Exploration and Learning Algorithms Used for Predicting Casting Properties

Sandra Gajoch[1], Łukasz Marcjan[1], Mateusz Witkowski[1], Adam Bitka[1,2], Krzysztof Jaśkowiec[1] , Marcin Małysza[1,2], Dorota Wilk-Kołodziejczyk[1,2]

[1] AGH, University of Krakow, Kraków, Poland
[2] Łukasiewicz Research Network-Krakow Institute of Technology, Kraków, Poland

sgajoch@agh.edu.pl

Abstract.
This article presents an analysis of the application of machine learning algorithms for predicting the mechanical properties of austempered ductile iron (ADI) castings. As part of the study, predictive models were developed and optimized to forecast strength parameters based on chemical composition, thickness, and heat treatment process parameters. A detailed analysis of the impact of hyperparameters on algorithm effectiveness was conducted, along with a comparison of different parameter space exploration methods.
The study evaluated the performance of various machine learning algorithms, identifying Gradient Boosting as the most effective for predicting mechanical properties. An additional outcome of this research is the development of a web application integrating the predictive models, allowing users to analyze the expected properties of castings based on input data.
This solution has potential applications in the foundry industry, enabling better control over production processes and reducing costs associated with experimental selection of technological parameters. The results confirm that applying machine learning algorithms can significantly improve the prediction of ADI iron's mechanical properties, paving the way for further automation and optimization of metallurgical production processes.

Keywords: Artificial Intelligence, cast iron, Grid search, Bayesian Search

1 Introduction

The use of exploration and learning algorithms helps solve complex practical problems. The casting production process consists of multiple stages, and an optimal solution for one stage may not be optimal for the entire process. Process planning involves many factors, such as the diversity of ordered components, variations in metal processing, and machining. The use of artificial intelligence and machine learning algorithms enables the optimization of the entire process in terms of cost reduction, better time management, and resource allocation.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_21

Process optimization is a key challenge in the era of Industry 4.0, where technology, automation, and data analysis play a crucial role in increasing efficiency and reducing company costs. The application of exploration and learning algorithms allows for minimizing human errors, generating automatic parameters and production schedules, detecting and eliminating casting defects, and optimizing resource and material management. Process optimization also contributes to reducing environmental impact by lowering energy and material consumption and minimizing the production of unnecessary components that fail to meet minimum requirements. Companies that embrace technological advancements become more competitive in the market and can better respond to customer demands.

The continuous development of artificial intelligence tools enables their application in materials engineering, discovering new solutions and techniques that contribute to better resource utilization. Analyses of machine learning algorithms, including neural networks and reinforcement learning, facilitate automatic knowledge acquisition and process improvement based on available data and observations. By applying data analysis and exploration techniques, individual process stages can be examined comprehensively, leading to optimal solutions.

The remainder of this article is organized as follows. Section 2 provides a review of the relevant literature related to machine learning applications in the foundry industry. Section 3 discusses the dataset, data processing methods, correlation analysis, and the machine learning algorithms applied in this study. Section 4 presents a series of experiments, including baseline model performance, hyperparameter optimization using Grid Search and Bayesian Search, and the impact of data augmentation. The deployment of the developed models as a web application is also described. Finally, Section 5 summarizes the findings and outlines future research directions.

2 Literature review

Process planning in metal processing is a key element of the metal industry's operations. Proper preparation of processes enables the maximization of production efficiency, reduction of downtime, minimization of energy losses, and lowering of manufacturing costs. The steel production process is characterized by high temperatures, strict interdependence of various stages, significant material and energy consumption, the use of massive equipment, and complex operational control. Artificial intelligence methods have been adapted to optimize solutions in steel production [1, 2]. The authors of [1] focused on batch production planning, while in [2], the researchers addressed issues related to steel production planning and scheduling, emphasizing that solving these problems separately often leads to suboptimal results. Additionally, the specificity of conditions in individual production plants poses further challenges. Artificial intelligence methods have also been applied to resource consumption prediction based on planned castings using a modifier in an automotive company [3], as well as in a laboratory setting based on weekly sample analysis plans [4]. Seeking to balance material properties and process efficiency, the authors of [5] utilized a Co-Evolutionary Algorithm (CEA) to optimize the chemical composition design of amorphous alloys and

