Islamic University of Technology, Bangladesh

An Investigative Approach to Estimate the Critical Temperature of Superconductors Using Machine Learning

By

Rashed Hasan Ratul (180021116)
Ahnaf Islam Naf (180021323)
Fatin Abrar Shams (180021331)
Syed Shaek Hossain Samir (180021339)

A Dissertation

Submitted in consideration of Partial Fulfillment of the Requirement for the

Bachelor of Science in Electrical and Electronic Engineering
Academic Year: 2021-2022

Department of Electrical and Electronic Engineering (EEE)

Islamic University of Technology (IUT)

The Organization of Islamic Cooperation (OIC)

Gazipur-1704, Dhaka, Bangladesh

Undergraduate Thesis on

An Investigative Approach to Estimate the Critical Temperature of Superconductors Using Machine Learning

Supervised by

Dr. Md. Ashraful Hoque

Professor

Department of Electrical and Electronic Engineering

Islamic University of Technology (IUT)

Dhaka, Bangladesh

Co-Supervised by

Mr. Mirza Muntasir Nishat

Assistant Professor

Department of Electrical and Electronic Engineering

Islamic University of Technology (IUT)

Dhaka, Bangladesh

An Investigative Approach to Estimate the Critical Temperature of Superconductors Using Machine Learning

A thesis presented to
The Academic Faculty
by

Rashed Hasan Ratul (180021116)
Ahnaf Islam Naf (180021323)
Fatin Abrar Shams (180021331)
Syed Shaek Hossain Samir (180021339)

Approved by

Dr. Md. Ashraful Haque

Dr. Md. Ashraful Haque

Professor

Department of Electrical & Electronic Engineering

Declaration of Authorship

This is to certify that the work presented in this report is the outcome of research carried out by the candidates under the supervision of Dr. Md. Ashraful Hoque, Professor, Department of Electrical and Electronic Engineering (EEE), Islamic University of Technology (IUT). It is also declared that neither this report nor any part thereof has been submitted anywhere else for the reward of any degree or any judgement.

.....

Dr. Md. Ashraful Haque

Professor

Department of Electrical & Electronic Engineering

Fatin Abrar Shams (180021331)

Rashed Hasan Ratul (180021116)

Ahnaf Islam Naf (180021323)

Syed Shaek Hossain Samir (180021339)

Dedicated to

The exceptional individuals in our lives who have consistently offered boundless love and unwavering support, encouraging us on our path of research and enabling us to accomplish this significant milestone.

Acknowledgments

In the name of the Almighty Allah (SWT), the Most Gracious and Merciful, we humbly express our profound gratitude for the bestowed gift of reason and the opportunity to acquire worldly knowledge. It is through His divine guidance that we have been blessed with the resilience and perseverance necessary to successfully complete our project.

Our pursuit of a bachelor's degree would have been an insurmountable task without the unwavering support of numerous individuals. At this moment, we wish to extend our heartfelt appreciation to these exceptional individuals who have been instrumental in our journey. Their invaluable guidance, unwavering support, and priceless advice have been constant companions throughout this transformative experience.

First and foremost, we are deeply grateful to our esteemed academic and research mentor, Dr. Md. Ashraful Hoque. His unwavering support, motivation, and invaluable recommendations have been the driving force behind our project. Our accomplishments owe a great debt to his invaluable guidance and expertise, which have played a truly pivotal role in our success.

We would also like to express our heartfelt thanks to Mr. Mirza Muntasir Nishat for his invaluable technical assistance throughout our project. His guidance and expertise have been vital in achieving our goals. His words ignite a genuine enthusiasm for scientific exploration, urging us to venture into uncharted territories. We consider ourselves immensely fortunate to have such an extraordinary mentor leading our endeavors. We sincerely appreciate his unwavering support and his instrumental role in helping us overcome challenges.

Lastly, we extend our deepest appreciation to our beloved families, whose unwavering presence and encouragement have been our pillars of strength. Their attentive ears, kind words, and boundless support have filled our hearts with joy and motivation. We also express our sincere gratitude to our friends, who have been constant companions on this remarkable journey, offering support and camaraderie.

With profound gratitude, we acknowledge the immense contributions of these remarkable individuals and the blessings bestowed upon us by the Almighty. Their unwavering support and guidance have shaped our academic lives and contributed to our growth and success.

ACRONYMS

RMSE	Root Mean Square Error
MAPE	Mean Absolute Percentage Error
MAE	Mean Absolute Error
Tc	Critical Temperature
LSTM	Long Short-Term Memory
CNN	Convolutional Neural Network
HNN	Hybrid Neural Network
WOA	Whale Optimization Algorithm
HPO	Hyper-parameter Optimization
SVR	Support Vector Regression
MLP	Multi-Layer Perceptron
ANN	Artificial Neural Network

Table of Contents

List of tables

List of Figures

Abstract

Ever since the initial discovery of superconductivity, the fundamental concept and the complex relationship between critical temperature and superconductive materials have been subject to extensive investigation. However, identifying superconductors that exhibit such behavior at normal temperatures remains a significant challenge, and there are still significant gaps in our understanding of this unique phenomenon, particularly regarding the fundamental criteria used to estimate critical temperature. To address this knowledge gap, a plethora of machine learning techniques have been developed to model critical temperatures, given the inherent difficulty in predicting them using traditional methods. Additionally, the limitations of the standard empirical formula in determining the temperature range require the development of more advanced and viable methods. This thesis presents an advanced machine learning-based approach that utilizes the intricate properties of superconductive materials to accurately predict critical temperatures. The proposed model showcases impressive performance, as reflected by the Root Mean Squared Error (RMSE) of 9.68, R^2 score of 0.922, Mean Absolute Error (MAE) score of 5.383, and Mean Absolute Percentage Error (MAPE) score of 4.575, surpassing the performance of existing research works. The findings of this thesis shed new light on the effective implementation of a stacking ensemble method with hyper-parameter optimization, providing a promising avenue for accurate critical temperature estimation. The findings of this study have significant consequences for the decision-making involved in the synthesis of superconductors, as the viability of this complex and resource-intensive process significantly depends on the accuracy of the critical temperature estimation. This thesis contributes to the advancement of superconductivity research by proposing an approximation technique with remarkable precision for the key function of superconductors.

Keywords: *superconductor; critical temperature; machine learning; stacking ensemble method.*

Chapter 1

Introduction

Superconductivity has been a topic of great interest in condensed matter research for many years [1]. The phenomenon of zero electrical resistance that is exhibited by some metals below a critical temperature, Tc, has been observed in several metals such as indium, mercury, lead, tin, and niobium. The high-temperature superconductors, which possess the ability to conduct electricity without any resistance, have the potential for vast technological applications, including efficient energy distribution, transportation, and magnetic confinement of particles in nuclear fusion plants [2].

Despite over a century of research, determining how a material's composition and structure correlates with its superconducting properties remains a challenging task. The complexity of high-temperature superconductors arises from the fact that they cannot be fully characterized by the existing theories of superconductivity, which can only describe a small subset of actual superconductors [3]. This has led researchers to explore other methods, such as machine learning, to gain a deeper understanding of the relationships between superconductivity and a material's chemistry and structure.

Machine learning algorithms offer a unique opportunity to extract patterns and relationships from large datasets that traditional methods may miss. Additionally, for electron-phonon paired superconductors, for which the theory is rather well established, predicting the critical temperatures Tc of emerging superconductors is a notoriously tough task [4]. Previous attempts to develop a Tc formula compatible with strong coupling theory by McMillan, Allen, and Dynes resulted in closed-form approximations of relations between Tc and several metrics of the electron-phonon interaction that were identified in Eliashberg theory [5]. However, recent research has shown that the Tcs for more newly discovered superconducting materials, which have a higher two-dimensional electron-phonon interaction, do not match Allen and Dynes' formulation.

This paper aims to explore some of the proposed machine learning techniques that can be utilized to enhance the existing and traditional methods. Specifically, we aim to introduce a new descriptor that goes beyond the traditional empirical methods to characterize the properties

of superconductors in terms of critical temperature. The numerical characterization of the material is a necessary first step in such methods, after which various machine learning algorithms are employed to test and compare the predictive model. The use of machine learning has the potential to revolutionize the field of superconductivity, providing a more efficient approach to understanding the relationships between a material's chemistry and structure and its superconducting properties.

