Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

School of Engineering and Sciences

Surface defect detection with predictive models in the galvanizing process

By Baruc Emet Pérez Benítez

Submitted to the
School of Engineering and Sciences,
in partial fulfillment of the requirements for the degree of

Master of Science In Engineering Sciences

Monterrey, México, December 4th of 2020

Dedication

For my family, who make the effort to keep supporting me in this journey towards my professional preparation. Despite the distance, they were always present and supportive.

For my partner-in-life, Juan Carlos García Valenzuela. You, more than nobody else, gave me the strength, the confidence and the courage to keep pushing forward. You are my advisor, and my rock. Thanks for your unconditional support.

For my three little dogs, Poncho, Totopo and Oliva. You transform the most stressful afternoon into moments of joy and happiness. Thank you.

Acknowledgements

Thanks to Dr. José Luis Preciado Arreola, who believed in me since the very beginning and gave me the opportunity to start this journey. Thanks not only for your technical advice on the area, but for your friendship and support.

Thanks to my friend and master's colleague Diego Villarreal Garza, who always covered my back and was always there to support me.

Thanks to the committee members of this thesis: M.Sc. Carlos Arnoldo Chee González and Dr. Víctor Gustavo Tercero Gómez, for your advice and feedback needed to round up this thesis.

Thanks to the institutions that granted me the scholarships that allowed me to do my master's studies: to Tecnológico de Monterrey for the full tuition scholarship and Consejo Nacional de Ciencia y Tecnología (CONACyT) for the support for living along the past two years.

Surface defect detection with predictive models in the galvanizing process

by

Baruc Emet Pérez Benítez

Abstract

Hot-dip galvanizing is a widely used process worldwide to provide metal products with a protective layer that enhances its corrosion resistance. The effectiveness of such layer relies on the uniformity of the coverage, thus, any alteration in the galvanizing layer may be considered as a defect. These defects are catalogued as surface defects where two groups are identified: Bare Spots and Dross-Derived defects. Currently, these defects are detected at the end of the line where no preventive actions can be performed. Consequently, the surface defects' occurrence is not avoided, increasing in turn the expenses of the company. For that reason, a project oriented to these defects' prediction is proposed.

This project consists on a set of predictive models, which are tested to be able to predict these defects' occurrence at an early stage that let the people of the galvanizing line to design and unleash preventive actions that could alleviate the surface defects' incidents. Four models are studied: Stepwise Logistic Regression, Random Forest Classifier, Gradient Boosting Classifier, and Low FNR Low FPR Random Forest Classifier (LFNR-LFPR RFC) ensemble. LFNR-LFPR RFC is a custom-made multi-objective ensemble designed in this project, which basic learners are two Random Forest Classifiers. To test the models' performance, the False Negative Rate (FNR) and False Positive Rate (FPR) scores are employed, where the acceptance criteria is to at most have a 15% of FNR and a 25% FPR. From the models tested, LFNR-LFPR RFC was able to outperform the others while achieving FNR and FPR scores under the acceptance criteria for most of the studied cases (two out of three for Bare Spots and one out of two for Dross-Derived defects). Furthermore, the importance of the variables selected for the LFNR-LFPR RFC model was evaluated. As a result, variables from different sources, such as the galvanizing line per se, the chemistry of the coil and from upstream processes, were obtained. In turn, these lists of variables can provide insights on how to design preventive actions that could decrease the surface defects' occurrence. Finally, the economic impact of the defects and the predictive models is assessed, where, according to the LFNR-LFPR RFC ensemble's results, savings are possible.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Background

Hot-dip galvanizing is a process where a metal product, such as steel or iron, is covered with a zinc-based coating that enhances the corrosion resistance of the material (Sahoo & Paulo, 2017). This protection allows the product to resist the environmental conditions at which it will be exposed, without changing its mechanical properties (Jiang, 2004; Azimi, 2012b); thus, assuring a correct performance of the metal product for a prolonged period of time. Since Tadeusz Sandzimir patented it in 1934, the practice of hot dip galvanizing has grown significantly, to the point that nowadays there are more than 600 continuous galvanizing lines around the world producing approximately 100 million tons of galvanized steel products per year (Ajersch, 2017). Industries like the automotive, construction and domestic appliances have taken advantage of galvanized steel materials to enhance the quality of their final products. For instance, galvanized high strength steel coils are of great interest for the automotive industry due to their lightweight, high impact endurance, and, thanks to the galvanizing layer, high corrosion resistance (Wu, 2018; Ajersch, 2017), which help the manufacturers to build safe and lasting transports.

The galvanizing layer provides the steel products with a three-fold protective layer, which consists first on a physical layer, the galvanizing layer per se, that isolates the steel sheet from its surroundings; then, a zinc's sacrificial protection, where corrosion happens on zinc instead on the steel; and finally, a natural layer created by the same weathering process at which the galvanized steel product is exposed (Sahoo & Paulo, 2017). This three-fold protection depends on a smooth and full coverage of the steel material with the galvanizing layer; otherwise, the desired corrosion resistance will not be met. Following this idea, it is important to galvanized products manufacturers to apply high-quality standards to their processes to warranty a uniform coverage in the final products. In fact, according to various authors, what makes a galvanized products manufacturer competitive in the current market is the ability of applying uniform galvanizing, that is to say, to produce non-defective products (Azimi, 2012b; Luo & He, 2016; Wu, 2018).

Any alteration on the galvanizing layer uniformity is considered as a defect, which are usually referred as surface defects due to their appearance on the surface of the galvanized steel product (Luo & He, 2016; Ajersch, 2017). Depending on the morphology and physical characteristics of the surface defects, they are classified into two categories: Bare Spots and Dross-derived defects. Bare spots referred to those areas in the galvanizing layer where the zinc coating is missing, and the causes are the presence of contaminants in the steel product before being immersed in the galvanizing bath, producing a poor adhesion, or mechanical damage on the product, provoking the coating to fall (Azimi, 2012b; Wu, 2018). On the other hand, dross-derived defects (also named as pimples, black spots, dross inclusions, lumps, among others), emerge as swelled spots in the galvanizing layer that are composed of ashes, dross particles and contaminants coming from upstream processes or the galvanizing line per se (Azimi, 2012a; Ajersch, 2017). An example of these defects is presented in Figure 1. Surface defects, far for representing an aesthetic defect, can damage the corrosion resistance that the galvanizing layer confers to the steel product. Both defects

damages have been evaluated by Azimi et al. (2012b) with a standard salt spray test, which speeds up the natural oxidation process in metal products, determining that bare spots decrease the corrosion resistance by 39% ± 1%, and dross-derived defects by 10% ± 1%. Such effects are not desirable in the finished products, since they can compromise their minimum quality-standards, and special attention must be paid to design and develop solutions that avoid their presence.

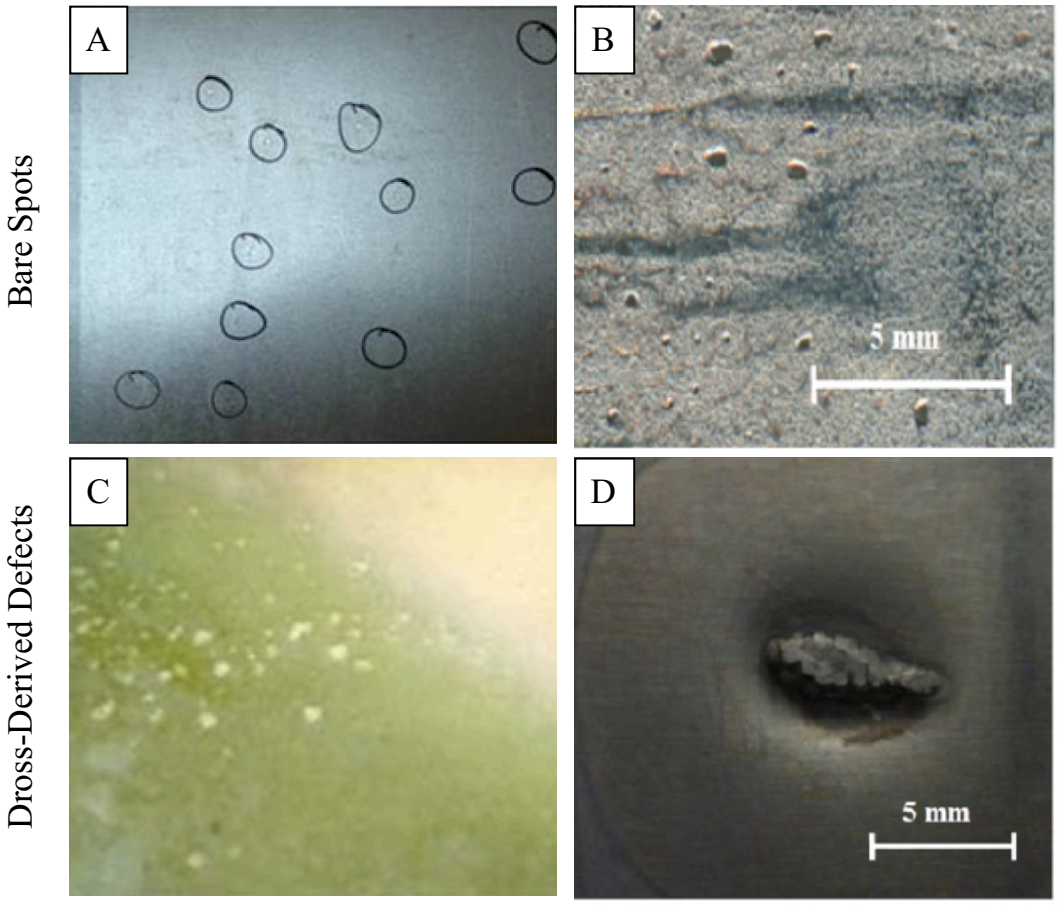

Figure 1. Bare Spots and Dross-Derived defects.
A) Bare spots in a steel coil section; B) bare spots microscopic image; C) dross-derived defects in a steel coil section; D) dross-derived defects microscopic image. B & D are taken from the work by Azimi, A. et al. (2012b).

Nowadays, continuous efforts from academia and industry on these defects' characterization have been focused on understanding their possible causes, such as the effects of the temperature at different locations in the galvanizing line, or the chemistry composition of the galvanizing bath (Bellhouse & McDermid, 2008; Li & Sun, 2014; Jiang, 2014); and the design of modifications in the galvanizing line that can decrease their appearance (Félix, 2013; Dubois & Bordignon, 2018; Sirin, 2020). However, the occurrence of these surface defects remains prevalent, and new research avenues have to be explored to come up with new solutions, e.g. the prediction of these defects' occurrence with machine learning models. Some approaches using this type of models are on the line of surface defects recognition using image classification with machine learning models (Luo, 2020). On one hand, these models decrease the probability of letting pass a defective product, enhancing the quality standards of the company. On the other hand, these models are still not taking any kind of action to reduce the occurrence of these defects. This raises the concern that

there is not an avenue to unleash preventive actions to avoid these defects. For that reason, the main objective of this work is to design and develop a machine learning model, specifically an ensemble model, capable of predicting the occurrence of surface defects on an early stage in the galvanizing line.

An ensemble model is a combination of a set of learners, which are classification or regression models, that are trained to solve a problem (Zhou, 2012). A basic illustration of the steps an ensemble model follows is presented in Figure 2. There is sufficient evidence on the literature that ensemble models are able to outperform a single model in prediction accuracy (Zhou, 2012; Li & Sun, 2014; Ren & Zhang, 2016), being the key behind these ensembles, the introduction of diversity among their base learners. In other words, diversity means that it is desired to have uncorrelated learners in the ensemble, to assure a variety of solutions that permits to explore a wider area of the hypothesis space in comparison of a single model, which in turn increases the chances to reach the right solution (Dietterich, 2000; Ren & Zhang, 2016). According to Ren & Zhang (2016) there are six strategies to earn diversity on an ensemble:

- Data Diversity: To partition the original dataset into training subsets to obtain a variety of solutions for a problem.
- Parameter Diversity: To test a set of parameters on the learners to obtain different predictors.
- Structural Diversity: To try different architectures of the predictors.
- Divide and conquer: To divide the problem into subtasks that each learner has to solve.
- Multi-Objective Optimization: To include more than one predictor in the problem's solution.
- Fuzzy ensemble: To apply fuzzy logic in the training process of the learners.

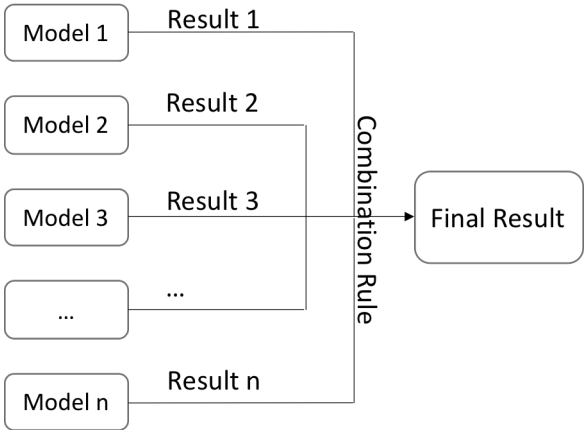

Figure 2. Basic functioning of an ensemble model.
First, the individual learners are train with a dataset or a subset coming from the original dataset. Then, each learner makes a decision, which are further fuse together into one final result using a combination rule.

