Advances in Science and Technology Research Journal

Advances in Science and Technology Research Journal 2022, 16(5), 120–128
https://doi.org/10.12913/22998624/154963
ISSN 2299-8624, License CC-BY 4.0

Received: 2022.08.09
Accepted: 2022.10.17
Published: 2022.11.01

Industrial Application of Deep Neural Network for Aluminum Casting Defect Detection in Case of Unbalanced Dataset

Michał Awtoniuk[1*], Dariusz Majerek[2], Artur Myziak[3], Cyprian Gajda[4]

[1] Institute of Mechanical Engineering, Warsaw University of Life Sciences, Nowoursynowska 166, 02-787 Warsaw, Poland

[2] Faculty of Fundamentals of Technology, Lublin University of Technology, Nadbystrzycka 38, 20-618 Lublin, Poland

[3] MyFee Sp. z o. o., Wojrowicka 49/1, 54-436 Wrocław, Poland

[4] IoTSolution Sp. z o. o., Zana 39 A, 20-634 Lublin, Poland

[*] Corresponding author's e-mail: michal_awtoniuk@sggw.edu.pl

ABSTRACT

We have developed a deep neural network for casting defect detection. The approach is original because it assumes the use of data related to the casting manufacturing process, i.e. measurement signals from the casting machine, rather than data describing the finished casting, e.g. images. The defects are related to the production of car engine heads made of silumin. In the current research we focused on the detection of defects related to the leakage of the casting. The data came from production plant in Poland. The dataset was unbalanced. It included nearly 38,500 observations, of which only 4% described a leak event. The work resulted in a deep network consisting of 22 layers. We assessed the classification accuracy using a ROC curve, an AUC index and a confusion matrix. The AUC value was 0.97 and 0.949 for the learning and testing dataset, respectively. The model allowed for an ex-post analysis of the casting process. The analysis was based on Shapley values. This makes it possible not only to detect the occurrence of a defect but also to give potential reasons for the appearance of a casting leak.

Keywords: machine learning, deep neural network, classification, casting defect detection

INTRODUCTION

The use of artificial intelligence methods finds practical application in many different scientific and industrial fields such as electrical engineering [1], mechanical engineering [2,3], materials engineering [4], environmental engineering [5], and many more. Artificial intelligence methods, and machine learning in particular, are also indicated as an important element of the popular Industry 4.0 concept [6]. A key issue within the Industry 4.0 concept is collection and processing of the data, e.g. through the use of Internet of Things technology [7]. The data can then be used to build machine learning models. Some examples of the most popular models are: shallow neural networks, deep neural networks (DNN), decision trees, random forests, and Support Vector Machines.

The mentioned models can be used for regression, classification and clustering. A very common phenomenon associated with the practical application of machine learning models is working with an unbalanced dataset. This is most often the case when the task of the model is classification, i.e. identifying standard cases from unusual ones, e.g. manufacturing defect detection, fraud detection, medical diagnosis. These cases are called classes. Thus, in the case of an unbalanced dataset, there will be a majority class and one or more minority classes. In order to perform the entire model building process correctly, data balancing must be applied. There are two main data balancing techniques, i.e. undersampling and oversampling [8,9]. Undersampling, also called downsampling, involves removing samples from majority class. The most undersampling methods are based on the k-nearest neighbours algorithm. In contrast, oversampling, another name is upsampling, involves adding samples to a minority class. Popular upsampling techniques include Adaptive Synthetic Sampling Approach (ADASYN) [10], Random Oversampling Examples (ROSE) [11], Synthetic Minority Oversampling Technique (SMOTE) [12].

The problem with insufficient data is not limited to the case of numerical data only, but also extends to images. There are many papers describing the use of image analysis in defectoscopy. In many works deep neural networks are used to detect production defects in castings. Most of them are based on image analysis. The source images can be X-rays [13,14], light microscope [15] but also metallographic images taken with a scanning electron microscope [16]. A number of data augmentation techniques are used to solve the problem of unbalanced datasets, e.g. geometric transformation, histogram equalization, generation of synthetic defects to images [17]. Defect detection methods based on computer vision show high efficiency, but require the preparation of an additional measuring workstation and time to perform accurate, often multiple, measurements.