Accurately estimating the critical temperature (Tc) is a crucial aspect of the complex and resource-intensive process involved in synthesizing superconductors [6]. The viability of the synthesis process depends significantly on the accuracy of the Tc estimation. Therefore, the findings of this study have significant implications for decision-making in the synthesis of superconductors [7]. The proposed approximation technique offers a reliable way for researchers to estimate the Tc of newly discovered superconducting materials. By accurately estimating Tc, researchers can efficiently identify materials with desirable superconducting properties, expediting the development of new superconductors.

This thesis presents a significant contribution to the field of superconductivity research by proposing a highly precise approximation technique for the key function of superconductors. The proposed method utilizes machine learning algorithms to predict the Tc of superconducting materials based on their chemical and structural properties. By leveraging this approach, researchers can gain a more comprehensive understanding of the intricate relationships between a material's chemistry, structure, and superconducting properties. This, in turn, can help researchers identify materials with the potential to be high-temperature superconductors, facilitating the development of innovative superconductors with practical applications. The proposed technique has the potential to drive advancements in the field of superconductivity, enabling researchers to explore new and exciting avenues for future research.

This study proposes a novel approach to estimate the critical temperature of superconducting materials using a combination of the stacking ensemble method and the hyperparameter optimization algorithm. This strategy has not been explored before, making it a unique and appealing approach. The results obtained from the proposed stacking model indicate superior and stable performance compared to other machine learning models at four different stages, which is a significant contribution of this study.

Chapter 2

2.1 Literature review

In recent years, there has been a surge of interest in using machine learning algorithms to predict the critical temperature of superconductors. To predict Tc from diverse sets of input characteristics, several models such as random forest, support vector machines, and artificial neural networks have been used [8]. Some research has also concentrated on finding important factors that contribute the most to Tc prediction. Despite substantial advances in this field, precisely predicting Tc for complex superconductors with multiple elements and disorder in their crystal structure remains a difficulty. The objective of this literature review is to provide a comprehensive summary of the current advancements in utilizing machine learning techniques for the purpose of estimating the critical temperature (Tc) of superconductors. The review will examine the various machine learning models that have been developed for this objective and their respective merits and limitations.

Hamidieh created a model in [9], that employs a combination of linear regression, gradient boosting, and neural networks to make predictions. They also used feature engineering to extract relevant information from the chemical formulas and improve the accuracy of the predictions Their statistical model performs fairly well with an RMSE of 9.5K. Despite having better RMSE and R2 score than us, our approach enables us to identify the features that are more crucial for predicting the Tc, but their approach is unable to do so.

The study in [10] suggests a technique that describes materials using atomic vectors [38,39] and predicts Tc using a hybrid neural network model that combines a convolutional neural network (CNN) and a long short-term memory neural network (LSTM). The LSTM recovers the long-dependence feature interactions between atoms, while the HNN model employs CNNto extract the short-dependence feature relationships. This deep learning-based approachperforms pretty well with R2 score of 0.899 and MAE 5.023K; however, they only manage a poor RMSE of 83.565. Consequently, we achieve results that are better than this paper in termsof RMSE and R2.

A novel method was developed by Paulino et al. [11] by fusing the MARS approximation and the whale optimization algorithm (WOA). This may be an appealing methodology that had not previously been explored. In addition to that Ridge, Lasso and Elastic-net regression was used for comparison purpose. The results show that all four machine learning techniques are capable of predicting Tc with reasonable accuracy but this hybrid WOA/MARS based model outperforms the rest three model with an RMSE value of 15.14, R2 score of 0.80 and MAE of 10.75. However, compared to this, in the context of RMSE, R2, and MAE, our work performs.

Two popular regression methods, linear and simple linear regression models, were utilized in the research of Babu et al. [12] to compare various performance metrics. Better results are obtained by their linear regression model, which has an RMSE of 17.68, MAE of 13.42, and R2 of 0.7396. Once more, the RMSE, R2, and MAE results for our work are better.

In the study of Mohammad N. Haque et al. [13], they introduced a new model for multivariate regression that involves the iterative fitting of a continued fraction alongside additive spline models. To assess its efficacy, they compared it with different established techniques, including AdaBoost, Kernel Ridge, Linear Regression etc. They evaluated the performance of these methods in predicting the critical temperature of superconductors based on their physical-chemical properties, which is a crucial problem in the field. They obtained RMSE of 10.989, which our work managed to out-perform.

2.2 Key Contributions

One of the major contributions of this study is the stability and consistency of the performance parameters obtained from the proposed stacking ensemble method, even after applying a feature reduction technique. This finding is crucial because it shows that the proposed approach can effectively reduce the dimensionality of the input data without sacrificing the model's performance.

Additionally, it is worth noting that the combination of these four performance metrics, namely RMSE, MAE, MAPE, and R2 score, has not been considered together for validating the accuracy of critical temperature estimation in any of the previously accessible research to our knowledge. The stability and consistency of the performance parameters obtained from the proposed stacking ensemble method even after employing a feature reduction technique add to

the novelty of this study. These results suggest that the proposed approach has the potential to enhance the accuracy and reliability of critical temperature estimation, which can have significant implications for the development of new superconductors with practicalapplications.

Overall, the novelty of this study lies in the proposed strategy that combines the stacking ensemble method and the hyperparameter optimization algorithm to estimate the critical temperature of superconducting materials accurately. The stable and consistent performance parameters obtained from the proposed approach, even after employing feature reduction, and the superior performance metrics compared to other machine learning models make this study a significant contribution to the field of superconductivity research.

Chapter 3

3.1 Data Preprocessing

To compare the proposed stacking model with other regressor models, three feature selection methods (f_regression, f_classif, mutual info regression) were used. The top 50 features were chosen out of the original 81 attributes. This enhances performance by focusing on the most relevant attributes and reducing dimensionality. The selection of 50 features was made with the intention of striking a balance between retaining enough information for accurate modeling and reducing the dimensionality of the dataset. Choosing a smaller subset of features helps to eliminate noise and irrelevant information that could potentially hinder the model's performance. Table 1 shows the overall progression of dataset pre-processing [14].

Feature Selection Method	**Scaling Method**	**Feature Count**	**Status of hyper parameter**
None	Min-Max Scaler	81	Without optimization
None	Min-Max Scaler	81	Optimized
f_regression	Min-Max Scaler	50	Without optimization
f_regression	Min-Max Scaler	50	Optimized
Mutual_info_regression	Min-Max Scaler	50	Without optimization
Mutual_info_regression	Min-Max Scaler	50	Optimized
f_classif	Min-Max Scaler	50	Without optimization
f_classif	Min-Max Scaler	50	Optimized

Table 1: Dataset pre processing

3.2 Feature Selection Tables

The inclusion of feature selection tables in this research article enhances the discussion by providing valuable insights into the hyperparameter optimization process. The tables specifically highlight the results obtained from optimizing hyperparameters on the 50 features selected through f_regression, mutual_info regression, and f_classif.

These tables effectively demonstrate the influence of feature selection on hyperparameter tuning, as different subsets of features may require varying hyperparameter configurations to achieve optimal performance. By selecting the most relevant features, the models can focus on the attributes that contribute most significantly to accurate predictions, leading to improved accuracy and efficiency. It underscores the importance of feature selection in refining the feature space and provides valuable guidance for optimizing algorithms on the selected subsets to achieve optimal model performance.

The most significant outcome from the three feature selection tables is that different feature selection methods (f_regression, mutual_info regression, and f_classif) lead to different optimal hyperparameters for the regression algorithms. This suggests that the choice of feature selection technique impacts the configuration of hyperparameters required for optimal model performance. Therefore, selecting the most appropriate feature selection method is crucial for achieving accurate and efficient regression models.

Chapter 4

4.1 Workflow

In this study, the effectiveness of three feature selection techniques, namely f_regression, f_classif, and mutual info regression, was evaluated both individually and in comparison to the baseline. This research takes an innovative approach as these specific feature selection methods have not been previously utilized in the preprocessing of this dataset, opening up possibilities for future research.

The study is divided into four stages. The first stage involves obtaining baseline results without applying any feature selection techniques. This step provides a reference point for evaluating the performance of the subsequent feature selection methods. Before proceeding with the feature selection stages, the data is preprocessed. Min-max scaling is employed as the chosen data preprocessing technique. This method scales the data to a fixed range, typically between 0 and 1. By applying min-max scaling, the data is normalized and brought within a standardized range, facilitating further analysis.