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_21

vacuum casting process parameters. Data analysis of die-cast aluminum [6] using the Support Vector Regression (SVR) algorithm for predicting material properties—including yield strength (YS), ultimate tensile strength (UTS), and elongation (EL)—demonstrated the relevance of such research due to the model's high predictive accuracy. In [7], an adaptive neuro-fuzzy inference system (ANFIS) was applied to predict the mechanical properties of A357 aluminum castings, produced using the low-pressure permanent mold casting method. The ANFIS model was employed to analyze the impact of chemical composition (elements such as Si, Mg, Fe, Ti, and Cu) on UTS, YS, and elongation (ε). The ANFIS model achieved high accuracy (>85%) in identifying key variables affecting casting quality and in predicting material properties. The prediction of mechanical properties was also addressed by a research team studying automotive cast parts [8]. Algorithms such as Random Forest (RF), K-Nearest Neighbor (KNN), and Extreme Gradient Boosting (XGBOOST) were used to predict UTS and YS. The studies highlight the potential of exploration and learning algorithms in improving casting process design, reducing the number of destructive tests, and increasing production efficiency. Therefore, integrated systems for process design and material preparation represent a crucial step in the advancement of materials engineering.

3 Materials and methods

3.1 Dataset

The input data consists of 513 records containing information on ADI cast iron, though some records lack complete data. Each record includes details on the chemical composition of the cast iron, heat treatment process parameters, product thickness in millimeters, and mechanical properties. The parameters related to chemical composition, heat treatment, and product thickness are fully recorded. However, the mechanical properties dataset is incomplete - (Fig. 1) illustrates the distribution of non-empty parameters.

The data set is uneven and has significant gaps with the following available values: Ultimate tensile strength (UTS) – 386 recorded values, Yield strength (YS) – 279 recorded values, Elongation (EL) – 360 recorded values, Brinell hardness (HB) – 309 recorded values, Impact toughness (measured on unnotched samples) – 314 recorded values. The dataset is a combination of multiple databases, analyzing alloys with varied chemical compositions. Due to the diverse data sources, the input parameters often exhibit significant variability. For example, the extreme values of magnesium or phosphorus content in the chemical composition are twenty times higher than their average values, whereas nickel and copper content percentages do not show significant deviations from the mean. Regarding heat treatment parameters, which include austenitization and austempering, austempering time and temperature do not have significantly outlying values, whereas austenitization time and temperature do. The average product thickness is approximately 30 mm, but outliers reaching up to 100 mm are present. Most physical parameter values do not deviate significantly from their average values, except for impact toughness and Brinell hardness, which exhibit larger deviations.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_21

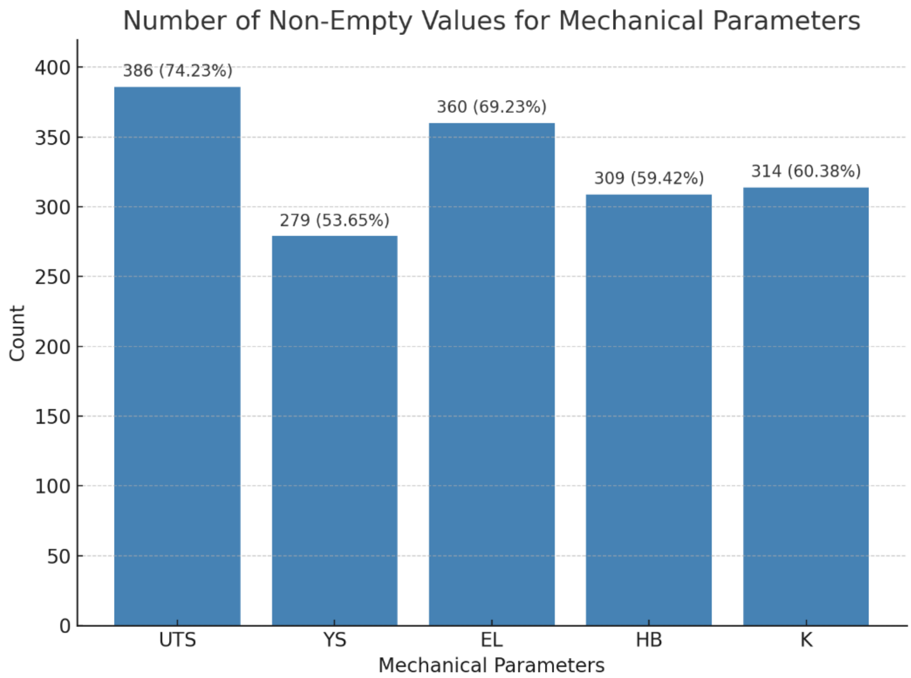

Fig. 1. Proportional Distribution of Physical Parameter Data

3.2 Data Correlation

The correlation matrix (Fig. 2) illustrates the relationships between various process variables and mechanical properties of the material. Correlations are expressed on a scale from -1 (strongly negative correlation) to 1 (strongly positive correlation), with a value of 0 indicating no correlation.