The aim of this paper is to build a deep neural network to detect a casting defect associated with a leakages. The main novelty of our research is a completely different methodology for building a machine learning model. The dominant approach in the literature is to detect a defect from an image. Unlike this approach we have proposed a method that uses numerical data from the production process to evaluate the casting. As a result, once the casting has been produced, no additional operations, such as X-ray images acquisition, are required. Another novelty is the use of the model interpretation method to identify a potential cause or set of causes of the defect. To our knowledge, no study to date has utilised the Shapley values to analyse the cause of a casting leak.

MATERIALS AND METHODS

Casting process parameters

The data came from production plant located in Poland. Due to production plant's know-how protection policy, we are not allowed to reveal its full name. The production process of car engine heads is well measured by a traceability system. The collected data has both qualitative and quantitative character. Among qualitative data, one can find information such as casting machine number, mould number or operator id. However, from the point of view of defect detection, quantitative data was more useful. In the feature selection process, we selected 31 features or, in other words, predictors that we would include in the model. We do not describe the feature selection process in the article for the reason that it was mainly based on an expert analysis performed by a technologist. The selected data can be divided into four groups. These were data related to the mould, mould cooling system, material and cycle time. A block diagram of the model is shown in Figure 1. A more detailed description is shown in the following list, unfortunately, due to

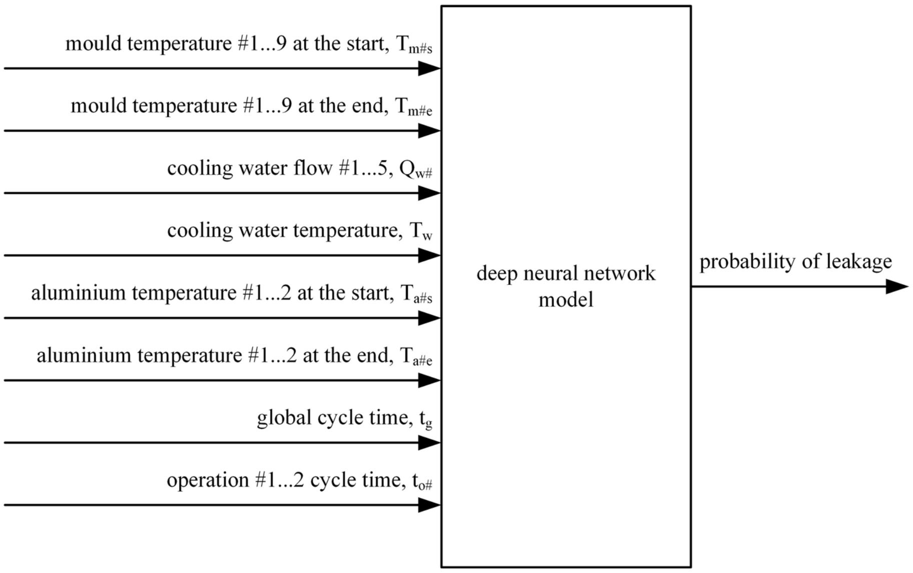
mould temperature #1...9 at the start, $T_{m\#s}$
mould temperature #1...9 at the end, $T_{m\#e}$
cooling water flow #1...5, $Q_{w\#}$
cooling water temperature, T_w
aluminium temperature #1...2 at the start, $T_{a\#s}$
aluminium temperature #1...2 at the end, $T_{a\#e}$
global cycle time, t_g
operation #1...2 cycle time, $t_{o\#}$
deep neural network model
probability of leakage

Fig. 1. Block diagram of the analysed DNN model know-how protection policy, we cannot provide more detailed technological information:

- mould:
  - nine measuring points of mould temperature, the data reading occurred twice, once at the start and once at the end of the casting process;
- mould cooling system:
  - five measuring points of cooling water flow;
  - one measuring point of cooling water temperature;
- material:
  - two measuring points of aluminum temperature, the data reading occurred twice, once at the start and once at the end of the casting process;
- cycle time:
  - one measuring of production global cycle time;
  - two measuring of production selected operations cycle time;

At this stage of the research, we focused on detecting only one casting defect, i.e. leakage. Thus, this is an example of binary classification with specific two classes ‘ok’ (normal casting) and ‘leakage’ (defective casting). The dataset included 38 491 observations and was unbalanced. There were 1 529 cases describing the occurrence of leakage, which is less than 4%.