Usually, an ensemble model incorporates more than one diversity strategy into its design, increasing the chances to improve the accuracy of a single model. In addition to these characteristics, ensembles use combination rules to combine the individual decisions of the learners to provide a final answer. These rules aids on the reduction of bias and variance that the

learners may have individually (Zhou, 2012). There are several options that serve as combination rules such as majority voting (the most repeated solution is chosen as the final one), winner-takes-all, (the best solution is chosen without taking into account the others), weighted average (weighting the decisions of each learner differently to construct the final one), among others (Zhou, 2012). These characteristics of low variance, low bias and high diversity make ensemble models good candidates to solve classification problems in a safe and supervised way.

This project results from a collaboration between the people of data science from the University of Tecnológico de Monterrey and the data science team and the galvanizing engineers from a metallurgical company. This company is focused on the production of steel materials used in diverse industries such as the automotive, industrial, domestic appliance, and construction fields, comprising a products' portfolio of coils, wires, profiles, among others. In order to manufacture these products, the company owns different production lines that makes feasible the production of each steel material attending the market's demands. The specific object of study in this thesis is the galvanized steel coils produced in the company's galvanizing line. A more detailed description of the problem, along with the main characteristics of this project are presented below.

1.2 Problem Statement

During the galvanizing line of this company, two defects known as surface defects (i.e., bare spots and dross-derived defects) are still present despite the current quality control processes. These defects impact directly the corrosion resistance characteristic that the galvanizing layer confers to the steel products. The issue is that, since the inspection of the steel materials for such defects is manual, there are cases where defective materials pass through the quality control assurance station in the line unnoticed, hence, arriving to the final client, which jeopardizes the relationship with the client and the company's image. Moreover, the quality assurance station, with the current design of the galvanizing line, is located at the end of the line, making no place to detect defective products at early stages of the line. Then, defective materials, whether detected or not, are expensive for the company since there is no way to stop the line once the product presenting a defect, to avoid the cost of the posterior industrial processes, or to provide insights on possible modifications on the line to prevent the occurrence of such defects.

Therefore, given the characteristics of this problem, the ability to predict the occurrence of these surface defects in the galvanizing line represents a promising solution, not only because it will permit to save money but also improve the product's quality. In addition, this prediction, if occurring at early stages in the line, will provide enough time and insights for the experts of the process to design and implement preventive actions that could decrease their appearance on their products.

1.3 Research Questions

According to the characteristics of this project, and based on the lack of classification models employed to predict the occurrence of surface defects in the galvanizing line (see Section 2.1 for more details), the following research questions are stated:

- Is there an off-the-shelf binary classification model capable of reaching a low predictive False Negative Rate (FNR) and False Positive Rate (FPR) performance for the bare spots and dross-derived defects in the galvanizing process?
- To what extent can a multi-objective ensemble model based on one of these classifiers outperform their single-objective version?
- Is the selected binary classification model capable to predict the occurrence of these defects at early stages of the galvanizing line to provide a sufficient time gap to perform preventive actions?
- Is the selected binary classification model capable to identify actionable variables that presents a role over these surface defects occurrence in the galvanizing line?

To find a binary classification model that solve these questions is no easy task; however, if answered properly, they will improve the current company's panorama related to these two surface defects of the galvanizing line.

1.4 Research Hypothesis

The hypothesis of this thesis is that using a custom-made binary classification multi-objective ensemble model, specific to each surface defect of the galvanizing line, can predict the defects' occurrence at an early stage on the line. Moreover, such model's performance surpasses that of off-the-shelf approaches reaching a low predictive FNR and FPR. Finally, the model can provide insights on actionable variables involved in the surface defects' occurrence for the creation of preventive actions.

1.5 Objectives

1.5.1 General Objective

The general objective of this project is to develop a binary classification model for each surface defect (i.e., bare spots and dross-derived defects) of the galvanizing line, whose predictions occur at early stages of the line with a performance that complies with the company's standards, while, at the same time, provides a sorted list of important and actionable variables who has an impact on the problem, and who will aid on the design of preventive actions to decrease their incidences.

1.5.2 Specific Objectives

To accomplish the general objective, the following specific objectives are identified.

- Design and fit binary classification models for surface defects' occurrence prediction, comparing the performances of off-the-shelf approaches with a custom-made ensemble model to assess the best solution in terms of the company's acceptance criteria involving the False Negative Rate (FNR) and False Positive Rate (FPR) scores.
- Assess the performance of the best model at different stages of the line to select an early detection version of it, to enhance the potential economic value delivered to the company.

- Detect actionable and important variables that have a role over the surface defects' occurrence to provide insights of possible line modifications and adjustments that will avoid their appearance, i.e. the design of preventive actions in the galvanizing line.

1.6 Scope and Limitations

A challenge faced in this project, which can improve over time, is the short time span that comprises the dataset in combination with their low occurrence, that is to say, the degree of imbalance of the population is high. Hence, the performance results obtained with the predictive models analyzed in this project are limited to the available records, from August 2019 to February 2020. Moreover, no further inquiries on the validity of the target variable, the indicator of the surface defects' presence, are considered. Finally, results of the predictions online performed by the company to evaluate the performance of the models live, that is to say, to predict the occurrence of defects in completely new steel coils, are not included in this work.

1.7 Expected Results

The final product of this project is a binary classification model or a set of models that are capable of predicting, at early stages of the galvanizing line, the occurrence of surface defects with a minimum performance of 15% False Negative Rate (FNR) and 25% False Positive Rate (FPR), which are the performance scores employed as requested by the company (explained in following chapters). In addition, the model must be able to provide a sorted list of important and actionable variables that can support the design of preventive actions by the engineers of the company. Finally, the selected model has to outperform common off-the-shelf approaches used in this type of problems: Single-Objective Random Forest and Gradient Boosting Classifiers.

1.8 Motivation

There are two grand motivations for this project. On one hand, the market of galvanized materials is highly competitive, where the company who can attain high quality standards, that is to say, to produce non-defective galvanized materials, has more chances to increase its sales. That is the case of the company participating in this project, which seeks to improve their quality control processes to enhance the customer service they currently have and position itself internationally. On the other hand, talking about surface defects' occurrence prediction in the galvanizing line, there are no records of efforts from the Industry or the Academia to design and implement predictive models in this subject. Then, a binary classification model that is capable to attend all the company's requirements will represent a novelty not only on the metallurgical area but also in the ensemble models trained with high dimensional and imbalanced datasets, i.e. a complex problem to solve.

The organization of this document is as follows: Chapter 2 contains a literature review of related work on surface defects prediction and ensemble models employed for classification purposes; Chapter 3 contains the description of the methodology; Chapter 4 explained in detail the experimental case use to train and test the proposed predictive models; Chapter 5 provides the results and their discussion; and Chapter 6 gives the conclusions and future work.

Chapter 2

Literature Review

This chapter contains the literature review on two topics: 1) the development of a predictive model of surface defects on galvanized steel products, and 2) the design of a low false negative and low false positive custom made non-parametric ensemble model. For each of these topics, the related work is presented to identify the strengths and weaknesses of the proposed ensemble model of this project, along with its application in surface defects prediction in a galvanizing line.

2.1 Surface defects prediction

For surface defects prediction, in order to picture the current landscape on the advancements on the subject, a literature review is conducted. Such review is performed in Scopus, which is a specialized scientific literature database that puts together more than 75 million items that have been published since 1970 (Elsevier, 2020), with the following search query:

TITLE-ABS-KEY(galvaniz* AND (("superficial defect" OR "bare spot" OR "dark spot" OR "black spot" OR (dross AND defect) OR (pimple AND defect)))).

This query was constructed by including several synonyms of the surface defects of interest, bare spots and dross-derived defects, in conjunction with all the galvanize variants to make a general search. A total of 65 documents were retrieved and further analyzed to elucidate the existence of previous work on the topic of interest. A basic scientometric analysis was conducted to determine the main objective of each document, to later identify clusters of documents that are related to surface defects prediction. The results of the scientometric analysis are presented in Table 1. Based on the content of the document, six categories were established:

1. **Bare spot & Dross – derived defect characterization:** Comprehends those documents were the main goal was to characterize the surface defects produced in the galvanizing line, by describing their morphology, the chemical compounds present on then, and their possible causes.
2. **Dross characterization:** Refers to the documents were the objective was to characterize the chemical composition and the main causes of the different types of dross formed inside the galvanizing pot.
3. **Line modifications to avoid surface defects:** Enumerates the documents that introduce a new process or machine that modifies the current architecture and methodology followed in the galvanizing line, in order to decrease the surface defects occurrence by addressing their possible causes.
4. **Dross formation modeling:** Encloses the documents where a numerical model is presented, which is capable of predicting the rate of dross formation and its possible accumulation areas inside the galvanizing pot.

5. **Zinc bath characterization:** Encompasses those documents where the main objective was to characterize the optimal physical conditions (temperature and chemical composition) of the zinc bath used to galvanized metal products, in order to avoid surface defects and warranty a high corrosion resistance.
6. **Galvanized products' surface defects assessment:** Contains the documents that assessed the consequences derived from the presence of surface defects on galvanized steel products, or the effect on the surface defects appearance by changing the chemical composition of the raw material (steel piece).

Table 1. Documents content classification distribution.

Categories	Documents Count	%
Bare spot & Dross - derived surface defect characterization	32	49%
Dross characterization	11	17%
Line modifications to avoid surface defects	7	11%
Dross formation modeling	6	9%
Zinc bath characterization	5	8%
Galvanized products' surface defects assessment	4	6%
Total	**65**	**100%**

Since the objective of this project is to develop a surface defect predictive model, special attention was given to the fourth category where numerical models were proposed to identify possible efforts on the subject. As stated by the authors of these documents (Ajersch, 2018; Ajersch, 2004; Ilinca, 2004; Ajersch, 2013; Sun, 2017a; Sun, 2017b), all the numerical models described were focused on creating simulations of the dross formation inside the galvanizing pot under different circumstances. Those documents were based on the analysis of the turbulence created by the immersion of the steel products (e.g. rolls, sheets, coils, etc.) inside the pot, and the temperature dynamics between the steel product and the zinc bath per se. As an output of these models, the prediction of the dross rate formation and the possible areas where it will tend to accumulate are given. These outputs are further translated into red flags that advise the operator to manually remove the dross clusters, or into instructions to an automatic system to robotically remove the dross.

The numerical models presented in these works, although attending the need of dross removal to avoid surface defects, were not built to predict their occurrence. Then, according to this literature review, there is no record of the development of a predictive model concentrated on the early identification of galvanizing surface defects on steel products. For that reason, the state-of-the-art exploration on this topic is broadened to find other approaches that have been taken to identify defective steel products on these defects.

Different from surface defects prediction, efforts from academia and industry have been focused on the identification of these defects to avoid the distribution and sale of defective products. On this matter, Automated Optical Inspection (AOI) systems, also referred as Automated Visual Inspection (AVI) systems, consisting on high-resolution cameras, in combination with machine

learning models, have been employed in the recognition of patterns on the steel products surface that are related with a surface defect (Luo, 2020). This AOI systems are placed in more than one location inside a metallurgical factory, to warranty the surface defects recognition in several sites inside it; for example, in the hot-rolling line, the cold-rolling-line, the galvanizing line, etc. (Henrich, 2019; Luo, 2020). Works on this line are presented in Henrich et al. (2019), Luo & He (2016), and on the review by Luo et al. (2020), where the main difference between these systems lies on the accuracy of classification they can provide.

Luo et al. (2020) made a review on AOI systems for surface defect recognition considering 120 publications over the past two decades on steel products. The authors classified the works in four categories based on their characteristics: statistical, spectral, model-based and emerging machine learning; all framed on the imaging context. The statistical approach is based on the pixel intensity distribution (e.g. Clustering, gray-level statistic, local binary pattern, etc.); the spectral approach considers a domain transformation of the image (e.g. Fourier transform, Gabor filtering, wavelet transform, etc.); model-based approaches includes models that are capable of read textures on the images to outperform the other approaches (e.g. Markov Random Field, Weibull Model, etc.); machine learning approaches use supervised, unsupervised and reinforcement learning to recognize and classify these defects (e.g. Support Vector Machine (SVM), Convolution Neural Network (CNN); Convolutional Autoencoder (CAE), Generative Adversarial Networks (GAN), etc.). An example on the recognition and classification of surface defects with these AOI systems is presented in Figure 3, which was taken from Luo & He (2016).

