Int j simul model 23 (2024) 4, 634-643
Original scientific paper

APPLICATION OF MACHINE LEARNING TO REDUCE CASTING DEFECTS FROM BENTONITE SAND MIXTURE

Breznikar, Z.*; Bojinovic, M.** & Brezocnik, M.*
* University of Maribor, Faculty of Mechanical Engineering, Smetanova 17, 2000 Maribor, Slovenia
** Pro Labor d.o.o., Mestni trg 7a, 3310 Žalec, Slovenia
E-Mail: ziga.breznikar@student.um.si, marko.bojinovic@prolabor.si, miran.brezocnik@um.si

Abstract
One of the largest Slovenian foundries (referred to as Company X) primarily focuses on casting moulds for the glass industry. In collaboration with Pro Labor d.o.o., Company X has been systematically gathering defect data since 2021. The analysis revealed that the majority of scrap caused by technological issues is attributed to sand defects. The initial dataset included information on defect occurrences, technological parameters of sand mixture and chemical properties of the cast material. This raw data was refined using data science techniques and statistical methods to support classification. Multiple binary classification models were developed, using sand mixture parameters as inputs, to distinguish between good casting and scrap, with the *k*-nearest neighbours algorithm. Their performances were evaluated using various classification metrics. Additionally, recommendations were made for development of a real-time industrial application to optimize and regulate pouring temperature in the foundry process. This is based on simulating different pouring temperatures while keeping the other parameters fixed, selecting the temperature that maximizes the likelihood of successful casting.
(Received in August 2024, accepted in November 2024. This paper was with the authors 2 weeks for 1 revision.)

Key Words: Gravity Casting, Machine Learning, Defect, Classifier, Data Science

1. INTRODUCTION

Machine learning algorithms are essential across various scientific fields, enabling the analysis of complex data, outcome prediction, and process optimization. In engineering and other disciplines, they are widely used to refine product design, simulate dynamic systems, and forecast equipment failures. Algorithms like genetic algorithms, genetic programming [1], neural networks [2], random forests, *k*-nearest neighbours [3], and reinforcement learning offer practical solutions in areas such as computer-aided modelling, production optimization, automation, material science [4], logistic and various management systems [5, 6]. These methods reduce development time and costs while driving sustainable, innovative engineering advancements.

In this paper, we aim to apply machine learning techniques to minimize casting defects, improving the overall quality and efficiency of the casting process. During casting, numerous technological parameters determine whether the final product will meet quality standards. The key technological parameters influencing the casting quality are numerous, however, this study specifically focuses on the following categories [7]:

- Casting parameters (e.g., pouring temperature, pouring time),
- Chemical composition of the cast material (i.e., the proportions of elements in alloy),
- Parameters of bentonite sand mixture (e.g., sand temperature, moisture content).

As indicated above, determining the quality of product based on its technological parameters is a complex multivariable problem. Solving such problems typically requires extensive knowledge and experience, but even this approach often only provides a rough approximation. However, if a sufficient amount of high-quality data is available, it can be leveraged to develop knowledge model using machine learning algorithms [8].

Supervised machine learning algorithms build empirical equations based on a dataset, linking a set of independent variables to its corresponding dependent variable [8]. When these corresponding variables are discrete, the problem is framed as a classification task [9]. A well-constructed knowledge model could serve as a foundation for optimizing process parameters in real-time, thus reducing scrap rates in production.

The quality and selection of the knowledge model primarily depend on the quality and quantity of input data [9]. Since the data collected for this study was not originally intended for classification purposes, the dataset was limited in both quantity and quality. As a result, we opted for a relatively simple classifier, to prevent overfitting [10]. Based on this work, it is possible to develop a more advanced classifier, which could then be applied in an industrial setting to ensure optimal casting parameters and maximize the likelihood of producing high-quality products. Accurate and comprehensive data is crucial, as it allows the model to capture complex relationships between input parameters and casting outcomes. Without sufficient and reliable data, even the most sophisticated classifier will struggle to deliver consistent and accurate predictions [10, 11].

The rest of the article is structured as follows. Section 2 describes the methods used to acquire and prepare data for the classification task. Preprocessing steps involved removing outliers, normalizing features, and balancing the dataset with synthetic data for minority class. In section 3, multiple binary classification models were developed with the KNN algorithm to predict casting outcomes based on the sand parameters. Different KNN configurations were created, analysed and evaluated using classification metrics. Section 4 discusses the classifier's performance and recommends improved data collection along with larger, more comprehensive datasets for better reliability. It also proposes a real-time optimization system that could dynamically adjust casting temperatures. The key contributions and findings of the study are summarized in the conclusion, along with recommendations for future research directions.

