree structure is determined, its outputs for every leaf j are calculated with equation (5):

$$\omega_j = -\frac{G_j}{H_j+\lambda}. \qquad (5)$$

Once the complete decision tree is generated during training, it can be exploited for predicting the target based on a given set of features. Details about the algorithm with which the decision trees are created can be found in [28, 29]. In order to further clarify the decisional mechanism of the trained model of this paper, one of the trees is shown in figure 4. It should be mentioned here that this is only a small part of the decision tree created during the training stage and actually, can be considered a small branch of it. Finally, the green blocks in figure 4 represent the leaves of the tree, namely the possible values that the model can predict.

2.3. KRR

KRR [30] is a linear transform method that allows working with too complex data to be directly addressed through a linear relationship. It uses the kernel trick [31] to transform the dataset to the image space in which it performs a ridge regression [32–34]. The particularity of the ridge regression

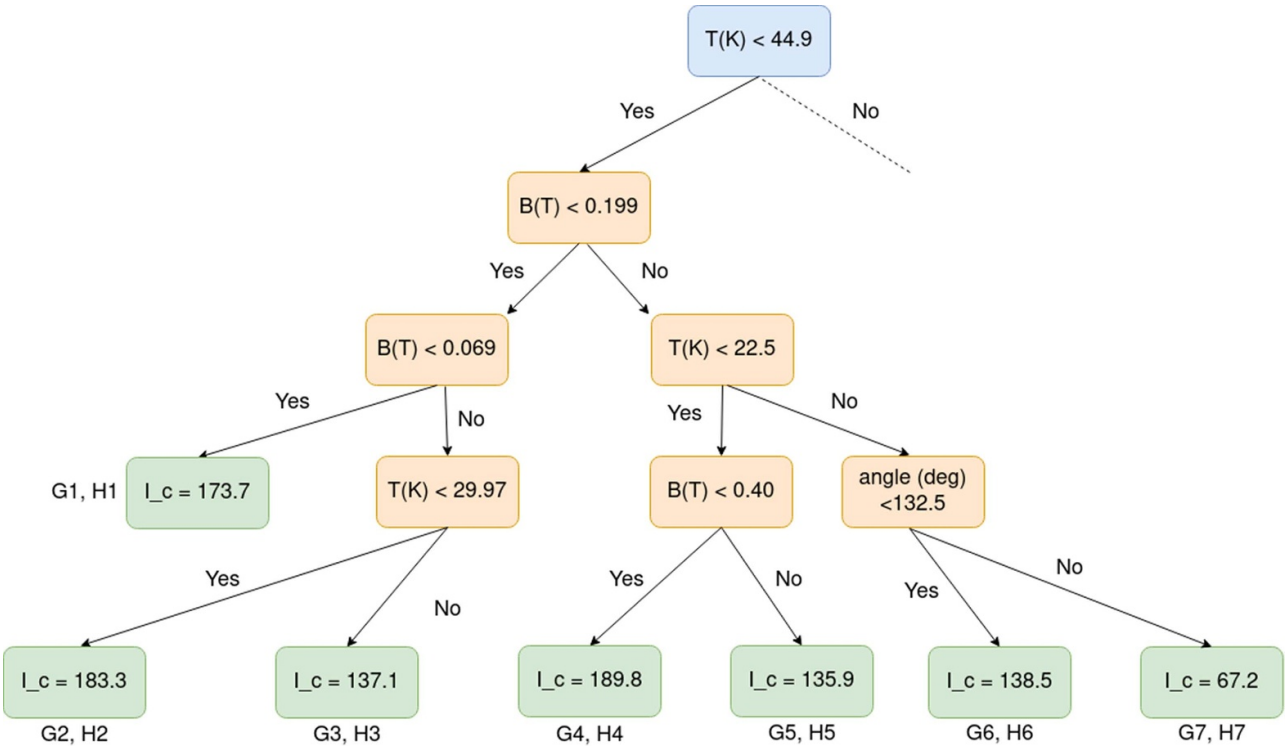


Figure 4. One decision tree of the XGBoost regressor trained model.

that differentiates it from the simple linear regression is the l2-norm regularisation [35], namely the addition of a regularisation term which is the sum of the squares of the model parameters. The objective function that the ridge regression minimises is shown in equation (6):

$$\mathrm{Obj}_{\mathrm{KRR}} = \frac{\sum_{i=1}^{n_s}(y_i - x_i)^2}{n_s} + \sum_{i=1}^{m}\beta_i{}^2 \tag{6}$$

where β_i is the ith model parameter, m is the total number of parameters, and y_i, x_i, and n_s are the same as explained in equation (1).

The regularisation term penalises the minimization of the sum of the squared residuals of the first term but prevents the model parameter from becoming very large and therefore, limits the model bias. Moreover, the appropriate kernel choice for the problem is fundamental to make the KRR operate properly. In this paper, the rational quadratic kernel, k_{RQ} [36], was used, which is shown as follows:

$$k_{\mathrm{RQ}}(x_i, x_j) = \left(1 + \frac{d(x_i, x_j)^2}{2\alpha l^2}\right)^{-\alpha} \tag{7}$$

where α is the scale mixture parameter and determines the relative weighting of large-scale and small-scale variations, l is the length scale of the kernel, and $d(\cdot,\cdot)$ is the Euclidean distance [37]. The best combination of α and l parameters was exhaustively looked for over a wide range of possible values using the grid-search technique [20]. More details about the KRR can be found in [38, 39], whereas another representative case study for such a method is reported in [40].

Table 1. Performance metrics comparison for the I_c per cm width prediction of the three AI models.

Model	RMSE	R^2
ANN	0.0076	0.999 98
XGBoost	14.01	0.999 51
KRR	73.77	0.986 43

3. Results and discussion

In the following section, the results obtained with the three AI techniques are reported. The numerical and visual results are divided into three subsections to present the performances of the proposed models for each chosen target (I_c per unit width, n-value and I_c). It should be noted that, for each of the three target cases, the quantities B, θ, and T always remain the features of the models, but every target requires dedicated training to reproduce the corresponding trend.

The evaluation metrics, i.e. RMSE and R^2 for ANN, XGBoost, and KRR obtained for each of the three targets are reported in tables 1, 2 and 3, respectively. Then, the results of the models are graphically displayed for the I_c per cm width and the n-value, but not for the I_c to avoid redundancy (since the corresponding plots would be highly similar to the I_c per cm width ones). Next, the linear regressions between the experimental values of the target belonging exclusively to the testing subset and the predictions of the AI models are shown for the same combinations of B, θ, and T.