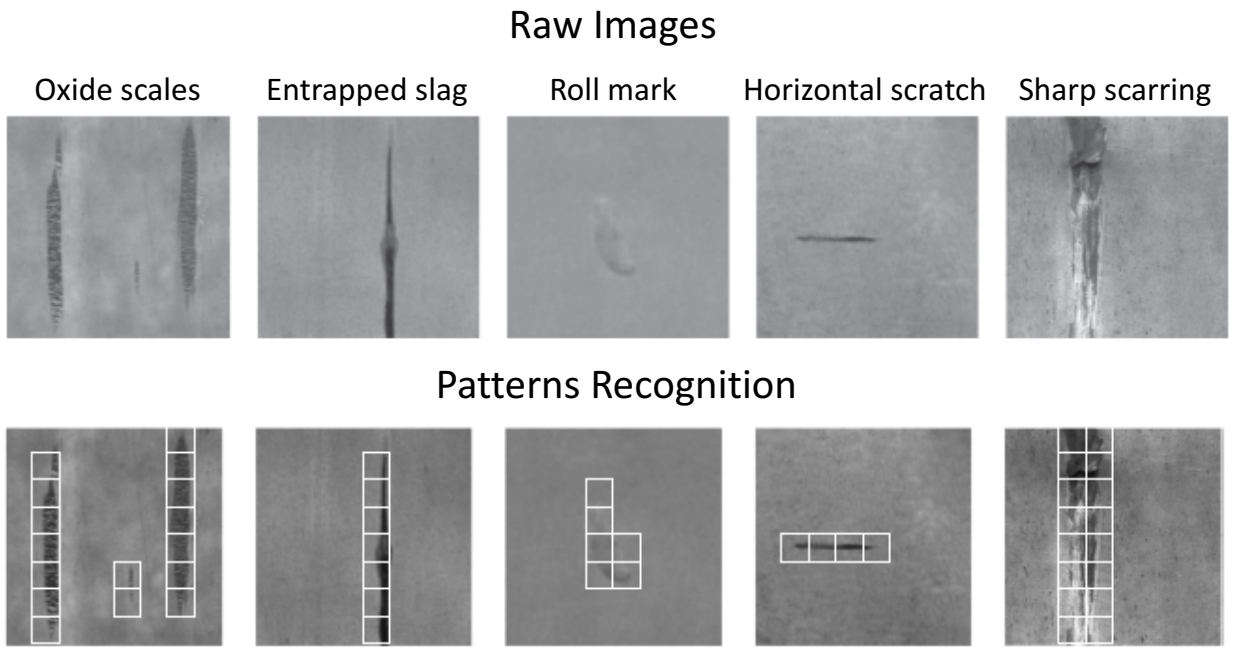

Figure 3. Automated Optical Inspection (AOI) systems in surface defects recognition.
In the upper lane appear the raw images of surface defects taken from the hardware of the system. In the lower lane, the same pictures are presented, but this time including a pattern recognition performed by the software of the system. Image taken from the work by Luo & He (2016).

An example of machine learning approaches using supervised learning is presented by Henrich et al. (2019). The authors proposed an AOI system that innovates on the software employed in the defects visual recognition process. They comment that, usually, these systems consist on a camera equipment (hardware), which detects the presence of an anomaly on the surface of a steel product, that is in constant communication with the trained classification model (software), which determines first the presence of a defect, and then, the type of defect. Both components of the system have their disadvantages that can hinder model performance. In the case of the hardware, obstructions on the lenses such as dust or debris particles can produce false positives, by implying the presence of a defect where there is none. On the other hand, software disadvantages rely on the completeness of the training dataset employed in the classification model construction, which, if not trained correctly, may lead to false positives and, worse, false negatives (letting pass defective products). One example of these dataset difficulties are rare defects of low occurrence that may not be captured properly by the dataset, or, by picking the subdatasets at random when training the model, the incorrect learning of defect occurrence on a certain type of steel product. To avoid at least the software difficulties, the authors developed the Smart Steel Technologies (SST) Training Set Optimizer, which includes: 1) A wide pool of defect images to capture all type of surface defects possible, 2) a train-test correlation method to assure the proper training of the classifier model, and 3) interconnectivity between different AOI systems inside a factory to create feedback loops that enhance the future model's performance after a re-training process. With these characteristics, the authors claimed to have a competitive AOI system that outperforms others already existing in the global market.

A disadvantage of the implementation of AOI systems is the high initial investment fee. According to Luo & He (2016), a single commercial AOI system cost is around $800,000 and $2,100,000 USD, without contemplating the maintenance, which can be expensive since it requires the intervention of technical experts. Another weakness is the lack of insights for preventive actions that could decrease the surface defects occurrence. Since this system's objective is to recognize rather than predict these defects, it does not provide any clues or directions on possible variables or machines in the line that could have a significant effect on their occurrence. However, not everything is lost since these systems present a high accuracy on recognizing surface defects, guaranteeing the capturing of defective materials; thus, avoiding their distribution and sale to the final clients.

2.2 Multi-Objective Ensemble Model

The second topic discussed is the design of a classification predictive model with an acceptable accuracy and precision, based on the criteria of FNR and FPR raised in this project. Specifically, this literature review focuses on classification ensemble models.

According to the review on ensemble models conducted by Ren, Y. et al. (2016), there are eight categories in which these ensembles, whether employed for classification or regression, can be grouped:

1) Conventional Ensemble Models: This category applies for the ensemble models that include on its design a bagging, boosting or stacking strategy.

2) Decomposition Based Ensemble Models: Here lies the ensemble models that use a divide-and-conquer technique, which disaggregate a problem into small parts with the intention to solve them with individual learners, and later combine or sum the outputs to get the final prediction.

3) Negative Correlation Learning Based Ensemble Models: This category encompasses the ensemble models that use a Negative Correlation Learning (NCL) approach, which consists on adding diversity to the learners of the ensemble, in spite of being trained with the same dataset.

4) Multi-Objective Optimization Based Ensemble Models: It includes the ensemble models that are designed to obtain the Pareto frontier of the analyzed predictors.

5) Fuzzy Ensemble Models: As its name says, this category covers the ensemble models that use fuzzy logic to enhance the overall performance of the ensemble.

6) Multiple Kernel Learning Based Ensemble Models: Under this category are the ensemble models that use a Multiple Kernel Learning (MKL) strategy, which consists on putting together with a predefined combination function more than one kernel that represents the studied dataset.

7) Deep Learning Based Ensemble Models: This category covers the ensemble models that use deep learning techniques, which combine a multiple layer structure with non-linear operations defined on each, to outperform individual learners.

8) Ensemble Regression Converted to Ensemble Classification: It encompasses the ensemble models that transform a regression problem into a series of individual classifiers to instead use a classification algorithm (It only applies when a high resolution is not required).

These eight categories are neither exclusive nor rigid. In other words, an ensemble model can belong to more than one category, depending on how it is structured, and with the evolution and advancements in this area, new categories are going to be needed to cover the whole spectrum.

Since the intention of this section is not to provide an extensive review on ensemble models, the literature review will be focused on the fourth category, Multi-Objective Optimization Based Ensemble Models, which is the main target of the ensemble model introduced in this thesis since it is design to deal with the trade-off between the two objectives of the project: to low both the FNR and FPR.

Ensemble models can be used to solve multi-objective problems, where a trade-off between objectives usually happen (Ren, 2016). Under this line, the learners of the ensemble can be trained to solve individually an objective, or to solve all objectives at a time. Ensemble models in this category are still under-researched, however, some proposals have been made on the line of Neural Networks generalization with the support of Genetic Algorithms, and few reports on classification problems combining Multi-Objective Decision Trees (MODT), Support Vector Machines (SVM), Random Forests and Bagging.

On the line of multi-objective problems solved with ensembles of evolutionary algorithms, we have the work by Tan, C.J. et al. (2014), where a modified micro genetic algorithm (MmGA) is used to optimize three objective functions related to the number of features selected and the classifier performance (sensitivity and specificity). A MmGa, developed by Tan, C.J. et al. (2013), is an evolutionary algorithm specialized on tackling multi-objective problems by finding the Pareto frontier set of parameters that result from solving the objective functions of the problem. They proposed an application for the MmGA to form an ensemble method in combination with a classifier that directly uses the parameters enlisted in the Pareto set that were chosen by the MmGA, in this case a Neural Network. Now, the individuals or possible solutions, in order to survive across generations, must be selected from a voting-based elite selection, where only the best individuals, in terms of the classifier performance (the objective was to maximize both parameters) and the number of features used (the objective was to minimize this parameter), are selected. They tested this method with available datasets from the University of California at Irvine (UCI), proving to surpass the classifier performances already reported in the literature (UCI, 2020).

Another ensemble model created to generalize neural networks in a less complex manner is proposed by Chandra & Yao (2004). This ensemble, named as DIVACE (DIVerse and ACcurate Ensemble learning), also puts together an evolutionary algorithm with a neural network, with the intention to have as a result a generalized network that could manage the trade-off of being accurate and diverse at the same time. These two characteristics are the objective of the ensemble, specifically to the evolutionary process, where the accuracy objective is solved by using a Memetic Pareto Artificial Neural Network (MPANN), and the diverse objective with a penalization given by the Negative Correlation Learning (NCL) algorithm. MPANN is an evolutionary algorithm that seeks the Pareto set of parameters that optimizes the objective functions of the problem (Abbass, 2001); while, the NCL algorithm incorporates a penalty term into the error function in order to have as a result non-correlated individuals in the evolutionary process, i.e. it encourages the diversity among individuals (Liu & Yao, 1999). To decide which set of parameters will represent the final solution of DIVACE, three different combination rules were tested: simple averaging, majority voting and winner-takes-all. In summary, DIVACE is an ensemble method that generates a set of generalized neural networks in one run, which are potential solutions to the classification problems intended to be solved.

Another work on the line of neural networks generalization and features selection is presented by Chen & Yao (2006). Their work consists on an ensemble that combines a Multi-Objective Evolutionary Algorithm (MOEA), which locates the Pareto optimal set of individuals where each one encloses a neural network configuration, with a Bayesian inference method, which helps in the parameter tuning of the neural networks' configurations and in feature selection. The MOEA employed in this ensemble was the elitist Non dominated Sorting Genetic Algorithm (NSGA), which searches for the best individuals that dominate over the rest in terms of fitness on both the objective functions: 1) maximize the neural network accuracy, and 2) minimize the number of features selected (Deb, 2002). Now, about the Bayesian inference method used, the Bayesian Automatic Relevance Determination (ARD) method was selected to generate the individuals to test with the ensemble. The Bayesian ARD has two tasks: The first one is to determine different possible configurations for the neural networks by setting three characteristics (weight vector w, variance of weights vector α, and the noise parameter β) at the same time. To decide the value for w, α and β, the Maximum Log-likelihood Estimation for each one is calculated by assuming they

present a normal distribution, followed by their estimation using an iterative process where one characteristic is calculated at a time by fixing the other two, and so on. The second task is to select different subfeature spaces based on the relevance that each feature presents on the Bayesian ARD evaluation, where the relevance of a feature is determined by the posterior probability, obtained from Bayes' Rule, by modeling a zero-mean Gaussian prior function (Neal, 1995; Tipping, 2001). At the end of the evolutionary process, the Pareto optimal set of individuals are subjected to a pruning process by using a Logistic Regression that assigns weights to each individual, and only those ones that have a weight greater than a preset threshold could be chosen for the final ensemble. As a result, this ensemble method provides a framework to feature selection and neural networks generalization, but, as the authors mentioned on their conclusions, further analysis has to be made to elucidate the strengths and weaknesses it may present.

Posterior to their work in 2006, Chen & Yao (2010) proposed a novel ensemble method that supports the generalization of neural networks for classification purposes, by improving the Negative Correlation Learning (NCL) algorithm to obtain a diverse array of competent neural networks ensembles. The resulting algorithm is baptized as the Multi-objective Regularized Negative Correlation Learning (MRNCL) algorithm. NCL, as previously stated, introduces a correlation penalty term that seeks to reduce the mean square error (MSE) of each, for this case, neural network that comprehends the whole ensemble, and reduce the correlation among these networks to ensure a diverse and robust ensemble for the current problem of study (Chen & Yao, 2010). However, a drawback of NCL is the optimization of the penalty term per se, which is, according to the authors, computationally expensive to get due to the need of cross-validation, and, if not chosen correctly, it might lead to overfitting that results in a non-generalized ensemble. To alleviate this, they enclose the neural networks ensembles production inside a Multi-Objective Evolutionary Algorithm (MOEA) that add one more regularization term, specifically the weight decay term, which furtherly penalizes each network to implement a correction to the decisions made by the NCL part of the algorithm by limiting the weights grow (Krogh & Hertz, 1992). Then, the MOEA (an NSGA) is in charge of optimizing the trade-off between the three objectives of this problem: 1) The objective of performance, that minimizes the MSE of a network based on the training set; 2) the objective of correlation, that minimizes the penalty term for each network to provide variability, or diversity, among the components of the ensemble; 3) the objective of regularization, which minimizes the regularization term introduced in the MRNCL algorithm that copes with large weights of the networks, a common result from employing NCL outside the proposed ensemble model, that promotes overfitting, i.e. a non-generalized neural network. As a final step of the algorithm, after the evolutionary process has ended, the Pareto set of neural networks are identified and combine equally (with the same weights) to form the final ensemble. The MRNCL performance was compared with other popular existing algorithms such as MNCL (Multi-objective NCL, without the extra regularization step), MoNN (Multi-objective Neural Networks), AdaBoost, Bagging, Support Vector Machines, among others, with 16 benchmark datasets, most of them coming from the UCI database. MRNCL was capable of outperform the other algorithms in 10 of the 16 cases, proving the robustness and generalization ability it has.