Modelling procedure

Data preprocessing consisted of several steps, the order of which is important. The first step was to divide the data into a learning and testing dataset. From the entire dataset, we randomly selected 2/3 of the observations as the teaching set (25 660 cases), while the remaining 1/3 of the data formed the testing set (12 831 cases). In order to maintain the proportion of classes in both datasets, we used stratified random selection. As a result, the learning dataset included 1 016 and the testing dataset included 513 cases of the leakage class, which represented 4% of both dataset.

Several predictors were characterized by some asymmetry. In some cases, bimodality and concentration in a narrow range of values were also noticeable. For this reason, we applied the Yeo-Johnson transformation [18]. The next step was to standardize the data in both datasets.

Due to unbalanced data, the models could be characterized by a burden of high efficiency in predicting the class of regular castings and low efficiency in detecting casting defects. Therefore, in order to remove the imbalance, the SMOTE method was used. This method consists of upsampling performed by simulating data created as linear combinations of existing observations. Then the newly formed observations have a similar multivariate distribution to that of the original data. It is important that upsampling is applied after splitting the learning and testing datasets. Otherwise, so-called data leakage could occur. In that case, a certain amount of information about how new observations were generated would be transferred to the testing dataset by the random division. This could artificially improve the quality of the classification. Similarly, data standardisation should also be carried out after splitting into a learning and testing dataset, but before balancing the learning dataset [19]. As a result of balancing, the learning dataset has grown to 49 288 observations. Details of the size of each dataset are shown in Table 1.

The deep neural network model was prepared in the Keras framework [20] implemented in the RStudio environment [21]. The procedure for finding the optimal model was to gradually expand the network structure. We started with a network with one hidden layer with different combinations of the number of neurons. In subsequent trials, we increased the number of layers and again tested different combinations of the

Table 1. The size of learning and testing dataset

Specification	Original dataset	Learning dataset		Testing dataset
		before SMOTE	after SMOTE	
Class ‘ok’ number of observations (dataset percentage)	36 962 (96%)	24 644 (96%)	24 644 (50%)	12 318 (96%)
Class ‘leakage’ number of observations (dataset percentage)	1 529 (4%)	1 016 (4%)	24 644 (50%)	513 (4%)
Total number of observations	38 491	25 660	49 288	12 831

number of neurons. We considered three types of hidden layers: dense, dropout and regularization layers. Without using the last two layers, the model always came out overfitted. The dropout layer removes a certain proportion of neurons (in our case it was up to 50%). The regularization layer can use the L1 and L2 norm. Using the L2 norm reduces the values of the model weights, while the L1 norm removes less important neurons. The activation function for each layer was ReLU. The exception was the last layer for which we adopted a sigmoidal activation function. The reason for this was that we wanted to achieve a casting defect probability in the output of the network.

Classification performance and model interpretability

We assessed the quality of the model using the Receiver Operating Characteristic (ROC curve) [22]. The ROC shows the relationship between true positive rate (TPR) and false positive rate (FPR) at all levels of threshold. The values of TPR and FPR vary from 0 to 1. As the ROC curve is closer to the point (TPR = 1, FPR = 0), the better is the classifier. The TPR and FPR rates can be calculated as:

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

where: TP – true positive, denotes the number of observations for which the model classified the casting as faulty when the casting was actually faulty (these are the cases we expect to see most),

TN – true negative, denotes the number of observations for which the model classified the casting as correct when the casting was actually correct,

FP – false positive, denotes the number of observations for which the model classified the casting as faulty when the casting was actually correct,

FN – false negative, denotes the number of observations for which the model classified the casting as correct while the casting was actually faulty (these are the cases we expect to have the fewest).