The content of individual chemical elements can significantly affect the mechanical properties of the alloy. Carbon content shows a negligible correlation with tensile strength and elongation, while higher carbon levels slightly decrease yield strength and increase hardness and impact toughness. Silicon influences yield strength similarly, but elongation increases with higher silicon content, while hardness decreases; other parameters do not show significant relationships. Manganese content generally has no effect on impact toughness and lowers other strength parameters. A notable decrease in tensile strength occurs with an increase in magnesium percentage, which also negatively impacts other properties. Higher copper and nickel content leads to increased elongation and slightly reduces impact toughness and tensile strength. Molybdenum decreases tensile strength and, to a lesser extent, yield strength. Phosphorus has a positive correlation with tensile strength, elongation, and impact toughness, without significantly affecting yield strength or hardness. The correlations of other elements (S, Ti, Cr, Sn, Al) are minor, indicating their minimal influence on mechanical properties. In summary, magnesium (Mg) and molybdenum (Mo) have the most significant

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_21

negative impact on tensile strength and yield strength. Silicon (Si), copper (Cu), nickel (Ni), and phosphorus (P) positively affect ductility (A5). Carbon (C) has the strongest effect on increasing hardness. Impact toughness increases with higher carbon (C) and phosphorus (P) content but decreases with higher copper (Cu) content.

Heat treatment also affects material properties. Austenitization temperature and time show varied correlations with mechanical properties. Temperature has a positive correlation with tensile strength, yield strength, and hardness, but a negative correlation with elongation and impact toughness. Time has a positive correlation with elongation but minimal effect on other mechanical parameters. The austempering process has a greater influence on mechanical parameters. Temperature shows a strong negative correlation with yield strength (-0.65), tensile strength (-0.5), and hardness (-0.79), indicating a significant reduction in strength with increasing temperature. Time has weak correlations with mechanical properties, suggesting that its impact is less significant compared to other parameters.

Fig. 2. Correlation Matrix

The mechanical properties of cast iron generally exhibit strong correlations with one another. Tensile strength and yield strength have a very strong positive correlation (0.89), indicating that an increase in one property is typically accompanied by an increase in the other. Elongation is strongly negatively correlated with tensile strength

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_21

(-0.42) and yield strength (-0.57), suggesting that materials with higher strength generally exhibit lower elongation. Brinell hardness shows a significant positive correlation with tensile strength (0.42) and yield strength (0.53), while it negatively correlates with elongation (-0.54). Among all mechanical properties, impact toughness is the least correlated with the others, with the lowest correlation observed with tensile strength (-0.37).

3.3 Data Completion and Augmentation

Due to the relatively high correlation coefficients between various mechanical parameters and the presence of missing values in the dataset, it is possible to impute missing values based on the correlated parameters available in the dataset. However, this approach carries the risk of introducing erroneous relationships between attributes. Creating a predictive model based on data that has been partially generated by other predictive models may lead to a situation where it is unclear whether the model is truly solving the intended problem.

For this reason, the study opted to use data augmentation techniques to expand the dataset. Data augmentation involves artificially increasing the training dataset by generating new, modified examples from existing data. The goal is to improve model performance by providing a larger number of diverse samples, which can help prevent overfitting and enhance the model's generalization ability.

3.4 Machine Learning Algorithms Used

Machine learning is widely used to solve prediction problems. This paper analyses the applicability of various machine learning algorithms for predicting the mechanical properties of ADI cast iron. To compare the performance of various artificial intelligence models, the following algorithms were evaluated:

- Linear Regression [9, 10]: A relatively simple and interpretable machine learning algorithm used in tasks involving the prediction of numerical values based on input data. It allows the modelling of linear relationships between independent and dependent variables. It assumes the existence of a linear relationship between independent variables and dependent variables.