2. MATERIALS AND METHODS

2.1 Data acquisition and preparation

Technological parameters were captured by sensors, which transmitted digital and analogue signals to PLCs. Sand parameters are captured within the filling drum at an average interval of two minutes, while the chemical composition of the alloy is recorded for each casting batch. These values were subsequently stored in a database using TCP/IP protocols. Since 2021, the company's quality control department has documented data on the number of defective castings and reason for scrap. Table I presents the reasons and the distribution of scrap included in the database. Initially, the database comprised 21130 datapoints, each representing an instance of scrap.

Table I: Reasons and distribution of scrap in the database.

Scrap reason	**Percentage** (%)
Overproduction/Remelt	77.62
Sand defect	10.38
Other	3.20
Gas porosity	2.90
Damaged casting	2.11
Cold shut	1.60
Core defect	0.81
Shrinkage porosity	0.62
Core misalignment	0.53
Machining defect	0.26
Subcontractor defect	0.01

Database was implemented in the MS SQL environment and consisted of multiple tables. Due to the lack of an explicit connection in the database between technological parameters of sand used in mould production and the quality of casting, it was essential to interpolate these parameters. To facilitate this process, we aggregated the scrap based on common batch number, assuming uniform sand parameters within each batch. This method produced 7115 data points. Therefore, a single data point contained information about number of castings inside that batch, the number of specific scrap and the type of cast used. The chemical composition of casting used in batches were explicitly linked to batch numbers, however, anomalies, including missing values, were presented in the chemical composition table. To address this, we employed a *k*-nearest neighbours method. The algorithm identified five nearest data points using Euclidean distance. The missing values were then filled with the average of the parameters from these five closest data points. Data points from these two tables were combined based on batch numbers. We identified 5147 common batches across both scrap and chemical composition tables. Since different alloys are composed of distinct elements, they yield varying input data for the classifier. Consequently, this led to classification dataset being segmented into several smaller subsets. To link the sand mixture parameters with scrap data, we utilized the Mould Production Report table and Sand Parameters table. These parameters are recorded on the mould production line at intervals of *n* minutes, with *n* varying according to production dynamics.

The next step involved linking the sand parameters of moulds used in casting with the scrap data. This was achieved using three primary datasets:

- **Scrap-Chemistry dataset**: This dataset provided the batch identification number, the chemical composition of the alloy and the associated scrap data for each casting batch.
- **Mould production report dataset**: This dataset contains batch numbers and an *EndOfBatch* timestamp (in date-time format), indicating the time at which mould preparation was completed. This timestamp allows us to connect casting batches with the corresponding sand parameters recorded around the mould preparation time.
- **Sand parameters dataset**: This dataset includes the sand parameters measured by sensors and the *Time* field (in date-time format), which logs when sensors captured these parameters. By comparing the *Time* and *EndOfBatch* values, we could establish a connection between sand parameters and casting batches.

The basic algorithm for creating a dataset for classification purposes is shown on a flowchart in Fig. 1 The parameters *T* and *dt* depend on the dynamics of production. The use of the algorithm to obtain approximate sand parameters used in mould production is necessary because the database does not contain an explicit connection between these parameters and casting batches. This approach significantly impacts the quality of the classifier.

Each data point thus contains 25 features, creating a 25-dimensional feature space, which is classified into five categories:

- Good casting (0),
- Sand defect (1),
- Shrinkage porosity (2),
- Cold shut (3),
- Gas porosity (4).

Using the steps outlined in the previous section, we obtained a dataset containing 392 data points. Developing a classifier with such a high-dimensional feature space relative to a small dataset was impractical. Therefore, we opted for a binary classification, where the input data consisted of sand parameters:

- Temperature,
- Moisture,
- Compression test,
- Compactness,

- Bentonite mass.

The classification classes were:

- Good casting (0),
- Sand defect (1).