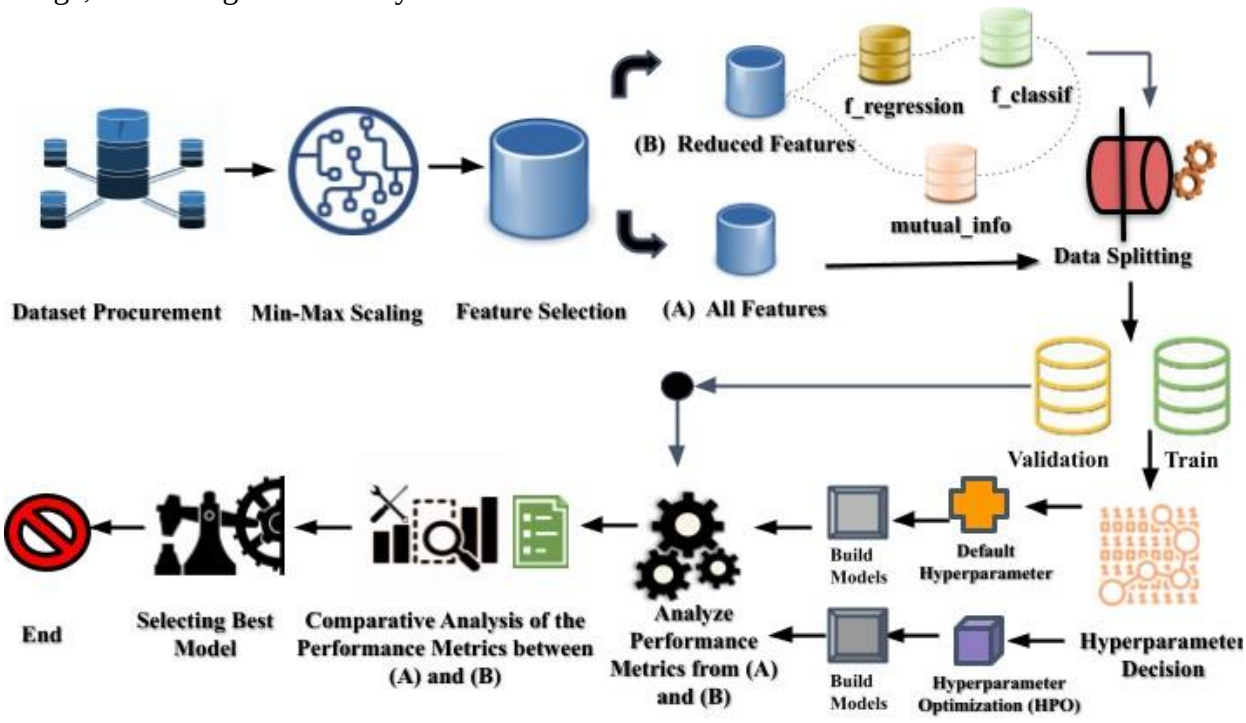

Figure 1: Overall workflow diagram.

In the next three stages, each of the feature selection methods is applied separately. The f_regression and f_classif methods are used to calculate the correlation between each predictor and the target variable. These methods capture linear interactions between predictors and the target. On the other hand, mutual_info_regression is capable of capturing various types of relationships, including linear, quadratic, and exponential. Compared to f_regression and f_classif, mutual_info_regression is generally considered more reliable and adaptable, particularly when the relationships between predictors and the target are unclear.

Feature selection is a process of reducing the number of input variables to include only those that are most beneficial to the model [15]. It can enhance model efficiency and reduce computational costs. By selecting the most relevant features, the model can focus on the attributes that have the greatest impact on accurate predictions. Overall, the workflow of this research article involves pre-processing the data using min-max scaling, followed by evaluating the performance of three feature selection techniques individually and in comparison to the baseline. This approach showcases the potential of these feature selection methods and their impact on the predictive accuracy of the model.

4.2 Baseline

In order to compare model performance with fewer features, a baseline was established using all 81 features. The study employed five standalone machine learning models, two stacked regression models, and one voting regression model. Initially, default hyperparameters were used for training the models. Subsequently, hyperparameter optimization was conducted using the Random-Search CV algorithm to find the best combinations [16]. The resulting optimal hyperparameter configurations for each model can be found in Table 2.

To evaluate the models, four performance metrics (RMSE, R2 score, MAE, and MAPE) were employed. The table presents the results for all models using the complete set of 81 features both before and after hyperparameter optimization. The objective was to assess the impact of hyperparameter optimization on model performance when utilizing reduced feature sets. The aim was to enhance predictive accuracy and overall performance by fine-tuning the hyperparameters.

The table provides a clear comparison of the results, highlighting the effectiveness of hyperparameter optimization in improving the models' performance. It demonstrates the benefits of refining the hyperparameters and selecting optimal combinations for different algorithms, leading to enhanced predictive capabilities.

Algorithms	**Hyperparameters passed**		**Best Hyperparameters**
Ridge Regressor	alpha	0.1, 1, 10, 0.001, 100	0.001
	tol	0.001, 0.0001, 0.01, 0.1, 0.00001	0.00001
	solver	auto, svd, cholesky, lsqr, sparse_cg, sag, saga	sparse_cg
Lasso Regressor	alpha	0.1, 1, 10, 0.001, 100	0.001
	tol	0.001, 0.0001, 0.01, 0.1, 0.00001, 0.000001, 0.0000001	0.0000001
KNN	n_neighbors	2, 5, 10, 25, 50	2
	leaf_size	10, 20, 30, 60, 90, 105, 120, 150	20
	algorithm	auto, ball_tree, kd_tree, brute	brute
	p	1, 2, 3, 5, 10, 20, 40, 80, 100, 200	2
SVR	epsilon	0.01, 0.1, 1, 10, 100	1
	C	0.5, 1, 5, 10, 100, 0.05	100
	cache_size	0.2, 2, 20, 200, 2000	2000
	coef0	0.01, 0.1, 0, 1, 10	10
	degree	1, 2, 3, 4, 5,	1
MLP	activation	logistic, relu	relu
	learning_rate_init	0.01, 0.1, 0.001	0.1
	hidden_layer_sizes	(55, 52, 78, 30), (56, 32, 25), (57, 40, 52, 75, 60)	(56, 32, 25)
RFR	n_estimators	20,40,60,80,100,120	120
	max_depth	5,10,15,20	20
	min_samples_split	2,4,8,10	2

Table 2: Hyperparameter optimization on all the 81 features.

4.3 F_Classif

This initial feature selection method employed in this study is a univariate technique. It utilizes univariate statistical tests to select the most relevant features, making it a preprocessing step for estimators [17]. From this method, the top 50 features were chosen. The same set of five standalone models from the baseline section were utilized, along with stacking and voting models. Hyperparameter optimization was performed using RandomSearch CV, resulting in the best hyperparameter combinations presented in Table 3. The tables display the results for RMSE, MAPE, MAE, and R2 score, comparing the performance with both the default and optimized hyperparameters.

Algorithm	**Hyperparameters passed**		**Best Hyperparameters**
Ridge Regressor	alpha	0.1, 1, 10, 0.001, 100	0.001
	tol	0.001, 0.0001, 0.01, 0.1, 0.00001	0.0001
	solver	'auto', 'svd', 'cholesky', 'lsqr', 'sparse_cg', 'sag', 'saga'	svd
Lasso Regressor	alpha	0.1, 1, 10, 0.001, 100	0.001
	tol	0.001, 0.0001, 0.01, 0.1, 0.00001, 0.000001, 0.0000001	0.0000001
KNN	n_neighbors	2, 5, 10, 25, 50	5
	leaf_size	10, 20, 30, 60, 90, 105, 120, 150	150
	algorithm	auto, ball_tree, kd_tree, brute	kd_tree
	p	1, 2, 3, 5, 10, 20, 40, 80, 100, 200	1
SVR	epsilon	0.01, 0.1, 1, 10, 100	0.1
	C	0.5, 1, 5, 10, 100, 0.05	100
	cache_size	0.2, 2, 20, 200, 2000	20
	coef0	0.01, 0.1, 0, 1, 10	0.1
	degree	1, 2, 3, 4, 5,	3
MLP	activation	logistic, relu	relu
	learning_rate_init	0.01, 0.1, 0.001	0.001

	hidden_layer_sizes	(55, 52, 78, 30), (56, 32, 25), (57, 40, 52, 75, 60)	(57, 40, 52, 75, 60)
RFR	n_estimators	20, 40, 60, 80, 100, 120	100
	min_samples_split	2, 4, 8, 10	2
	max_depth	5, 10, 15, 20	20

Table 3: Hyperparameter optimization on 50 features selected by f_classif.