One improvement for the ensemble methods that uses an evolutionary process as a framework to warranty the generalization of neural networks is presented by Chen et al. (2014). Two popular requirements for computational programs or algorithms are how long it takes to find an accurate solution and how sure the user is that the said solution is the best or nearly the best. For that reason,

the algorithms, in this case the one that will create the final ensemble, must ensure it is capable of covering, in the shortest time possible, the search space efficiently, i.e. since it is not possible to prove every single combination of parameters, the algorithm must provide an initial population, or a diversification step inside it, that encompasses as many neighborhoods of data as possible of the entire set of combinations available. Here lies the proposal of the authors, where a novel diversification step is introduced in genetic algorithms to produce generalized individual neural networks or an ensemble of them. They suggest the use of Random Array Local Search (RALS), a variant of Ortogonal Arrays (OA), as a diversification step in an evolutionary algorithm framework. An OA is employed to represent all the possible combination of factors or parameters of an experiment with a small number of combinations; however, a single one of these arrays is time consuming and difficult to develop, increasing the computational time. For that reason, they recommend using only one OA at the beginning, to produce the initial population, and then varied them with the RALS. This algorithm, RALS, helps to search within each neighborhood other solutions that may have a better performance than the ones in the initial population. If it is the case, the algorithm stores the new individual, and it keeps iterating until the optimal solution is reached or until the stopping criteria programmed for the whole ensemble is met. To build the final ensemble the best individuals are selected, and to provide an answer, the majority voting approach is used. At the end, the whole algorithm, baptized as a Unified Evolutionary Training Scheme (UETS), is capable of efficiently search the whole sample space, in a short time. They validate this last statement with the n-bit parity problem (a popular and difficult classification problem) (Hohil, 1999) by comparing the time needed to find the optimal solution by UETS, with an algorithm that encloses the use of OAs to get the optimal solutions, Two-Phase Genetic Local Search (TPGLS) algorithm (Tseng & Chen, 2006). The results of the UETS were astonishing, since it was capable of reach the optimal solution 11 times faster, and, in some cases, it was the only one to come up with a valid solution without getting trapped in local optima.

As mentioned before, there is evidence in literature that prove the superiority in accuracy of ensemble models over individual learners; however, this statement cannot be true unless the individual predictions of the learners of an ensemble are diverse. By following this idea, Gu & Jin (2014) proposed an ensemble method, which uses genetic algorithms in combination with a set of Support Vector Machines (SVM) classifiers, to optimize the trade-off between accuracy and diversity, and, at the same time, try to outperform individual classifiers. SVM is a clustering tool that uses a hyperplane to separate data into classes by maximizing the distances between the data points and itself (Bennett & Campbell, 2000). The ensemble is composed of a Multi-Objective Evolutionary Algorithm (MOEA), specifically the elitist non-dominated sorting genetic algorithm (NSGA-II), and a set of SVM classifiers that were trained with subsets of the whole dataset or with subfeature sets from the same. As for the individuals that were given to the ensemble, they were represented as binary strings (0: the classifier is not in the ensemble; 1: the classifier is in the ensemble) with a length equal to the number of SVM classifiers available. In the evolutionary process, the NSGA-II is in charge of selecting the best individuals that optimizes all the objective functions (elitist selection) for them to survive across generations, and at the end, find the Pareto set of individuals that provide the best candidates to be the optimal solution according to it (Deb et al., 2002). For this multi-objective problem, the accuracy is measured with the classification rate, which is obtained by: 1) First combining the individual predictions of the SVM classifiers, in this case, the combination rule is given by majority voting for each sample to be labeled, and then 2) calculating the proportion of samples that were correctly classified; while, diversity is measured

with three different approaches: Coincident Failure Diversity (CFD), Disagreement (DIS) and Hamming Distance (HD), where all of them have as a result a value between 0, meaning that all learners have the same prediction for a sample, and 1, when all learners have unique predictions for the same. This ensemble method was tested using available datasets from the University of California at Irvine (UCI), and they showed some cases where it presents a higher accuracy than a single classifier. Nevertheless, the superiority of the ensemble was not unanimous, since there were cases where the single classifier still presented a better performance. Therefore, more research must be done to come up with new restrictions that could strengthen the current algorithm.

Different from the previous works presented in this section, Kocev et al. (2007) proposed an ensemble model for classification and regression purposes based on Multi-Objective Decision Trees (MODT). A MODT, as its name indicates, deals with more than one objective at a time, modifying its growth behavior based on a vector of objectives optimization (Haimes, 1990; Todorovski, 2002; Castin & Frénay, 2018). This optimization is achieved by minimizing the variance function for the set of objectives; thus, maximizing the clusters homogeneity, created by the model, and enhancing its predictive performance (Kocev, 2007). To demonstrate the superiority of MODT ensembles over Simple-Objective Decision Trees ensembles (SODT), they analyze the performance of both in 21 benchmark datasets, consisting of 11 classification problems and 10 regression problems, and tested their accuracy and model size needed to learn. The number of samples, variables and targets in each dataset were varied to test the robustness and versatility of the ensembles, covering from 85 to over 10,000 samples, 4 to 142 variables, and 2 to 14 targets. What they found was that MODT ensembles are equally good or better than SODT ensembles, and the model size, which can be translated of how fast the model learn, was smaller for MODT ensembles in all 21 datasets. The final product of this research was an ensemble model with MODT as learners, which are capable of outperform SODT ensembles, providing accurate solutions in less time both in classification and regression problems.

Finally, Makhtar at al. (2015) proposed a classification ensemble model for medical diagnosis purposes. They implement an ensemble model, whose objectives are the maximization of its accuracy (correctly diagnosed with the disease), sensitivity (incorrectly diagnosed with the disease) and specificity (correctly diagnosed as healthy), on a public dataset of diabetes (Pima Indian Diabetes) by the University of California at Irvine (UCI). This dataset is of low dimensionality (9 variables) and it is not greatly imbalanced (768 patients, where 500 are healthy and 268 are ill). Their ensemble model consists on a pool of classifiers created by the WEKA library in Java (Frank, 2016), which are combined using a majority voting combination rule. The models created with such library were a set of K-nearest neighbor classifiers, decision trees and numerical prediction algorithms, and they were compared with other classification models such as Bagging, AdaBoost and Naïve Bayes. To select those classifiers that will be part of the final ensemble, a diversity metric named as double-fault is included. This metric is used to identify those classification models that are less correlated, based on the correct and incorrect classifications made by two models, to warranty diversity on the final voting. As a result, the authors could outperform the other models in accuracy, while maintaining high levels of the other two objectives: sensitivity and specificity. However, the improvement was not groundbreaking, since it improved the accuracy in 1% in comparison with AdaBoost. For that reason, the authors mention in their conclusions that the ensemble has potential, but future work is needed to enhance the final ensemble performance. Some interesting lines of study that they express are the

exploration of other diverse metrics, and the inclusion of other classification models into the pool employed.

In summary, most of the multi-objective ensemble models found on the literature are intended to generalize Neural Networks using Genetic Algorithms to avoid overfitting when new data is tested, along with other classification models involving MODT, SVM, RF, among others. Usually, the work on ensembles is centered on classification problems that, based on the datasets that were analyzed with them, present low-dimensional and balanced data. The objectives are designed to assess the performance of the model, or, in the case of Neural Networks, to assess the complexity of the Neural Networks architecture. A more detailed summary of the main characteristics tested in each work is presented in Table 2, along with a comparison between these models (stated as "related work") and the final ensemble model proposed in this project. As shown in the table, our model shares various characteristics such as the type of learners (Random Forest), the number and type of objectives (two performance objectives) and the combination rule (weighted average). However, in some aspects, our model deals with more complex situations such as high dimensional (7750 in contrast with 142 variables on the related work) and highly imbalanced problems (more than 97% in two classes classification, in comparison to 89% on the related work). A final remark, ensemble models that are enclosed inside a Genetic Algorithm framework cannot deal with overly high dimensional problem, such as the one posed by our dataset, since, according to the literature, the time needed to reach a solution increases exponentially depending on the number of objectives and variables involved (Eiben, 2003; Yang, 2014).

Table 2. Summary on the related work to multi-objective ensemble models.

	Type of Learners	**Classification / Regression**	**Number of Variables**	**Degree of Imbalance***	**Number of Objectives**	**Combination Rule**
Tan, C.J. et al. (2014)	GA, NN	Classification	10 - 13	56%; 62%	3: Sensitivity, Specificity, # Features	Majority Voting
Chandra, A. & Yao, X. (2004)	GA, NN	Classification	8 - 14	56%; 65%	2: Accuracy, Diversity	Majority Voting, Winner-takes-all, Weighted Average
Chen, H. & Yao, X. (2006)	GA, NN, Bayesian ARD	Classification	8 - 14	56%; 62%; 65%	3: Weight Vector, Variance of Weights Vector, Noise Parameter	Weighted Average
Chen, H. & Yao, X. (2010)	GA, NN	Classification	3 – 60	51%; 50%; 56%; 61%; 65%; 65%; 68%; 70%; 33 – 33%; 10 – 24%; 25 – 50%; 14 – 70%; 0.5 – 79.5%	3: MSE, Diversity, Regularization term for NN (Weight Decay)	Average
Chen, W. et al. (2014)	GA, NN	Classification	4 - 21	50%; 56%; 62%; 65%; 70%; 89%; 33 – 33%	1: Accuracy	Majority Voting
Gu, S. & Jin, Y. (2014)	GA, SVM	Classification	7 - 33	8 – 50%; 2 – 93%;	2: Accuracy, Diversity	Majority Voting

Kocev, D. et al. (2007)	MODT	Classification & Regression	4 - 142	50%; 9 – 42%; 2 – 76%; 0.5 – 79.5%;	1: Accuracy	Probability Distribution Vote
Makhtar, M. et al. (2015)	K-Nearest Neighbor Classifier, DT, Numerical Prediction Algorithms	Classification	8	65%	3: Accuracy, Sensitivity, Specificity	Majority Voting
Our Model	RF	Classification	7750	97%; 97%; 98.4%; 98.5%; 98.6%	2: FNR, FPR	Weighted Average

*On two-class problems, the degree of imbalance presented is the dominant one. On multiclass problems, the value presented is the range between the less represented class and the dominant one. GA: Genetic Algorithm; NN: Neural Network; SVM: Support Vector Machine; ARD: Automatic Relevance Determination; MODT: Multi-Objective Decision Trees; DT: Decision Trees; RF: Random Forests; FNR: False Negative Rate; FPR: False Positive Rate

2.3 Knowledge Gap Analysis

Both literature reviews, surface defects prediction and multi-objective ensemble models, present opportunity areas in the context of this project's problem. Said areas are discussed in this section as follows.

First, according to the surface defects prediction literature review, efforts from the Industry and the Academia have been focused on the recognition of surface defects, with the use of AOI systems, or in understanding possible causes that the metallurgical scientific field have been studied, such as the dross formation dynamics, temperature profiles, etc.. However, different from the numerical methods developed to model the dross formation as contaminant inside the galvanizing pot, there are no insights on possible causes inside a metallurgical factory that may cause or enhance the surface defects' occurrence; thus, the problem has not been addressed. Moreover, even though these recognition systems decrease the number of unnoticed defective products, the related costs from the galvanizing process in conjunction with upstream processes are still present, representing a monetary loss for the company. By taking these into account, a robust predictive model capable of properly classifying defective products in these surface defects will not only represent an innovation in the metallurgical area, since no other attempts' records were found, but an opportunity of improvement to enhance the products' quality, while saving money.

Secondly, as it can be seen from the multi-objective ensemble models literature review and the information presented in Table 2, the majority or, in some cases, the totality of the ensembles were not tested with datasets containing a large number of variables (no more than 142 variables were modeled) or with a high degree of imbalance between the modeled classes. In a few words, the problems or datasets that do not comply with these characteristics, which is the case of this project's problem, have been unattended by the research area of ensemble models. Since ensembles have proved to outperform their individual counterparts in some cases, an opportunity

area is presented to design and develop new ensemble models capable of properly classify data with these characteristics.

