

The Optimization of the Zinc Electroplating Bath Using Machine Learning and Genetic Algorithms (NSGA-II)

Ramazan KATIRCI^{1*}, Bilal TEKİN²

¹Sivas University of Science and Technology, Faculty of Engineering and Natural Sciences, Department of Metallurgy and Materials Engineering, Türkiye

²Sivas University of Science and Technology, Faculty of Engineering and Natural Sciences, Department of Computer Engineering, Türkiye

(ORCID: [0000-0003-2448-011X](https://orcid.org/0000-0003-2448-011X)) (ORCID: [0000-0002-6690-3152](https://orcid.org/0000-0002-6690-3152))



Keywords: Machine learning, Zinc electroplating, Genetic algorithm, Optimization, Image processing, Surface detection.

Abstract

In this study, our aim is to predict the compositions of zinc electroplating bath using machine learning method and optimize the organic additives with NSGA-II (Non-dominated Sorting Genetic Algorithm) optimization algorithm. Mask RCNN was utilized to classify the coated plates according to their appearance. The names of classes were defined as “Full Bright”, “Full Fail”, “HCD Fail” and “LCD Fail”. The intersection over union (IoU) values of the Mask RCNN model were determined in the range of 93–97%. Machine learning algorithms, MLP, SVR, XGB, GP, RF, were trained using the classification of the coated panels whose classes were detected by the Mask RCNN. In the machine learning training, the additives in the electrodeposition bath were specified as input and the classes of the coated panels as output. From the trained models, RF gave the highest F1 scores for all the classes. The F1 scores of RF model for “Full Bright”, “Full Fail”, “HCD Fail” and “LCD Fail” are 0.95, 0.91, 1 and 0.80 respectively. Genetic algorithm (NSGA-II) was used to optimize the compositions of the bath. The trained RF models for all the classes were utilized as the objective function. The ranges of organic additives, which should be used for all the classes in the electrodeposition bath, were determined.

1. Introduction

Electrodeposition is extensively utilized in the world because it is cheap and can be applied to the wide surfaces easily [1]. Many metallic coatings such as nickel, chrome, zinc, bronze, brass, copper and etc. can be performed using this method [2]–[6]. Zinc coating among the metallic coatings is the most vastly used method to protect the steels from the corrosion [7]. Two types of zinc coating baths are present. One is acidic and the other is alkaline zinc electroplating bath [7], [8]. Many parameters are present to perform the coating in the electrodeposition bath [8]. Except for parameters such as temperature and current, the other parameters are constantly changing during operation. Hence, they should be kept under control to ensure the stability of the electroplating bath.

Organic additives are very important factor to enhance the physical properties (brightness, throwing/covering power and etc.) of the coating [9]. In the industry, the lack of organic additives [10] and the other needs in the electroplating bath is estimated by the engineer using the Hull-cell panels or examining the coated parts. The defects formed on the surface of the panels during the coating are closely related to the deficiencies in the electroplating bath. Thanks to this relationship, the experienced engineer can solve the problem of the electroplating bath by examining the surface of the coated panels.

Machine learning principles have widely been used to detect objects and faces in many areas [11]. Especially, material scientists have utilized these methods to identify surface properties of the materials. For instance, in order to estimate the

*Corresponding author: ramazankatirci@sivas.edu.tr

Received: 06.09.2022, Accepted: 23.12.2022

adhesion force of the materials to the sublayer by utilizing a scratch test technique (DIN EN ISO 20502:2016–11) [12]. Wang and et al. also used CNN technique to estimate the factors influencing particles in plasma spray [13]. The method is a promising method which can be used to estimate the content of electroplating bath in the field of the electroplating process. Katırcı et al. utilized Mask RCNN method to extract the related part from the whole image and machine learning method was used to predict the quality of the Hull-cell panel [1]. Also, the same group used machine learning algorithms such as MLP (Multilayer perceptron), SVR (Support vector machine regression) and XGB (eXtreme gradient boosting) to estimate ZnNi thickness and Ni% ZnNi alloy coating [14].

The aim of this study is to adapt the artificial intelligence methods to the electroplating process to keep the variable parameters under control during operation. In this study, the Mask RCNN technique was implemented to define the deposited panels according to the appearance of the coating. To measure the additives in the electrocoating bath affecting the view of the coating, machine learning algorithms were used. Genetic algorithm (NSGA-II) was implemented to optimize the amount of organic additives in the working electroplating bath.