4.4 F_Regression

The F_regression method is a recommended feature selection criterion for identifying potentially predictive features, regardless of their association's sign with the target variable. This method provides p-values as a measure of feature significance. In this section, the top 50 features were selected using the F_regression method. Standalone models including Ridge, Lasso, KNN, SVR, and MLP were utilized. The hyperparameters of these algorithms were optimized using RandomSearch CV. The optimized hyperparameters were also applied to build the stacking and voting models [18].

To gain a general understanding of each algorithm's performance, all standalone models were initially trained using their default hyperparameters. Similarly, the stacking and voting ensemble methods were initially constructed with default parameters. Table 4 provides a list of the parameters used in this case. These results offer insights into the impact of hyperparameter optimization on model performance.

Algorithm	Hyperparameters passed		Best Hyperparameters
Ridge Regressor	alpha	0.1, 1, 10, 0.001, 100	0.001
	tol	0.001, 0.0001, 0.01, 0.1, 0.00001	0.1
	solver	auto, svd, cholesky, lsqr, sparse_cg, sag, saga	svd
Lasso Regressor	alpha	0.1, 1, 10, 0.001, 100	0.001
	tol	0.001, 0.0001, 0.01, 0.1, 0.00001, 0.000001, 0.0000001	0.000001
KNN	n_neighbors	2, 5, 10, 25, 50	2
	leaf_size	10, 20, 30, 60, 90, 105, 120, 150	20
	algorithm	auto, ball_tree, kd_tree, brute	ball_tree
	p	1, 2, 3, 5, 10, 20, 40, 80, 100, 200	1
SVR	epsilon	0.01, 0.1, 1, 10, 100	1
	C	0.5, 1, 5, 10, 100, 0.05	100
	cache_size	0.2, 2, 20, 200, 2000	20
	coef0	0.01, 0.1, 0, 1, 10	0.01
	degree	1, 2, 3, 4, 5,	1
MLP	activation	logistic, relu	relu
	learning_rate_init	0.01, 0.1, 0.001	0.001
	hidden_layer_sizes	(55, 52, 78, 30), (56, 32, 25), (57, 40, 52, 75, 60)	(55, 52, 78, 30)
RFR	n_estimators	20, 40, 60, 80, 100, 120	120
	min_samples_split	2, 4, 8, 10	4
	max_depth	5, 10, 15, 20	20

Table 4: Hyperparameter optimization on 50 features selected by f_regression.

4.5 Mutual Info Regression

Mutual information is a nonnegative measure of the interdependence between two random variables. Mutual information quantifies variable dependence, with zero indicating independence and higher values indicating stronger dependence. This section utilized mutual information regression. This method utilizes nonparametric algorithms based on k-nearest neighbor distances to estimate entropy [19]. Optimal hyperparameters were determined through the RandomSearch CV method after initially using default hyperparameters. Table 5 summarizes the best hyperparameters for each model.

Algorithm	**Hyperparameters passed**		**Best Hyperparameters**
Ridge Regressor	alpha	0.1, 1, 10, 0.001, 100	0.001
	tol	0.001, 0.0001, 0.01, 0.1, 0.00001	0.0001
	solver	'auto', 'svd', 'cholesky', 'lsqr', 'sparse_cg', 'sag', 'saga'	auto
Lasso Regressor	alpha	0.1, 1, 10, 0.001, 100	0.001
	tol	0.0010, 0.0001, 0.01, 0.1, 0.00001, 0.000001, 0.0000001	0.01
KNN	n_neighbors	2, 5, 10, 25, 50	2
	leaf_size	10, 20, 30, 60, 90, 105, 120, 150	105
	algorithm	'auto', 'ball_tree', 'kd_tree', 'brute'	brute
	p	1, 2, 3, 5, 10, 20, 40, 80, 100, 200	20
SVR	epsilon	0.01, 0.1, 1, 10, 100	1
	C	0.5, 1, 5, 10, 100, 0.05	100
	cache_size	0.2, 2, 20, 200, 2000	200
	coef0	0.01, 0.1, 0, 1, 10	0.01
	degree	1, 2, 3, 4, 5,	2
MLP	activation	logistic, relu	relu

	learning_rate_init	0.01, 0.1, 0.001	0.01
	hidden_layer_sizes	(55,52,78,30), (56,32,25), (57,40,52,75,60)	(56, 32, 25)
RFR	n_estimators	20, 40, 60, 80, 100, 120	100
	min_samples_split	2, 4, 8, 10	4
	max_depth	5, 10, 15, 20	20

Table 5: Hyperparameter optimization on 50 features selected by mutual_info.

4.6 Regressor Model Description

4.6.1 Lasso regression

Less absolute shrinkage and selection operator, abbreviated as LASSO, is a technique for conducting regression investigation that is utilized in the disciplines of machine learning and statistics. This technique combines the processes of variable regularization and selection in order to develop the likelihood and comprehension of the final statistical version [20]. Initial development of Lasso focused on linear regression. Many features of the estimator are shown off in this relatively straightforward example. As an illustration, it is related to ridge regression, lasso coefficient approximations, optimal subset selection, and so-called soft thresholding. Additionally, it shows that if the covariates are collinear, estimates of the coefficients are not required to be distinct. This is different from traditional linear regression. It generates less complex models with fewer predictors that can be more readily understood. LASSO regression, which employs regularization, is centered on simple models with fewer factors and parameters. We can better analyze the models using the shrinkage technique. Finding variables that are closely related to variables that correspond to the aim is another benefit of the reducing process. By penalizing the overall size of the regression coefficients, LASSO regression can be used to improve precision and decrease variance for linear regression models [21].

$$L_{lasso}(\beta) = \sum_{i=1}^{n} (y_i - x_i\beta)^2 + \lambda \sum_{j=1} |\beta_j| \quad (1)$$

4.6.2 KNN regression

An intuitive method for estimating the association between self-determining varia-bles and the uninterrupted result by averaging data from the identical neighborhood is K-Nearest Neighbor (KNN) Regression [22]. Although there are many advantages to this method, it quicklybecomes impractical when numerous independent elements are pre-sent. Simply take the meanof the numbers you're aiming for among your K nearest neighbors for a forthright and simple KNN regression implementation. An alternate approach weights the mean of inversed distanceof K closest. KNN regression uses distance functions in a manner similar to how KNN classification does. It is advised to first look at the data before determining what value of K to employ. Most of the time, a higher K value reduces background noise and increases accuracy, but at the cost of obscuring the clear differences between individual features. Cross-validationis an additional technique you can use to determine a suitable K value after the fact by validatingyour K value against a different data set [23]. K typically needs to be more than 10 to work at its best on the majority of datasets. The findings are significantly better than 1-NN. The formulato calculate Euclidean distance is [24]:

$$d(a,b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \quad (2)$$

The formula to calculate Manhattan distance is [25]:

$$d(a,b) = \left(\sum_{i=1}^{n} |a_i - b_i|^c\right)^{\frac{1}{c}} \quad (3)$$

The prediction *E(y)* will be for *x* will be:

$$E(b) = \frac{1}{k}\sum_{i=1}^{k} b_i \quad (4)$$

4.6.3 Support Vector Regression

SVR was built using the same technology as the widely-used Support Vector Machine (SVM). It is a very popular machine learning strategy for categorizing data that does not readily lend itself to linear categorization. The SVR method identifies the line (or hyperplane in higher dimensions) that best matches the data after allowing us to select a tolerance for model error

[26]. The SVR aims to fit the best line within a set tolerance as opposed to closing the gap

between real and predicted, in contrast to other regression models (the distance between the hyperplane and boundary line).