Afterwards, the reconstruction of the critical surfaces of the targets, obtained by interpolation between the experimental points of the complete dataset (which are also included in the

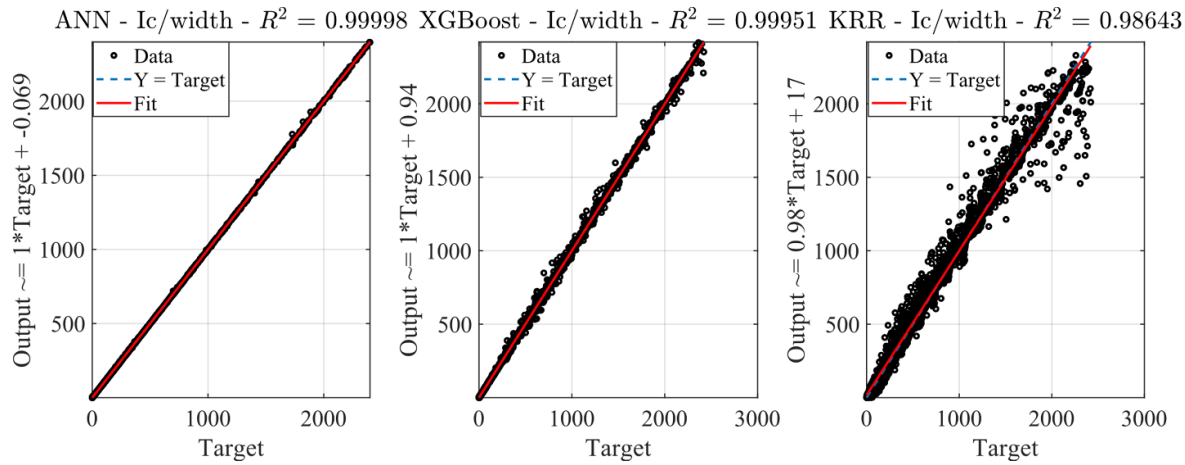


Figure 5. Regression plot for I_c per cm width comparison between experimental values (targets) and predictions made by AI models.

plots for comparison) are reported. This shows the capability of the models in accurately estimating the values of I_c, per width I_c, and n-value for a set of data points that differ from the ones which were fed into AI models during the training process. These critical surfaces are reconstructed at 20 and 65 K, as they are considered relevant and strategic operating temperatures in large-scale power applications in applied superconductivity. Many large-scale power applications would operate at subcooled liquid nitrogen, and therefore their operating temperatures would be essentially 65 K. On the other hand, with the emerging trend of integration of liquid hydrogen (LH_2) in many transportation applications of superconductors, including cryo-electric aircraft [41–43], the characterization of HTS material at boil off temperature of LH_2, i.e. 20 K become more critical.

Next, a comparison between the experimental values of the target belonging exclusively to the testing subset and the predictions of the AI models for the same combinations of B, θ, and T, under given operating conditions are illustrated on 2D plots. The absolute value of the relative error is also included in these figures.

Finally, the absolute value of the relative error of the AI models over the whole B–θ grid at 20 K and 65 K are shown to quantify and detect the weak regions for the operation of such models. Similar to previous plots, these only include testing data.

3.1. Critical current per cm width surfaces

By using ANN, the I_{cw} targets can be estimated with high accuracy at any combination of features, as table 1 reports that RMSE is only 0.0076, R^2 is extremely close to 1 (i.e. 0.999 98), and figure 5 shows that most of the estimated data are located on the Y = Target line (where Target stands for the experimental value of the target and Y for its corresponding prediction). XGBoost and KRR are less accurate, resulting in RMSE equal to 14.01 and 73.77, and R^2 equal to 0.999 51 and 0.986 43, respectively.

Figure 5 also shows that the KRR performance is negatively impacted by the relatively large error for I_{cw} values over 1000 A cm^{-1}, as many predictions do not lay on the Y = T line. Nevertheless, this model can effectively reconstruct the critical surface, as it is shown in figure 6. It is worth mentioning again that the red points labelled as 'Model' are predicted for different feature values than the ones of the experimental dataset, which means they are in positions that the model has never encountered during training. ANN and XGBoost also produce critical surfaces in very good agreement with the experimental data. However, the critical surfaces produced by the XGBoost are not smoothly reconstructed but are composed of a series of bushes of predictions nearby the available experimental data, each separated by a considerable discontinuity from one another. For instance, at T = 20 K, with low field magnitude and angle, the prediction of the I_c per cm width changes by about 500 A cm^{-1} over a short range of a few tens of mT. Such a trend should be no surprise since it was explained in section 2.2 that the XGBoost decision tree and, in turn, its leaves are derived to reduce a set of residuals between the prediction and the dataset target. It is noted that the nature of the dataset itself might have therefore played a major role in producing such a bias of the model over certain target values. In fact, the experimental dataset used in this study follows a particular pattern according to which the B field is kept constant over a certain number of tests during which the field angle is iteratively changed from 0° and 240°. After a set of experiments at a constant B field is complete, the B field is changed and kept constant again until all the whole 0°–240° range of field angle values are explored. The fact that the XGBoost is only trained for certain values of B might have biased the model towards the production of the bushes. A more detailed analysis of the topology of the data is discussed in section 3.5. It is also worth mentioning that the experimental dataset does not provide any data for magnetic field magnitudes below 0.2 T at 20 K, therefore the three models are extrapolating to reconstruct the critical surfaces. Figures 7 and 8 show that the absolute value of the relative error of the ANN model is below 1% for most of the predicted data, and reaches 2%–3% at 2 T and 65 K. The absolute value of the relative error of the KRR is usually included in a range that varies from some units and a few tens of per cent. However, figure 9 underlines the limits

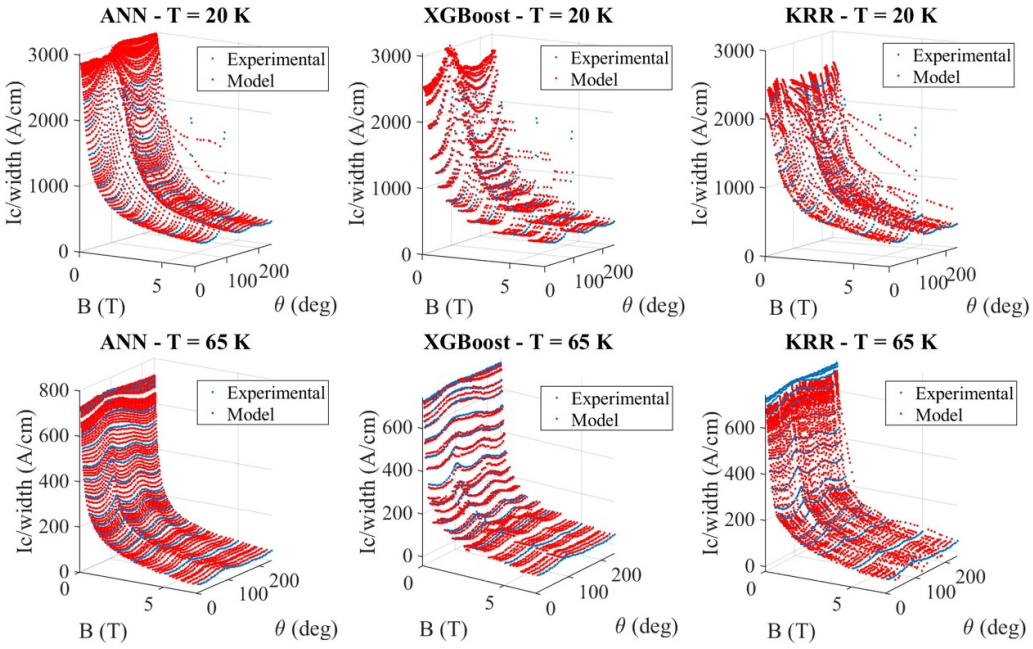


Figure 6. Critical currents per cm width predictions against the experimental data at 20 K and 65 K.