2. Material and Method

Zn electrodeposition was carried out utilizing a Thurlby 30V 1A-model DC model direct-current generator. The composition of the basic Zn electroplating bath is presented in Table 1 and named basic-Zn. The pH of the bath was maintained at 4.5. All electroplating was performed in the room temperature. KOH was added to dissolve ZnO and to rise the conductivity of the Zn electroplating bath. The additives were included to the bath to enhance the surface properties, such as brightness, color and etc., of the coating. Potassium silicate (PS), imidazole-epichlorohydrin (IME) and mirapol (MP) were used as carrier. Carriers ensure that the coating thickness is homogeneous [15]. The orthochloro benzaldehyde chloride (OCB) and benzyl pyridinium 3-carboxylate (BPC) chemicals were used as a brightener. The brighteners reduce the crystal size to increase in the brightness of the coating [16]. Hull-cell was used to deposit the panels; whose surface area is 1 dm² (Figure 1).

Table 1. The basic Zn electroplating bath

Chemicals	Quantity (g/L)
ZnO	12.5
KOH	170
K ₂ CO ₃	50

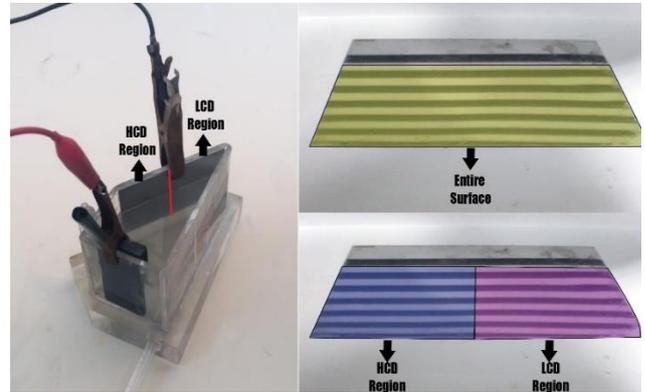


Figure 1: Hull-cell (left) and panels (right).

The images of the plates were taken to generate the dataset for training and test. The code implementations were performed on Anaconda platform and Truba HPC system. Python language was used to write the codes. Sklearn libraries were utilized for machine learning (ML). Mask RCNN codes were fitted to this study. Tensorflow 1.14 and keras 2.2.4 libraries were utilized in the Mask RCNN. The images of the Zn deposited plates were acquired and resized to 128*128 pixel. The via-2.0.10 software and polygon frame were utilized for labeling. The parts within the electrodeposition bath were labeled and classified according to the appearance of the panels. Four classes were used for classification. These are “Full Bright”, “Full Fail”, “LCD Fail” and “HCD Fail”. “Full Bright” indicates that the entire surface of the coating is bright, “Full Fail” indicates that the entire surface is either mat or has defects. “HCD Fail” and “LCD Fail” depict defects in HCD (High Current Density) and LCD (Low Current Density) regions respectively. The experimental studies were performed in 38 different electroplating baths to acquire the images from the deposited panels. The images were augmented with the image processing techniques such as blur, brightness, contrast, gaussian blur, median, salt and pepper noise, saturation and sharpen. For the test dataset, 114 “Full Bright”, 76 “Full Fail”, 114 “LCD Fail” and 114 “HCD Fail” images were generated. The Zn electrodeposition bath having the different content was trained versus the class of the coated panels using MLP (Multi-layer perceptron), SVC (Support vector

classifier), XGB (eXtreme gradient boosting), GP (Gaussian process) and RF (Random forest) machine learning algorithms. The achievement of the models was measured with F1 score. The leave-one-out (loo) method was utilized for cross-validation because it is favored for small dataset [17]. In this technique, one data is extracted from the dataset and the rest of data

is utilized for training. Afterwards, the extracted data is predicted by the model trained. This event is reiterated for each data. The hyperparameters providing the best model were investigated using the grid search technique. The flowchart of the whole study is summarized in Figure 2.

