ISSN: 2710-0588 DOI: 10.29556/FSE.202109_1(6).0021

Prediction of Materials Property based on Machine Learning

Ying Cao[1,*]

[1]Department of Culture Technology, Jeonju University, Jeonju 55069, Korea.

Abstract

Two feature extraction methods and three machine learning methods are used to predict the coercivity properties of NdFeB magnetic materials. First, use PCA, Auto-encoder feature extraction methods to reduce high-dimensional features, and use Random Forest, GBRT and ANN to predict the coercivity properties of magnetic materials. Second, identify the most effective model that combines feature selection and machine learning to predict the magnetic properties of the material, the performance of the combination of different feature extraction methods and machine learning models is compared. The experimental results show that: (1) the auto encoder+neural network method helps to obtain the best prediction performance; (2) the machine learning model has a greater impact on the prediction results than the feature selection method. This research provides a way to find a more effective prediction method for NdFeB magnetic materials.

Keywords

Machine Learning; Magnetic Material; Performance Prediction.

1. Introduction

The development of materials has long been valued by various countries, and the discovery of new materials to be able to drive social progress by leaps and bounds [1]. Due to the complexity of the inherent relationship of materials The traditional materials research method has a long period, high cost and certain Occasionality and other deficiencies cannot meet the current demand for material research and development [2]. Since the 1990s, thanks to the innovation of computer technology, data-driven science based on data and machine learning as the core has become the fourth paradigm of scientific discovery following experimental science, theoretical induction, and computational simulation (as shown in Figure [3].] Shown), which can handle hidden correlations between large amounts of data while ignoring complex internal mechanisms, thereby shortening the development cycle and reducing costs. The implementation of the "Material Genome Initiative (MGI)" has promoted the application and development of data-driven technology in materials. Its core is to generate large amounts of data through high-throughput experiments and calculations, and apply big data technology for data storage. Excavate to obtain valuable laws [4-5], and transform them into reliable knowledge under the guidance of expert experience and theory, so as to obtain the relationship between material composition-process-structure-performance [6-7], which is new provide basis for material design and material modification. How to use existing material data or purposefully generate data for material research and development is the key to data-driven materials science.

In machine learning problems, the essence of machine learning algorithms is to find a suitable mapping relationship between features and targets. Only when there is a potential relationship between features and targets, can better prediction results be obtained. When machine learning algorithms are used for material performance prediction, the features are different material properties. To make the prediction effect good, it is necessary to find suitable material properties (descriptors) for prediction. Therefore, choosing the right descriptor is also one of the main problems in material performance prediction.

ISSN: 2710-0588 DOI: 10.29556/FSE.202109_1(6).0021

The paper mainly uses experimental methods to find the best features for different machine learning models. Through the fusion of feature extraction methods and machine learning models, it compares different feature selection methods (PCA [8], AutoEncoder [9]) in different machine learning models (Random Forest [10], GBRT [11], Neural Network [12]), summed up the best combination model, and provides a reference for the development of a more effective material performance prediction model for NdFeB magnetic materials.

2. Methodology

2.1. Material performance prediction method based on machine learning

Figure 1 shows the predictive modeling process of NdFeB magnetic materials based on machine learning. First, collect a dataset of NdFeB materials with known coercivity properties; then, standardize or normalize the original data to eliminate the influence of each data feature on the prediction results due to different orders of magnitude; then, use different the feature extraction method selects a specified number of feature subsets and provides them as input to the machine learning model; after that, ten-fold cross-validation and different machine learning models are used to predict, and the prediction results are represented by the specified evaluation; finally, the comparison is The best combination model of feature selection and machine learning. The experiment mainly involves two methods of feature extraction and three machine learning regression models. These feature extraction methods and machine learning models will be introduced below.