Our proposed ensemble model attempts to fill these gaps by representing the first classification model for probable defective materials in surface defects in the galvanizing line, which is capable to model a high-dimensional problem with a high degree of imbalanced in the defective sample available. Hence, if proved better than current off-the-shelf approaches, it will represent an important advance in binary classification problems with these characteristics, and the first effort to rather than recognize, predict the surface defects' prediction.

Chapter 3

Methodology

This section is organized following the specific objectives of this work: 1) Database assembly and cleansing, 2) designing and fitting predictive models for surface defects, and 3) detecting actionable and significant variables for preventive actions in the galvanizing line. Full development of each objective is presented below.

3.1 Database Assembly and Cleansing

The first step before starting to design a modeling strategy for the given problem is the assembly and cleansing of the database. This step not only helps to build the training / testing database of this project, but also helps to understand the available data in terms of the dimensionality of the problem, the types of variables and the size of the population. To achieve this step, seven actions are executed.

1. Outlier Removal

A common problem in technological-acquired data is the presence of noise in form of atypical values, also known as outliers (Tukey, 1962). These values usually appear in the extremes of a variable distribution, representing technological artifacts product of malfunctioning or errors in sensors or measurement systems, representing false information that unnecessarily increase the variance of the same (Dixon, 1960; Tukey, 1962). Several techniques have been developed to carefully avoid these outliers without losing valuable information; one of them is winsorizing the data, which replace the extremes values with the next largest or smallest value, depending on the localization of the outlier (Dixon, 1960). In Python, this technique is already coded in the Scipy library (*winsorize*), and it allows the user to choose where to cut in each extreme of the variable, advising to use small values to avoid an over-winsorization that would lead to information loss (Scipy, 2020).

2. Imputation of Empty Cells

Another common problem in databases are empty cells, which cannot be handled by some predictive models; thus, they must be filled in. Based on the nature of each variable (type of variable, number of empty cells, etc.), there are a handful of imputation strategies that can be used to solve this missingness problem (Gelman & Hill, 2018). For this project, the variables with missing values where classified in three groups, for which a different imputation strategy is implemented:

a) *Punctual variables, punctual imputation strategies*. Some variables already have a defined imputation strategy by, in this case, the galvanized products manufacturers; then, no further analysis is needed. For example, a frequent strategy consists on impute

the blank cells with the average between the product before and after, chronologically speaking, if and only if it fulfills certain restrictions.

b) *Imputation based on rules*. This imputation strategy depends on the characteristics of the variable to impute, such as the type of variable (numerical or alphanumerical) and the number of unique values (continuous or discrete). The first distinction is the type of variable. If the variable has alphanumerical values, it is imputed with their mode and the imputation process is over. If the variable has numerical values, a distinction is made depending on if its values are considered as continuous or discrete based on the number of unique values it has. If the variable has continuous data (defined as having more than 10 unique values), the next step is to figure out if it is skewed or not. If not skewed, the variable is imputed with its mean; otherwise, if skewed, the variable is imputed with its median. To determine the skewness of the variable a skew test from the Scipy library is used, being skewed if the p-value is significant (less or equal to 0.05), and non-skewed otherwise. This test is obtained using the third central moment test ($\sqrt{b_1}$) described in the equations in Table 3, where X corresponds to the values of the variable, μ to the mean of the variable, n to the sample size, which has to be greater than eight, and $Z\left(\sqrt{b_1}\right)$ to the result of the test (D'Agostino, 1990; Scipy, 2020). On the other hand, if the variable is considered as discrete, the mode's frequency rate (MFR) is calculated with $MFR = Mode's\ Frequency/n$, where n is the number of samples. Then, if the MFR is equal or greater than 90%, the variable is imputed with the mode; otherwise, the variable is imputed with a random value coming from its existing unique values, where each value has a probability $P = Unique\ Value's\ Frequency/n$, proportional to its frequency on the variable. These set of rules are presented in Algorithm 1.

Table 3. Third central moment test equations to determine the skewness of a variable.

Third central moment test equations	
$\sqrt{b_1} = \frac{E[(X-\mu)^3]}{(E[(X-\mu)^2])^{3/2}}$	$Y = \sqrt{b_1}\left\{\frac{(n+1)(n+3)}{6(n-2)}\right\}^{1/2}$
$\beta_2\left(\sqrt{b_1}\right) = \frac{3(n^2+27n-70)(n+1)(n+3)}{(n-2)(n+5)(n+7)(n+9)}$	$W^2 = -1 + \left\{2\left(\beta_2\left(\sqrt{b_1}\right)-1\right)\right\}^{1/2}$
$\delta = \frac{1}{\sqrt{\ln W}}$	$\alpha = \left\{\frac{2}{W^2-1}\right\}^{1/2}$
$Z\left(\sqrt{b_1}\right) = \delta \ln\left(\frac{Y}{\alpha} + \left\{\left(\frac{Y}{\alpha}\right)^2 + 1\right\}^{1/2}\right)$	

c) *Complete case-analysis*. This strategy consists on exclude those samples where blank cells are present in at least one variable. This strategy is not implemented directly into the context of this problem, but it is reserved to a set of variables were no guessing is allowed to impute, such as the product chemistry, or target variables.

Algorithm 1. Imputation rules for group 2 variables.

For all v variables $\in V$ do
  $T = \text{d\_type}(v)$
  If $T \neq$ Numeric then
    Impute $\text{mode}(v)$
  Else if $T =$ Numeric then
    $u = \text{unique}(v)$
    If $u > 10$ then
      $p = \text{skew\_test}(v).\text{p\_value}$
      If $p > 0.05$ then
        Impute $\text{mean}(v)$
      Else if $p < 0.05$ then
        Impute $\text{median}(v)$
    Else if $u < 10$ then
      $f_{mode} = \text{mode}(v)/n$
      If $f_{mode} \geq 0.9$ then
        Impute $\text{mode}(v)$
      Else if $f_{mode} < 0.9$ then
        Impute $x_i \in v$

3. Time Series Analysis

Part of the data for this project, instead of being as punctual features, come in form of time series, which need a different treatment in order for them to be considered inside the predictive models. In this case, concrete statistics, such as the mean, the coefficient of variation, the 5% percentile, the 95% percentile and the range, are calculated to extract the main information for each of them. Furthermore, to avoid the inclusion of repeated information, the Pearson's Correlation Coefficient (PCC) (Weisstein, n.d.) is calculated for each pair of statistics, where the ones who has a squared-PCC greater than 0.9 are eliminated according to the following hierarchy:

Mean > Coefficient of Variation > 5% Percentile > 95% Percentile > Range.

The PCC is calculated with

$$PCC = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E(X)^2} \cdot \sqrt{E(Y^2) - E(Y)^2}},$$

where X and Y are the two variables that are evaluated. As a product, the overall behavior of the time series is captured with these five statistics, where the mean entrap the average behavior, the coefficient of variation the variability of the series, the 5% percentile the minimum value avoiding possible outliers, the 95% percentile the maximum value avoiding possible outliers, and with the range, the range itself.

4. Label Encoding

The next action is the transformation of categorical variables with alphanumerical information into numerical categories for their inclusion in the final database. To accomplish this, a simple label encoding is conducted, which first identifies the number of categories in a variable, and then, it changes the alphanumerical value into a numerical one (Cohen, 2015). This technique does not take into account any extra relationship between the values of the variable or a special order denoting a hierarchy, and it assumes that each sample lies in only one category. This technique is employed as it is already coded in the Scipy library in Python (*LabelEncoder*) (Scipy, 2020). A visual aid for the label encoding process is depicted in Figure 4.

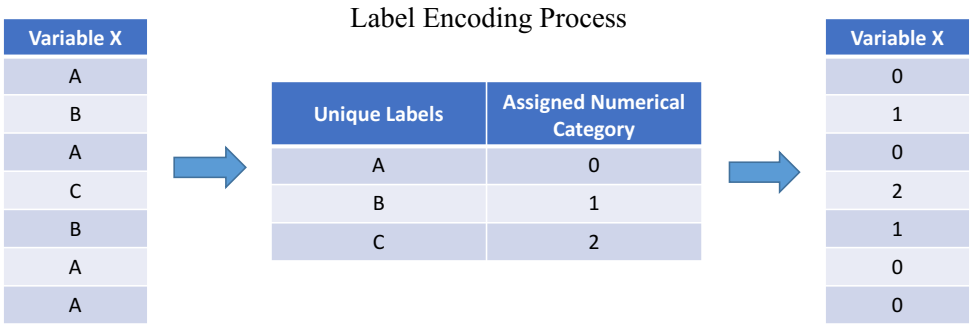

Figure 4. Label encoding process.
First, the unique values of the variable are identified; then, to each unique value, a numerical category is assigned. Finally, the values in the variable are replaced by their corresponding numerical category.

5. Feature Engineering

A technique employed to enrich the original set of variables of the database with new variables that could improve the effectiveness of the predictive models is feature engineering (Kuhn & Johnson, 2020). The new engineered variables are designed according to the knowledge of experts, in this case the galvanizing line experts, and they are a result of the combination of the existing ones. Another characteristic of these variables is that they result from combinations of variables that by no other means can be obtained by the predictive model itself, such as the basic mathematic operators ($+$, $-$, $\times$ and $\div$).

6. Removal of constant fields

Another step in the cleansing of the final database is the removal of constant fields. There is no gain in the predictive model's performance by including fields with a constant value for all samples in the population (Kuhn & Johnson, 2020). For that reason, all the fields with this characteristic are filtered out of the database.

7. Normalization

As a final step of the construction of the final database is the normalization of the data using the standard score

$$Z = \frac{X - E[X]}{\sqrt{Var[X]}},$$

where X are the values of a variable. In this case, normalization is used to have the same scale of magnitude and variability for all variables, in order to be able to compare the effects of them equally (Milligan & Cooper, 1988). After this action, the final database is ready to train and test the posterior predictive models.

3.2 Design and Fit of Predictive Models

A set of predictive models are employed in this project for the prediction of probable defective products. In total, four models are tested in order to find the model that is capable of minimizing the objective functions under the maximum limits established on the acceptance criteria: A maximum False Negative Rate (FNR) of 15% and a maximum False Positive Rate of (FPR). These two metrics are used to evaluate indirectly the recall and precision of the models, which are common scores for evaluating learning methods despite of the context of the problem (Zhou, 2012), using their confusion matrix produced in each iteration. The formulas to calculate the FNR and the FPR metrics are

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

and

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

where FN are the false negatives, FP the false positives, TN the true negatives and TP the true positives. The models tested are the off-the-shelf Forward Stepwise Logistic Regression (FS-LR), the Random Forest Classifier (RFC), the Gradient Boosting Classifier (GBC) and the custom-made Low FNR and Low FPR Random Forest Classifier Ensemble (LFNR-LFPR RFC).

A. Forward Stepwise Logistic Regression (FS-LR)

This model results from the combination of the Logistic Regression (LR) model with the Forward Stepwise (FS) algorithm for variable selection. LR is a linear model frequently used for binary classification where it assesses the underlying relationships between the individual effect of independent variables (i.e. marginal probability) and a categorical dependent variable (target variable) (Hosmer, 1978; Efron & Hastie, 2016). The linear LR curve is obtained with

$$\hat{\pi}(x) = \frac{1}{1 + e^{-(\widehat{\beta_0} + \widehat{\beta_1} x)}},$$

where $\hat{\pi}(x)$ is the binomial class probability estimator, $\widehat{\beta_0}$ and $\widehat{\beta_1}$ are the coefficient maximum-likelihood estimator (MLE), and x are the variable's values. LR can deal with high dimensional problems (large number of variables), however, the risk of including redundant information

without any kind of variables discrimination can harm the final fit of the model to the data (Efron & Hastie, 2016). Instead, a subset with less yet significant variables is desired. This is the reason why LR is coupled with the FS strategy for variable selection. FS is a selection procedure where the variables are added to the model one by one, and only the ones who are able to comply with a significance criterion are selected to form the final subset of variables (Hosmer, 1978; Efron & Hastie, 2016).

For this project, the significance criterion under which the variables are selected is based on the p-value that the model returns for each variable in each iteration. All the variables with a p-value greater than 0.2 are removed from the analysis; otherwise, they keep competing to stay in the winner subset. More than one set of variables comply with the significance criterion; thus, Akaike's Information Criteria (AIC) is calculated for each candidate in order to be able to compare them and select the subset with the smallest AIC score (Efron & Hastie, 2016). The formula that is used to calculate AIC is $AIC = 2K - \log(\mathcal{L}(\hat{\theta}|y))$, where K is the degrees of freedom of the tested model and $\mathcal{L}(\hat{\theta}|y)$ is the maximum likelihood estimator (MLE). Moreover, the algorithm of the FS-LR is depicted in Algorithm 2.