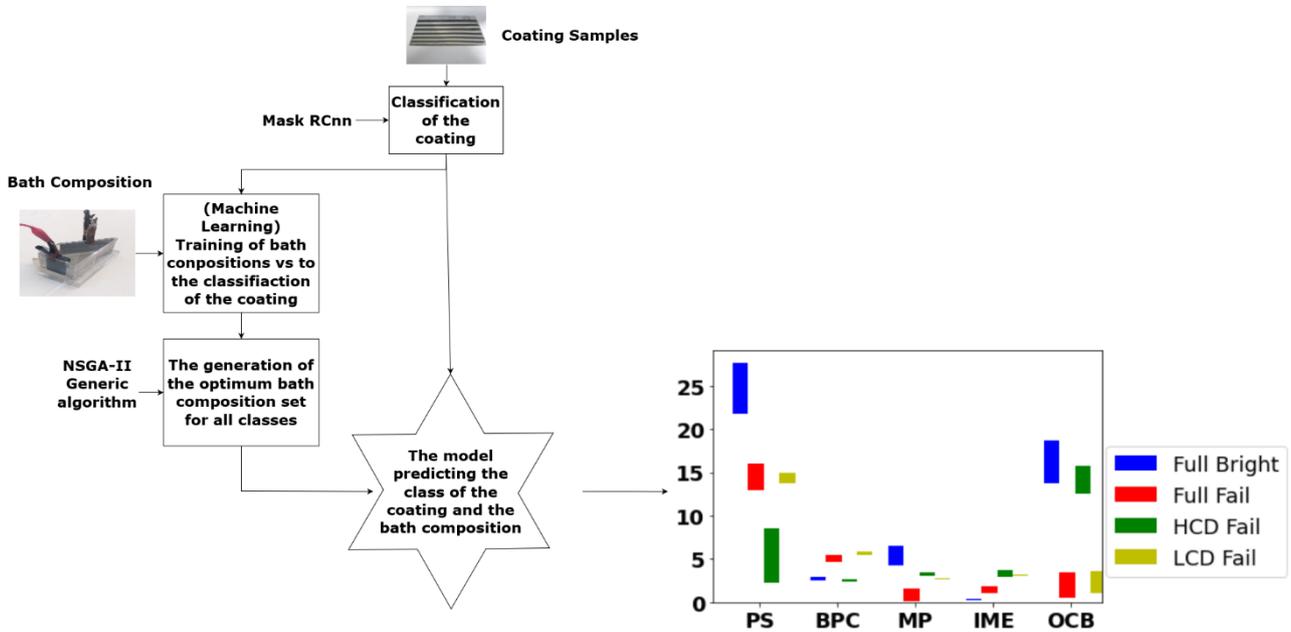


Figure 2. The flowchart of the whole study.

3. Results and Discussion

3.1. MASK RCNN

Figure 3 shows the samples coated Zn metals. The black lines on the plates are not actual but the reflection of the background. Thanks to the reflection, it is possible to define the level of the brightness of the coating. For example, in Figure 3a, the black lines on the plate are clearly seen, which means that the surface is fully bright. In Figure 3b, there are some defects and mat regions. As seen in Figure 3c, the black lines are not seen because the surface is fully mat. With this technique, it is possible to analyze the surface of the coating easily in terms of defects and brightness sequentially.



Figure 3. Zn coated panels obtained in the electroplating baths with varied compositions.

The electroplating baths are dynamic. The concentration of additives varies constantly in the operation. Therefore, it is crucial to keep them under control. The surface view of the coating provides the knowledge about the deficiencies in the electroplating bath. For example, the dullness at higher current density (HCD) region of the coated panel indicates the lack of brightener in the zinc electroplating bath. The cloudy deposit points out the metal contamination in the bath. The surface appearance of the coating was divided into 4 categories. These are “Full Bright”, “Full Fail”, “LCD Fail” and “HCD Fail”. The Mask RCNN algorithm was used to define which category the surface views belonged to. This step is crucial because these outputs were used in the machine learning algorithms to identify the related organic additives affecting the surface on the plate. The Mask RCNN algorithm detected all categories without error in the test dataset. This result is very promising to identify the additives in the electroplating bath. Figure 4 indicates the loss values of the training and validation dataset. The lower loss is expected for thriving model. Region proposal network (RPN) refers to a deep CNN technique for suggesting regions

in object recognition. ‘rpn_class_loss’ is the loss value computed during the classification, and ‘rpn_bbox_loss’ is the loss value computed during the bounding box detection. The three outputs emerge in the Mask RCNN structure: box delimiter, classification and masking. ‘mrcnn_bbox_loss’, ‘mrcnn_class_loss’, ‘mrcnn_mask_loss’ values show the errors of these outcomes. Figure 5 illustrates some examples for the ground truth and predicted mask from the test images. The results depict that the underfitting or overfitting was not observed because the trained model predicted all classes 100% correctly in the test dataset. The intersection over union (IoU) values of the object detection in the test dataset are in the range of 93–97%.

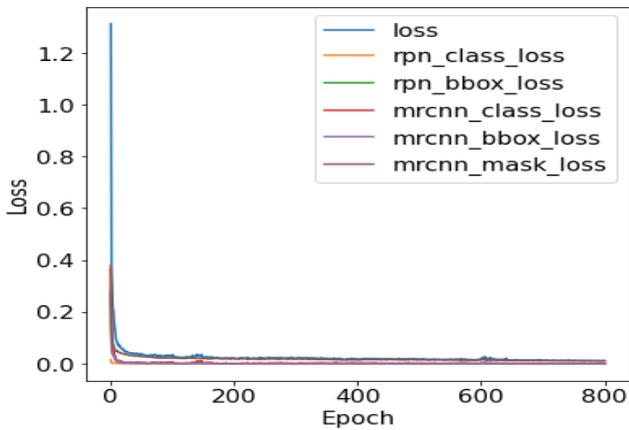


Figure 4. The loss vs. epoch number.