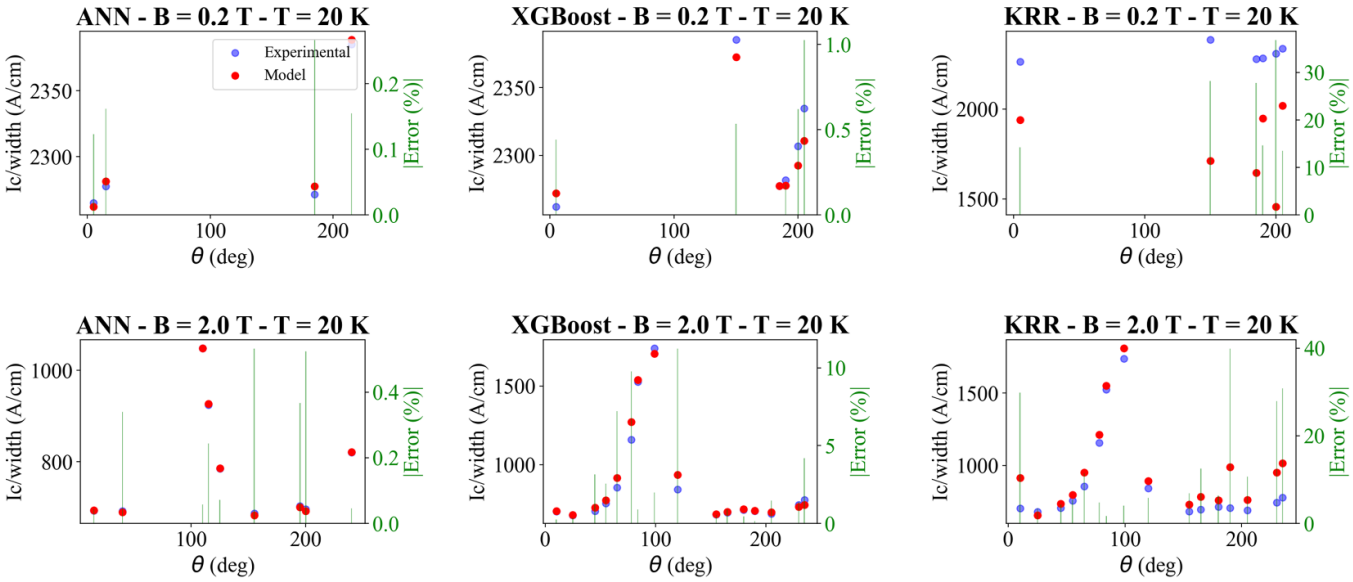


Figure 7. Experimental vs. predicted data and absolute values of the relative error of the I_c/width prediction at 20 K and 0.2 T and 2.0 T (only testing data is considered).

of this model in predicting the I_c per cm width in the regions of very large magnetic field magnitudes and field angles close to 240°, where it generates relative errors of over 50% at 20 K and over 150% at 65 K. It must be noted that, in the experimental dataset, I_c and I_c per unit width dramatically drop at high magnetic fields. For instance, at 65 K the maximum value of I_{cw} is 743.52 A cm^{-1} at $B = 0.02$ T and $\theta = 90°$, whereas its minimum value is 34.34 A cm^{-1} at $B = 7$ T and $\theta = 20°$. Therefore, the lack of accuracy of the model at high fields and temperatures can be explained by such a substantial drop in the value of the target; in fact, an absolute error of 30 A cm^{-1} would result in a relative error of 4% for its maximum value at 65 K and a relative error of 87% for its minimum value at the same temperature.

3.2. *n*-value

Table 2 and figure 10 show that, in terms of R^2, the performance of all three AI models in estimating the *n*-value is worse than the one achieved for the I_{cw}, showing that the trend of this target is more challenging to be intelligently reproduced. In fact, R^2 is calculated equal to 0.987 54, 0.982 63, and 0.911 00 for ANN, XGBoost, and KRR, respectively.

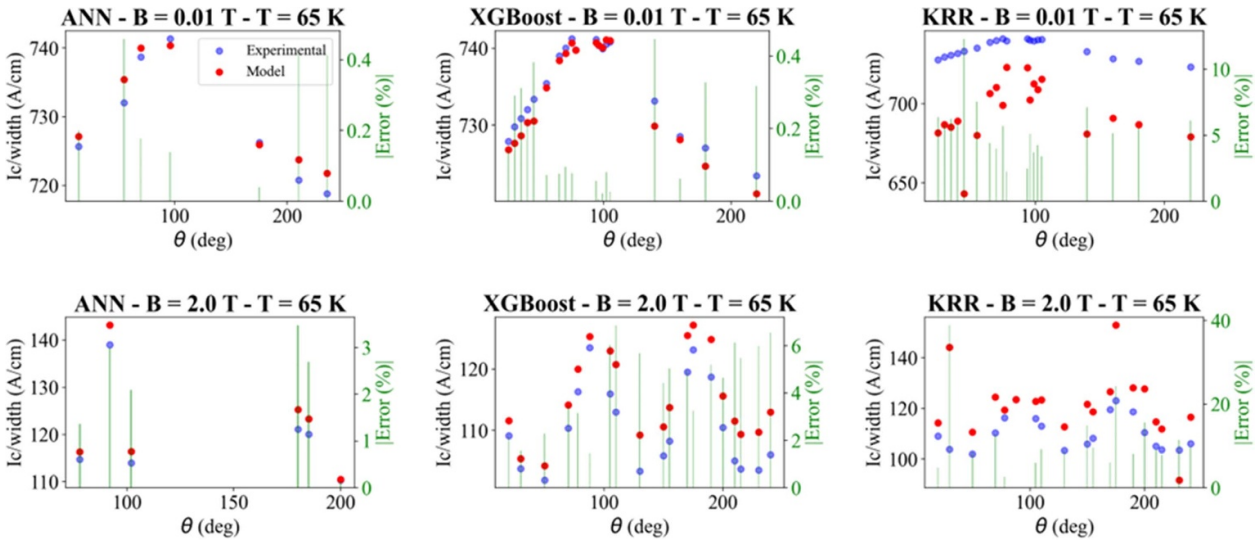


Figure 8. Experimental vs. predicted data and absolute value of the relative error of the I_c/width prediction at 65 K and 0.01 T and 2.0 T (only testing data is considered).