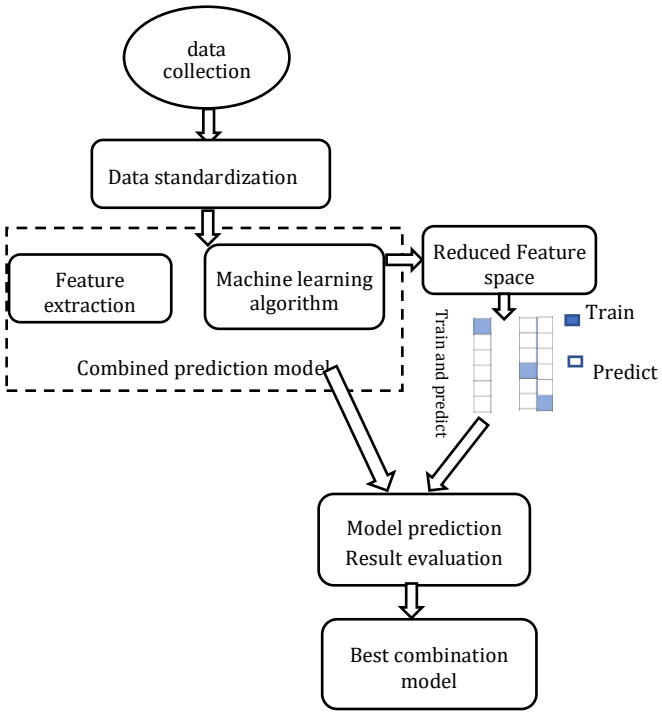

Figure 1. Flow chart of the prediction method

2.2. Feature selection method

2.2.1. Principle Component Analysis (PCA)

PCA [8]is a common feature extraction method in data science. Technically, PCA finds the eigenvectors of a covariance matrix with the highest eigenvalues and then uses those to project the data into a new subspace of equal or fewer dimensions. It tends to find the direction of maximum variation (spread) in data. PCA is more useful when dealing with 3 or higher-

ISSN: 2710-0588 DOI: 10.29556/FSE.202109_1(6).0021

dimensional data. First, identify the relationship among features through a Covariance Matrix. Through the linear transformation or eigendecomposition of the Covariance Matrix, then get eigenvectors and eigenvalues, transform our data using Eigenvectors into principal components. Lastly, quantify the importance of these relationships using Eigenvalues and keep the important principal components.

2.2.2. Auto-encoders Networks(AE) for feature extraction

Auto-encoder[9] is a generative model where an ANN(Artificial Neural Networks) is trained to reconstruct its own input in an unsupervised way. An autoencoder employs an asymmetric structure composed of two main blocks: an encoder part that compresses the input into a low dimensional representation that contains the informative content of the data; a decoder part that is trained to reconstruct the input from the features extracted by the encoder. Once the unsupervised pretraining is completed, the encoder part is thus a powerful automatic feature extractor that, completed with a suitable output layer, can be then fine-tuned [6] in a supervised way in order to obtain the desired estimator more in detail, fine-tuning allows to adjust the parameters of the encoder in order to extract features that are even more effective in addressing the specific target estimation problem. This operation is performed by adding a further layer to the encoder block. Based on the problem at hand, the encoder can be followed by either a classification or a regression output layer.

Our problem can be formalized as a regression one. Thus, in the proposed approach, a linear output layer is added to the encoder, then the resulting network is trained in a supervised fashion. Note that previously computed encoder weights provide the starting point to the stochastic gradient descent algorithm. As a consequence, fine-tuning is expected to refine the features extracted in an unsupervised fashion, so that they are even more valuable in tackling the regression problem. Autoencoders can be created using various Neural Network structures.

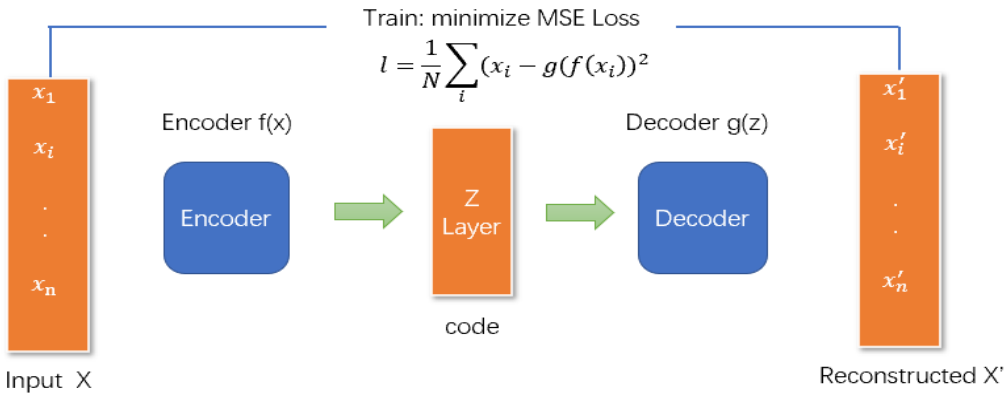

Figure 2. Auto-encoder network architecture

As shown in Fig. 2, suppose we have the dataset X, then apply an auto-encoder to first compress X into a low dimensional intermediate representation Z and then map it back to a reconstructed copy X'. The encoder function is $z_i = f(x_i)$ with non-linear neurons, the decoder function is $x'_i = g(z_i)$ with the antisymmetric shape of the encoder. The reconstruction error of $x_i$is the mean squared error:

$$error_i = ||x_i - g(f(x_i))||^2 \quad (1)$$

the auto-encoder network is learned by minimizing the reconstruction loss:

$$loss = \frac{1}{|X|}\sum_{x_i \in X} error_i = \frac{1}{|X|}\sum_{x_i \in X} ||x_i - g(f(x_i))||^2 \quad (2)$$