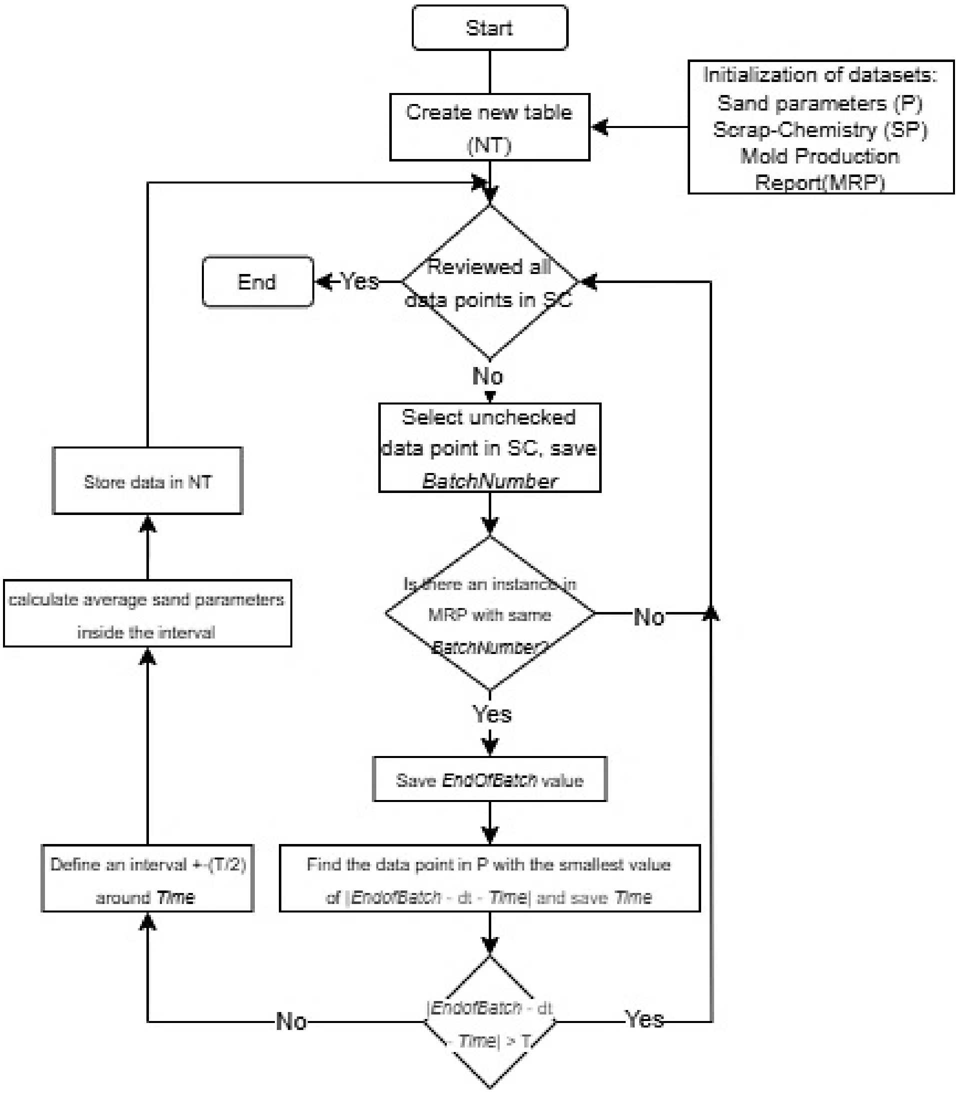

Figure 1: Basic algorithm for creating a dataset for classification purposes.

As noted above, we removed the parameters related to the alloy's chemical composition. This reduction minimized the feature space and limited the classification to two classes. Minority classes (2 to 4) would have negatively impacted classifier quality. The small dataset available for model training compelled us to simplify the classification problem. This simplification also increased the dataset size, resulting in the final dataset with 1099 data points. If a sand defect is present within the casting batch, the data point is classified as 1.

2.2 Preprocessing

Preprocessing involves cleaning, verifying and formatting data into a usable dataset, which enhances the accuracy of machine learning model development [12].

Abnormalities in the collected data include faulty sensor records, which could adversely affect classifier performance. To identify these anomalies, we used the interquartile range (IQR) method [13] shown in Fig. 2.