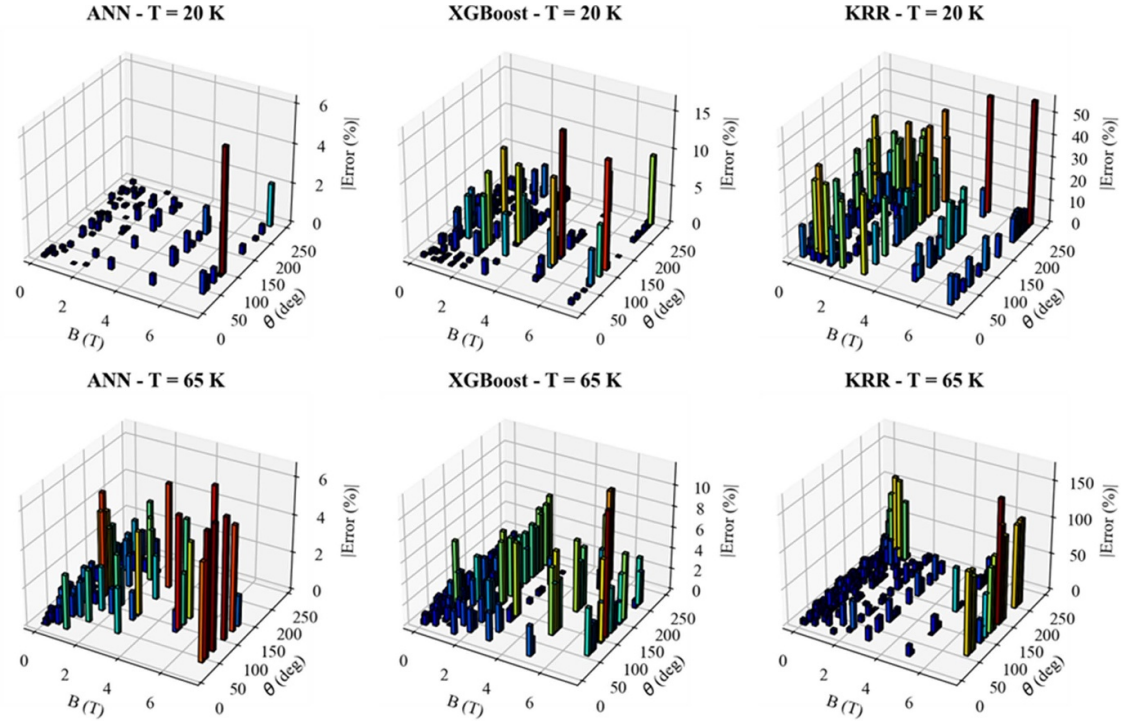


Figure 9. Absolute value of the relative error between I_c/width experimental and predicted data (only testing data is considered).

Table 2. Performance metrics comparison for the *n*-value prediction of the three AI models.

Model	RMSE	R^2
ANN	0.1510	0.987 54
XGBoost	0.8707	0.982 63
KRR	1.974	0.911 00

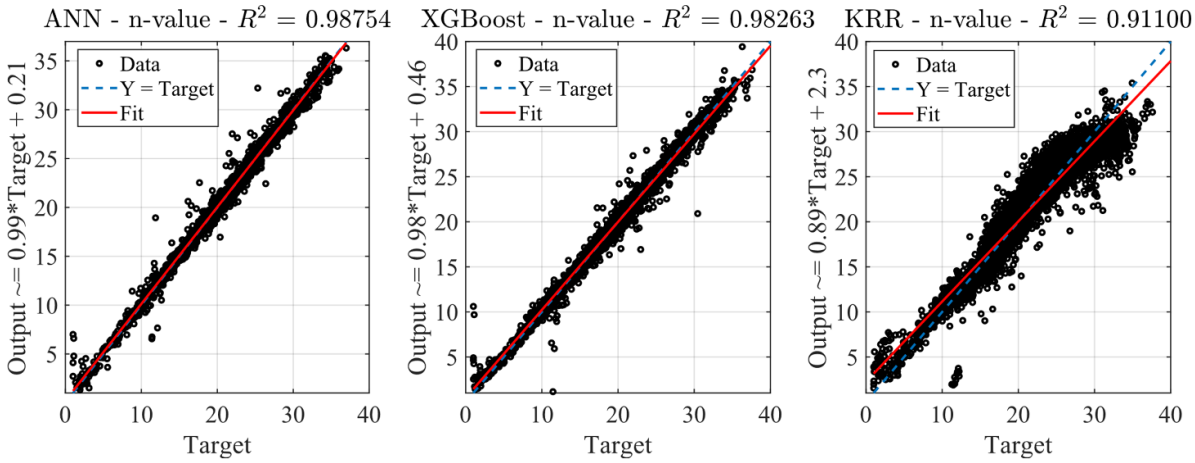


Figure 10. Regression plot—*n*-value comparison between experimental values (targets) and predictions made by AI models.

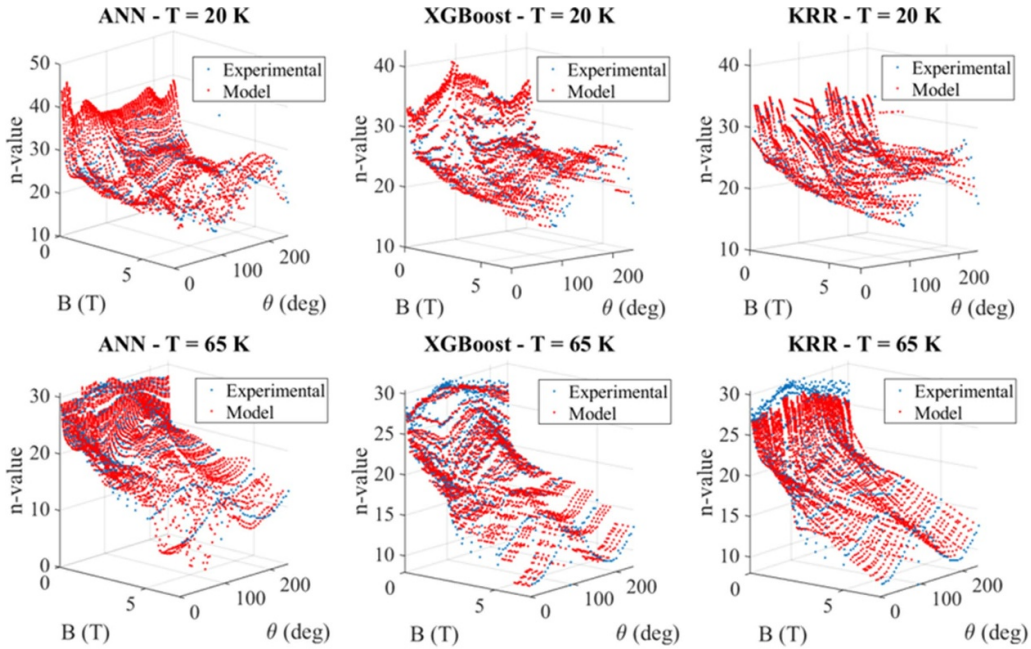


Figure 11. *n*-value predictions against the experimental data at 20 K and 65 K

Figure 11 shows that ANN predicts relatively large values at 20 K and very low fields if compared to the other two models at the same conditions. To explain this, it is reminded that the experimental data at 20 K temperature is only available above 0.2 T, which means the surface at this temperature is reconstructed even slightly outside of the range of the data that the model was trained with.

Nevertheless, figures 12–14 show that the absolute value of the relative error of the ANN model is in a range between only 1% and 6% for the large majority of the predicted data. Exclusively, for very few outliers at 7 T and field angle close to 90°, the relative error reaches a few tens percentages, that is related to topology of the experimental data rather than ANN performance. It will be discussed in detail in section 3.5. On the other hand, the most challenging region for the KRR in predicting the *n*-value is at low magnetic fields, as figure 11 reports relative errors up to 25%.

Finally, the relative error of the XGBoost model is below 10% for the majority of the predictions, except for a few spread outliers at 65 K.

3.3. Critical current

Broadly and qualitatively speaking, the performance regarding the I_c target is very similar to the I_{cw} one for every model, as it is summarised in table 3.

3.4. Sensitivity analyses

In this section, the results of a series of sensitivity analyses on controlling parameters of the three proposed AI-based models are shown.