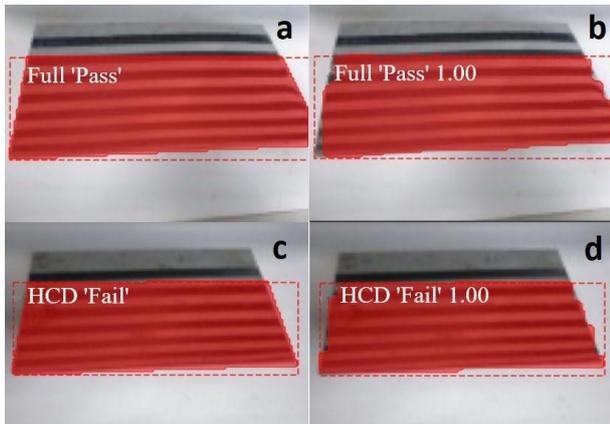


Figure 5. a) Ground truth and b) predicted mask for “Full Bright” class c) ground truth and d) predicted mask for “HCD Fail” class.

3.2 Machine Learning

Organic additives are the most important parameters in the electroplating bath because they influence the appearance of the coating directly. The concentration

of organic additives is constantly changing in the running bath, so they should be kept within a certain range. To achieve this, the machine learning algorithms are a promising technique to estimate the organic additives in the electroplating bath. Some organic additives with different concentration were added to the Hull-cell and coatings were performed in these electroplating baths. The coatings were labelled as “Full Bright”, “Full Fail”, “HCD Fail” and “LCD Fail” according to the quality of the coating. The design of experimental studies is presented in the excel file named as contents of the solution in the supplementary material. The SVC [18], [19], MLP [20], [21], XGB [22], [23], GP [24], [25] and RF [26], [27] algorithms were used to train the dataset generated from the experimental studies. The hyperparameters of all models were optimized at two levels. For example, the hyperparameter of ‘n_estimators’ in RF model were tested in 10 and 50 values. The other hyperparameters in all models were tested in the same way. The highest F1 score was used to select the best hyperparameters in all models. The hyperparameters giving the highest F1 score were printed in the txt files. The accuracy, F1 score, Precision and Recall of all the models are presented in Table 2. The model having the highest F1 score was selected as the best model. As seen in Table 2, the best model is RF model. It predicted all classes in the test dataset with high accuracy. RF model was selected for further optimization. The equations to calculate the accuracy, F1 score, Precision and Recall metrics are indicated in Eq.1. The meanings of the terms of TP, TN, FP and FN are depicted in Table 3.

Table 2. The accuracy, F1 score, Precision and Recall of all models.

	ML Methods	Accuracy	F1 score	Precision	Recall
Full Bright	MLP	0.76	0.77	0.71	0.83
	SVC	0.66	0.68	0.61	0.78
	GP	0.58	0.58	0.55	0.61
	XGBoost	0.84	0.82	0.88	0.78
	RF	0.95	0.95	0.90	1.00
Full Fail	MLP	0.60	0.48	0.00	0.62
	SVC	0.61	0.48	0.54	0.44
	GP	0.39	0.38	0.33	0.44
	XGBoost	0.58	0.60	0.50	0.75
	RF	0.92	0.91	0.88	0.94

HCD Fail	MLP	0.95	0.50	0.50	0.50
	SVC	0.95	0.00	0.00	0.00
	GP	0.95	0.00	0.00	0.00
	XGBoost	0.95	0.00	0.00	0.00
	RF	1.00	1.00	1.00	1.00
LCD Fail	MLP	0.95	0.00	0.00	0.00
	SVC	0.95	0.00	0.00	0.00
	GP	0.95	0.00	0.00	0.00
	XGBoost	0.95	0.00	0.00	0.00
	RF	0.97	0.80	0.67	1.00

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

$$F1\ score = \frac{2 * Precision * Recall}{(Precision + Recall)}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{1}$$

$$Recall = \frac{TP}{(TP + FN)}$$

Table 3. The meanings of the terms of TN, TP, FP and FN.