- **Decision Trees [11–13]:** An easy to interpret and visualize machine learning algorithm for solving classification and regression problems. A hierarchical graphical representation in which each node represents the results of a test on a variable and each branch reflects the result of that test. The algorithm selects one of the independent variables that has the greatest influence on the partitioning of the data.

- Random Forest [14–16]: Also known as Bagged Decision Trees, it is based on creating a model consisting of multiple decision trees to achieve better performance and generalization. It is used for both classification and regression problems. It is based on creating multiple decision trees trained on random data with repetition. Each tree receives a different data set and a random subset of independent variables, increasing diversity and reducing the risk of overfitting.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_21

- Boosting Algorithms [17–20]: A technique used in machine learning to reduce prediction errors by combining weaker models into one. It focuses on reducing systematic errors and improving efficiency. There are many implementations of boosting, including:

- Adaptive boosting (AdaBoost): One of the earliest boosting methods. Initially, it assigns the same weight to each set, which are then adjusted. It assigns greater weight to misclassified observations in order to improve them in the next iteration.

  - Gradient boosting (GB) - A sequential algorithm that optimizes loss functions by sequentially generating base models so that the current one is more effective than the previous one.
  - Extreme gradient boosting (XGBoost) - An improved version of GB in terms of speed and scalability, using multithreading and distributed computing.

- Support Vector Regression (SVR) [21, 22]: A machine learning technique that uses the SVM (Support Vector Machines) algorithm to solve regression problems. It is used to predict numerical values by modeling the relationship between independent and dependent variables. In contrast to the SVM, which seeks to maximize the margin between data points, this approach aims to minimize regression error. The utilization of kernel functions enables the transformation of input data into higher-dimensional spaces.

- K-Nearest Neighbors (KNN) [23, 24]: A machine learning algorithm is used in both classification and regression. It works on the assumption that objects with similar characteristics are adjacent to each other in feature space. The distances between the sample and all samples in the dataset are calculated. The K value determines how many nearest neighbors will be considered when predicting classes or values for a new sample.

3.5 Selected Model Evaluation Metrics

During the study on the performance of machine learning models, only the R^2 and RMSE metrics were chosen, as they are the most intuitive to interpret. Additionally, cross-validation was applied during model training to identify potential overfitting and obtain more realistic results.

R^2 Score (Coefficient of Determination): This regression metric indicates how well the model fits the data. It represents the proportion of data variability explained by the model. The R^2 score ranges from 0 to 1, where 1 means the model perfectly fits the data (all points lie on the regression line), while 0 indicates that the model does not explain any data variability and its predictions are equivalent to a simple average.

RMSE (Root Mean Squared Error): This metric measures the average size of the error predicted by the model compared to actual values. RMSE is expressed in the same unit as the data, making it easy to interpret. A lower RMSE value indicates better model performance, as it signifies smaller prediction errors.

Cross-validation is a technique used to assess the performance of machine learning models by splitting the dataset into multiple subsets. The model is trained on a portion

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_21

of the data and tested on the remaining part. This process is repeated for different data splits, and the final results are averaged, providing a more stable and reliable model evaluation.

3.6 Hyperparameter Search Algorithms

One of the ways to improve model performance is hyperparameter optimization, which is the process of adjusting their values to achieve the best configuration for a specific task. Hyperparameters are model parameters that are not learned during training but must be specified before the learning process begins. Hyperparameter optimization is a crucial step in creating an effective machine learning model, as it can significantly impact its performance. The goal of this process is to find settings that minimize the model error or improve other performance metrics. A decision was made to apply the Grid Search and Bayesian Search strategies during the research. This decision was based on the fact that both approaches allow for systematic searching of the hyperparameter space, which is essential to finding the best model settings. [25, 26]

4 Application

Due to the number of algorithms being analyzed, the number of mechanical parameters to be determined, and the methodology of the solutions, the study was divided into several research scenarios. All the experiments in this work were conducted on a machine with an Intel Core i7-4790K 4 x 4.00GHz processor and 16 GB of physical RAM.

4.1 Study of Models with Default Parameters, Without Data Augmentation and Hyperparameter Tuning

The goal of this experiment was to assess the model performance using default parameters in order to establish a baseline for comparisons in later analyses. Table 1 summarizes the algorithms that best adapted to the training and test sets, as well as the algorithms that adapted the least for the specific mechanical parameters.