The threshold level determines at what value of the leak probability (model output signal) the casting will be considered as a leak. The default value for the threshold level is 0.5, but this can be adjusted during model calibration processes. Whether we care more about detecting leaks or detecting correct castings, the threshold will vary. The model output signal is the probability of leakage. If that probability exceeds the threshold then we assigned such an observation as a class 'leakage' otherwise as a class 'ok'. In our case, the threshold level was selected on the basis of a distance criterion, i.e. we chose the point on the ROC which was closest to the point (1, 0). This threshold level was a trade-off between maximizing TPR and minimizing FPR. The distance criterion [23] can be calculated as

$$d = (1 - TPR)^2 + FPR^2$$

After determining a specific threshold on the ROC curve, we calculated the accuracy (ACC). The ACC value varies from 0 to 1. This index is a very common measure of the quality of a classifier; however, it can easily take on high values in the case of an unbalanced dataset. The ACC can be calculated as

$$ACC = \frac{TP + TN}{TP + TN + TP + FN}$$

The last index of classifier performance we used was the Area Under ROC Curve (AUC). This index, unlike ACC, does not depend on the threshold level and evaluates the model in a more general way. It can also be used in the case of an unbalanced dataset.

We performed the model interpretability using Shap analysis. The method allowed to perform a local analysis (so-called instance level) for single observations as well as a global analysis (so-called dataset level) by averaging Shapley values over all observations [24]. The formal definition of Shapley value describes it as the contribution ϕ_j of the j-th feature to the prediction of the value $\hat{f}(x)$ according to the following formula

$$\varphi(x, j) = \frac{1}{p!} \sum_J \Delta^{j|\pi(J,j)}(x)$$

where: $\Delta^{j|J}(x) = E_X\left[f(X)\middle|X_{j_1} = x_{j_1}, \ldots, X_{j_K} = \right.$ $= x_{j_K}, X_j = x_j\big] - E_X\left[f(X)\middle|X_{j_1} = \right.$ $\left. = x_{j_1}, \ldots, X_{j_K} = x_{j_K}\right]$

and $\pi(j|J)$ the set of the indices of the variables that are positioned in J before the j-th variable. The Shaley value is therefore the effect of removing the information carried by the j-th variable from the conditional expectation value of the prediction. To perform the analysis, we used the Shapley Additive Explanations (SHAP) library available in Python [25]. We divided the analysis into two stages. In the first stage, we calculated the average absolute Shapley value for each predictor. The higher it was, the more impact the feature had on the model's prediction. As a result we determined the global feature importance for the entire test dataset. In a second step, we analysed the distribution of Shapley values for several of the most significant features. This allowed us to investigate the influence of the values of the individual predictors on the probability of a casting defect.

RESULTS AND DISCUSSION

Casting defect detection performance

Table 2 shows the final structure of the network. The learning process followed several initial assumptions. We used binary cross-entropy as the loss function. The maximum training length was 100 epochs. However, the learning process stopped automatically if the value of the loss function in two consecutive epochs did not decrease. The batch size was equal to 256. We used the NAdam algorithm as the optimiser [26]. The validation dataset was 30% of the learning dataset. The progression of the learning is shown in Figure 2. In the figure, we have shown the change in the ACC index instead of the loss function for the reason that we use the ACC index to evaluate the quality of the classifier also on the test dataset. Learning was automatically stopped at epoch 26 according to the early stopping criterion.

The ROC curves for the learning and testing datasets are shown in Figure 3. The AUC value was 0.97 and 0.949 for the learning and testing datasets respectively. These values indicate very good predictive capabilities of the model. Using the distance criterion, the threshold points were selected as 0.714 (learning) and 0.701 (testing). For these threshold point, we calculated the confusion matrices presented in Tables 3 and 4. To improve the clarity in both confusion matrices, we marked the correct classification cases, i.e. TN

Table 2. Structure of DNN model

Layer number	Layer type	Additional information*
1	input	n = 31
2	dense	n = 32
3	regularization	L2 = 0.001
4	dense	n = 32
5	dropout	rate = 0.5
6	dense	n = 32
7	dropout	rate = 0.5
8	dense	n = 32
9	dropout	rate = 0.4
10	dense	n = 32
11	dropout	rate = 0.5
12	dense	n = 32
13	dropout	rate = 0.5
14	dense	n = 32
15	dropout	rate = 0.4
16	dense	n = 20
17	dropout	rate = 0.4
18	dense	n = 10
19	dropout	rate = 0.4
20	dense	n = 5
21	regularization	L1 = 0.001
22	output	n = 1