The optimal classification decision function can be defined as:

$$f(x) = w.x + b \tag{5}$$

4.6.4 Multi-Layer Perceptron

MLP (multilayer perceptron) refers to a category of artificial neural network that uses convolutional layer to learn new information (ANN). The term multilayer perceptron (MLP) can mean either a network composed of multiple layers of perceptron or any ANN with a feedback control mechanism. An MLP includes a minimum of three distinct layers of nodes: output, hidden, and input [27]. All nodes—aside from the input nodes—are neurons with nonlinear activation functions. As a supervised learning technique, backpropagation is used in MLP's training process. MLP is distinct from a linear perceptron since it employs non-linear activation and has many more layers. It can tell data apart and make distinctions between them. With the help of the tanh (Hyperbolic Tangent) function, a given element's value can be transformed to lie in the range -1 to 1.

$$\tanh(a) = \frac{1-e^{-2a}}{1+e^{-2a}} \tag{6}$$

4.6.5 Random forest

By utilizing many decision trees in conjunction with the "bagging" and "bootstrapping" processes, Random Forest is an ensemble method capable of doing both regression and classification. The main concept is to aggregate many decision trees to regulate the final output, rather than relying on individual trees [28]. The learning models of Random Forest are built on top of many decision trees. To assess the efficacy for each model, we generate sample datasets by selecting rows and characteristics at random. Bootstrap refers to this particular part of the document. The decision function is:

$$H(a) = \arg\max_{Y} \sum_{i=1}^{k} I(h_i(a) = Y) \tag{7}$$

4.6.6 Stacking Method

Stacking is a method that can be used to ensemble a number of different classification or regression models [29]. Ensemble models can be created in a variety of ways; however, bagging and boosting are the most common approaches. The variance can be reduced using the bagging technique by averaging the results of numerous similar models with a high volatility. Boosting is the process of building numerous incremental models in order to reduce bias while maintaining a low variance. When used to a problem involving classification or regression,

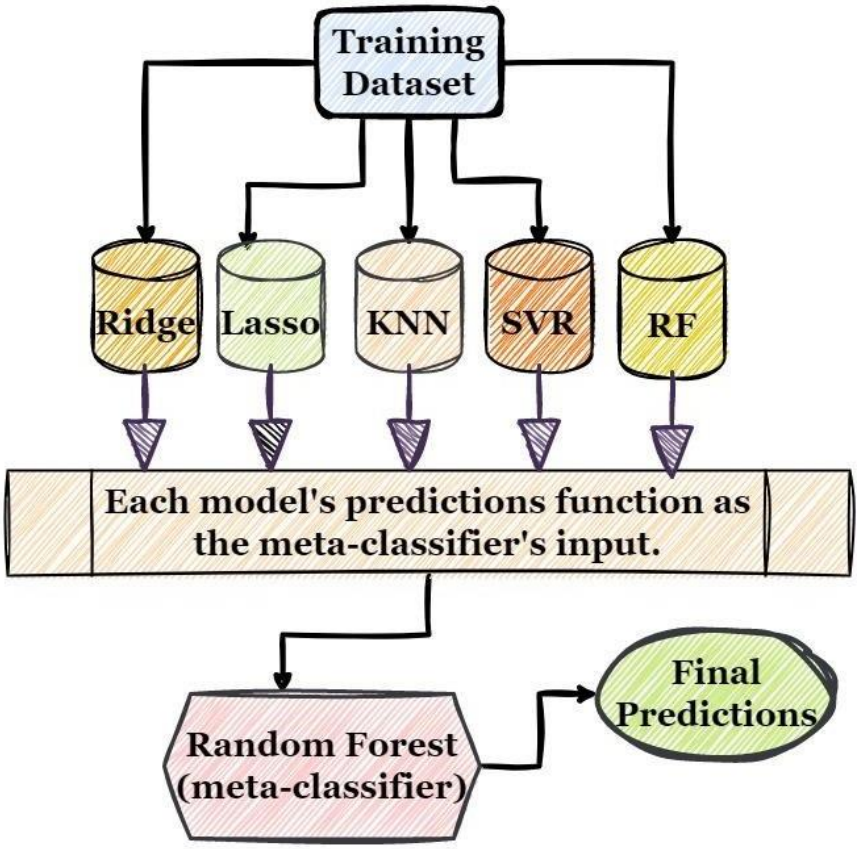

Figure 2: Stacking model proposed structure

stacking has the advantage of combining the most successful features of multiple different efficient models. This, in turn, produces predictions that are superior to those produced by any one individual model in the ensemble. A random division into J sections of the same size is performed using this method on the dataset. One set is utilized for the testing phase of the j-fold cross-validation, while the remaining sets are put to use in the training phase.

Because of these training testing pair subsets, it is able to obtain the predictions of several learning models, which are subsequently utilized as the meta-data in order to construct the meta-model. The ultimate forecast is determined by the meta-model, which is also referred to as the

winner-takes-all technique [3]. In our approach, we also optimized the hyper-parameters of the final estimator while evaluating the performance of the proposed algorithm. In our suggested model, we used five algorithms as estimators, with the Random Forest regressor with the default hyper-parameters, serving as the final estimator. Five algorithms were used as the base estimators, which were Random Forest, KNN, SVR, Ridge and Lasso. All these base estimators were once used with the default hyper-parameters and once with the hyper-parameters found while optimising each of those standalone models with RandomSearch CV.

4.6.7 Voting

In Voting Classifiers, there are several models of the various machine learning algorithms that are present. These models are fed the entire dataset, and after being trained on the data, each algorithm will make a prediction [31]. After all of the models have made their predictions for the sample data, the method that has been employed the most often will be utilized to obtain

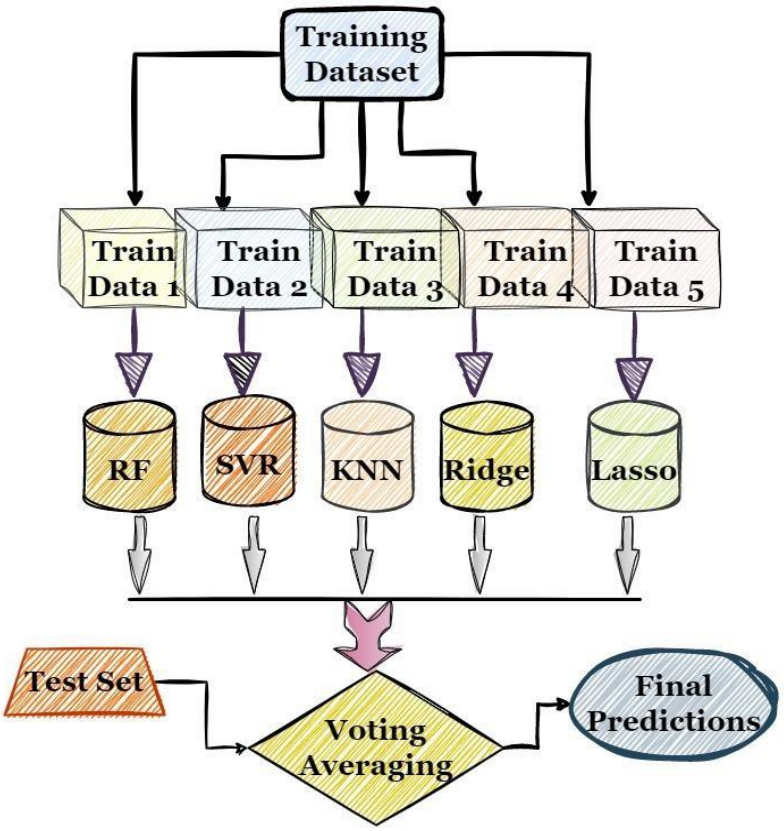

Figure 3: Voting model proposed structure.

the final forecast from the model. In this section, the category that is predicted the most

accurately by the various algorithms will be utilized as the model's final prediction. The voting ensemble method used in this work consists of five standalone models. These models are the Random Forest, KNN, SVR, Ridge and Lasso. All of the models were first used with the default hyper-parameters. In order to optimize the voting ensemble model further, the best hyper-parameters which were found using RandomSearch CV were used for each of the standalone models while using them as the estimators.

4.7 Hyper-parameter optimization

In machine learning, the task of determining the ideal collection of hyperparameters for a learning algorithm is referred to as hyper-parameter optimization, or tuning. A parameter whose value is utilized to guide the learning process is termed as hyperparameter. For this process, we used a random search CV to find out the best hyper-parameter. RandomizedSearchCV applies a "score" and a "fit" method. Cross-validated search across parameter settings is used to optimize the estimator's parameters, which are then used to implement these methods.

4.8 Evaluation Metrics

4.8.1 RMSE

From the RMSE we can find the standard deviation of the error in predictions. Residuals or these prediction errors are the measure of how far from the regression line data points are. The RMSE measures how spread these residual values are.