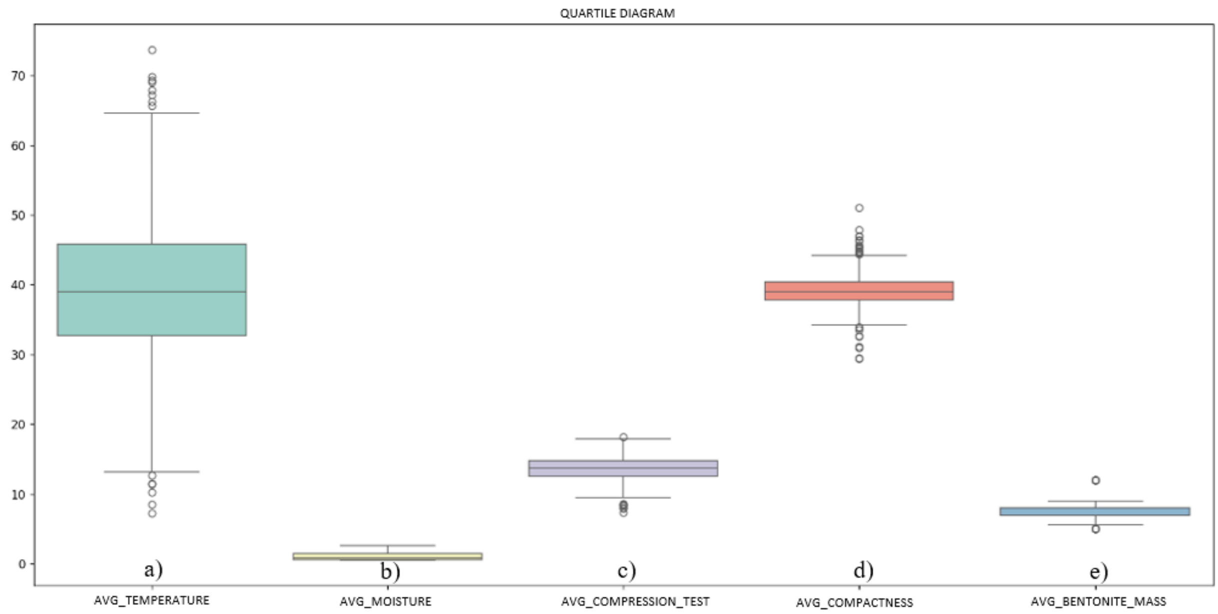

Figure 2: IQR plot of features: a) average temperature, b) average moisture, c) average compression test, d) average compactness, e) average bentonite mass.

As shown in Fig. 2, the properties b) and e) are asymmetrically distributed around the median, while other properties display symmetry. The dots in Fig. 2 indicate outliers, which represent a relatively small portion of the dataset and were therefore removed. After cleaning, the revised dataset contained 1028 elements.

Normalization of datasets is beneficial for learning algorithms [12]. However, the collected data spans different ranges, which complicates achieving an optimal computation state for machine learning models. Using different normalization methods for data with varying distributions is essential because it ensures that each feature is transformed appropriately, preserving its underlying characteristics and enhancing model performance [12]. Therefore, analysis of distribution for each feature was necessary. Figs. 3 and 4 represent Quantaile-Quantile (Q-Q) diagrams and histograms respectively [13].

Figs. 3 and 4 show that features b) and e) deviate significantly from a normal distribution, while the remaining features can be approximated as normally distributed. The discrete values observed in e) stem from the proportional addition of bentonite relative to the mass of sand. When the sand weight is set, the corresponding amount of bentonite is added, resulting in bentonite mass values that are not continuous but rather discrete. Moisture content represented in b) is usually around the lower limit because, under typical conditions, the moulds are relatively dry. However, the outliers on the right side of the distribution can be attributed to changes in weather conditions. When there's higher humidity in the air, the sand absorbs more moisture, leading to occasional spikes in moisture content. For the normally distributed features, we applied the StandardScaler method to transform the data [12]. In contrast, we used a logarithmic transformation for the others to better manage their skewed distribution [9].

The dataset was divided into three subsets in a 60:20:20 ratio using stratification, ensuring that class distribution remained consistent across all subsets [12]. The largest subset was used for training the model, while the other two were used for validation and testing. Data transformation parameters were determined based on the training set.

Using the SMOTE method, we generated synthetic data for the minority class and incorporated it into the training set [14]. Initially the training set contained 616 data points, which increased to 1170 after applying SMOTE.