Note: *n is the number of neurons; rate is the neuron dropout rate; L1, L2 are the norm L1 and L2 regularization parameters

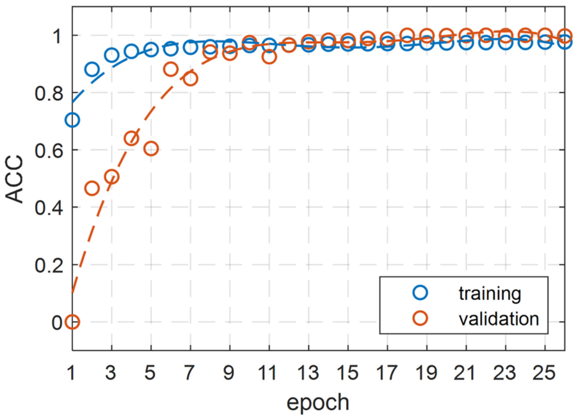

Fig. 2. Change in ACC value during DNN learning

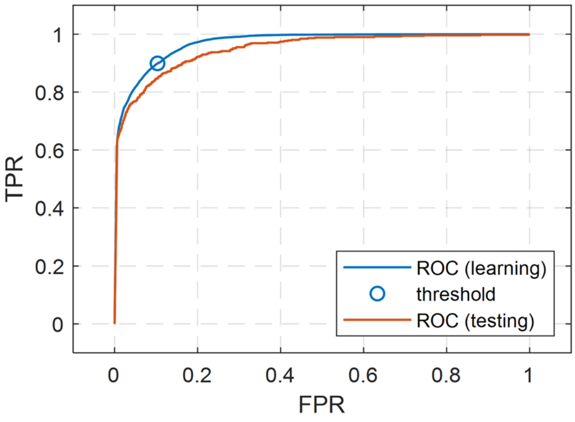

Fig. 3. The ROC curves for trained model

and TP, in green and the misclassification cases, i.e. FN and FP, in red. Tables 3 and 4 also present classification performance index such as ACC, FPR, and TPR.

It is difficult to compare the achieved results with other works due to the completely different dataset. For example, in the paper [15], the model achieved an ACC of 0.94. However, the authors do not describe in detail the issue of balancing the dataset. It should be remembered in a case of unbalanced dataset the ACC index is not a good measure of the quality of the classifier. In contrast, the paper [14] compares a number of models whose ACC was in the range 0.833-0.955 with completely balanced dataset. It can therefore be concluded that the ACC indices for both datasets are satisfactory.

The other indexes are also at a satisfactory level and demonstrate high leak detection performance (TPR = 0.865 for the testing dataset). The number of misclassified observations, especially of the FN type, can be reduced by choosing a different threshold point on the ROC curve. In practice, this will reduce the probability for which a casting is considered defective. On the one hand, this will actually improve the detection of true leaks, but on the other hand it will increase the FPR, which in practice will mean more falsely detected leaks. It is important to remember that the model is intended to assist the production process by indicating which castings should pass additional quality tests. Too many wrongly indicated leaks will result in unnecessary inspection activities. Thus, the distance criterion appears to be an appropriate compromise solution in this case.

Casting defect cause interpretability

Figure 4 shows the global feature importance. It illustrates the strength of the effect of each feature on the leakage probability, but does not indicate whether the feature has positive or negative impact on it. By far the most important feature in the model was temperature T_{m8e} changing the predicted leakage probability on average by 9.7 percentage points. It can also be seen that the features related to the temperature of the casting mould dominate (the first six features). In addition, the group of most important features includes two temperature measurement points, i.e. #9 and #3, taken at both the beginning and end of the casting process. Features related to the flow of water cooling are ranked at the 7th position and below. For a more detailed analysis, we selected the two most significant features, i.e. mould temperatures T_{m8e} and T_{m9e} with the average absolute Shapley value of 0.097 and 0.054 respectively.

Figure 5 shows the distribution of Shapley value over the selected feature. Note that the dataset has been normalised and transformed with the Yeo-Johnson transformation, so that the measurement ranges are modified. For this reason, we have not included a description of the units on the horizontal axes. However, this does not prevent us from drawing interesting conclusions.