Table 1 Experiment Results for Models with Default Parameters

Result	Tensile Strength	Yield Strength	Elongation	Brinell Hardness	Impact Toughness
Best Fit to the Training Set	Decision Tree Regressor	Decision Tree Regressor oraz Extra Tree Regressor	Decision Tree Regressor oraz Extra Tree Regressor	Decision Tree Regressor oraz Extra Tree Regressor	Decision Tree Regressor oraz Extra Tree Regressor
Best Fit to the Training Set with 5-Fold Cross-Validation	Extra Trees Regressor	Extra Trees Regressor	Extra Trees Regressor	XGBRegressor	Extra Trees Regressor
Best Fit to the Test Set	Extra Trees Regressor	XGBRegressor	XGBRegressor	Gradient Boosting Regressor	XGBRegressor
Worst Fitting Model	SVR	SVR	SVR	SVR	SVR

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_21

During the analysis of the basic experiment results, it was observed that decision tree-based algorithms tend to overfit. This is particularly evident in the case of the DecisionTreeRegressor and ExtraTreeRegressor models, which in most cases achieved a 100% fit to the training set. However, when verifying the results on the test set, these same models performed significantly worse. The worst-performing model was the SVR, which is most likely due to the default value of the "C" parameter set to 1. Such a low value causes the model to be highly tolerant of prediction errors and attempts to find a solution close to linear regression, which does not favor accurate predictions for non-linear data.

4.2 Study of Models Using Hyperparameter Tuning - Grid Search

The goal of this experiment was to examine the impact of specific hyperparameter values on the performance of predictive models. The shortest average training time for a given model was achieved by the DecisionTreeRegressor, ExtraTreeRegressor, and KNeighborsRegressor algorithms. On the other hand, the most time-consuming models were the RandomForestRegressor, AdaBoostRegressor, and ExtraTreesRegressor algorithms. The GradientBoostingRegressor algorithm achieved top results for each of the mechanical parameters, except for impact toughness. For impact toughness, the most efficient model was the ExtraTreesRegressor. Among the top models were also XGBRegressor, XGBRFRegressor, and RandomForestRegressor—all of which achieved results around 0.8 R^2. The other models performed significantly worse. Table 2 presents a summary of the algorithms with hyperparameter tuning that achieved the highest average and maximum R^2 scores, as well as the worst average and maximum values.

Table 2 Results of the studied machine learning models for mechanical parameters with hyperparameter tuning

Result	Tensile Strength	Yield Strength	Elongation	Brinell Hardness	Impact Toughness
Highest Average R^2 Score	Gradient Boosting Regressor	Gradient Boosting Regressor	Gradient Boosting Regressor	AdaBoost Regressor	Gradient Boosting Regressor
Best Maximum R^2 Score	Gradient Boosting Regressor	Gradient Boosting Regressor	Gradient Boosting Regressor	Gradient Boosting Regressor	Extra Trees Regressor
Lowest Average R^2 Score	Linear Regression	Linear Regression	SVR	Linear Regression	Linear Regression
Worst Maximum R^2 Score	Linear Regression	SVR	Linear Regression	Linear Regression	Linear Regression

The analyses indicate that the most effective and stable model is usually the GradientBoostingRegressor, which in most cases showed the highest average accuracy and the best maximum results. On the other hand, the worst results were obtained by the linear regression algorithm, whose performance did not improve in any case and, in most tests, did not exceed $R^2 = 0.6$.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_21

4.3 Study of Models with Data Augmentation

The goal of this experiment was to examine the impact of various data augmentation techniques on the overall performance of individual models. This experiment used settings similar to those in section 4.2. The following changes were made to the dataset:

- The Fe parameter was introduced — the iron content calculated based on the total content of other elements subtracted from 100%.

- Dimensionality reduction was performed — data regarding the content of elements such as "S", "P", "V", "Cr", "Ti", "Sn", and "Al" were excluded from the analysis.

- The ausferritization time was represented as the logarithm of the time in seconds (this approach was inspired by the article [27]).