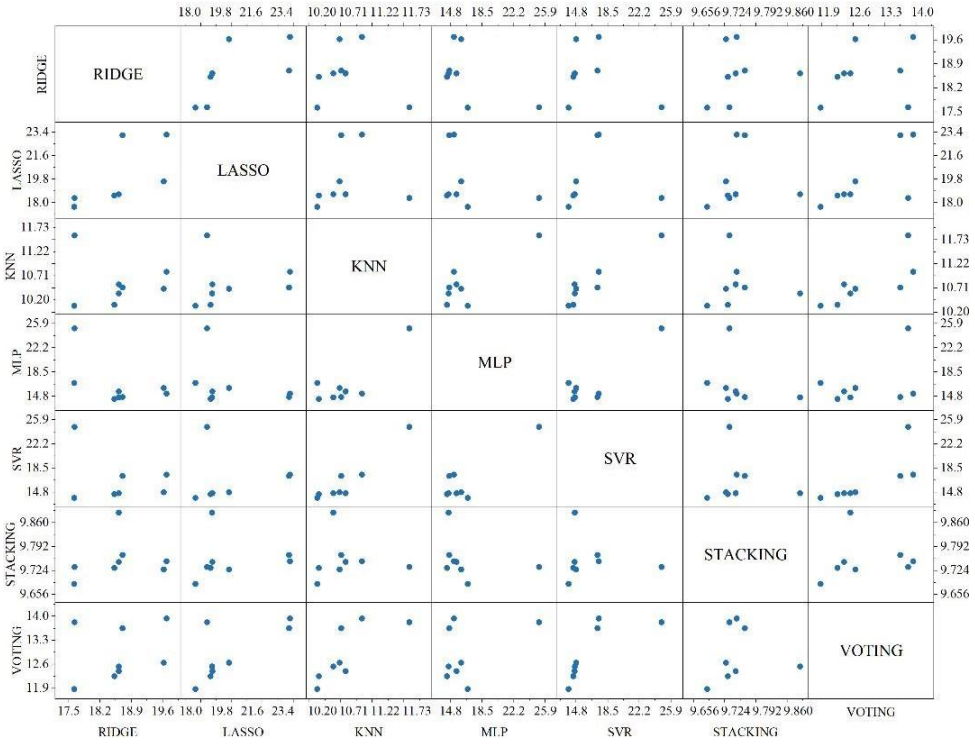

Figure 4: Scatter matrix plot for all the ML models.

The underlying assumption when presenting the RMSE is that the errors are unbiased and follow a normal distribution. The RMSE can be defined by the following equation [32]:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_i^2} \quad (8)$$

where n is the number of samples and e is the error term.

4.8.2 MAE

Mean Absolute Error (MAE) is a measure of the average magnitude of the errors in a set of predictions, without taking into account their direction. It is the average absolute difference between the predicted and actual values and is used to evaluate the performance of a regression model. MAE can be represented by the following equation [33]:

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |e_i| \quad (9)$$

Where e is the error term and n is the number of samples. MAE presents itself to be the most natural measure of average error magnitude, and that (unlike RMSE) it is an unambiguous measure of average error magnitude.

4.8.3 MAPE

Mean absolute percentage error is the most common error analysis technique used for forecasting. It measures accuracy as percentage. MAPE can be represented by the following mathematical equation [34]:

$$M = \frac{1}{n}\sum_{t=1}^{n} \left| \frac{A_{t-F_t}}{A_t} \right| \quad (10)$$

Where *n* is the number of fitted points, At is the actual value, Ft is the forecast value. MAPE is generally used when the the quantity to be predicted remains much higher than zero.

4.8.4 R2 score

A measure of how well a linear regression model fits the data is called the R-squared. This statistic expresses, as a percentage, the percentage of the variation in the dependent variable that can be attributed, as a whole, to the effects of the independent variables. On a scale that ranges from 0 to 100 percent, the coefficient of determination, or R-squared, provides an important measurement of the strength of the relationship between the model and the dependent variable [35]. The R2 score can be defined by the following equation

$$R^2 = \frac{\text{Variance Explained By Model}}{\text{Total Variance}} \quad (11)$$

Chapter 5

Results

In terms of RMSE, R2, MAE, and MAPE scores, the proposed model performed noticeably better than the other regressor models. Furthermore, there were fewer random variations in the results under multiple conditions. The suggested model therefore predicts a significantly more stable and consistent result, whereas all the other relevant ML models showed fluctuations at various situations for RMSE, as seen in Fig 6. The suggested stacking model achieves the best result of RMSE 9.686 after hyperparameter optimization with all 81 characteristics taken into account, as shown in Table 7, which displays the average RMSE values under all eight distinct conditions.

The proposed stacking model has also shown significant stability and achievement in terms of R2 scores, with an average R2 Score that was close to 0.91 across eight different situations. The appropriate comparison of R2 score of the model with all the additional ML methods is shown in Table 8 and Table 9 where both cases for optimized hyper-parameters and default hyper-parameters are considered.

The stacking model once again has the best MAE score, coming in at 5.383, with scores hovering around 5.4 throughout all eight situations. This is the lowest MAE achieved among all of the algorithms tested. The average value of MAE for all the algorithms for each of the feature selection technique used, with HPO and without HPO are shown in Table 10 and 11.

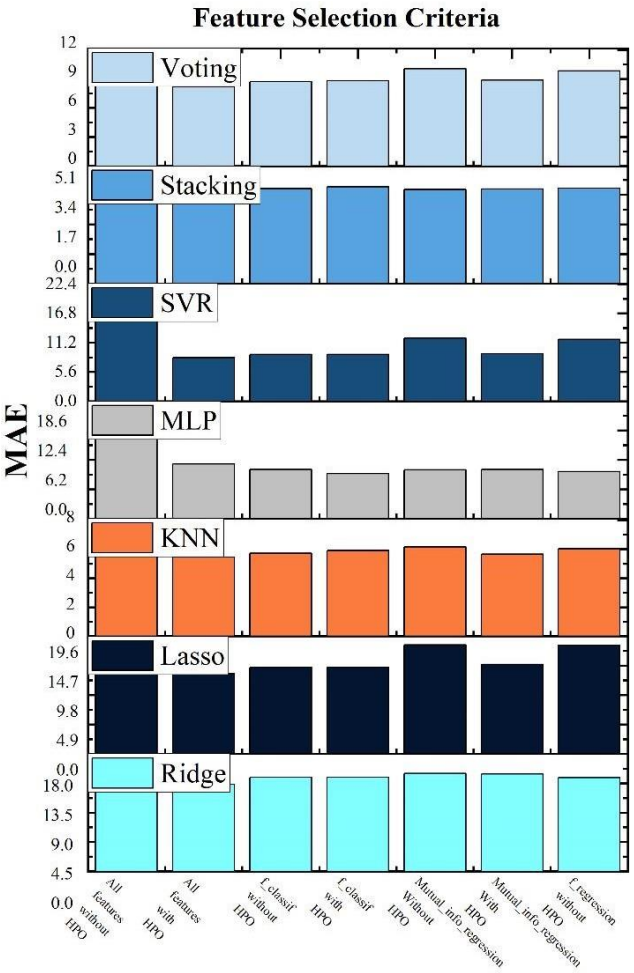

Figure 5: MAE for different ML models in terms of feature selection methods

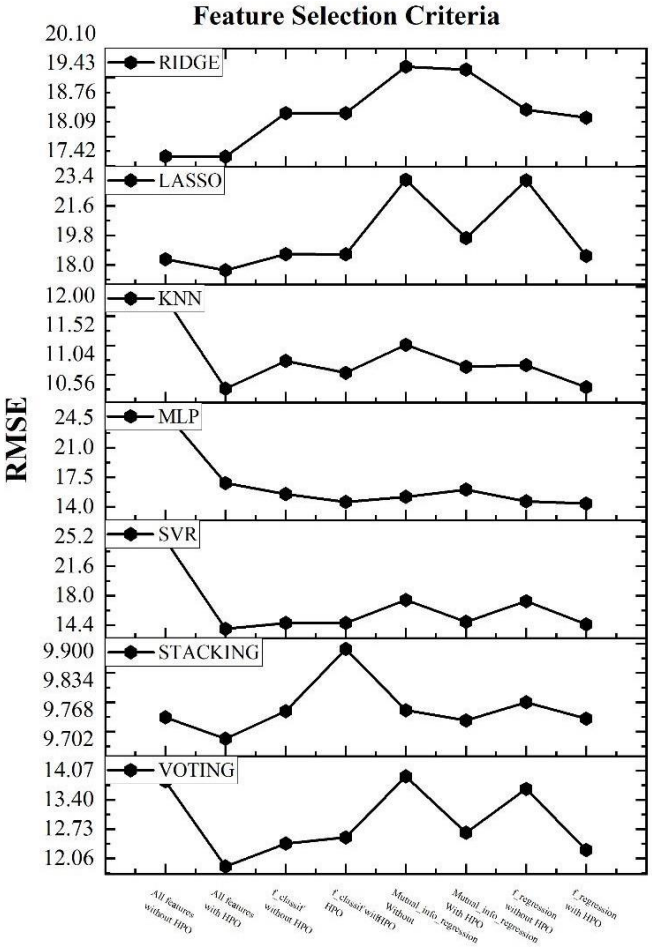

Figure 6: MAE for different ML models in terms of feature selection methods

Similar to the previous three performance criteria, the proposed stacking model once again achieves the best MAPE score, which is 4.575 as shown in Table 12. It is clear that for MAPE the stacking model performs the best when the default hyper-parameters for all the algorithms in the stacking model is used. In Fig. 7 the values of MAPE is denoted by the width of each of the individual sections. It is clear from the figure that for all of the feature selection techniques used here, the width of the stacking model for MAPE is the lowest compared to other models.