Confusion Matrix		Y-predicted	
		0	1
Y-true	0	True Negative (TN)	False Positive (FP)
	1	False Negative (FN)	True Positive (TP)

Recall indicates that how much of the proportion of true positives (TP) is described correctly. Precision shows how much of the ratio of predicted positives are actually 1. F1 score is the weighted average of precision and recall values [28], [29]. The confusion matrix of all models is presented in CM excel file in the supplemental materials. In the next step, the main and interaction effects of hyperparameters of RF model were investigated. Figure 6 shows the main effects of hyperparameters

for “Full Bright” class in RF. As seen in figure 6, ‘n_estimators’, criterion and ‘warm_start’ is the most efficient hyperparameters affecting the F1 score. While ‘n_estimators’ and ‘warm_start’ hyperparameters indicates the enhancing effect, ‘criterion’ hyperparameter depicts the decreasing effect. It is significant to examine the interaction effects of these hyperparameters to each other. Figure 7 indicates the interactions of hyperparameters. It is seen from the figure that the small interaction among ‘n_estimators’, ‘warm_start’ and ‘criterion’ hyperparameters are present. An important point which can be seen in this figure is that ‘warm_start’ hyperparameter indicates robust effect. When it is kept in True, the influence of ‘n_estimators’ and ‘criterion’ disappear and the maximum F1 score is acquired. The similar results were observed in other classes. In all the classes, ‘warm_start’ emerged as the most important hyperparameter. The task of ‘warm_start’ hyperparameter in the algorithm is that when ‘warm_start’ hyperparameter set to True, the previous solution is used to fit the trees. Otherwise, it fits a whole fresh forest. The warm start hyperparameter is an effective way to add more trees until the training reaches a satisfying accuracy [30]. The optimum hyperparameters for all classes in the RF model to acquire the highest F1 score are presented in Table 4.

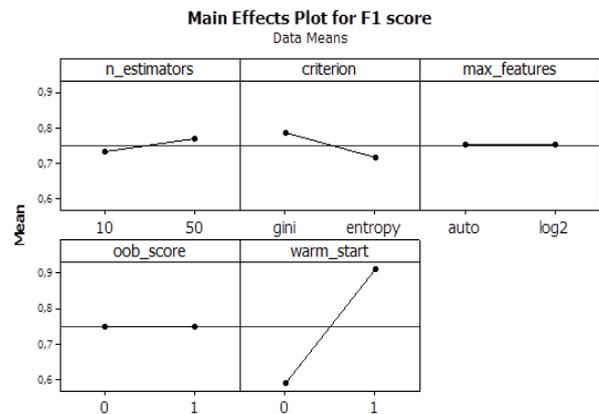


Figure 6: The main effects of hyperparameter for “Full Bright” class.

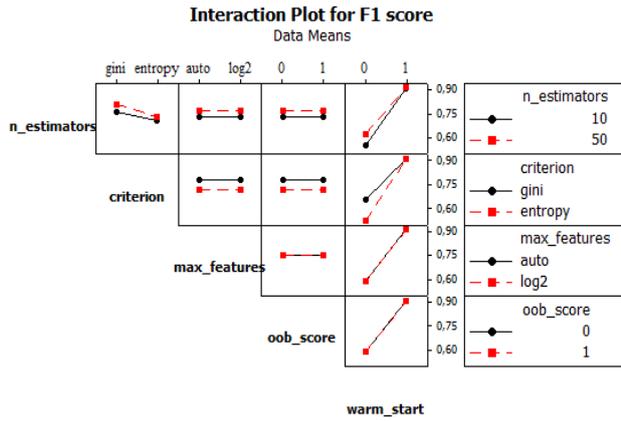


Figure 7: The interaction of hyper parameters for “Full Bright” class.

Table 4: The optimum hyper parameters for all classes in RF model.

Classes	Hyperparameters
Full Bright	n_estimators: 10, criterion: gini, max_depth: None, min_samples_split: 2, min_samples_leaf: 1, min_weight_fraction_leaf: 0.0, max_features: sqrt, max_leaf_nodes: 17, min_impurity_decrease: 0.0, bootstrap: False, oob_score: False , warm_start: True, class_weight: None, ccp_alpha: 0.0, max_samples: None
Full Fail	n_estimators: 10, criterion: gini, max_depth: None, min_samples_split: 2, min_samples_leaf: 1, min_weight_fraction_leaf: 0.0, max_features: sqrt, max_leaf_nodes: None, min_impurity_decrease: 0.0, bootstrap: True, oob_score: False , warm_start: True, class_weight: None, ccp_alpha: 0.0, max_samples: None
HCD Fail	n_estimators: 10, criterion: entropy, max_depth: None, min_samples_split: 2, min_samples_leaf: 1, min_weight_fraction_leaf: 0.0, max_features: sqrt, max_leaf_nodes: None, min_impurity_decrease: 0.0,

bootstrap: False, oob_score: False , warm_start: True, class_weight: None, ccp_alpha: 0.0, max_samples: None

LCD Fail n_estimators: 10, criterion: gini, max_depth: None, min_samples_split: 2, min_samples_leaf: 1, min_weight_fraction_leaf: 0.0, max_features: sqrt, max_leaf_nodes: None, min_impurity_decrease: 0.0, bootstrap: True, oob_score: False , warm_start: True, class_weight: None, ccp_alpha: 0.0, max_samples: None