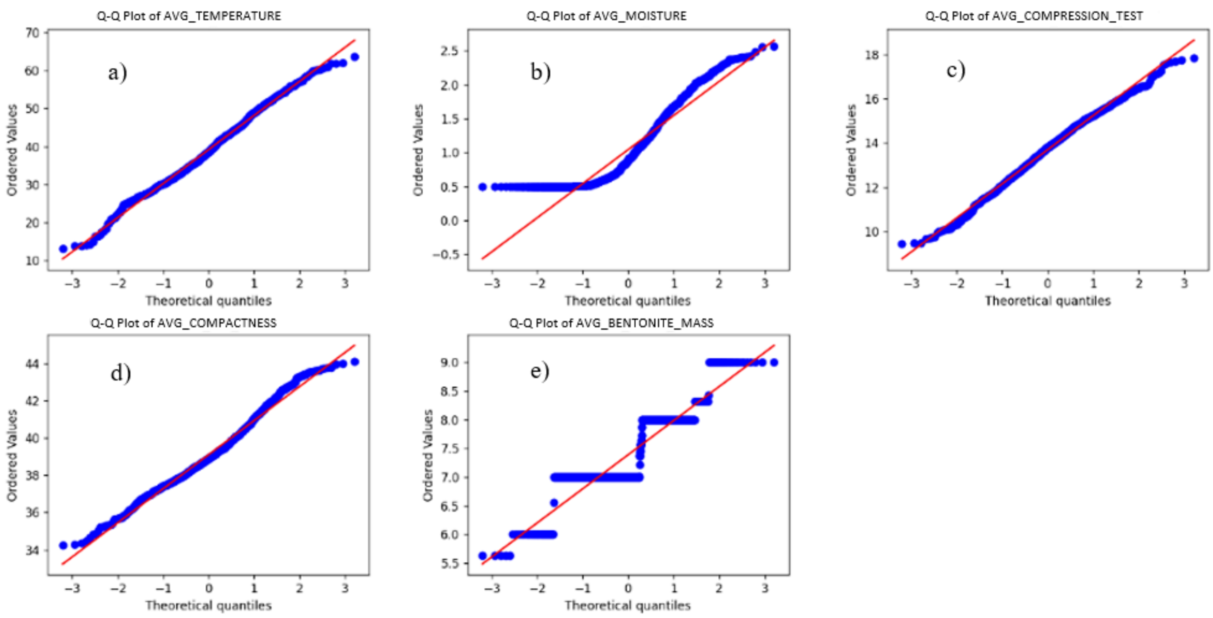

Figure 3: Quantile-Quantile plot of features: a) average temperature, b) average moisture, c) average compression test, d) average compactness, e) average bentonite mass.

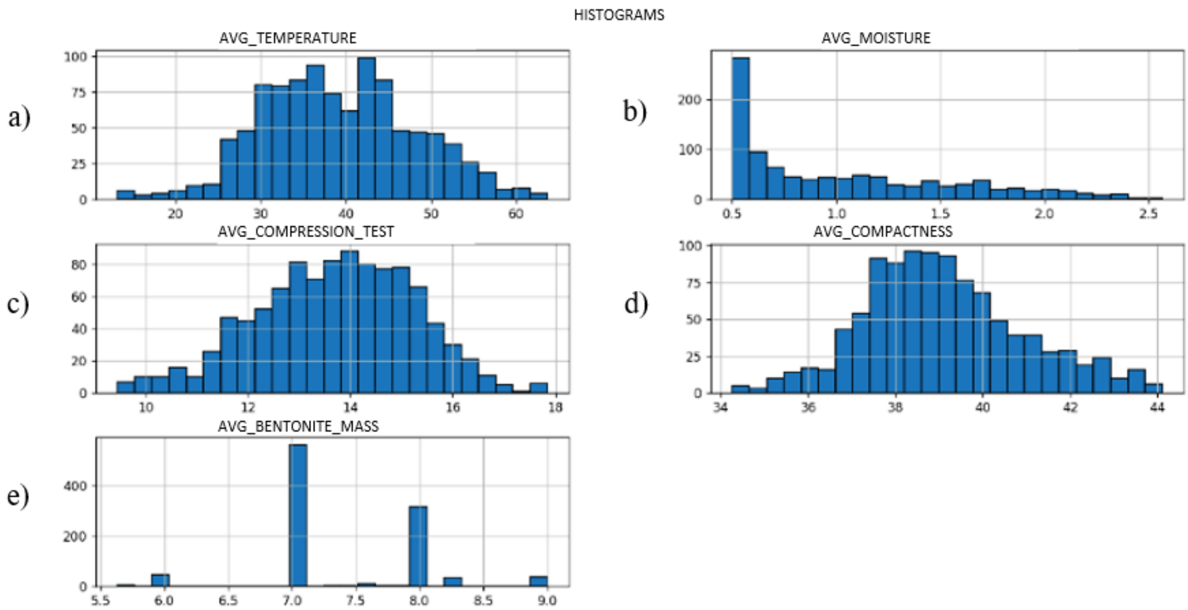

Figure 4: Histogram plot of features: a) average temperature, b) average moisture, c) average compression test, d) average compactness, e) average bentonite mass.