Criteria	Ridge	Lasso	KNN	MLP	SVR	Stacking	Voting
All features	17.636	18.345	11.821	25.158	24.812	9.734	13.826
f_classif	18.624	18.655	10.788	15.513	14.649	9.748	12.400
Mutual_info_regression	19.689	23.204	11.052	15.193	17.477	9.750	13.938
f_regression	18.705	23.159	10.718	14.657	17.315	9.768	13.652

Table 6: Average RMSE under all conditions without hyperparameter optimization.

Criteria	Ridge	Lasso	KNN	MLP	SVR	Stacking	Voting
All features	17.627	17.677	10.332	16.826	13.938	**9.686**	11.875
f_classif	18.624	18.651	10.591	14.595	14.661	9.888	12.540
Mutual_info_regression	19.622	19.639	10.691	16.05	14.803	9.727	12.648
f_regression	18.521	18.548	10.356	14.388	14.487	9.731	12.252

Table 7: Average RMSE under all conditions with hyperparameter optimization.

Criteria	Ridge	Lasso	KNN	MLP	SVR	Stacking	Voting
All features	0.7348	0.7131	0.8808	0.4041	0.4751	0.9191	0.8370
f_classif	0.7043	0.7033	0.9006	0.7947	0.8170	0.9189	0.8689
Mutual_info_ regression	0.6695	0.5410	0.8958	0.8032	0.7396	0.9189	0.8344
f_regression	0.7017	0.5428	0.9020	0.8167	0.7444	0.9186	0.8411

Table 8: Average R2 under all conditions without hyperparameter optimization.

Criteria	Ridge	Lasso	KNN	MLP	SVR	Stacking	Voting
All features	0.7351	0.7336	0.9089	0.7571	0.8343	0.919958	0.8797
f_classif	0.7043	0.7034	0.9043	0.8166	0.81674	0.916636	0.8659
Mutual_info_ regression	0.671792	0.6712	0.9025	0.7774	0.81316	0.919301	0.8636
f_regression	0.707593	0.7067	0.9085	0.2169	0.8210	0.91922	0.8720

Table 9: Average R2 under all conditions with hyperparameter optimization.

Criteria	Ridge	Lasso	KNN	MLP	SVR	Stacking	Voting
All features	13.3535	13.981	6.6822	20.406	18.524	5.475659	10.130
f_classif	14.3864	14.408	5.7242	10.375	8.9589	5.454676	8.7256
Mutual_info_ regression	14.9859	18.142	6.1600	10.349	12.065	5.405764	10.035
f_regression	14.377	18.110	6.0186	9.9280	11.841	5.465831	9.8239

Table 10: Average MAE under all conditions without hyperparameter optimization.

Criteria	Ridge	Lasso	KNN	MLP	SVR	Stacking	Voting
All features	13.34874	13.3659	5.44815	11.49106	8.31017	5.383053	8.1803
f_classif	14.38648	14.4049	5.89576	9.47307	8.92848	5.557045	8.8195
Mutual_info_ regression	14.90684	14.9273	5.64775	10.38217	9.04408	5.438871	8.8572
f_regression	14.22273	14.2468	5.46085	24.37746	8.85529	5.440037	8.5893

Table 11: Average MAE under all conditions with hyperparameter optimization.

Criteria	Ridge	Lasso	KNN	MLP	SVR	Stacking	Voting
All features	12.86335	16.6023	5.464846	26.62253	15.40685	4.815816	10.3292
f_classif	16.81786	16.3090	5.488384	9.007473	7.637216	4.575237	9.732087
Mutual_info_ regression	16.17585	19.2875	5.071434	7.340627	10.02875	4.984084	9.649217
f_regression	15.81139	19.5317	5.687521	7.916581	9.335178	4.747403	9.386212

Table 12: Average MAPE under all conditions without hyperparameter optimization.

Criteria	Ridge	Lasso	KNN	MLP	SVR	Stacking	Voting
All features	12.95154	13.003	5.05690	11.038	7.4839	4.80839	8.3418
f_classif	16.81786	16.351	5.5626	9.4737	7.2858	4.7112	9.7490
Mutual_info_ regression	17.03013	16.545	4.5454	9.5143	8.0557	5.250034	9.7261
f_regression	17.49597	16.945	4.3979	38.199	7.7147	4.818958	9.8588

Table 13: Average MAPE under all conditions with hyperparameter optimization.

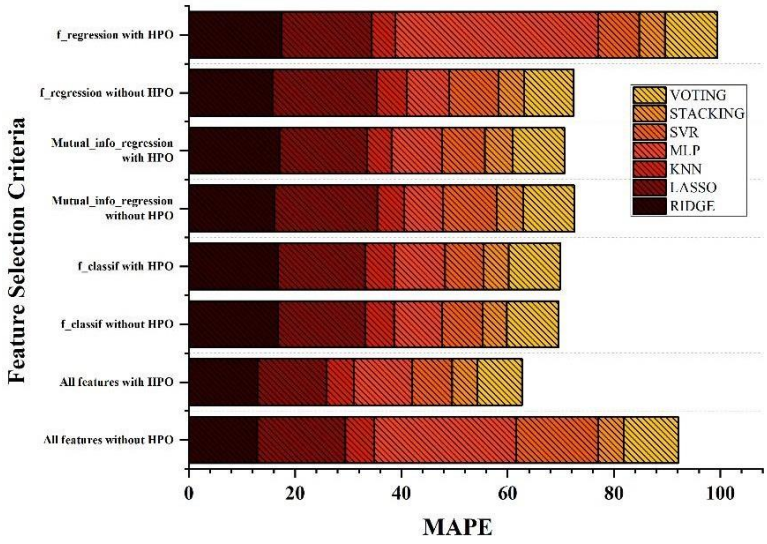

Figure 7: Feature select criteria vs. MAPE

Conclusion

This thesis introduces a novel approach based on machine learning for accurately predicting critical temperatures in superconducting materials. The study addresses the challenges associated with conventional techniques and identifies the need for innovative techniques for precisely estimating critical temperatures. Utilizing a stacking ensemble method with hyperparameter optimization, the proposed method outperforms previous research in terms of performance.

Evaluation of the model's efficacy using a variety of metrics, such as RMSE, R2 score, MAE, and MAPE, yields insightful information. The results indicate that the incorporation of hyperparameter optimization improves the estimation of critical temperature's precision and dependability. The average RMSE, R2 score, MAE, and MAPE values obtained from models with hyperparameter optimization consistently outperform those without hyperparameter optimization, demonstrating the significance of optimizing model parameters.

The study also investigates the effect of feature reduction on model performance. Even after employing feature reduction techniques, the proposed method maintains stability and consistency in performance metrics, according to the findings. This result is significant because it demonstrates that the model can reduce the dimensionality of input data without compromising precision.

Notably, this study introduces the combined use of four comprehensive performance metrics, RMSE, R2 score, MAE, and MAPE, which were not previously considered together. This comprehensive evaluation strategy enhances the originality and significance of this work.

Overall, the proposed method contributes to the advancement of superconductivity research by providing a method for estimating critical temperatures with high precision. The results demonstrate the capability of the stacking ensemble method with hyperparameter optimization to improve the accuracy and dependability of critical temperature estimation. As the accuracy of critical temperature estimation has a direct impact on the viability and practical applications of new superconducting materials, these developments have significant implications for decision-making in the synthesis of superconductors.

In conclusion, this thesis provides a promising method for estimating the critical temperature of superconductors precisely. Even after feature reduction, the stability, consistency, and superior performance metrics attained by the proposed method make it a significant contribution to the field of superconductivity research. Future research can build upon these findings and further refine the proposed method to open up new avenues for the practical development of high-temperature superconductors.

Reference

[1] Fradkin, Eduardo. *Field theories of condensed matter physics*. Cambridge University Press, 2013.

[2] Zhang, Hongye, Zezhao Wen, Francesco Grilli, Konstantinos Gyftakis, and Markus Mueller. "Alternating current loss of superconductors applied to superconducting electrical machines." *Energies* 14, no. 8 (2021): 2234.