3.3 Optimization

NSGA-II (Non-dominated Sorting Genetic Algorithm) algorithm were utilized to optimize the electroplating bath composition. The NSGA-II algorithm is a multi-objective optimization algorithm based on non-dominated sorting and crowding proposed by Deb et al [31], [32]. It was designed based on genetic algorithm. The algorithm was created to eliminate the shortcomings of the NSGA algorithm developed by Srinivas and Deb [33]. In addition to the steps of the genetic algorithm, non-dominated sorting and crowding distance calculations are performed. Since NSGA-II is an algorithm with low computational complexity and fast, it has many applications in the literature [34]–[36]. After the bath composition of electroplating bath was trained versus the appearance of the surface, the best models, having the highest F1 score, were used as the objective function in the optimization algorithm. The optimization problem is generally defined as follows.

$$\begin{aligned}
 \min f_m(x) & \quad m = 1, \dots, M \\
 g_j(x) \leq 0 & \quad j = 1, \dots, J \quad \text{Inequality constraints} \\
 h_k(x) = 0 & \quad k = 1, \dots, K \quad \text{Equality constraints} \\
 x^L &= [x_1, x_2, \dots, x_i] \\
 x^U &= [x_1, x_2, \dots, x_i] \\
 x &\in \Omega
 \end{aligned}$$

where x_i shows the i -th variable to be optimized, x_i^L and x_i^U depict the upper and lower bounds. $f_m(x)$, $g_j(x)$ and $h_k(x)$ illustrate the m -th objective function, j -th inequality constraint and k -th equality constraint respectively [37]. The models generated separately for “Full Bright”, “Full Fail”,

“HCD Fail” and “LCD Fail” classes were used as the objective function computing the output. In the “Full Bright” objective function, the maximum was defined as the best. No constraint was determined. The problem definition was made as follows,

$$\begin{aligned} \max f_m(x) & \quad \text{models generated} \\ & \quad \text{in machine learning} \\ g_j(x) & \quad \text{no constraint} \\ h_k(x) & \quad \text{no constraint} \\ x_l = [1, 0, 0, 0, 0, 0] & \quad \text{lower bound} \\ x_u = [2, 30, 6, 10, 5, 20] & \quad \text{upper bound} \end{aligned}$$

The bath composition ranges for all classes were determined and their results are illustrated in Figure 8. As shown in the figure, to acquire the “Full Bright” surface, PS, BPC, MP, IME and OCB materials should be kept in the ranges of 21.8-27.7, 2.5-2.9, 4.3-6.5, 0.2-0.4 and 13.8-18.7 respectively. OCB and BPC chemical amounts are overlapped in “HCD Fail” class, so the amount of PS chemicals should not fall below 20 and IME chemical should not exceed 0.5. The composition of the bath can be estimated from this figure showing the ranges of the bath compositions versus the class of the coating, which is determined by Mask RCNN algorithm.

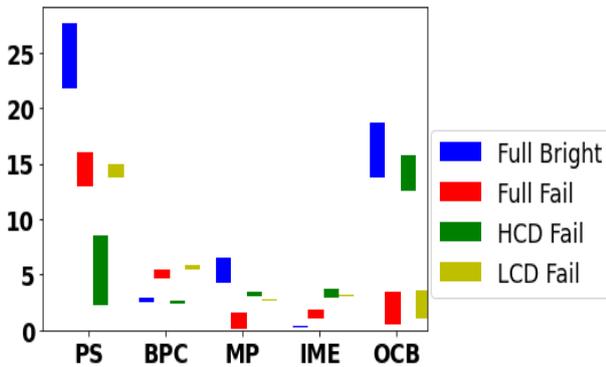


Figure 8: Optimized electroplating bath composition for all classes.

4. Conclusion and Suggestions

In this study, the optimum electroplating bath composition for “Full Bright” surface was investigated. In the first step, Mask RCNN were utilized to classify the coating, which was evaluated for the four classes, “Full Bright”, “Full Fail”, “HCD Fail” and “LCD Fail”. These classes were generated in accordance with the appearance of the coating, which were affected by the organic additives in the

electrodeposition bath. In the second step, the electroplating bath compositions were trained versus the classes of the coating. The scope of this step is to generate the model predicting the class of the coating from the bath composition. RF algorithm was detected as the best model for all the classes. The hyperparameters of RF model were further optimized to increase the F1 score. The optimum RF models for each class was generated. These models were used as the objective function in the NSGA-II optimization process. In the last step, the bath composition ranges for all classes were found by using NSGA-II algorithm. The flow chart summarizing the whole study is presented in Figure 2.