3. CLASSIFICATION MODELS

The machine learning algorithm used to create the classification model is called *k*-nearest neighbours (KNN). *K*-nearest neighbours is a relatively simple algorithm that does not assume a normal distribution of data [8]. Its simplicity helps prevent model overfitting, which is beneficial when training on a relatively small dataset.

Knowledge models built upon KNN algorithms vary based on two parameters [12]:

- Number of neighbours (*k*): The letter *k* represents the number of nearest data points the algorithm considers to classify a new data point.
- Distance metric: In data analysis, the distance between two points indicates their similarity. Various metrics are available to measure this distance.

Four different knowledge models were developed through iterative work in Python. The code was designed to find optimal KNN parameter combinations for maximizing values across various classification metrics. The independent variables for KNN included:

- Number of neighbours: $k \in N \mid k \leq 100$
- Distance metrics:
  - Euclidean,
  - Cosine,
  - Manhattan.

The dependent variables were the classification metrics:

- Accuracy,
- Precision,
- Recall,
- *F*-score.

The iterative loop to find the parameters for optimal metrics values generated and compared 300 knowledge models. The flowchart of the algorithm is represented in Fig. 5.

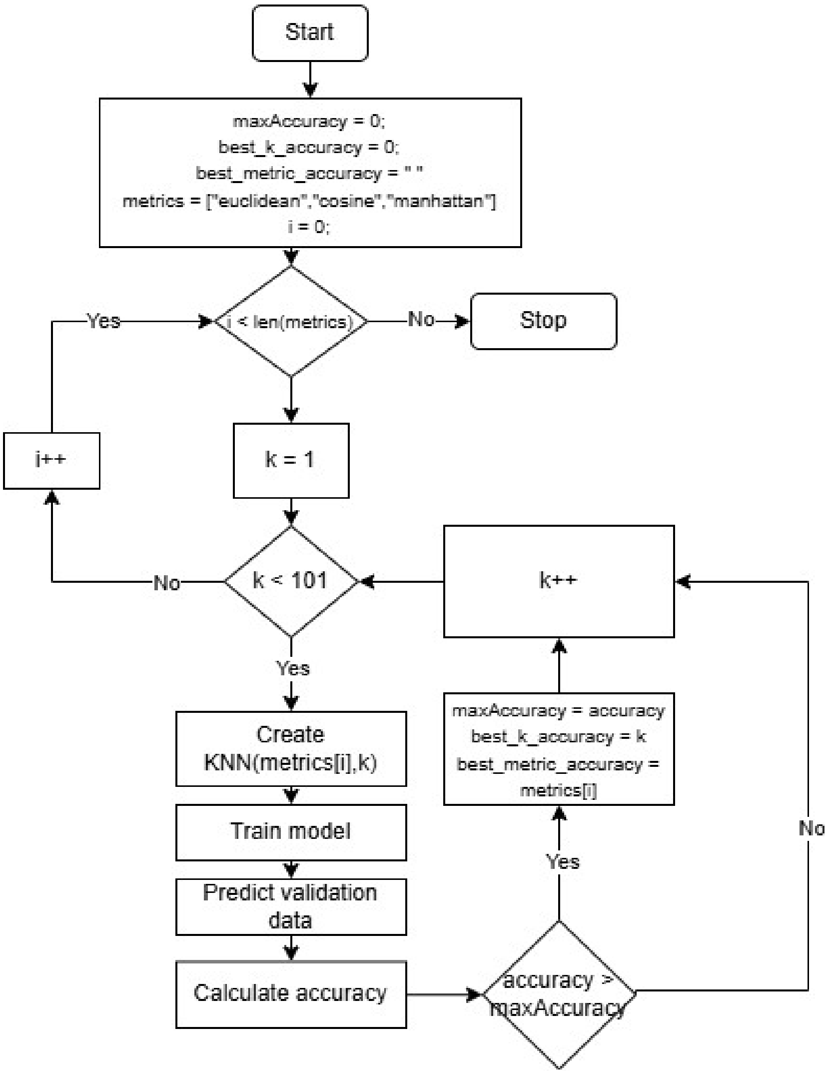

Figure 5: Flowchart of an algorithm to determine optimal KNN parameters for maximizing accuracy.