Table 3. The confusion matrix for learning dataset

Specification		Target class		Classification performance
		‘ok’	‘leakage’	

Predicted class	‘ok’
	no. of observations
	(dataset percentage)
		TN
		22 096
		(44.8%)
			FN
			2 484
			(5%)
				FPR = 0.103
				TPR = 0.899
				ACC = 0.899

	‘leakage’ no. of observations (dataset percentage)	FP 2 548 (5.2%)	TP 22 160 (45%)	

Table 4. The confusion matrix for testing dataset

Specification		Target class		Classification performance
		‘ok’	‘leakage’	

Predicted class	‘ok’
	no. of observations
	(dataset percentage)
		TN
		10 878
		(84.8%)
			FN
			70
			(0.5%)
				FPR = 0.117
				TPR = 0.865
				ACC = 0.882

	‘leakage’ no. of observations (dataset percentage)	FP 1 440 (11.2%)	TP 443 (3.5%)	

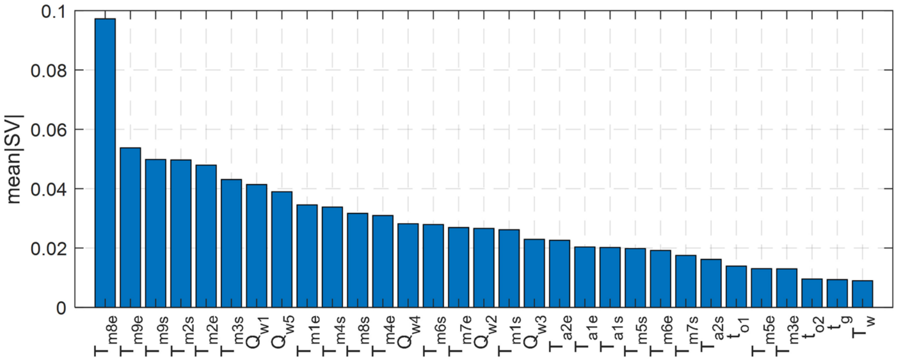

Fig. 4. Global feature importance measured as the mean absolute Shapley value (SV)

Figure 5 (left) suggests that the relationship between the probability of leakage is inverse. In addition, the observations cluster into three groups. This indicates that the distribution of this feature is multi-modal. This is confirmed by the histogram (Figure 6), in which we have shown the temperature distribution of the T_{m8e} mould over the entire original dataset.

Each group is characterised by a different Shapley values range, but only for the lowest temperatures Shapley values are positive. This means that lower temperatures of the mould T_{m8e} increase the probability of leakage. The other two groups are concentrated in a narrower temperature range. It is also worth noting there is a large spread of Shapley values for certain temperature values. For example, for a normalised temperature value of 0.9, the Shapley values vary from -0.2 to 0. Thus, such a mould temperature value can significantly reduce the probability of leakage (by up to 20 percentage points) as well as have a neutral effect. This is an indication that there are some more complex interactions between features in the process.

Figure 5 (right) shows that the higher the temperature is, the higher the Shapley value. In contrast to the mould temperature of T_{m8e}, here a more regular distribution of observations can be seen. The Shapley values vary between -0.2 and 0.3. Higher temperatures are associated with a higher probability of leakage, while lower temperatures reduce this probability.

CONCLUSIONS

This study proposed the original deep neural network model for detection the aluminum casting defect in automotive industry. The research focused on leak detection of engine heads made of silumin. The results of the study lead to the following conclusions:

1. The model performance is on the acceptable level given that the relationships existing between the predictors and the outcome variable