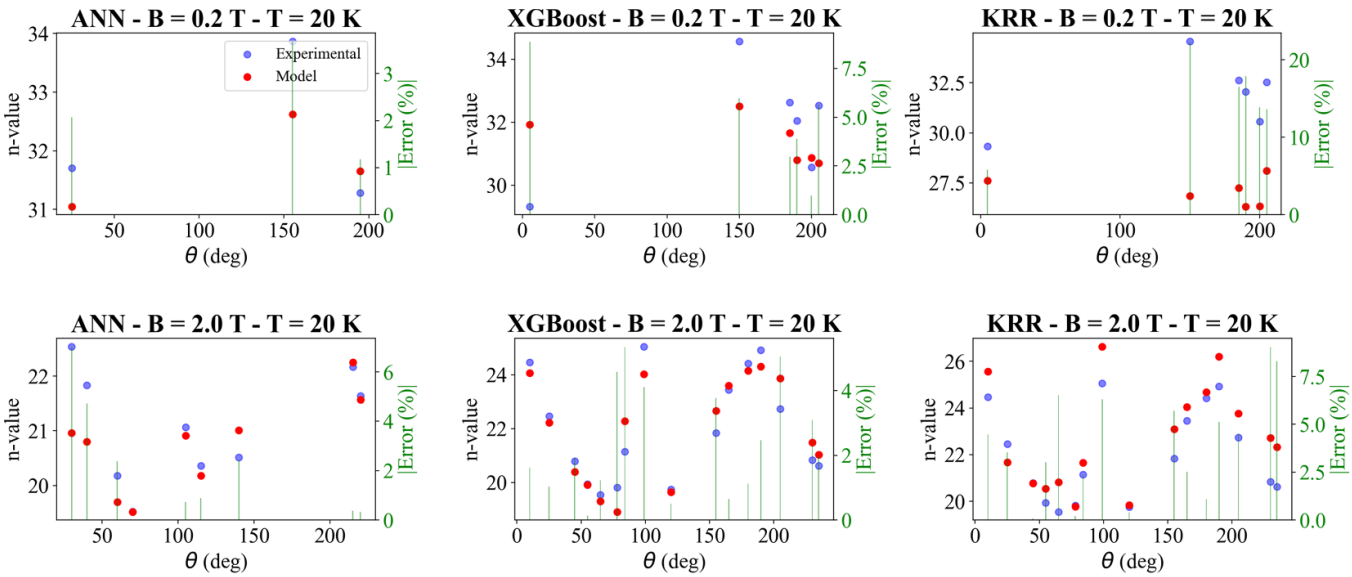


Figure 12. Experimental vs. predicted data and absolute value of the relative error of the *n*-value prediction at 20 K and 0.2 T and 2.0 T (only testing data is considered).

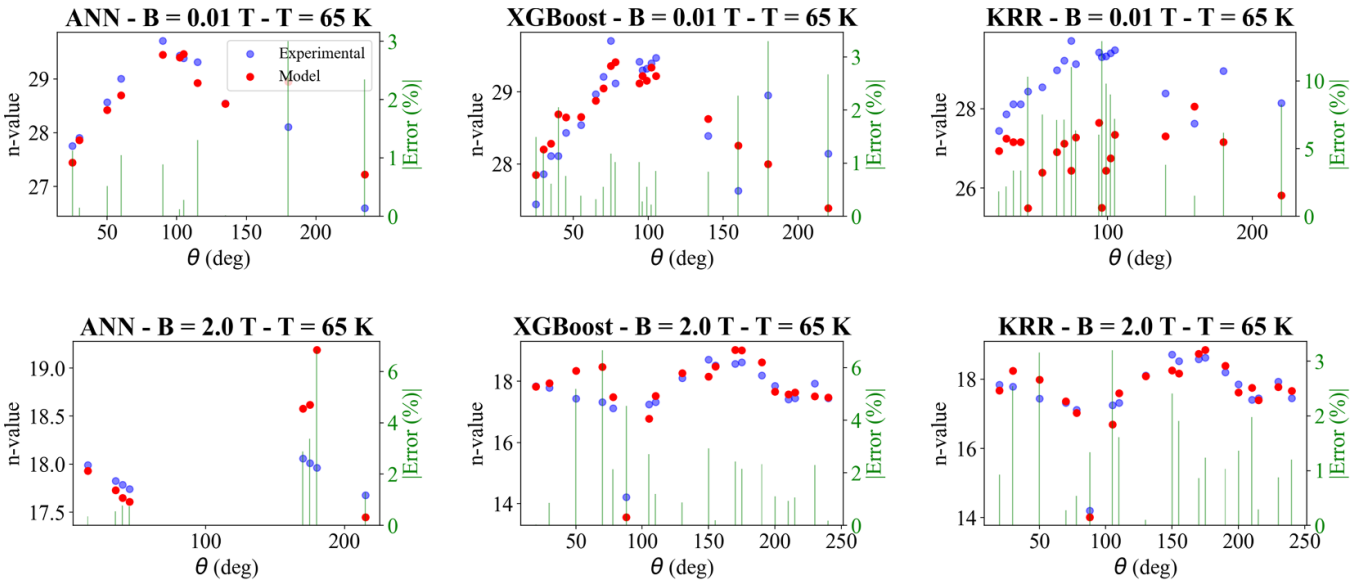


Figure 13. Experimental vs predicted data and absolute value of the relative error of the *n*-value prediction at 65 K and 0.01 T and 2.0 T (only testing data is considered).

3.4.1. ANN. A sensitivity analysis is also conducted to evaluate the impact of changes in controlling parameters of ANN, including the number of neurons and number of hidden layers on the final estimation results, for I_c at 20 K temperature. In general, the higher the number of neurons and hidden layers get, the longer the estimation computation time for testing the neural network would be. With reference to a specific range of number of neurons and hidden layers, accuracy would be increased as well, and RMSE value reduced. Also, it can be conceived that the increase in the number of hidden layers makes the simulations more accurate but slower in comparison to the situation that numbers of hidden layers remain constant and only the number of neurons increases. Table 4 shows the sensitivity analysis results, which are consistent with all previous statements. The selection of the number of neurons and hidden layers is more related to the requirements of the application. For instance, if the ANN model is applied to characterise a superconducting device in design stage, the highest possible accuracy is needed, and so the most complex ANN model with multiple hidden layers and 15–20 neurons in each layer could be selected without any concerns about estimation computation time.

On the other hand, if the ANN model is going to be used for real-time condition monitoring purposes or real-time modelling of the HTS device, the ANN with lower numbers of hidden layer(s) and 10–20 neurons in this layer could be