The bottleneck z layer in the middle is the compressed feature space which we want to calculate

$$z = f(x) \quad (3)$$

ISSN: 2710-0588 DOI: 10.29556/FSE.202109_1(6).0021

2.3. Machine learning regression model

2.3.1. Random Forest regression model

Breiman [10] proposed random forests, which add an additional layer of randomness to bagging In addition to constructing each tree using a different bootstrap sample of the data, random forests change how the classification or regression trees are constructed. In standard trees, each node is split using the best split among all variables. In a random forest, each node is split using the best among a subset of predictors randomly chosen at that node. This somewhat counterintuitive strategy turns out to perform very well compared to many other classifiers including discriminant analysis, support vector machines, and neural networks, and is robust against overfitting .In addition, it is very user-friendly in the sense that it has only two parameters (the number of variables in the random subset at each node and the number of trees in the forest) and is usually not very sensitive to their values.The random forests algorithm is as follows:

1. Draw tree bootstrap samples from the original data.

2. For each of the bootstrap samples, grow an unpruned regression tree, with the following modification: at each node, rather than choosing the best split among all predictors, randomly sample entry of the predictors and choose the best split from among those variables.

3. Predict new data by aggregating the predictions of the ntree trees (i.e., majority votes for classification, the average for regression). An estimate of the error rate can be obtained.

4.Based on the training data. At each bootstrap iteration, predict the data not in the bootstrap sample (what Breiman calls "out-of-bag", or OOB, data) using the tree grown with the bootstrap sample.

5. Aggregate the OOB predictions. Calculate the error rate, and call it the OOB estimate of error rate.

2.3.2. Gradient Boosted Regression Trees (GBRT)

Similar to Random Forests. GBRT [11] is a machine learning technique that is also based on tree averaging. However, instead of training many full high variance trees that are averaged to avoid overfitting,

In gradient boosting decision trees, it combine many weak learners to come up with one strong learner. The weak learners here are the individual decision trees. All the trees are conncted in series and each tree tries to minimise the error of the previous tree. Due to this sequential connection, boosting algorithms are usually slow to learn, but also highly accurate. In statistical learning, models that learn slowly perform better.

The weak learners are fit in such a way that each new learner fits into the residuals of the previous step so as the model improves. The final model aggregates the result of each step and thus a strong learner is achieved.

A loss function is used to detect the residuals. For instance, mean squared error (MSE) can be used for a regression task and logarithmic loss (log loss) can be used for classification tasks. It is worth noting that existing trees in the model do not change when a new tree is added. The added decision tree fits the residuals from the current model.

2.3.3. Aritifical Neural Network

Artificial Neural Networks (ANN) [12] are relatively crude electronic models based on the neural structure of the brain. The brain basically learns from experience. ANN are computers whose architecture is modeled after the brain. They typically consist of hundreds of simple processing units which are wired together in a complex communication network. Each unit or node is a simplified model of a real neuron which sends off a new signal or fires if it receives a sufficiently strong Input signal from the other nodes to which it is connected. Neural network (NN) layers are independent of one another; that is, a specific layer can have an arbitrary

ISSN: 2710-0588 DOI: 10.29556/FSE.202109_1(6).0021

number of nodes. This arbitrary number of nodes is called a bias node. The bias nodes are always set as equal to one. In analogy, the bias nodes are like the offset in linear regression given as; y = ax + b, where "a" is the coefficient of independent "x" and then "b" is called slope. A bias major function is to provide nodes with a constant value that is trainable, in addition to the normal inputs received by the network node. Importantly, a bias value enables one to move the activation function either to the right or the left, which can be analytical for ANN training success. When the NN is used as a classifier, the input and the output nodes will match input features and output classes. However, when the NN is used as a function approximation, it generally has an input and an output node. However, the number of designed hidden nodes is essentially greater than those of input nodes.