Algorithm 2. Stepwise Logistic Regression.

1. A null model M_o is initialized with no predictors.
2. For $k = 0$ to $p - 1$:
    a. Consider all $p - k$ models that increment the predictors in M_k with one additional predictor.
    b. Select the best model among the $p - k$ models and name it M_{k+1}. The best model is defined as having the smallest Residual Sum of Squares (RSS) or highest R^2.
3. Choose a single best model from among $M_o, \dots, M_p$ using a AIC as a prediction error.

B. Random Forest Classifier (RFC)

Random Forests were introduced by Breiman (2001) as a combination of individual decision trees for both classification and regression purposes. A decision tree is a non-parametric supervised learning algorithm that creates a set of decision rules based on the input data (Efron & Hastie, 2016). Each decision rule is based on a splitting criterion that decides which variable to select as the initial or following node to keep going down in the tree growth. In the context of the RFC, the splitting criterion is the Gini Index (GI), where for each level of the tree the variable with the smallest GI is selected to become the next node where two branches will depart (Nembrini, 2018). The GI of a variable X_m is calculated with

$$GI(X_m) = 1 - \sum_{j=1}^{J} {p_j}^2, \quad (1)$$

where p_j is the portion of samples classified under a class j of the target variable out of the total number of samples at a certain node.

Random Forests are capable of delivering accurate results with low bias and variance due to the diversity strategies they implement on their algorithm, no matter the dimensionality or the size of the population. For instance, Random Forests' good performance rely on the randomness they introduce with the aid of the bootstrap algorithm, which makes a sampling with replacement from the original data to form a new subdataset, in order to create independent subsets of both samples and features to train each individual tree (Breiman, 2001; Efron & Hastie, 2016). At the end of the Random Forest algorithm, the individual decisions of each tree are combined to provide the final answer by applying average as a combination rule. The algorithm with the steps followed by a RFC is presented in Algorithm 3. For this project, the RFC are designed with 400 individual trees, and they are restricted to have at least four samples per branch after a split, and at least two samples in each split.

Algorithm 3. Random Forest Classifier.

1. For a given dataset $d = (X, y)$ with p variables, fix $m \leq p$ and a B number of trees.
2. For $b = 1$ to B:
    a. From the training data, draw a bootstrap sample d^* of size N by randomly sampling the n rows with replacement n times.
    b. Using d^*, grow a tree T_b by doing the following steps for each terminal node of T_b, until the minimum node size (n_{min}) is reached.
        i. For m in p variables:
            1. Calculate Gini Index (GI)
        ii. Pick the variable m with the smallest GI: best split-point.
        iii. Split the node into two daughter nodes.
    c. Save each tree (T_b) predictions and the bootstrap samples for each of the training observations.
3. Using the generated ensemble of trees $\{T_b\}_1^B$, make a prediction at a testing point x for its classification:
    a. Let $\hat{C}_b(x)$ be the prediction of the class of the b^{th} random forest tree. Then, average the predictions:

$$\hat{C}_{rf}^B(x) = \frac{\sum_1^B \hat{C}_b(x)}{B}.$$

To estimate the accuracy of the RFC model, the Out-Of-Bag (OOB) error estimates are considered. This error is estimated using the loss function (L) of interest in combination with the OOB RFC's predictions ($\hat{r}_{RF}^{(i)}(x_i)$). The OOB predictions are calculated with

$$\hat{r}_{RF}^{(i)}(x_i) = \frac{1}{B_i} \sum_{b:w_{bi}^*} \hat{r}_b(x_i)\,,$$

where B_i are the number of times the sample was not in the bootstrap sampling, $\hat{r}_b(x_i)$ the prediction of the individual trees, and w_{bi}^* the frequency vector for a bootstrap iteration that contains how many times each sample is repeated. These predictions are already calculated by the

RFC models in the Sklearn Python library (Scikit-learn, 2020), and they consist on averaging all the random-forests trees where the ith sample was not included by the bootstrap sampling (Efron & Hastie, 2016). Then, to compute OOB error, each $\hat{r}_{RF}^{(i)}(x_i)$ is used to compute the loss function (L) of interest, such as misclassification or squared-error loss, to calculate the total error

$$err_{OOB} = \frac{1}{n}\sum_{i=1}^{n} L[y_i, \hat{r}_{RF}^{(i)}(x_i)],$$

where y_i are the target values, n the sample size and L the loss function of interest.

Prior to the RFC training in this project, the Principal Component Analysis (PCA) was used to reduce the high dimensionality of the problem without any major loss of information (Jolliffe, 1990).The PCA algorithm identifies from a set of variables its principal components (PC) that are those variables with high variance and uncorrelated between each other. The first PC is the variable with the highest variance from the set of variables; the second PC is the variable with the next highest variance, but subject to being uncorrelated to the first PC; the third PC is the next variable with the following highest variance and subject to being uncorrelated to the first and second PC, and so on (Jolliffe, 1990). At the end, instead of the n original variables, the resulting subset is reduced to a size m, which is at least of size $n - 1$. Another characteristic of the PCs is the amount of variance each one explained from the original set of variables, which is given by the eigenvalues of the PC (Jolliffe, 1990). For this project, the PCA algorithm was employed to safely reduce the dimensionality of the problem by enclosing the 90% of the variance between the resulting PCs.

A final consideration for this model is the modification of the final answer of the RFC by using flexible cutoff values. The RFC delivers a continuous output that represent the probability of class membership of a sample, which is a value between 0 and 1. The common way to translate this prediction into a binary response, belonging to class A or to class B, is using a default cutoff point of 0.5, where below 0.5 correspond to class A and above 0.5 to class B. However, this assumption is not necessarily correct for all cases; instead, to decide a different cutoff value based on the Receiver Operating Characteristics (ROC) curve, which is the visual representation for a binary classification between the FNR and FPR metrics, is preferrable (Fawcett, 2006). In order to avoid the visual exploration of a series of ROC curves, a grid of cutoff values is tested in the model limited in the range of $(0 - 0.5]$.

C. Gradient Boosting Classifier (GBC)

Gradient Boosting Classifier was introduced by Friedman (2001) as a way to improve the performance of weak learners, such as small individual decision trees. This approach shares similarities with Random Forest as it is an ensemble of decision trees; thus, preserving the low bias and variance properties (Efron & Hastie, 2016). However, they are not the same since Gradient Boosting tries to amend the errors of previous trees by giving higher weights to those samples that were misclassified (Friedman, 2001; Efron & Hastie, 2016). Gradient Boosting, in some cases, can be more powerful than Random Forest, but at a certain cost, such as the time-consuming need of parameter tuning and cross-validation rounds (Efron & Hastie, 2016). There are three parameters to tune for Gradient Boosting, which, depending on the values they receive, can change

considerably the final result of the model. These parameters are the number of trees (B), the maximum tree depth (d) and the shrinkage rate (ϵ). It is important to give conservative values for these parameters to decrease the potential risk of overfitting (Friedman, 2001). For example, high B, d and ϵ, which means a low shrinkage, could lead to overfitting the data. The GBC algorithm consists on an additive model where the target are the residuals (r) instead of the target variable (y), except in the first iteration where $r = y$. The first step is the initialization of the parameters B, d and ϵ, the initial fit at zero ($\hat{G}_o = 0$) and the residual vector equal to the target variable ($r = y$). At each iteration, given by the number of trees B, a decision tree of maximum depth d is fitted with the input variables X and the target value r. The fitted model $\tilde{g}_b$ is added to the fitted value $\hat{G}$ as a shrunken version, where $\tilde{g}_b$ is multiplied by the shrinking rate ϵ, which slows down the learning rate to avoid high validation set errors. Also, according to the predictions of the current tree, the residuals vector is updated by subtracting the samples that were correctly classified. In the following iterations, instead of working with the entire target variable y, they work only with those samples that are still incorrectly classified. Finally, all the individual classifiers created with the algorithm are combined using voting as a combination rule to form the final classifier (Friedman, 2001; Efron & Hastie, 2016). The GBC algorithm is presented in Algorithm 4. For this project, the GBC's parameters are designed as follows:

- Number of Trees (B) equal to 200 estimators.
- Maximum Tree Depth (d) between three and seven.
- Shrinking Rate (ϵ) between 0.02 to 0.16, with steps of 0.02.

Algorithm 4. Gradient Boosting Classifier.

1. For a given dataset $D = (X, y)$ with p variables, fix $m \leq p$, a B number of trees, a maximum tree depth d and a shrinking rate ϵ. Also, set the initial fit $\hat{G}_o = 0$ and the residual vector $r = y$.
2. For $b = 1$ to B:
    a. Using D, grow a tree T_b by doing the following steps for each terminal node of T_b, until the maximum tree size (d) is reached.
        i. For i in d:
            1. For m in p variables:
                - Calculate Gini Index (GI).
            2. Pick the variable m with the smallest GI: best split-point.
            3. Split the node into two daughter nodes.
        ii. Update the fitted model $\hat{G}_b$ with the shrunken tree T_b:
        $$\hat{G}_b = \hat{G}_{b-1} + (T_b \cdot \epsilon).$$
        iii. Update the residual vector r with the tree's T_b predictions.
        $$r_i = r_i + T_b(x_i), \;\; i = 1, \dots, n.$$
    b. Save each tree (T_b) predictions and the bootstrap samples for each of the training observations.
3. Return the fitted functions $\hat{G}_b$ fitted.

The estimation accuracy of this model is performed using k-fold cross-validation (CV). This technique is used to validate the accuracy of the predictive model using the same available data, that is to say, without the need of new data (Efron & Hastie, 2016). The k-folds CV segments the original dataset into k mutually exclusive subsets of the same size to provide a training set (D) and a testing set (D_T). For each iteration, given by the number of folds k, the testing set D_T is rotated, modifying, in turn, the training set D, which is the remaining data. Then, for every fold, the prediction probability for each test set is stored to, at the end, calculate the accuracy error based on a loss function of interest (Kohavi, 1995). For this project, based on the low occurrence of defects, five folds were tested to avoid the dilution of the defects, and, as a loss function, the misclassification error was used for being able to calculate the FNR and FPR scores.

For the experiments using GBC, the use of PCA as a dimensionality reduction strategy and flexible cutoff values to modify the model's final decision prevails with the same parameters: 90% of variance entrapment, and cutoff values in the range of $(0 - 0.5]$. Besides these characteristics, class weights are introduced to alleviate the class imbalance of the dataset, where the non-defective samples are the dominant class and the defective ones the non-dominant class. Class weights are employed to balance the classes by multiplying by a factor each of them (Scikit-learn, 2020). Normally, the dominant class remains unchanged, while the non-dominant class is modified by the specified class weight. For this model, class weights from 1 to 20, with steps of 5, for the non-dominant class are tested.

D. Low FNR and Low FPR Random Forest Classifier Ensemble (LFNR-LFPR-RFC)

The LFNR-LFPR-RFC is a custom-made multi-objective ensemble based on the performance goals of this project: minimize FNR and FPR. It is designed using a stacking strategy to train its basic learners, which are RFCs, and a weighted average as a combination rule. The stacking procedure consists on using a trained learner to combine the subsequent individual learners (Zhou, 2012). In more detail, stacking uses two types of learners, first-level learners and second-level learners, where the former is used to fit the original training set, while the latter fits the feedback set provided by the first-level learner. In this case, the first-level learner is a RFC that minimizes the FNR, and the second-level learner is a different RFC that boosts (i.e. amend the errors made by the first model, which are a high number of false positives) the high FPR created by the first model. After fitting the models, the individual decisions are combined using weighted averaging as a combination rule. This rule is selected when the outputs of the learners of the ensemble have a different importance (Zhou, 2012). In this case, the best results for both metrics are desired; however, due to the existing trade-off between both metrics, to minimize FNR is prioritized over FPR minimization. The algorithm of this ensemble is described in Algorithm 5, and the steps followed by the ensemble are presented below.

- Fit the original set with the first RFC, where the dependent variable is the target variable y that indicates the occurrence of a surface defect.
- Calculate the confusion matrix to identify the false positives generated by the first model. The new field created with the false positive indicator is named as y_{FP}.
- Fit the second RFC with the same independent variables and, instead of y, y_{FP} as the dependent variable.

- Combine the predictions of both RFC models using a weighted average as a combination rule, where the weights employ add up to one.
- Translate the combined prediction using the predefined cutoff value.

To complement this model, flexible cutoff values (within the range of $(0 - 0.5]$) and different class weights (1, 10, 100 and 1000), for the non-dominant class, are used to deal with the dataset imbalance. Also, as in the RFC models presented in this chapter, the RFC learners of this ensemble are designed with 400 individual trees, and they are restricted to have at least four samples per branch after a split, and at least two samples in each split. The accuracy estimation of this model is evaluated using a k-fold cross-validation with 5 folds to avoid the dilution of the defective products in each segment. For the weighted averaging, weights between 0.5 and 1 are tested for the first RFC model, and the weight complement $(1 - w)$ for the second RFC.