Table 3 Results of the studied machine learning models for mechanical parameters with hyperparameter tuning and data augmentation

Result	Tensile Strength	Yield Strength	Elongation	Brinell Hardness	Impact Toughness
Highest Average R^2 Score	Gradient Boosting Regressor	AdaBoost Regressor	Gradient Boosting Regressor	AdaBoost Regressor	Gradient Boosting Regressor
Best Maximum R^2 Score	Gradient Boosting Regressor	XGBRegressor	Gradient Boosting Regressor	ExtraTrees Regressor	Gradient Boosting Regressor
Lowest Average R^2 Score	Linear Regression	Linear Regression	SVR	Linear Regression	Linear Regression
Worst Maximum R^2 Score	Linear Regression	Linear Regression	Linear Regression	KNeighbors Regressor	Linear Regression

The GradientBoostingRegressor algorithm achieved top results for three of the mechanical parameters. Among the best models were also AdaBoostRegressor, XGBRegressor, and ExtraTreesRegressor. The remaining models performed significantly worse. Table 2 presents a summary of the algorithms with hyperparameter tuning that achieved the highest average and maximum R^2 scores, as well as the worst average and maximum values.

The conducted data augmentation did not yield the expected results, and in many cases, it had a negative impact on the performance of more advanced models. In a few instances, the maximum R^2 coefficient improved for simpler algorithms, such as SVR, where for the K parameter, the value increased to 0.64 compared to the previous study, where it was only 0.53. Such a slight improvement in the model's response to changes in the data could be due to the fact that the models were overfitting due to the insufficient amount of data.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_21

4.4 Study of Bayesian Optimization - Bayesian Search

The goal of this experiment was to find the most optimal hyperparameters for the GradientBoostingRegressor model, given the best overall results from Table 2 and Table 3. The optimization is initiated by the function _train_models, in which the objective function is specified and used by the gp_minimize method from the skopt library [28], implementing optimization based on a Bayesian approach. Table 4 presents the top three optimization results for each mechanical parameter using Bayesian Search. For tensile strength, the model obtained through Bayesian Search is 3 percentage points better than the best result using Grid Search. For yield strength, the model's performance improved by 1 percentage point. For the elongation parameter, the optimization was ineffective, as the results decreased by 3 percentage points compared to the best model found using Grid Search. For the HB parameter, Bayesian optimization generated the best predictive model observed during the conducted research. For the K parameter, a decrease of about 1 percentage point was also noted.

Table 4 Top Three Bayesian Optimization Results for Mechanical Parameters

Parameter	n_estimators	Max depth	Min samples split	Min samples leaf	Max features	Learning rate	RMSE	R2
Tensile Strength	263	2.0	6	1	sqrt	0,19747	97,1913	0,82125
	440	2.0	5	2	sqrt	0,15191	94,24301	0,81911
	500	2.0	10	1	sqrt	0,17631	99,71942	0,81772
Yield Strength	500		15	16	5	0,41487	107,3134	0,7692
	363	5.0	12	15	log2	0,33678	117,2393	0,76763
	449		15	4	10	0,01534	111,0111	0,76205
Elongation	500	4.0	20	3	10	0,0643	1,94019	0,68966
	500		20	5	10	0,01	1,96385	0,68818
	500		19	14	1	0,21465	1,97934	0,68665
Brinell Hardness	276	20.0	6	14	log2	0,25239	30,67184	0,84162
	226	20.0	10	14	sqrt	0,26227	31,01598	0,83934
	500	5.0	20	16	5	0,29529	31,96409	0,82974
Impact Toughness	500	5.0	16	17	10	0,0628	20,93412	0,71911
	253	3.0	12	13	5	0,21654	21,34153	0,71249
	426	20.0	20	13		0,06868	20,83767	0,70993

Optimization using the Bayesian Search algorithm was a major success, as it successfully selected the best parameter sets in terms of model performance. For the models related to the elongation and impact toughness parameters, the situation is not entirely clear, as it cannot be definitively stated that these models are worse. Models generated using Bayesian optimization were evaluated using 5-fold cross-validation, while Grid Search only applied 3-fold validation. This may suggest that, in reality, the models selected by Bayesian Search are more efficient. Therefore, when evaluating the results for the A5 and K parameters, models obtained using both Bayesian optimization and Grid Search will be considered.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_21

4.5 Summary of Results

During the evaluation of results, all models selected during the study were assessed based on the test set, just as in the first experiment with default parameters. This allowed for the selection of the best model for each parameter, with the summary of selected models shown in Table 5.