In Fig. 5, a flowchart illustrates the process for calculating optimal KNN parameters to achieve maximum accuracy. The same approach was applied for calculating the maximum values of other classification metrics. The values are presented in Table II.

Table II: Classifier tuning parameters and corresponding classification metric values.

Number of neighbours (k)	**Distance metric**	**Classification metric**
2	Cosine	Accuracy = 0.85
4	Euclidean	Precision = 0.086
41	Euclidean	Recall = 0.64
5	Cosine	F-score = 0.015

The classification metrics are based on the assumption that the sand defect (labelled as 1) is the positive class, while good casting (labelled as 0) is considered the negative class.

4. RESULTS AND DISCUSSION

In the previous section, the method for obtaining four classification models was described. The next step was to decide which of these models to use. From a process perspective, the most critical error would be incorrectly predicting that a casting will be of good quality. Given our labelling of the positive and negative classes, it is essential to minimize the number of false negatives (FN). Based on this criterion, we chose to use recall as the evaluation metric. Recall is defined as the ratio of correctly predicted positive classes to the total instances of all positive classes [10], as shown in Eq. (1).

$$recall = \frac{TP}{TP + FN} \tag{1}$$

Eq. (2) applies to the given dataset.

$$TP + FN = const. \tag{2}$$

This implies that as the recall value increases, the number of FN instances decreases. In other words, the higher the recall value, the lower the number of FN predictions will be. Based on this logic, we selected a KNN classifier with parameters k = 41 and cosine distance.

The chosen classifier was evaluated on the test set and the classification metrics are shown in Table III.

Table III: Classification metrics of the selected classifier.

Accuracy	**Precision**	**Recall**	***F*-score**
0.58	0.13	0.50	0.20

Confusion matrix is shown in Table IV.

Table IV: Confusion matrix of the classifier.

	Actual 1	**Actual 0**
Predicted 1	10	70
Predicted 0	10	100

Given the information from Table III, the model demonstrates moderate performance with areas that can be refined further with a more suitable and larger dataset. This classifier is suitable for scenarios where capturing as many positives as possible is a priority. Based on our classification of positives being scrap, this is a desired scenario. The relatively low values of precision and F-score (F-score is defined as the harmonic mean of precision and recall [8]) can generally be attributed to an imbalanced data set. Other factors that we couldn’t take into consideration are environmental factors such as temperature and humidity, which are mainly conditioned by weather and seasonal variations. These factors can directly affect the casting process and subsequently the quality of the cast.

As previously mentioned, the quality and size of the dataset are paramount for an effective classifier. The reasons for relatively low-quality classifier include:

- Imbalanced dataset,
- Relatively small dataset,
- Aggregated data,
- Data anomalies,
- Missing data (e.g., casting parameters, environmental factors).

By implementing a data gathering project at Company X aimed at enhancing the classifier, we believe its quality can be significantly improved, resulting in a reduction of scrap. The outline of data collection process and its implications are described below.

The final parameter that can be adjusted in the casting process is the casting temperature [15]. By implementing a high-quality knowledge model, it would be possible to optimize the casting temperature parameter in real-time, considering the other determined parameters in the casting process. An industrial PC could simulate different casting temperature values and select the one with highest probability of producing a high-quality casting. This optimal casting temperature value would then be communicated to the furnace temperature regulator, which would heat the alloy to the specified temperature.

In Company X, they would select casting for which they aim to reduce scrap rates. A dedicated database would be created for the casting, capturing data necessary for the classifier. Data collection would follow steps bellow:

- **Sand parameter collection**: A machine vision system would be attached to the filling drum, with an algorithm detecting the presence of a marked mould using computer vision and recording the time in a date-time format. An identification number would be affixed to the mould frame, and the algorithm would log the timestamp and mould number in a table. Since Company X already has a sand parameter collection process, the sand parameters would be linked through the timestamp, thus creating an explicit connection between the mould number and the sand parameters in the table.
- **Alloy chemical composition collection**: There is no need to modify the data collection method for chemical composition. The chemical composition of the alloy would be linked to casting through the casting batch number.
- **Casting parameter collection**: A machine vision system would be attached to the casting ladle, and the algorithm would record the time and identification number of the mould beneath it. A temperature sensor would need to be installed in the furnace or casting ladle, logging the time and temperature in the data table. The alloy temperature and mould identification number could then be connected via the timestamp.
- **Casting classification**: Marked moulds would need to be separated from the cooling line since mould destruction process typically removes the information linking each casting to its originating mould. Marked moulds would need to be manually broken, and the classification of the resulting casting would be recorded.