Algorithm 5. Low FNR and Low FPR Random Forest Classifier Ensemble.
For a given dataset $d = (X, y)$ with p variables, fix $m \leq p$ and a B number of trees. Also, state the weight for each RFC model to form the final ensemble with weighted average as a combination rule: w_1 and w_2, where $w_1 + w_2 = 1$.

1. For $b = 1$ to B:
    a. Apply Algorithm 3 to fit a RFC.
    b. Obtain the false positives generated by the first RFC to state the target of the second RFC (y_2).
    c. Apply the algorithm 3 to fit a second RFC with the new target y_2.
    d. Combine the predictions of both RFC ($\hat{p}_1$ and $\hat{p}_2$) using weighted averaging:

$$\hat{p} = w_1 \cdot \hat{p}_1 + w_2 \cdot \hat{p}_2.$$

For the dimensionality reduction strategy, PCA is no longer considered since, despite producing a smaller number of explanatory variables into a smaller number of PCs, these are still composed by a large number of original variables. This would still complicate translation of modeling insights into operational instructions. Therefore, a new supervised dimensionality reduction strategy oriented to the performance goals of this project (FNR and FPR) is proposed. This strategy consists on a variable preselection according to their ability to capture a certain portion of defective materials as "quick" as possible. The first step is to standardize the variables, beyond their normalization with the standard score (Z), by dividing each of them in a configurable number of categories (num_bins). After this standardization, in each category of a variable, the counting of the total number of samples along with the number of defective ones is made to further sort these categories in a descending manner based on the number of defective samples they contain. Furthermore, the accumulated sum of defective and non-defective samples captured by each sorted category is computed, in such a way that it allows to visualize at which point each variable is capable of capturing a configurable target quantity of defective samples indicated by a fraction (fnr_obj). The next step is to sort again the variables, but now in an ascendent manner, based on the number of non-defective samples that have to be marked to comply with the configurable target quantity of defective samples. This sorting guarantees that the variables that will be selected are the ones who present the best individual performance in efficiently capturing the defective samples. Finally, the number of variables desired for both RFC models in the ensemble

(num_vars_1 for the first RFC and num_vars_2 for the second RFC) are selected from the sorted variables list.

For example, let's suppose these parameters for the variables preselection: $num\_bins = 5$, $fnr\_obj = 0.01$, $num\_vars\_1 = 30$, $num\_vars\_2 = 40$ and a sample of $n = 100$ with 10 defective samples (n_{def}). First, the algorithm will divide each variable in five categories (using the *cut* function from the Python's library Pandas). For each category the number of defective samples is obtained to sort the categories in a descendent manner. Then, the accumulated sum of defective and non-defective samples is calculated to identify on which category the fnr_obj is accomplish. Supposing we have the distribution presented in Table 4 for a certain variable, based on the fnr_obj, the number of defective samples to capture is 9, given by $\#\,def = (1 - fnr\_obj) * n_{def} = (1 - 0.1) * 10 = 9$. For this case, the objective is accomplished in the second category, so a total of 36 non-defective samples are marked as defectives (false positives). Next, the variables are sorted again in a ascendent manner according to the number of false positives they produce. Finally, the first 30 variables are selected for the first RFC model, and 40 variables for the second RFC.

Table 4. An example of the distribution of a certain variable divided by categories.

# Cat.	Defective Samples	Non-defective Samples	Total Samples
1	6	14	20
2	3	22	25
3	1	14	15
4	0	30	30
5	0	10	10

For this preselection strategy, values between 4 and 12 are tested for num_bins, between 0.05 and 0.01 for fnr_obj, between 2 and 40 for num_vars_1, and between 2 and 100 for num_vars_2.

3.3 Variable Importance

The third specific objective is oriented to the importance each variable has over the defects' occurrence. This measurement aids to sort the list of variables used by each predictive model in order to identify which variables have the highest impact on the models' target. According to the technique used by the models to obtain the variables importance, it is possible to divide them into two groups: Odds Ratio technique, for the FS-LR, and Gini Importance, for the RFC, GBC and the ensemble model LFNR-LFPR RFC.

1) Odds Ratio

This technique is employed to assess the at how extent two events are related by calculating the ratios between the odds of the occurrence of event A given event B, and vice versa (Efron & Hastie, 2016). In the context of the effect a variable has over the defects' occurrence, the odds ratio calculates the defect probability ratio multiplier (PRM) that shows if the variable has a defect

enhancing role (PRM>1), a defect diminishing role (PRM<1) or no effect at all (PRM=1). In this project, the Odds Ratio (OR) is calculated for the Logistic Regression since this model does not consider interactions between the modeled variables, hence, the individual effects are the only available. Logistic Regression is already a modified version of the OR, where the β coefficients calculated by the Logit are the natural logarithm form of the OR:

$$\beta = \log\left\{\frac{p}{1-p}\right\},$$

where p is the binomial class probability estimator. Then, to calculate the OR for Logistic Regression (OR_{LR}), the β coefficients are raised to an exponential function, $PRM = OR_{LR} = e^{\beta}$. The last step is to sort the variables according to their effects over the defects' occurrence. However, a final transformation to the PRM is performed in order to be able to compare those PRM lesser than 1 to those greater than 1. This transformation (PRM_T) consists on adding to the PRM its reciprocal, that is to say

$$PRM_T = PRM + \frac{1}{PRM}.$$

2) Gini Importance

In models such as Random Forest and Gradient Boosting, applied to classification tasks, the variable importance is calculated based on their accumulated impurity decrease power according to the followed splitting criterion (Louppe, 2013; Nembrini, 2018). In a two-class classification context, impurity measures the correct labeling of the data in a binary class scenario, where purity is reached when said measurement results in a 0, indicating that all data belong to one class only (Louppe, 2013). In the context of Random Forest and Gradient Boosting classifiers, the splitting criterion used to measure the impurity decrease is the Gini Index (GI) presented in equation 1. Based on the GI, the impurity decrease is calculated with

$$\Delta i(\tau) = GI(\tau) - p_l \cdot GI(\tau_l) - p_r \cdot GI(\tau_r),$$

where $GI(\tau)$ is the Gini Index of node τ, $GI(\tau_l)$ and $GI(\tau_r)$ are the Gini Index of the left and right child nodes respectively, and p_l and p_r are the portion of samples belonging to each child node out of the total samples presented in node τ (Menze, 2009). Then, the variables' importance, also known as Gini Importance (Imp_G), is calculated by

$$Imp_G(X_m) = \sum_{b=1}^{B} \sum_{\tau \in \mathrm{T(b)}} \Delta i_{X_m,b}(\tau),$$

where $\Delta i_{X_m,b}(\tau)$ is the impurity decrease generated every time X_m is selected for a split at a certain node τ in the bth tree of the ensemble model (Breiman, 2001; Menze, 2009; Louppe, 2013; Nembrini, 2018), and T(b) is the set of all nodes belonging to the bth tree. In a few words, the Imp_G of a variable results from accumulating its impurity decrease achieved every time it is selected for a split. Finally, the variables' importance is presented as a fraction between 0 and 1 or

as a percentage, where the sum of all variables' importance must add up to 1 or 100%, depending on the case.

Chapter 4

Experimental Case

As stated above, the methodology presented in Chapter 3 is tested using an experimental case of defect occurrence prediction. The problem is briefly exposed in Chapter 1, and it consists of a binary classification problem to determine the presence of surface defects on galvanized steel coils. Two surface defects are modeled separately, bare spots and dross-derived defects, which are of low occurrence and unpredictable by the manufacturer, since they are identified once the steel coil has left the galvanizing line. Currently, the identification of defective materials occurs at the end of the line, where an operator visually inspects the steel coil before being rolled up for packaging. If a defective material is detected, it is submitted to a more detailed revision where the coil is accepted or rejected based on a list of specifications. If the coil is rejected, a corrective action is performed, where it can be washed and re-galvanized, or degraded to scrap. Both corrective actions generate undesired expenses that, right now, cannot be avoided due to the lack of further insights that can unleashed preventive actions in the future. For that reason, a classification system for early surface defects prediction, which in turn provides a list of possible variables involved in the defect occurrence for preventive actions design, is desired.

4.1 Dataset Overview

The assembled dataset used for this project covered the steel coils produced in a timespan of six months, which contains 6,837 entries, where each of them corresponds to a steel coil that passed through the galvanizing line. The dimensionality of the dataset is high since it contains 7,750 variables, which combine punctual features and time series coming from different sources in the galvanized steel coil production. Said sources comprise upstream processes, product segmentation tables, product chemistry tables, and the galvanizing process per se. As mentioned in Chapter 3, the time series from upstream processes and from the galvanizing line are decomposed in five statistics (mean, coefficient of variation, 5% percentile, 95% percentile and range) to trap the behavior of the complete series in punctual features. In addition, feature engineering is employed to create new variables suggested by experts in the topic. These new variables encompass the heat profile of the coils along the galvanizing line, differentials between adjacent coils in certain variables (galvanizing pot temperature, line speed, air nozzle distances, among others), the coil volume ($length \times width \times thickness$), the mass production per minute ($(volume \times density)/time$), chemical relationships, dew points' trends, and oxygen concentration trends. Furthermore, from the total of entries, 86 of them are catalogued as defective with bare spots and 58 with dross-derived defects, meaning that both defects are of low occurrence, hence, producing an imbalanced dataset.

The total entries of the dataset enclose six main products that are differentiated based on the steel family they belong (CQ, CR-HSLA, DS, EDDS, HR-HSLA, and REFOS), where such products can be further segmented into subgroups based on other characteristics such as their metallurgic practice, steel grade, width, length and thickness. Each product has different chemical

characteristics, and require distinct treatments in the production line, involving temperatures and line speeds, which makes hard to model them altogether. For that reason, for each surface defect, a set of clusters is designed, based on the product segmentation characteristics, to decrease the product variability and also concentrate the defect occurrence by removing all the products that have not presented defective materials so far. In the case of the bare spots, three clusters are created and named after the steel families they contain: CR-HSLA I, CQ I, and CR-HSLA II, CQ II, EDDS & DS. While for dross-derived defects, two clusters are designed: CR-HSLA, and CQ, EDDS & REFOS. In the bare spots' clusters, nine defects are left out, since, if they are included, the total size of the cluster increases approximately in 50% promoting the dilution of defective coils proportion. The size of the clusters along with the number of defective coils they contain is presented in Table 5. Table 6 exposes the detailed product segmentation specifications considered to build each cluster. Each of this clusters are modeled independently, thus, the final product of this project are five predictive models.

Table 5. Bare spots and dross-derived defects clusters' content.

Bare Spots Clusters			Dross-Derived defects Clusters		
Cluster	**# Coils**	**# Defective Coils**	**Cluster**	**# Coils**	**# Defective Coils**
CR-HSLA I	1,128	35	CR-HSLA	2,414	40
CQ I	772	23	CQ, EDDS & REFOS	1,282	18
CR-HSLA II, CQ II, EDDS & DS	1,333	19			

Table 6. Clusters' steel coils characteristics.
For confidentiality purposes, the clusters characteristics are censored, and only the number of different groups is presented. Cat.: Category

Defect	Cluster	Metallurgical Practice (MP)	Steel Grade (SG)	Width Range	Thickness Categories	Length Range
Bare Spots	CR-HSLA I	1 MP	1 SG	Cat. 6	Heavy	-
	CQ I	2 different MP	11 different SG	Cat. 6	Heavy	-
	CR-HSLA II, CQ II, EDDS & DS	12 different MP	5 different SG	-	-	Cat. 2
Dross-Derived	CR-HSLA	4 different MP	6 different SG	Cat. 3, Cat. 4, Cat. 5, Cat. 6	Light, Medium, Heavy	-
	CQ, EDDS & REFOS	8 different MP	5 different SG	Cat. 3, Cat. 4, Cat. 6	Light, Medium, Heavy	-

Finally, the clusters are submitted to a cleansing process where outliers are winsorized, blank cells are imputed (based on strategies described in Chapter 3), alphanumerical variables are label-encoded, constant fields are filtered out, and all the variables are normalized with the standard score (Z).

4.2 Predictive Models' specifications

As it was mentioned in Chapter 3, three off-the-shelf and one custom-made predictive models are chosen for this project. Such models are fitted with the data contained on each cluster designed for each surface defect. Hence, five different models were created in every experiment, except for the FS-LR where, instead of using the clusters, six different family-specific datasubsets are tested. The performance of all models is evaluated based on the FNR and FPR scores that indirectly measures the precision and recall accomplished by each model. An acceptance criterion, based on both scores, is determined by the metallurgical company, which consists on achieving at most 15% of FNR and 25% of FPR, where having a false negative is much more alarming and expensive than a false positive, which only implies only a moderate, although not negligible, additional effort. As a final remark, the four selected models are not tested in parallel, but they are chosen one after the another when the previous attempt fail to comply with the acceptance criterion. Finally, for each model, the FNR and FPR values are recorded and plotted in a scatter plot, and the important variables are enlisted and plotted in individual histograms.