Table 5 Summary of the Best Models for Parameters

Parameter	Model	N estimators	Max depth	Min samples split	Min samples leaf	Max features	Learning rate
Tensile Strength	bs_rm	263	2	6	1	sqrt	0.19746561089365053
Yield Strength	bs_a5	500	4	20	3	10	0.0643048881887624
Elongation	gs_a5	500	None	15	2	sqrt	0.01
Brinell Hardness	bs_a5	500	4	20	3	10	0.0643048881887624
Impact Toughness	bs_a5	500	4	20	3	10	0.0643048881887624

4.6 Application

The set of models forms the basis of a web application providing a graphical interface that allows the end user to quickly estimate the parameters of the produced casting. For this project, a microservices architecture was chosen. A key aspect considered when making this decision was the ability to separate the project into distinct modules for the user interface and the server-side component. This approach allows each component to be developed independently, which makes it easy to expand specific functionalities without affecting the rest of the system.

The main task of the client application is to display data retrieved through requests to the server and to allow the user to input data describing the chemical composition, thickness, and heat treatment parameters of the cast iron. Upon reaching the main page of the client application, the user will see a table of standards based on PN-EN 1564:2012 in the central part of the view. The navigation bar on the main page of the application contains the application name and three functional buttons: Predict Params, Read CSV, API Specification.

The main task of the server is to process the HTTP requests received from the client (the user can input the chemical composition, thickness, and heat treatment process of the designed cast iron) via the exposed API (Application Programming Interface), which includes three endpoints:

- GET /CastIronApi/norms – returns a list of standards based on EN-PN 1564:2012.

- POST /CastIronApi/ – makes predictions of mechanical properties based on the received parameters of the cast iron and assigns the appropriate standard.

- POST /CastIronApi/csv – makes predictions of mechanical properties, in this case, the query contains a .csv file with a list of input parameters for one or more cast irons, which is then processed and a list of predictions is returned.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_21

Communication between the main components is realized using the REST API exposed on the server side. The client application, when interacting with the user, sends HTTP requests to the server, which processes the received queries, loads the models, calculates the results, and sends a response to the frontend application, which then presents the obtained results.

5 Conclusion

Research on the use of machine learning algorithms in the context of ADI cast iron production has enabled the identification of optimal models for predicting its mechanical properties. As a result, a set of optimized predictive models and a web application for interactive work with these models were developed. An analysis was conducted, covering the impact of hyperparameters on the effectiveness of algorithms, different parameter space search strategies, and the assessment of the efficiency of both simpler and more complex models after applying data augmentation to a limited dataset. The research results allowed for the selection of the most efficient models and the optimization of their hyperparameters. It was found that GradientBoostingRegressor from the scikit-learn library achieves the best results for all analyzed mechanical properties of ADI cast iron, which is why models based on this method were used in the developed system.

Despite the use of augmentation techniques, no significant improvement in model accuracy was observed. This may be due to both the data specification and the insufficient size, diversity, and completeness of the dataset. Data on the mechanical properties of ADI cast iron are strongly related to actual physical and technological processes. Augmenting such data may not introduce new data, which may lead to overfitting.

Future work should focus on the applicability of artificial neural networks and other data augmentation techniques. The tested dataset was very limited, so other data generation techniques need to be tested to expand the dataset. Data augmentation did not bring any improvement, so in future research, we plan to use synthetic data to verify the models on a larger dataset.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Wang YZ, Zheng Z, Zhu MM, et al (2022) An integrated production batch planning approach for steelmaking-continuous casting with cast batching plan as the core. Comput Ind Eng 173:108636. https://doi.org/10.1016/J.CIE.2022.108636
2. Lee M, Moon K, Lee K, et al (2024) A critical review of planning and scheduling in steel-making and continuous casting in the steel industry. Journal of the Operational Research Society 75:1421–1455. https://doi.org/10.1080/01605682.2023.2265416