Consequently, thanks to this study, it has been possible to predict the content of the electroplating bath from the appearance of the coating. The suggested method indicates that it is possible to keep the organic additives in the electroplating process under control using artificial intelligent methods. If this study is implemented in the industry, it is expected that time and operating cost decrease abruptly. Also, the process will be automated. By extending the number of class and samples in the dataset, the achievement in estimating the content of the electroplating bath, ie organic additives, can be increased.

Acknowledgment

The experiments reported in this paper were partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources) and some computing resources were provided by the National Center for High Performance Computing of Türkiye (UHem).

Contributions of the Authors

Ramazan Katırcı: Conceptualization, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Visualization, Resources.

Bilal Tekin: Software, Investigation, Validation, Data curation, Visualization.

Conflict of Interest Statement

There is no conflict of interest between the authors.

Statement of Research and Publication Ethics

The study is complied with research and publication ethics.

References

- [1] R. Katirci, E. K. Yilmaz, O. Kaynar, and M. Zontul, "Automated evaluation of Cr-III coated parts using Mask RCNN and ML methods," *Surf. Coat. Technol.*, vol. 422, no. 127571, p. 127571, Sep. 2021.
- [2] E. Sezer, B. Ustamehmetoğlu, and R. Katirci, "Effects of a N,N-dimethyl-N-2-propenyl-2-propene-1-ammonium chloride-2-propenamide copolymer on bright nickel plating," *Surf. Coat. Technol.*, vol. 213, pp. 253–263, Dec. 2012.
- [3] R. Katirci and U. Yilmaz, "Statistical studies of Zn–Ni alloy coatings using non-cyanide alkaline baths containing polyethyleneimine complexing agents," *Transactions of the IMF*, vol. 92, no. 5, pp. 245–252, Sep. 2014.
- [4] R. Katirci, "A chrome coating from a trivalent chromium bath containing extremely low concentration of Cr³⁺ ions," *Int. J. Surf. Sci. Eng.*, vol. 10, no. 1, pp. 73–85, Jan. 2016.
- [5] L. N. Bengoa, W. R. Tuckart, N. Zabala, G. Prieto, and W. A. Egli, "Bronze electrodeposition from an acidic non-cyanide high efficiency electrolyte: Tribological behavior," *Surf. Coat. Technol.*, vol. 253, pp. 241–248, Aug. 2014.
- [6] Z. Lai et al., "Computational analysis and experimental evidence of two typical levelers for acid copper electroplating," *Electrochim. Acta*, vol. 273, pp. 318–326, May 2018.
- [7] M. Kul, S. U. of Science, and 58050 Sivas Turkey Technology Department of Aeronautical Engineering, "Effect of process parameters on the electrodeposition of zinc on 1010 steel: Central composite design optimization," *Int. J. Electrochem. Sci.*, vol. 15, pp. 9779–9795, Oct. 2020.
- [8] M. Chotirach et al., "Systematic investigation of brightener' s effects on alkaline non-cyanide zinc electroplating using HPLC and molecular modeling," *Mater. Chem. Phys.*, vol. 277, no. 125567, p. 125567, Feb. 2022.
- [9] N. Sorour, W. Zhang, E. Ghali, and G. Houlachi, "A review of organic additives in zinc electrodeposition process (performance and evaluation)," *Hydrometallurgy*, vol. 171, pp. 320–332, Aug. 2017.
- [10] R. Katirci, E. Sezer, and B. Ustamehmetoğlu, "Statistical optimisation of organic additives for maximum brightness and brightener analysis in a nickel electroplating bath," *Transactions of the IMF*, vol. 93, no. 2, pp. 89–96, Mar. 2015.
- [11] K. Hameed, D. Chai, and A. Rassau, "Score-based mask edge improvement of Mask-RCNN for segmentation of fruit and vegetables," *Expert Syst. Appl.*, vol. 190, p. 116205, Mar. 2022.
- [12] B. Lenz, H. Hasselbruch, A. G. Holger, and Mehner, "Application of CNN networks for an automatic determination of critical loads in scratch tests on a-C:H:W coatings," *Surf. Coat. Technol.*, vol. 393, p. 125764, Jul. 2020.
- [13] J. Zhu, X. Wang, L. Kou, L. Zheng, and H. Zhang, "Prediction of control parameters corresponding to in-flight particles in atmospheric plasma spray employing convolutional neural networks," *Surf. Coat. Technol.*, vol. 394, p. 125862, Jul. 2020.
- [14] R. Katirci, H. Aktas, and M. Zontul, "The prediction of the ZnNi thickness and Ni % of ZnNi alloy electroplating using a machine learning method," *Transactions of the IMF*, vol. 99, no. 3, pp. 162–168, May 2021.
- [15] M. P. M. Schlesinger, *Modern Electroplating*. Elsevier, 2011.
- [16] Y. Nakamura, N. Kaneko, M. Watanabe, and H. Nezu, "Effects of saccharin and aliphatic alcohols on the electrocrystallization of nickel," *J. Appl. Electrochem.*, vol. 24, no. 3, Mar. 1994.
- [17] A. Kirtis, M. Aasim, and R. Katirci, "Application of artificial neural network and machine learning algorithms for modeling the in vitro regeneration of chickpea (*Cicer arietinum* L.)," *Plant Cell Tissue Organ Cult.*, Mar. 2022.
- [18] T. V. Dinh, H. Nguyen, X.-L. Tran, and N.-D. Hoang, "Predicting rainfall-induced soil erosion based on a hybridization of adaptive differential evolution and support vector machine classification," *Math. Probl. Eng.*, vol. 2021, Feb. 2021.
- [19] A. Rizwan, N. Iqbal, R. Ahmad, and D.-H. Kim, "WR-SVM model based on the margin radius approach for solving the minimum enclosing ball problem in support vector machine classification".
- [20] Y. Wei, J. Jang-Jaccard, F. Sabrina, A. Singh, W. Xu, and S. Camtepe, "AE-MLP: A hybrid deep learning approach for DDoS Detection and Classification," *IEEE Access*, vol. 9, pp. 146810–146821, 2021.