5. CONCLUSION

This study demonstrates the potential of machine learning algorithms to optimize the casting process in foundries by predicting the likelihood of casting defects based on technological parameters. Despite challenges such as imbalanced and relatively small dataset, as well as the absence of direct links between some of the key parameters, the results suggest that a predictive classifier could be valuable in real-world applications. The binary classification model focusing on sand defects showed a moderate performance, with recall being the primary evaluation metric to minimize falsely predicting good quality casts. While the current classifier's performance is moderate, the findings underscore the importance of high-quality, comprehensive data in enhancing the model accuracy.

To further improve the classifier's performance and reduce scrap, a robust data collection system, as outlined in this study, is essential. By integrating real-time measurements of sand parameters, alloy composition, and casting, and feeding this data into a more advanced machine learning model, it is possible to continuously optimize casting parameters. Particularly, pouring temperature can be optimized to minimize defects. Future work should focus on expanding the dataset and refining the data gathering process to provide a stronger foundation for the development of a more accurate and robust industrial application.

REFERENCES

[1] Kovačič, M.; Lešer, B.; Brezocnik, M. (2021). Modelling and optimization of sulfur addition during 70MnVS4 steelmaking: an industrial case study, *Advances in Production Engineering & Management*, Vol. 16, No. 2, 253-261, doi:10.14743/apem2021.2.398

[2] Han, L. N.; Ma, X. Z.; Tan, J. D.; Li, J. H.; Dong, Y. Q. (2023). Balancing material supply-demand with ARIMA and neural networks, *International Journal of Simulation Modelling*, Vol. 22, No. 4, 712-722, doi:10.2507/IJSIMM22-4-CO18

[3] Zhang, Y. D.; Liao, L.; Yu, Q.; Ma, W. G.; Li, K. H. (2021). Using the gradient boosting decision tree (GBDT) algorithm for a train delay prediction model considering the delay propagation feature, *Advances in Production Engineering & Management*, Vol. 16, No. 3, 285-296, doi:10.14743/apem2021.3.400

[4] Guo, K.; Yang, Z.; Yu, C.-H.; Buehler, M. J. (2021). Artificial intelligence and machine learning in design of mechanical materials, *Materials Horizons*, Vol. 8, No. 4, 1153-1172, doi:10.1039/D0MH01451F

[5] Avci, İ.; Bidollahkhani, M. (2023). Real-time building management system visual anomaly detection using heat points motion analysis machine learning algorithm, *Technical Gazette*, Vol. 30, No. 1, 318-323, doi:10.17559/TV-20220417151954

[6] Azhar Ali, S. E.; Rizvi, S. S. H.; Lai, F.; Faizan Ali, R.; Jan, A. (2021). Predicting delinquency on mortgage loans: an exhaustive parametric comparison of machine learning techniques, *International Journal of Industrial Engineering and Management*, Vol. 12, No. 1, 1-13, doi:10.24867/IJIEM-2021-1-272

[7] Elliott, R. (2014). *Cast Iron Technology*, Butterworth-Heinemann, London

[8] Müller, A. C.; Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*, O'Reilly Media, Sebastopol

[9] Goodfellow, I.; Bengio, Y.; Courville, A. (2016). *Deep Learning*, MIT Press, Cambridge, from *https://www.deeplearningbook.org/*, accessed on 22-8-2024

[10] Alpaydin, E. (2014). *Introduction to Machine Learning*, 3rd ed., MIT Press, Cambridge

[11] Siahaan, A. A.; Asrol, M. (2023). Development of a machine learning model for predicting hardness in the water treatment pharmaceutical industry, *International Journal of Industrial Engineering and Management*, Vol. 14, No. 2, 138-146, doi:10.24867/IJIEM-2023-2-329

[12] Karakatič, S.; Fister, I. (Jr.) (2022). *Strojno učenje: s Pythonom do prvega klasifikatorja*, University of Maribor, Maribor (in Slovenian), from *https://dk.um.si/Dokument.php?id=156302&lang=slv*, accessed on 10-7-2024

[13] Bruce, P.; Bruce, A. (2017). *Practical Statistics for Data Scientists*, O'Reilly Media, Sebastopol

[14] Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research*, Vol. 16, 321-357, doi:10.1613/jair.953

[15] Beeley, P. (2001). *Foundry Technology*, 2nd ed., Butterworth-Heinemann, London, doi:10.1016/B978-0-7506-4567-6.X5000-6