In this paper, we applied relu activation and batch normalization to all layers, except for the last layer. The Adam optimization algorithm was used to select the best model with the lowest loss. Set the training epoch is to 500 and iteratively minimizes the objective function.

3. Experiment and discussion

3.1. Dataset and preprocessing

The material data used in this article is the experimental data of the NdFeB material through the grain boundary diffusion process to improve the magnetic properties, including 214 data and 41 characteristic attributes. Characteristic properties include four aspects of material properties, such as basic magnet composition, diffusion compound composition, processing conditions, and magnetic properties. For real-world experimental data, manual preprocessing, missing value filling, outlier removal, feature selection, and feature reorganization are first required. This paper uses the median value of feature attributes to fill in missing values. Features with missing feature values exceeding 30% are deleted. If more than 5 feature values of a single piece of data are missing, then this row of data is deleted; for the outliers of a single feature, use Statistic mean square deviation, any data point more than 3 times the standard deviation, then these points are likely to be outliers or outliers, delete these entries. Feature selection and feature reorganization are analyzed by material professionals, and finally, 178 data, 32 features, and a target performance are collected, which constitute the data set used in the experiment.

The software used for experimental data processing and algorithm model in the experiment is python, and the software tool used for data processing and algorithm model calculation in the experiment is Python [13]. As open-source software, Python has a rich and powerful library, which has been used by more and more scientific researchers for machine learning research and experiments. The library Scikit-learn [14] contains a large number of machine learning methods, which can be efficient and convenient. The ground is used for experiments. The data preprocessing, feature selection, and machine learning algorithm models used in the article are also selected from the Scikit-learn machine learning library.

Data normalization[15]and standardization can not only eliminate the influence of different features on the prediction model when the magnitude of the difference is large but also keep the data in the original distribution, which is helpful to improve the prediction effect of the model. The original data has been normalized or standardized. When predicting the magnetic properties of NdFeB materials, this paper standardizes the feature data, that is, transforms each feature through the following formula:

$$\widehat{x_i} = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{4}$$

Each feature consists of a set of m samples, where x= (x1, x2,..., xm), xi∈x; μ represents the mean value of this feature set, σ^2 represents the variance, and ϵ points the random error, Through standardization, the mean value of the data is 0 and the standard deviation is 1.

ISSN: 2710-0588 DOI: 10.29556/FSE.202109_1(6).0021

3.2. Evaluation

In this paper, the mean square error MSE, the average absolute error MAE and the goodness of fit R^2 are mainly used as the performance evaluation indicators of the prediction model. MSE is defined as follows:

$$MSE = \frac{1}{|Y|}\sum_{y_i \in Y}(y_i - \hat{y}_i)^2 \quad (5)$$

MAE is defined as follows:

$$MAE = \frac{1}{|Y|}\sum_{y_i \in Y}|y_i - \hat{y}_i| \quad (6)$$

The maximum value of R^2 is 1. The closer the value is to 1, the better the fit. The calculation method is as follows:

$$R^2 = 1 - \frac{\sum_{y_i \in Y}(y_i - \hat{y}_i)^2}{\sum_{y_i \in Y}(y_i - \bar{y})^2} \quad (7)$$

In the above formulas, Y is the target property of the sample, and |Y| is the number of data samples, y_i and $\hat{y}_i$ are the true target value and predicted value of the i-th sample respectively, the average value

$$\overline{y_i} = \frac{1}{|Y|}\sum_{y_i \in Y} y_i \quad (8)$$

3.3. Experimental results and analysis

Since the target attributes of material prediction are continuous values, the prediction of magnetic properties is a regression problem. Most of the features in the data are continuous values, a small part is discrete values and does not have a certain order relationship, so these feature data are specially coded. The machine learning model parameters are obtained through the grid search GridSearchCV method, and finally, R^2, MSE, MAE is used as the evaluation. Using the above-mentioned 2 feature extraction methods and 3 machine learning models for 2 × 3 combinations, 6 combinations are obtained to predict material properties.