[3] Flores-Livas, José A., Lilia Boeri, Antonio Sanna, Gianni Profeta, Ryotaro Arita, and Mikhail Eremets. "A perspective on conventional high-temperature superconductors at high pressure: Methods and materials." *Physics Reports* 856 (2020): 1-78.

[4] Xie, Stephan R., Gregory R. Stewart, James J. Hamlin, Peter J. Hirschfeld, and Richard G. Hennig. "Functional form of the superconducting critical temperature from machine learning." *Physical Review B* 100, no. 17 (2019): 174513.

[5] Méndez-Moreno, R. M. "A Schematic Two Overlapping-Band Model for Superconducting Sulfur Hydrides: The Isotope Mass Exponent." *Advances in Condensed Matter Physics* 2019 (2019): 1-7.

[6] Lilia, Boeri, Richard Hennig, Peter Hirschfeld, Gianni Profeta, Antonio Sanna, Eva Zurek, Warren E. Pickett et al. "The 2021 room-temperature superconductivity roadmap." *Journal of Physics: Condensed Matter* 34, no. 18 (2022): 183002.

[7] Yazdani-Asrami, Mohammad, Wenjuan Song, Antonio Morandi, Giovanni De Carne, Joao Murta-Pina, Anabela Pronto, Roberto Oliveira et al. "Roadmap on artificial intelligence and big data techniques for superconductivity." *Superconductor Science and Technology* 36, no. 4 (2023): 043501.

[8] Stanev, Valentin, Corey Oses, A. Gilad Kusne, Efrain Rodriguez, Johnpierre Paglione, Stefano Curtarolo, and Ichiro Takeuchi. "Machine learning modeling of superconducting critical temperature." *npj Computational Materials* 4, no. 1 (2018): 29.

[9] Hamidieh, Kam. "A data-driven statistical model for predicting the critical temperature of a superconductor." Computational Materials Science 154 (2018): 346-354.

[10] Li, Shaobo, Yabo Dan, Xiang Li, Tiantian Hu, Rongzhi Dong, Zhuo Cao, and Jianjun Hu. "Critical temperature prediction of superconductors based on atomic vectors and deep learning." Symmetry 12, no. 2 (2020): 262.

[11] García-Nieto, Paulino José, Esperanza García-Gonzalo, and José Pablo Paredes-Sánchez. "Prediction of the critical temperature of a superconductor by using the WOA/MARS, Ridge, Lasso and Elastic-net machine learning techniques." *Neural Computing and Applications* 33 (2021): 17131-17145.

[12] Babu, R. Venkatesh, G. Ayyappan, and A. Kumaravel. "Comparison of Linear Regression and Simple Linear Regression for critical temperature of semicon-ductor."

[13] Moscato, Pablo, Mohammad Nazmul Haque, Kevin Huang, Julia Sloan, and Jon C. de Oliveira. "Learning to extrapolate using continued fractions: Pre-dicting the critical temperature of superconductor materials." arXiv preprint arXiv:2012.03774 (2020).

[14] Zebari, Rizgar, Adnan Abdulazeez, Diyar Zeebaree, Dilovan Zebari, and Jwan Saeed. "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction." Journal of Applied Science and Technology Trends 1, no. 2 (2020): 56-70.

[15] Chandrashekar, Girish, and Ferat Sahin. "A survey on feature selection methods." Computers & Electrical Engineering 40, no. 1 (2014): 16-28.

[16] Elgeldawi, Enas, Awny Sayed, Ahmed R. Galal, and Alaa M. Zaki. "Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis." In Informatics, vol. 8, no. 4, p. 79. Multidisciplinary Digital Publishing Institute, 2021.

[17] Yan, Chaokun, Jun Zhang, Xi Kang, Zhengze Gong, Jianlin Wang, and Ge Zhang. "Comparison and Evaluation of the Combinations of Feature Selection and Classifier on Microarray Data." In 2021 IEEE 6th International Conference on Big Data Analytics (ICBDA), pp. 133-137. IEEE, 2021.

[18] Olteanu, D. A., and M. J. Schleich. "F: Regression models over factorized views." Proceedings of the VLDB Endowment 9, no. 13 (2016).

[19] Van Dijck, Gert, and Marc M. Van Hulle. "Speeding up the wrapper feature subset selection in regression by mutual information relevance and redundancy analysis." In Artificial Neural Networks–ICANN 2006: 16th International Conference, Athens, Greece, September 10-14, 2006. Proceedings, Part I 16, pp. 31-40. Springer Berlin Heidelberg, 2006.

[20] Chen, Jie, Kees de Hoogh, John Gulliver, Barbara Hoffmann, Ole Hertel, Matthias Ketzel, Mariska Bauwelinck et al. "A comparison of linear regression, regularization, and machine learning algorithms to develop Europe-wide spatial models of fine particles and nitrogen dioxide." Environment international 130 (2019): 104934.

[21] Spencer, Bruce, Omar Alfandi, and Feras Al-Obeidat. "A refinement of lasso regression applied to temperature forecasting." Procedia computer science 130 (2018): 728-735.

[22] Wang, Yang, Yu Xiao, Jianhui Lai, and Yanyan Chen. "An adaptive k nearest neighbour method for imputation of missing traffic data based on two similarity metrics." Archives of Transport 54 (2020).

[23] Rodriguez, Juan D., Aritz Perez, and Jose A. Lozano. "Sensitivity analysis of k-fold cross validation in prediction error estimation." IEEE transactions on pattern analysis and machine intelligence 32, no. 3 (2009): 569-575.

[24] Wang, Liwei, Yan Zhang, and Jufu Feng. "On the Euclidean distance of images." IEEE transactions on pattern analysis and machine intelligence 27, no. 8 (2005): 1334-1339.

[25] Faisal, M., and E. M. Zamzami. "Comparative analysis of inter-centroid K-Means performance using euclidean distance, canberra distance and manhattan distance." In Journal of Physics: Conference Series, vol. 1566, no. 1, p. 012112. IOP Publishing, 2020.

[26] Bárcenas, Roberto, Ruth Fuentes-García, and Lizbeth Naranjo. "Mixed kernel SVR addressing Parkinson's progression from voice features." Plos one 17, no. 10 (2022): e0275721.

[27] Delashmit, Walter H., and Michael T. Manry. "Recent developments in multilayer perceptron neural networks." In Proceedings of the seventh Annual Memphis Area Engineering and Science Conference, MAESC. 2005.

[28] Dou, Jie, Ali P. Yunus, Dieu Tien Bui, Abdelaziz Merghadi, Mehebub Sahana, Zhongfan Zhu, Chi-Wen Chen, Khabat Khosravi, Yong Yang, and Binh Thai Pham. "Assessment of advanced random forest and decision tree algorithms for modeling rainfall-induced landslide susceptibility in the Izu-Oshima Volcanic Island, Japan." Science of the total environment 662 (2019): 332-346.

[29] Shu, Chang, and Donald H. Burn. "Artificial neural network ensembles and their application in pooled flood frequency analysis." Water Resources Research 40, no. 9 (2004).

[30] Xia, Rui, Yunpeng Gao, Yanqing Zhu, G. U. Dexi, and Cong Wu. "A Fast and Efficient Method Combined Data-Driven for Detecting Electricity Theft to Secure the Smart Grid with Stacking Structure." Available at SSRN 4019865.

[31] Sagi, Omer, and Lior Rokach. "Ensemble learning: A survey." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8, no. 4 (2018): e1249.

[32] Fortin, Vincent, Mabrouk Abaza, Francois Anctil, and Raphael Turcotte. "Why should ensemble spread match the RMSE of the ensemble mean?." Journal of Hydrometeorology 15, no. 4 (2014): 1708-1713.

[33] Choudhury, BhaskarJ. "Evaluation of an empirical equation for annual evaporation using field observations and results from a biophysical model." Journal of Hydrology 216, no. 1-2 (1999): 99-110.

[34] Tayman, Jeff, and David A. Swanson. "On the validity of MAPE as a measure of population forecast accuracy." *Population Research and Policy Review* 18 (1999): 299-322.

[35] Nugroho, Adi, Sri Hartati, and Khabib Mustofa. "Vector Autoregression (Var) Model for Rainfall Forecast and Isohyet Mapping in Semarang–Central Java–Indonesia." International Journal of Advanced Computer Science and Applications 5, no. 11 (2014).

Lupón, Josep, Hanna K. Gaggin, Marta De Antonio, Mar Domingo, Amparo Galán, Elisabet Zamora, Joan Vila et al. "Biomarker-assist score for reverse remodeling prediction in heart failure: The ST2-R2 score." International journal of cardiology 184 (2015): 337-343.