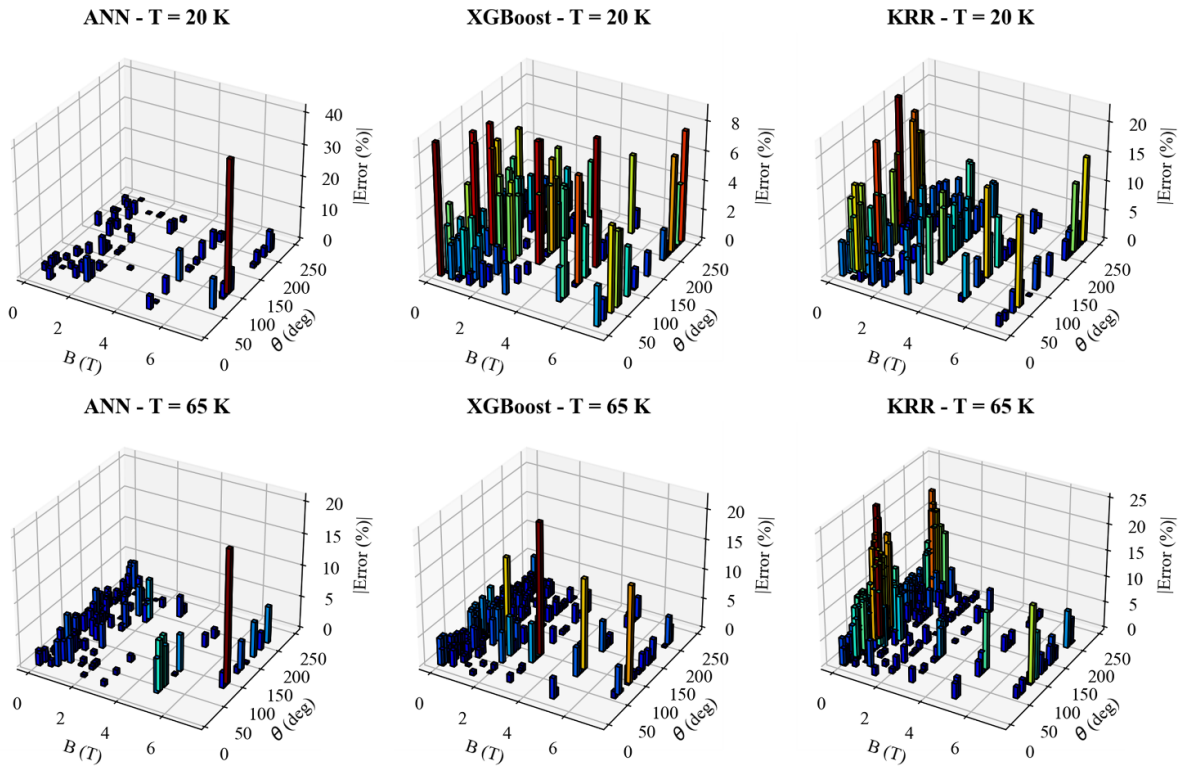


Figure 14. Absolute value of the relative error between n-value experimental and predicted data (only testing data is considered).

Table 3. Performance metrics comparison for the I_c prediction of the three AI models.

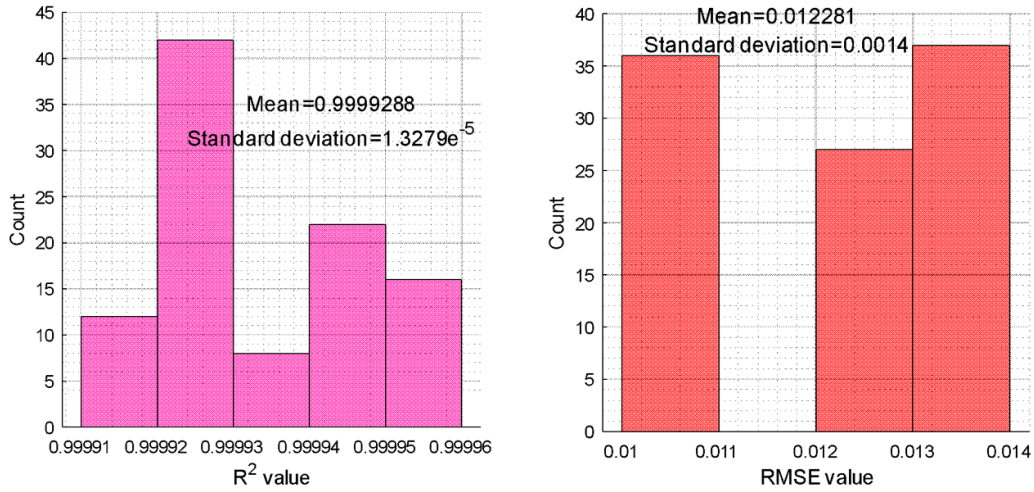
Model	RMSE	R^2
ANN	0.0085	0.999 92
XGBoost	5.884	0.999 46
KRR	29.51	0.986 43

Table 4. Impact of changes in controlling parameters of the ANN model on the results of I_c estimation.

Hidden layers number	Neurons	R^2	RMSE	Computation time (s)
1	5	0.988 56	0.1434	4.873
	10	0.992 87	0.1267	5.436
	15	0.994 15	0.1123	7.012
	20	0.995 54	0.0965	7.555
2	5	0.997 38	0.0752	6.559
	10	0.999 73	0.0235	8.504
	15	0.999 89	0.0152	17.069
	20	0.999 88	0.0156	30.778
3	5	0.998 97	0.0464	7.124
	10	0.999 89	0.0157	12.837
	15	0.999 93	0.0138	31.786
	20	0.999 95	0.0115	85.814
4	5	0.9985	0.0532	9.522
	10	0.999 82	0.0208	20.122
	15	0.999 95	0.0104	64.747
	20	0.999 95	0.0113	158.594
5	5	0.999 35	0.0358	9.476
	10	0.999 91	0.0141	27.186
	15	0.999 94	0.0114	89.734
	20	0.999 97	0.0092	287.384

Table 5. Testing computational time performance of ANN.

Structure	Temperature (K)	Field (T)	Field orientation (degree)	Input number	Testing time (s)
I_c	20	1	0	[1 1 1]	0.000 496
I_{cw}	20	1	0	[1 1 1]	0.000 326
n-value	20	1	0	[1 1 1]	0.000 298
I_c	[20:10:70]	−1.5:0.5:1.5	10:5:30	[6 7 5]	0.001 329
I_{cw}	[20, 30, 40, 50, 60, 70]	[−1.5 −1 −0.5 0 0.5 1 1.5]	[10 15 20 25 30]	[6 7 5]	0.001 2857
n-value	[20, 30, 40, 50, 60, 70]	[−1.5 −1 −0.5 0 0.5 1 1.5]	[10 15 20 25 30]	[6 7 5]	0.001 287
I_c	20:1:70	−2:0.5:2	1:1:360	[51 9 360]	0.002 045
I_{cw}	20:1:70	−2:0.5:2	1:1:360	[51 9 360]	0.002 042
n-value	20:1:70	−2:0.5:2	1:1:360	[51 9 360]	0.002 091



a) R^2 values b) RMSE values

Figure 15. The stability, repeatability, and reproducibility of ANN results after 100 times of repetition of I_c estimation at 20 K temperature.

selected (depends on the nature of the problem) which would not only give us a high accuracy but also characterise the superconductor behaviour in a much faster manner.