Table 1. Comparisons of RF, GBRT, ANN with various feature extraction methods

Feature exrtaction / Regression Algorithm	PCA			AE		
	R^2	MSE	MAE	R^2	MSE	MAE
RF	0.8078	2.7544	1.1797	0.8422	2.3768	1.3532
GBRT	0.8365	1.9933	1.2867	0.8216	2.4476	1.5432
ANN	0.8826	1.3512	**0.9876**	**0.9002**	**1.3132**	1.1287

As shown in Table 1, RF represents random forest regression, GBRT represents Gradient Boosted Regression Trees, and ANN represents the artificial neural network regression model. For the feature extraction method, PCA represents principal component analysis, AE represents the auto-encoder network. The performance of ANN+AE has significantly better than the other 5 combination algorithms, all three evaluations can perform best. Here we take the best-performing ANN regression algorithm as an example. In the experiment, a simple three-layer network structure is used where hidden nodes are 64,16,1, Relu as activation function, Adam as optimization, 0.005 as the learning rate.

Discussing the performance of the feature extraction algorithm for regression prediction, we found that for the same regression algorithm, the difference between the results of using PCA and AE is not very obvious, but overall the AE effect is still better. PCA selects the extracted feature dimensions according to the importance of the features. According to the optimal effect of the experiment, we find that the performance is best when the feature dimension is 20. AE for feature extraction, the results are generated by optimized the Hyperparameter, the number

ISSN: 2710-0588 DOI: 10.29556/FSE.202109_1(6).0021

of nodes in each layer is 30, 20, z, 20, 30, z is the number of nodes in the bottleneck layer, finally optimized as 4 through experiments.

4. Conclusion

In this paper, two feature extraction methods and three machine learning models are used. The magnetic properties of 178 pieces of NdFeB grain boundary diffusion data were predicted by the model. The analysis found that:

(1) The performance of different feature selection methods on the same machine learning model is not much different, while the same feature selection method is used in different machine learning models. There is a big difference in the above, and the PCA feature selection method is better;

(2) Among the three machine learning models, the neural network method has better results, and the prediction results of the pca+neural network combined model are generally better. This combined model uses The goodness of fit can reach 0.9002 in the prediction.

the research results in this article provide a useful reference for further research and development of effective material performance prediction features and models and provide a choice for the reverse design of material features with target properties that can be used.

References

[1] Wang H, Xiang Y, Xiang X D, et al (2015). Science&Technology Review. 2015, 33(10),13.

[2] He P, Lin P P(2019). Materials Review A: Review Papers. 2019, 33(1),156.

[3] Himanen L, Geurts A, Foster A S, et al (2019). Advanced Science, DOI: 10.1002 /advs.201900808.

[4] Agrawal A, Choudhary A(2016).Perspective: Materials informatics and big data: Realization of the “fourth paradigm” of science in materials science. APL Materials, DOI: 10.1063 /1.4946894.

[5] RAMPRASAD Rampi, BATRA Rohit, PILANIA Ghanshyam,et al (2017). Machine learning in materials informatics:recent applications and prospect [J] Computational Materials Science, 2017,54(3):1-13.

[6] Hosni M,Abnane I,Idri A,et al(2019).Computer Methods and Programs in Biomedicine,2019,177,89.

[7] Wang H,Xiang Y,Xiang X D,et al. Science & Technology Review, 2015,33(10),13.

[8] J. C. B. Melo, G. D. C. Cavalcanti and K. S. Guimaraes(2003). "PCA feature extraction for protein structure prediction," Proceedings of the International Joint Conference on Neural Networks, 2003., 2003, pp. 2952-2957 vol.4, doi: 10.1109/IJCNN.2003.1224040.

[9] Kramer, Mark A. (1991). "Nonlinear principal component analysis using auto associative neural networks" (PDF). AIChE Journal. 37 (2): 233-243. doi: 10.1002/aic.690370209.

[10] Breiman, “Random Forests”, Machine Learning, 45(1), 5-32, 2001.

[11] Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics, 29(5), 1189–1232.

[12] Anil K Jain, Jianchang Mao and K.M Mohiuddin, “Artificial Neural Networks: A Tutorial”, Michigan State university, 1996

[13] Hosni M,Abnane I,Idri A,et al.Computer Methods and Programs in Biomedicine,2019,177,89.

[14] SANNER M F.(1999) Python: a programming language for software integration and development[J]. Journal of Molecular Graphics and Modelling, 1999, 17(1): 57-61.

[15] PEDREGOSA Fabian, VAROQUAUX Gal, GRAMFORT Alexandre,et, al(2011). Scikit-learn: machine learning in Python [J]. Journal of Machine Learning Research, 2011, 12: 2825-2830.

[16] Dalwinder Singh, Birmohan Singh, Investigating the impact of data normalization on classification performance, Applied Soft Computing,Volume 97, Part B, 2020,105524, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2019.105524.