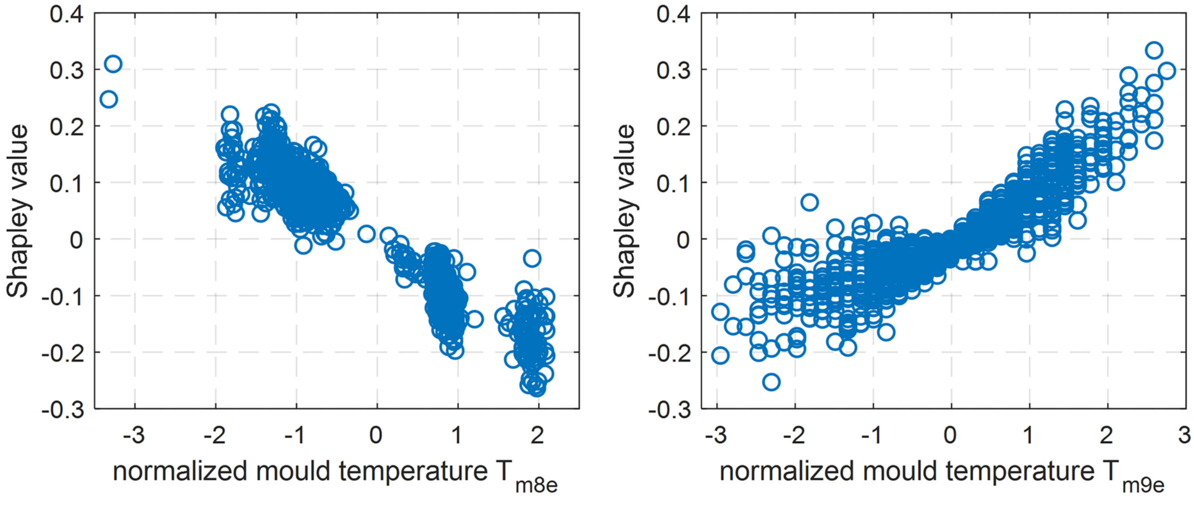

Fig. 5. Distribution of Shapley values over mould temperatures T_{m8e} (left) and T_{m9e} (right)

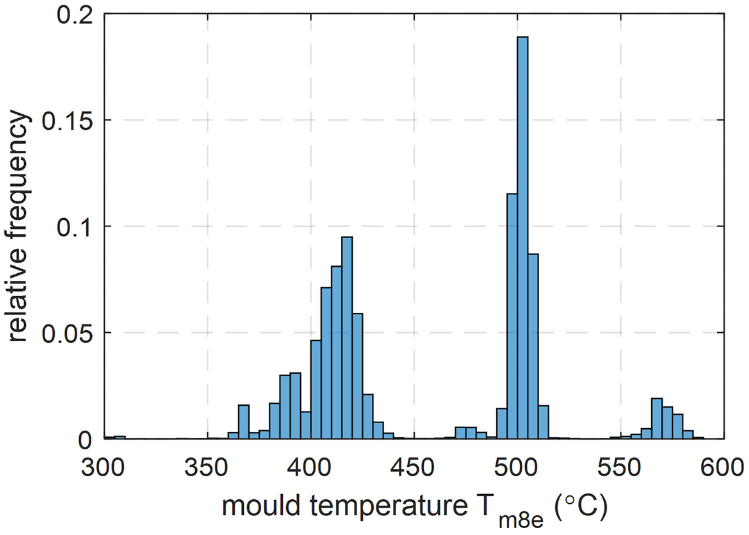

Fig. 6. Histogram of mould temperature T_{m8e}

are complex, often nonlinear, and there are also interactions of the efficiency of individual variables.

2. The most important features influencing the probability of leakage are the signals of the three mould temperature measurement points. One of the point showed a dominant influence. This is probably related to the way the aluminium is introduced into the mould and to the cooling process of the casting.
3. None of the considered features had a clear positive or negative influence on the probability of leakage. For each feature, it was possible to distinguish areas that either increased or decreased the occurrence of a defect.
4. The developed model can be used to automatically evaluate the casting immediately after the completion of the process. The assessment can be performed without additional quality tests. The interpretation of the model by means of Shap analysis indicates the potential causes of the casting defect.
5. Most of the considered features were characterised by an interaction with another feature or features. Finding these interactions could be an interesting direction for future research.
6. The sensitivity analysis of the model presented in the paper is the most interesting part, as it reveals how each predictor affects the probability of leakage. Indication of the strength of the influence and its shape helps to assess the course of the casting process and, consequently, can provide a basis for introducing a corrective procedure.