To show the fast estimation characteristic of the ANN model, three scenarios are considered in table 5. Firstly, just one input is fed into ANN model during testing phase to estimate I_c, I_{cw}, and n-value of any HTS tape. By doing this, the testing (estimation) time was about 0.2–0.5 ms to estimate the I_c, I_{cw}, and index value. Then, the number of inputs was increased to $6 \times 7 \times 5$ vector, under such circumstances, it takes about 1 ms for ANN model to estimate I_c, I_{cw}, and index value. At last, $51 \times 9 \times 360$ inputs were fed into ANN model and testing time have increased to about 2 ms. These results prove the fast computation testing time of the proposed ANN model to estimate the values, especially in comparison to training time that was in the range of seconds to tens of seconds for I_c, I_{cw}, and n-value. It should also be mentioned that the results for computation time are reported based on running the ANN model on a personal computer equipped with Core™ i7-3612QM 2.1 GHz CPU and 8.0 GB RAM, DDR 3. Thus, any improvements in computation resources could reduce the simulation time to even less than milliseconds without negatively impacting the accuracy of estimation. Moreover, the stability of estimated results is tested to show that the estimations are reproducible and repeatable. In fact, because of its stochastic nature, reproducibility of results by ANN cannot be simply taken for granted and it is therefore investigated. To do this, the results of I_c estimation at 20 K were studied after 100 runs. Figure 15 shows the distribution of RMSE and R^2 values after 100 repetitions. As seen in these figures, the value of R^2 is stable during 100 times of simulation and changes at a maximum of around 0.005%. On the other hand, for RMSE, it can be observed that the values of RMSE remain lower than 0.02, which is quite acceptable for such an estimation purpose. In figures 15(a) and (b), mean and standard deviation values are also reported. After analysing figure 15, stability, reproducibility, and repeatability of the estimations by the ANN model can be well guaranteed. This is solid proof of the fact that the reported estimated values of I_c, I_c per width, and n-value could be achieved by anyone who applies the presented structure of ANN to a reasonable number of data.

3.4.2. XGBoost and KRR. Tables 6 and 7 report the sensitivity analysis results for the XGBoost and the KRR, respectively. In both tables, the results corresponding to the tuned hyperparameters are highlighted in bold, i.e. $\lambda = 0.2$ and $\gamma = 0.6$ for the XGBoost model and $\alpha = 0.15$ and $l = 1.2$ for the KRR.

Therefore table 6 shows the impact of the hyperparameters λ and γ on the I_c per cm width prediction over all available

Table 6. Impact of λ and γ on the I_c per cm width prediction of the XGBoost model.

γ	λ	R^2	RMSE	Computation time (s)
	0.1	0.999 431	15.0568	0.849 808
0.5	0.2	0.999 497	14.1565	0.820 797
	0.3	0.999 419	15.2112	0.824 097
	0.1	0.999 435	14.9958	0.796 851
0.6	**0.2**	**0.999 507**	**14.01**	**0.792 583**
	0.3	0.999 417	15.2351	0.799 748
	0.1	0.999 434	15.0142	0.8126
0.7	0.2	0.999 497	14.1492	0.828 052
	0.3	0.999 434	15.0067	0.812855

Table 7. Impact of α and l on the I_c per cm width prediction of the KRR model.

α	l	R^2	RMSE	Computation time (s)
	0.7	0.981 097	86.7479	27.2499
0.05	1.2	0.983 074	82.0868	27.2099
	1.7	0.982 716	82.9488	27.8582
	0.7	0.984 281	79.1056	24.7503
0.15	**1.2**	**0.986 329**	**73.7715**	**26.3436**
	1.7	0.985 203	76.7509	25.7341
	0.7	0.978 944	91.556	26.0707
0.6	1.2	0.982 994	82.2801	26.6068
	1.7	0.981 495	85.8296	26.9218

data, whereas table 7 is referred to the impact of the hyperparameters α and l on the I_c per cm width. Both these two sensitivity analyses are carried out with a computer equipped with Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40 GHz 2.40 GHz (2 CPUs) and 256 GB RAM.

As one can see in the tables, differently from the ANN, for both of the two models changing hyperparameter values does not affect the computational time of the two algorithms. This is related to the fact that these hyperparameters are not responsible for the structure of the algorithm, but only for the final results of the performance.

Similar to the ANN model, for both XGBoost and KRR the R^2 metric is almost stable for all possible combinations of the hyperparameters, whereas the gap between the best and the worst RMSE is relatively greater.

3.5. Discussion on the topology of the data

One main reason for the poor estimation of the n-value for the proposed methods is related to the topology of experimental data and also the procedure of the n-value calculation. In other words, we know that there is a potential error for any experimental study when one measures a physical or electromagnetic value/parameter. However, here the source of input error is even more complicated, since despite direct measurement of I_c value, n-value needs to be calculated from measured experimental data, and it involves curve-fitting and mathematical calculation to find the value of n.

One reason that some of the mentioned AI models face trouble finding proper regression for the input data is the topology of the experimental data itself. Here we have an input subset consisting of temperature, magnitude, and orientation of the magnetic field. These experimental inputs are changed in a fixed manner according to pre-set intervals, and output which can be critical current or n-value is measured. However, for many fairly similar inputs, therefore, we have very different outputs. Now imagine what the AI model will see is a huge number of similar inputs while their corresponding output is different. Therefore, for AI models it would be very difficult to find a proper pattern and an accurate relationship/regression between inputs and output in such a way that the estimated output always closely follows the real experimental output.

One more interesting point to share with readers is that, for such examples of problems when many similar inputs have very different outputs, the larger the number of total data are, the more the chance of lower estimation accuracy will be. Because it further confuses the AI system with a large amount of data and makes it a bigger hassle to find the relationship/regression between inputs and output. In addition, such topology of experimental input data will be very likely to cause overfitting. Therefore, the problem investigated in this paper shows the importance of preparing appropriate data for an AI-based model. Often, researchers have this misconception that AI models only need 'many' data points, and if one prepares a huge amount of data, the accuracy of the AI model will increase. However, what is crucial for establishing

effective AI models for superconducting applications is the quality of data beyond the data size.

4. Conclusions

In this paper, we have modelled the $I_c(B,\theta,T)$ and $n(B,\theta,T)$ characteristics of HTS tapes using different AI techniques. The models were trained with experimental data obtained from a publicly accessible dataset for HTS tapes. The results of the AI models for reconstructing the aforementioned relations and trends are reported. Intensive research has also been applied in optimising these models to maximise the goodness of fit and minimise the error using controlling and hyperparameters tuning techniques through extensive sensitivity analysis studies. In order to provide a comprehensive study on these AI models and facilitate reproducibility, sensitivity analyses were carried out to investigate the impact of the models' parameters on their capability of reconstructing the $I_c(B,\theta,T)$ and $n(B,\theta,T)$ characteristics from the experimental dataset. It is found that the ANN model is highly accurate in predicting the critical current of HTS tapes depending on their operating conditions, performing R^2 very close to 1 and extremely low RMSE. Despite its stochastic nature, ANN reproducibility has also been tested and demonstrated. The XGBoost model has also proved the capability of tree-based regression models to reconstruct the critical surface of HTS tapes by performing the second highest R^2 metric in predicting the n-value parameter. Although the KRR models have shown worse performance when compared to the two previous models, it has proved to be an effective alternative for assessing the challenge of predicting the critical current and the n-value of HTS tapes. The models' performances have further been investigated by means of the visual reconstruction of the critical surfaces of the tape. The absolute value of the relative error of their predictions is also graphically reported to spot weak regions and analyse the results. Finally, a discussion about the topology of the experimental data is also carried out in relation to the models' results.

Data availability statement

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

Acknowledgments

The authors would like to thank Dr Wimbush and Dr Strickland for providing openly accessible experimental data for critical current and index value characteristics of HTS tapes on the public website. Indeed, this work was not accomplishable without having access to such a comprehensive and large number of data.

The authors would like to acknowledge the support of the COST Action CA19108 'High-Temperature SuperConductivity for AcceLerating the Energy Transition'.