- [21] M. Ramkumar, C. G. Babu, K. V. Kumar, D. Hepsiba, A. Manjunathan, and R. S. Kumar, "ECG Cardiac arrhythmias Classification using DWT, ICA and MLP neural networks," *J. Phys. Conf. Ser.*, vol. 1831, no. 1, p. 12015, Mar. 2021.
- [22] M. Ma et al., "XGBoost-based method for flash flood risk assessment," *J. Hydrol.*, vol. 598, p. 126382, Jul. 2021.
- [23] D. Zhang et al., "iBLP: An XGBoost-based predictor for identifying bioluminescent proteins," *Comput. Math. Methods Med.*, vol. 2021, p. 6664362, Jan. 2021.
- [24] C. Villacampa-Calvo, B. Zaldivar, E. C. Garrido-Merchán, and D. Hernández-Lobato, "Multi-class Gaussian process classification with noisy inputs," *J. Mach. Learn. Res.*, vol. 22, no. 36, pp. 1–52, 2021.
- [25] B. Gips, "Texture-based seafloor characterization using gaussian process classification," *IEEE J. Oceanic Eng.*, pp. 1–11, 2022.
- [26] Y. Chen, W. Zheng, W. Li, and Y. Huang, "Large group activity security risk assessment and risk early warning based on random forest algorithm," *Pattern Recognit. Lett.*, vol. 144, pp. 1–5, Apr. 2021.
- [27] K. Liu, X. Hu, H. Zhou, L. Tong, W. D. Widanage, and J. Marco, "Feature analyses and modeling of lithium-ion battery manufacturing based on random forest classification," *IEEE/ASME Trans. Mechatron.*, vol. 26, no. 6, pp. 2944–2955, Dec. 2021.
- [28] B.-C. Yan, H.-W. Wang, S.-W. F. Jiang, F.-A. Chao, and B. Chen, "Maximum f1-score training for end-to-end mispronunciation detection and diagnosis of 12 english speech".
- [29] B. Ćwiklinski, A. Giełczyk, and M. Choraś, "Who Will Score? A machine learning approach to supporting football team building and transfers," *Entropy*, vol. 23, no. 1, Jan. 2021.
- [30] F. Pedregosa et al., "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [32] E. D. Durmaz and R. Sahin, "NSGA-II and goal programming approach for the multi-objective single row facility layout problem," *Journal of the Faculty of Engineering and Architecture of Gazi University*, vol. 32, no. 3, pp. 941–955, 2017, doi: 10.17341/gazimmfd.337647.
- [33] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, Sep. 1994, doi: 10.1162/evco.1994.2.3.221.
- [34] A. Ala, F. E. Alsaadi, M. Ahmadi, and S. Mirjalili, "Optimization of an appointment scheduling problem for healthcare systems based on the quality of fairness service using whale optimization algorithm and NSGA-II," *Sci. Rep.*, vol. 11, no. 1, p. 19816, Oct. 2021.
- [35] S. Verma, M. Pant, and V. Snasel, "A Comprehensive review on NSGA-II for multi-objective combinatorial optimization problems," *IEEE Access*, vol. 9, pp. 57757–57791, 2021.
- [36] P. Wang, J. Huang, Z. Cui, L. Xie, and J. Chen, "A Gaussian error correction multi-objective positioning model with NSGA-II," *Concurr. Comput.*, vol. 32, no. 5, Mar. 2020.
- [37] J. Blank and K. Deb, "Pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89497–89509, 2020.