REFERENCES

1. Salat R., Awtoniuk M. Black box modeling of PIDs implemented in PLCs without structural information: a support vector regression approach. Neural Computing and Applications 2015; 26: 723–34.
2. Zając K., Płatek K., Biskup P., Łatka L. Modelling of hardfacing layers deposition parameters using robust machine learning algorithms. Journal of Physics: Conference Series 2021; 2130:012016.
3. Kulisz M., Zagórski I., Matuszak J., Kłonica M. Properties of the Surface Layer After Trochoidal Milling and Brushing: Experimental Study and Artificial Neural Network Simulation. Applied Sciences 2020; 10:75.
4. Szala M, Łatka L, Awtoniuk M, Winnicki M, Michalak M. Neural Modelling of APS Thermal Spray Process Parameters for Optimizing the Hardness, Porosity and Cavitation Erosion Resistance of Al2O3-13 wt% TiO2 Coatings. Processes 2020; 8:1544.
5. Szeląg B., Suligowski R., De Paola F., Siwicki P., Majerek D., Łagód G. Influence of urban catchment characteristics and rainfall origins on the phenomenon of stormwater flooding: Case study. Environmental Modelling & Software 2022; 150:105335.
6. Pizoń J., Kulisz M., Lipski J. Matrix profile implementation perspective in Industrial Internet of Things production maintenance application. Journal of Physics: Conference Series 2021; 1736:012036.
7. Awtoniuk M., Nowakowski T., Chlebowski J., Świętochowski A., Dąbrowska M., Klonowski J., et al. Internet of Things as an element of the frost protection system in orchards. Journal of Physics: Conference Series 2021; 2130:012015.

8. He H., Garcia EA. Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering 2009; 21:1263–84.
9. Mohammed R., Rawashdeh J., Abdullah M. Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. In: Proc. of 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan 2020, 243–248.
10. He H., Bai Y., Garcia EA., Li S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: Proc. of IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong 2008, 1322–1328.
11. Menardi G., Torelli N. Training and assessing classification rules with imbalanced data. Data Mining and Knowledge Discovery 2014; 28: 92–122.
12. Chawla NV., Bowyer KW., Hall LO., Kegelmeyer WP. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research 2002; 16: 321–357.
13. Mery D. Aluminum Casting Inspection Using Deep Learning: A Method Based on Convolutional Neural Networks. Journal of Nondestructive Evaluation 2020; 39: 12.
14. Jiang L., Wang Y., Tang Z., Miao Y., Chen S.. Casting defect detection in X-ray images using convolutional neural networks and attention-guided data augmentation. Measurement 2021; 170: 108736.
15. Nikolić F., Štajduhar I., Čanađija M. Casting Defects Detection in Aluminum Alloys Using Deep Learning: a Classification Approach. International Journal of Metalcasting 2022.
16. Lin J., Ma L., Yao Y. Segmentation of casting defect regions for the extraction of microstructural properties. Engineering Applications of Artificial Intelligence 2019; 85: 150–163.
17. Du W., Shen H., Fu J., Zhang G., He Q. Approaches for improvement of the X-ray image defect detection of automobile casting aluminum parts based on deep learning. NDT & E International 2019; 107: 102144.
18. Yeo I-K., Johnson RA. A New Family of Power Transformations to Improve Normality or Symmetry. Biometrika 2000; 87: 954–959.
19. Zheng A., Casari A. Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. O'Reilly Media, 2018.
20. Chollet F., others. Keras. GitHub 2015. Available from: https://github.com/fchollet/keras.
21. R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing 2020. Available from: https://www.R-project.org/.
22. Fawcett T. An introduction to ROC analysis. Pattern Recognition Letters 2006; 27: 861–874.
23. Perkins NJ., Schisterman EF. The Inconsistency of "Optimal" Cutpoints Obtained using Two Criteria based on the Receiver Operating Characteristic Curve. American Journal of Epidemiology 2006; 163: 670–675.
24. Biecek P., Burzykowski T. Explanatory Model Analysis: Explore, Explain and Examine Predictive Models. Chapman and Hall/CRC, 2021.
25. Lundberg SM., Lee S-I. A Unified Approach to Interpreting Model Predictions. In: Proc. of 31st Conference on Neural Information Processing Systems (NIPS'17), Long Beach CA, USA 2017.
26. Dozat T. Incorporating Nesterov Momentum into Adam. In: Proc. of International Conference on Learning Representations (ICLR 2016), San Juan, Puerto Rico 2016.