ORCID iDs

Giacomo Russo https://orcid.org/0000-0002-2011-8423
Mohammad Yazdani-Asrami https://orcid.org/0000-0002-7691-3485
Riccardo Scheda https://orcid.org/0000-0002-8872-6804
Antonio Morandi https://orcid.org/0000-0002-1845-4006
Stefano Diciotti https://orcid.org/0000-0001-8778-7819

References

[1] Gömöry F, Šouc J, Vojenčiak M and Klinčok B 2007 *Supercond. Sci. Technol.* **20** S271
[2] Majoros M, Glowacki B A and Campbell A M 2001 *Supercond. Sci. Technol.* **14** 353
[3] Pardo E, Vojenčiak M, Gömöry F and Šouc J 2011 *Supercond. Sci. Technol.* **24** 065007
[4] Viljamaa J, Rostila L, Kováč P, Melíšek T, Hinterberger A and Reissner M 2011 Comparison of different critical current density models for undoped monofilamentary Ti-sheathed MgB_2 tapes *J. Supercond. Nov. Magn.* **24** 287–97
[5] Zermeño V M R, Habelok K, Stępień M and Grilli F 2017 *Supercond. Sci. Technol.* **30** 034001
[6] Yazdani-Asrami M, Taghipour-Gorjikolaie M, Song W, Zhang M and Yuan W 2020 Prediction of nonsinusoidal AC loss of superconducting tapes using artificial intelligence-based models *IEEE Access* **8** 207287–97
[7] Leclerc J, Hell L M, Lorin C and Masson P J 2016 Artificial neural networks for AC losses prediction in superconducting round filaments *Supercond. Sci. Technol.* **29** 065008
[8] Zhang Z *et al* 2018 AC loss prediction model of a 150 kJ HTS SMES based on multi-scale model and artificial neural networks *IEEE Trans. Magn.* **54** 1–5
[9] Wen Z, Zhang H and Mueller M 2021 *Supercond. Sci. Technol.* **34** 125019
[10] Pan H and Shen G X 2003 Application of artificial neural network methods in HTS RF coil design for MRI *Concepts Magn. Reson.* **18** 9–14
[11] Richard L S, Bonifetto R, Carli S, Froio A, Foussat A and Zanino R 2014 Artificial neural network (ANN) modeling of the pulsed heat load during ITER CS magnet operation *Cryogenics* **63** 231–40
[12] Liu Q and Kim S 2022 Temperature-field-angle dependent critical current estimation of commercial second generation high temperature superconducting conductor using double hidden layer Bayesian regularized neural network *Supercond. Sci. Technol.* **35** 035001
[13] Tomassetti G, de Marzi G, Fiamozzi Zignani C, Giorgetti F and Della Corte A 2022 *Supercond. Sci. Technol.* **35** 014002
[14] Wimbush S *et al* 2017 *Robinson Research Institute* (available at: http://htsdb.wimbush.eu/)
[15] Strickland N M, Hoffmann C and Wimbush S C 2014 A 1 kA-class cryogen-free critical current characterization system for superconducting coated conductors *Rev. Sci. Instrum.* **85** 113907
[16] Russo G *et al* 2021 Critical surface reconstruction for HTS tapes using artificial intelligence techniques *Hi-Scale COST Meeting, COST Action CA19108* (*Bologna*)
[17] Pina J M *et al* 2020 CA19108—high-temperature superconductivity for accelerating the energy transition *(COST, European Cooperation in Science & Technology)* (available at: www.cost.eu/actions/CA19108/)

[18] Wimbush S C and Strickland N M 2017 A public database of high-temperature superconductor critical current data *IEEE Trans. Appl. Supercond.* **27** 1–5
[19] Yazdani-Asrami M, Sadeghi A, Seyyedbarzegar S M and Saadat A 2022 Advanced experimental-based data-driven model for the electromechanical behavior of twisted YBCO tapes considering thermomagnetic constraints *Supercond. Sci. Technol.* **35** 054004
[20] Pedregosa F *et al* 2011 sklearn.model_selection.GridSearchCV (available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html)
[21] Hastie T, Tibshirani R, Friedman J H and Friedman J H 2009 *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* vol 2 (New York: Springer) pp 1–758
[22] McCulloch W S and Pitts W 1943 A logical calculus of the ideas immanent in nervous activity *Bull. Math. Biol.* **5** 115–33
[23] Tino P, Benuskova L and Sperduti A 2015 *Springer Handbook of Computational Intelligence* (New York: Springer)
[24] Abiodun O I, Jantan A, Omolara A E, Dada K V, Mohamed N A and Arshad H 2018 State-of-the-art in artificial neural network applications: a survey *Heliyon* **4** e00938
[25] Graupe D 2013 *Principles of Artificial Neural Networks* (Singapore: World Scientific Publishing)
[26] Krogh A 2008 What are artificial neural networks? *Nat. Biotechnol.* **26** 195–7
[27] Chen T and Guestrin C 2021 *github* (available at: https://github.com/dmlc/xgboost)
[28] Chen T and Guestrin C 2016 XGBoost: a scalable tree boosting system *Association for Computing Machinery (New York, USA)* pp 785–94
[29] Public 2021 XGBoost documentation (available at: https://xgboost.readthedocs.io/en/latest/tutorials/model.html)
[30] Murphy K P 2012 *Machine Learning: A Probabilistic Perspective* (Cambridge, MA: The MIT Press) ch 14.4.3, pp 492–3
[31] Hofmann M 2006 Support vector machines-kernels and the kernel trick *Notes* **26** 1–16
[32] Cueto M D 2020 *Kernel Ridge Regression—Python Tutorial* (available at: www.mdelcueto.com/blog/kernel-ridge-regression-tutorial/)
[33] Pedregosa F *et al* 2011 *JMLR* **12** 2825–30
[34] Pedregosa F *et al* 2011 *scikit-learn* (available at https://scikit-learn.org/stable/modules/generated/sklearn.kernel_ridge.KernelRidge.html)
[35] Cortes C *et al* 2012 L2 regularization for learning kernels (arXiv:1205.2653)
[36] Pedregosa F *et al* 2011 *sklearn.gaussian_process.kernels.RationalQuadratic* (available at: https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.RationalQuadratic.html)
[37] Duvenaud D 2014 The Kernel cookbook: advice on covariance functions (available at: www.cs.toronto.edu/duvenaud/cookbook)
[38] Pedregosa F *et al* 2011 *Kernel ridge regression user guide* (available at https://scikit-learn.org/stable/modules/kernel_ridge.html#kernel-ridge)
[39] Vovk V 2013 Kernel ridge regression *Empirical Inference* (New York: Springer) pp 105–16
[40] Zhao Z-W, Del Cueto M, Geng Y and Troisi A 2020 Effect of increasing the descriptor set on machine learning prediction of small molecule-based organic solar cells *Chem. Mater.* **32** 7777–87
[41] Yazdani-Asrami M, Seyyedbarzegar S M, Zhang M and Yuan W 2022 Insulation materials and systems for superconducting powertrain devices in future cryo-electrified aircraft: part I—material challenges and specifications, and device-level application *IEEE Electr. Insul. Mag.* **38** 23–36
[42] Yazdani-Asrami M, Zhang M and Yuan W 2021 Challenges for developing high temperature superconducting ring magnets for rotating electric machine applications in future electric aircrafts *J. Magn. Magn. Mater.* **522** 167543
[43] Yazdani-Asrami M *et al* 2022 Selecting a cryogenic cooling system for superconducting machines: general considerations for electric machine designers and engineers *Int. J. Refrig.* **140** 70–81