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_21

3. Mete Ayhan H, Kır S (2024) Ml-driven approaches to enhance inventory planning: Inoculant weight application in casting processes. Comput Ind Eng 193:110280. https://doi.org/10.1016/J.CIE.2024.110280
4. Silva AJ, Cortez P (2021) An Automated Machine Learning Approach for Predicting Chemical Laboratory Material Consumption. IFIP Adv Inf Commun Technol 627:105–116. https://doi.org/10.1007/978-3-030-79150-6_9
5. Zuo D, Lu Y (2024) Design of Amorphous Alloy Composition and Optimization of Vacuum Die Casting Process Parameters Based on Co-Evolutionary Algorithm. IEEE Access 12:123883–123896. https://doi.org/10.1109/ACCESS.2024.3452676
6. Yang J, Liu B, Zeng Y, et al (2024) Data extension-based analysis and application selection of process-composition-properties of die casting aluminum alloy. Eng Appl Artif Intell 133:108514. https://doi.org/10.1016/J.ENGAPPAI.2024.108514
7. Al O, Candan F, Candan S, et al (2024) Adaptive neuro-fuzzy inference system approach for tensile properties prediction of LPDC A357 aluminum alloy. Comput Mater Sci 244:113275. https://doi.org/10.1016/J.COMMATSCI.2024.113275
8. Virdi JS, Peng W, Sata A (2019) Feature selection with LASSO and VSURF to model mechanical properties for investment casting. ICCIDS 2019 - 2nd International Conference on Computational Intelligence in Data Science, Proceedings. https://doi.org/10.1109/ICCIDS.2019.8862141
9. Su X, Yan X, Tsai CL (2012) Linear regression. Wiley Interdiscip Rev Comput Stat 4:275–294. https://doi.org/10.1002/WICS.1198
10. LinearRegression — scikit-learn 1.6.1 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html. Accessed 2 Feb 2025
11. Kingsford C, Salzberg SL (2008) What are decision trees? Nat Biotechnol 26:1011–1012. https://doi.org/10.1038/NBT0908-1011
12. DecisionTreeRegressor — scikit-learn 1.6.1 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html#sklearn.tree.Decisi%20onTreeRegressor. Accessed 2 Feb 2025
13. ExtraTreeRegressor — scikit-learn 1.6.1 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.tree.ExtraTreeRegressor.html. Accessed 2 Feb 2025
14. Biau G, Scornet E (2016) A random forest guided tour. Test 25:197–227. https://doi.org/10.1007/S11749-016-0481-7/METRICS
15. RandomForestRegressor — scikit-learn 1.6.1 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html#sklearn.ense%20mble.RandomForestRegressor. Accessed 2 Feb 2025
16. ExtraTreesRegressor — scikit-learn 1.6.1 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html#sklearn.ensembl%20e.ExtraTreesRegressor. Accessed 2 Feb 2025
17. Boosting Algorithms Explained. Theory, Implementation, and… | by Zixuan Zhang | Towards Data Science. https://medium.com/towards-data-science/boosting-algorithms-explained-d38f56ef3f30. Accessed 2 Feb 2025
18. AdaBoostRegressor — scikit-learn 1.6.1 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html#sklearn.ensemble%20.AdaBoostRegressor. Accessed 2 Feb 2025

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_21

19. GradientBoostingRegressor — scikit-learn 1.6.1 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html#sklearn.e%20nsemble.GradientBoostingRegressor. Accessed 2 Feb 2025
20. Python API Reference — xgboost 2.1.3 documentation. https://xgboost.readthedocs.io/en/stable/python/python_api.html#module-xgboost.sklearn. Accessed 2 Feb 2025
21. Awad M, Khanna R (2015) Efficient learning machines: Theories, concepts, and applications for engineers and system designers. Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers 1–248. https://doi.org/10.1007/978-1-4302-5990-9/COVER
22. SVR — scikit-learn 1.6.1 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html#sklearn.svm.SVR. Accessed 2 Feb 2025
23. Song Y, Liang J, Lu J, Zhao X (2017) An efficient instance selection algorithm for k nearest neighbor regression. Neurocomputing 251:26–34. https://doi.org/10.1016/J.NEUCOM.2017.04.018
24. KNeighborsRegressor — scikit-learn 1.6.1 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html#sklearnneighbors-kneighborsregressor. Accessed 2 Feb 2025
25. Liashchynskyi PB, Liashchynskyi P (2019) Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS. arXiv.org
26. Wu J, Chen XY, Zhang H, et al (2019) Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. Journal of Electronic Science and Technology 17:26–40. https://doi.org/10.11989/JEST.1674-862X.80904120
27. Yescas MA (2003) Prediction of the vickers hardness in austempered ductile irons using neural networks. International Journal of Cast Metals Research 15:513–521. https://doi.org/10.1080/13640461.2003.11819537
28. Bayesian optimization with skopt — scikit-optimize 0.8.1 documentation. https://scikit-optimize.github.io/stable/auto_examples/bayesian-optimization.html. Accessed 2 Feb 2025

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_21