Proceeding in order, FS-LR is the first attempted model, knowing that its performance would probably not be enough to meet the acceptance criteria, to set a performance baseline of the FNR and FPR scores as an initial exploration of the problem. In addition, FS-LR will provide the first insights on variables that may have a major effect on the defects' occurrence based on their probability ratio multiplier (MRP). Finally, since the order of the variables matters on the forward stepwise algorithm, the variables are arranged chronologically to select the earliest set of variables as possible to model the problem.

From this point on, the designed clusters are used to fit the models, and a set of scenarios comprising the different zones inside the galvanizing line are designed to see the prediction evolution throughout different spots in the line. In total, 27 scenarios are tested with different combinations of 18 zone choices (two sets of 8 zones comprising their setpoint variables and their feedback variables, and 2 zones without this variables' distinction). Table 7 gives a detailed description of the elements of each scenario.

Table 7. Scenarios description.
For confidentiality purposes, the real names of the zones are censored. s: setpoint variables; f: feedback variables.

Esc.	Zones	Esc.	Zones
0	Z1, Z2, Z3_s, Z4_s, Z5_s, Z6_s, Z7_s, Z8_s, Z9_s, Z10_s	14	Z1, Z2, Z3_f, Z4_f, Z5_f, Z6_s
1	Z1, Z2, Z3_s	15	Z1, Z2, Z3_f, Z4_f, Z5_f, Z6_f
2	Z1, Z2, Z3_s, Z4_s	16	Z1, Z2, Z3_f, Z4_s, Z5_s, Z6_s, Z7_s
3	Z1, Z2, Z3_s, Z4_s, Z5_s	17	Z1, Z2, Z3_f, Z4_f, Z5_s, Z6_s, Z7_s
4	Z1, Z2, Z3_s, Z4_s, Z5_s, Z6_s	18	Z1, Z2, Z3_f, Z4_f, Z5_f, Z6_s, Z7_s
5	Z1, Z2, Z3_s, Z4_s, Z5_s, Z6_s, Z7_s	19	Z1, Z2, Z3_f, Z4_f, Z5_f, Z6_f, Z7_s
6	Z1, Z2, Z3_f	20	Z1, Z2, Z3_f, Z4_f, Z5_f, Z6_f, Z7_f
7	Z1, Z2, Z3_f, Z4_s	21	Z1, Z2, Z3_f, Z4_f, Z5_f, Z6_f, Z7_f, Z8_s
8	Z1, Z2, Z3_f, Z4_f	22	Z1, Z2, Z3_f, Z4_f, Z5_f, Z6_f, Z7_f, Z8_f
9	Z1, Z2, Z3_f, Z4_s, Z5_s	23	Z1, Z2, Z3_f, Z4_f, Z5_f, Z6_f, Z7_f, Z8_f, Z9_s

10	Z1, Z2, Z3_f, Z4_f, Z5_s	24	Z1, Z2, Z3_f, Z4_f, Z5_f, Z6_f, Z7_f, Z8_f, Z9_f
11	Z1, Z2, Z3_f, Z4_f, Z5_f	25	Z1, Z2, Z3_f, Z4_f, Z5_f, Z6_f, Z7_f, Z8_f, Z9_f, Z10_s
12	Z1, Z2, Z3_f, Z4_s, Z5_s, Z6_s	26	Z1, Z2, Z3_f, Z4_f, Z5_f, Z6_f, Z7_f, Z8_f, Z9_f, Z10_f
13	Z1, Z2, Z3_f, Z4_f, Z5_s, Z6_s		

The following models (RFC, GBC and LFNR-LFPR RFC) are designed according to the parameters described in the previous Chapter. For these models the only new consideration is the prevention of the possible temporary effect the data may have. This is accomplished by shuffling the data before being fitted by the models.

4.3 Surface defects quality control

It is worth explaining the quality control process triggered by the identification of a defective material in surface defects by the galvanized products manufacturer. As stated before, the defective materials identification is made manually, where an operator at the end of the galvanizing line is visually inspecting the coils while they are passing by. This inspection seems naïve and questionable; however, the manufacturer trains their inspectors in advance in order to be able to recognize the different morphologies and characteristics of surface defects. Also, this quality control process does not end here, but it represents the tip of the iceberg that unleashes a robust process to validate the presence of the defect and, if present, the severity of it. This process is critical for the predictive models developed in this work, since they are trained with the company's available information, where the target variable is the classification made by the quality control department.

The quality control process begins with the detection of one or several surface defects in a coil at the end of the galvanizing line. This defective coil is marked in a computer program, along with the defect's qualitative description (defect localization and apparent severity magnitude), that creates a flag for the quality committee. This committee is in charge of double-checking the entire coil to validate the marked defect and search for others that were unnoticed. Then, the severity of the surface defects is evaluated according to specific internal metrics, and it is classified in a three-class system: mild, medium, severe. The characteristics of each class differ according to the assessed defect, which are presented in Table 8 for both surface defects. The final evaluation of a defective coil has two possible outputs: accepted or rejected, which depends on the severity of the defect along with the end use of the coil. For example, if the product will be employed for the automotive industry in the internal part of a car, mild defects can be accepted; however, if the coil is intended to be part of the exposed side of the car, the solely presence of the defect, despite its severity, is unacceptable. In total, four industries, with their respective subcategories, are distinguished: Automotive, industrial, domestic appliance, and construction. Finally, in spite of the final decision, whether accepted or rejected, the status of the coil regarding having defects stays on the database, which is the one used to identify in the predictive models of this project the defective materials.

Table 8. Bare Spots and Dross-Derived defects severity class characteristics.

Defect	Severity Class	Description
Bare Spots	Mild	Isolated defect, not perceptible to the touch, and not perceptible to painting test.
	Medium	Continuous defect, not perceptible to the touch, and not perceptible to painting test.
	Severe	Perceptible defect to the touch and to painting test.
Dross-Derived defects	Mild	Present in at most three isolated sections throughout the coil, and not perceptible relief (to the touch and to painting test).
	Medium	Present in at most ten isolated sections throughout the coil, and perceptible to the touch.
	Severe	Continuous defect throughout the coil and bulging grain on the coil's surface is observable with the naked eye.

Chapter 5

Results & Discussion

This Chapter is organized in a chronological fashion, where the first model's results and their discussion are presented together prior to showing the results of the following model, and so on. As mentioned in previous Chapters, the order of the models is the following: FS-LR, RFC, GBC, and LFNR-LFPR RFC.

5.1 Forward Stepwise Logistic Regression (FS-LR) Results

For each surface defect, the dataset is divided according to the six steel families that it comprises. In some cases, such divisions are merged since the number of defects for some of them is exactly one, being impossible for the model to run a test. Furthermore, steel families with no defects are not considered. Then, for bare spots, five steel families are tested, where three of them (CR-HSLA, CQ and EDDS) are analyzed individually, and other two where fused together (DS and REFOS) since one of them presents only one defective coil. On the other hand, for dross-derived defects, four steel families are tested individually (CR-HSLA, CQ, EDDS and REFOS). The distributions of total number of coils and total number of defective materials are given in Table 9.

Table 9. Coils distribution according of the steel family and the surface defect.

Defect	Steel Family	Number of coils	Number of defective coils
Bare Spots	CR-HSLA	3,323	48
	CQ	2,284	25
	EDDS	544	8
	DS / REFOS	690	5
Dross-Derived Defects	CR-HSLA	3,323	40
	CQ	2,284	4
	EDDS	544	9
	REFOS	455	5

The ROC curve for each group is obtained using a grid of cutoff values to portrait all the FNR-FPR scenarios from capturing all defects to not capturing a single defect at all. At each point of the cutoff values grid, the FNR and FPR scores are calculated with the confusion matrix obtained from the misclassification of the coils. The complete ROC curve is plotted as a continuous function, where a trade-off between the FNR and FPR scores is quite notorious. In addition to the ROC curve, the variable importance of the variables selected by the model is calculated with the Probability Ratio Multiplier (PRM). This PRM value denotes the impact the variable has over the analyzed defect, where a PRM of one represents no impact, lesser than one a defect's occurrence decremental effect, and greater than one a defect's occurrence incremental effect. Both results, the ROC curve and the PRM values, are presented for a group in each surface defect in Figure 5 (bare spot) and 6 (dross-derived defect).

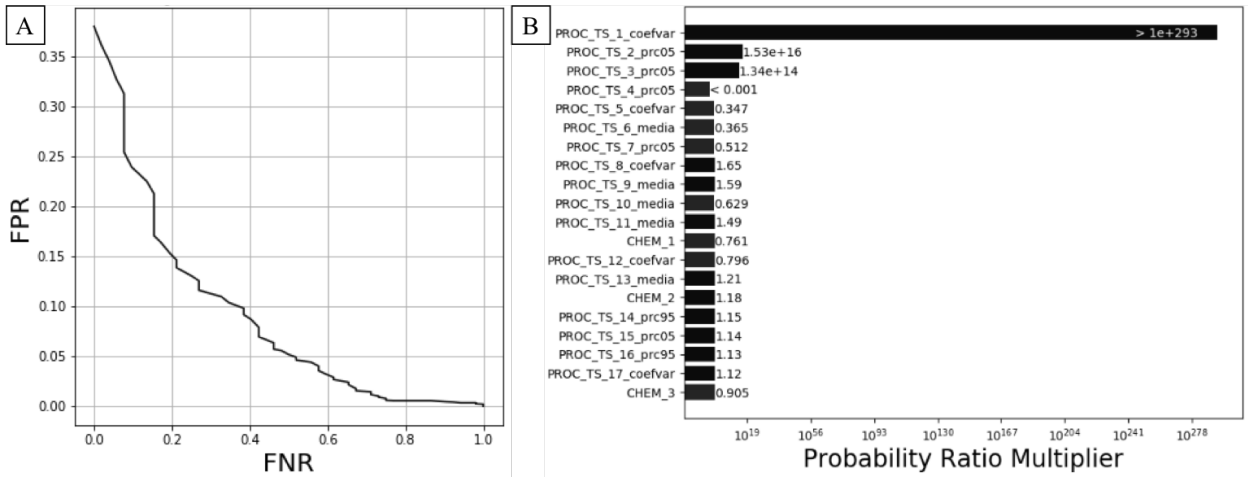

Figure 5. ROC curve and PRM values for the group CR-HSLA in Bare Spots defect.
A) ROC curve; B) PRM values. PROC_TS: Times series from the galvanizing line, CHEM: Steel coils chemistry.

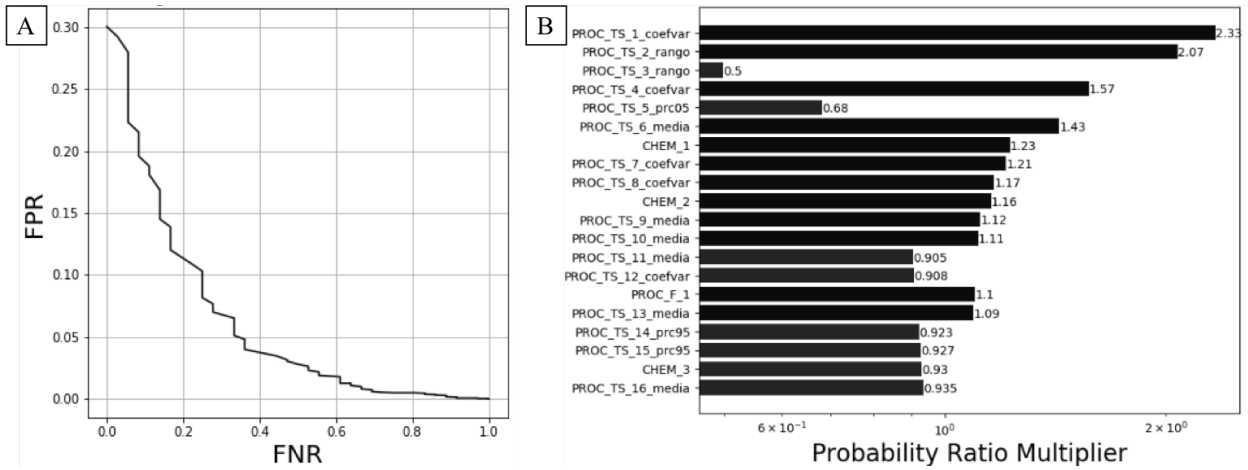

Figure 6. ROC curve and PRM values for the group CR-HSLA in Dross-Derived defects.
A) ROC curve; B) PRM values. PROC_TS: Times series statistics from the galvanizing line, CHEM: Steel coils chemistry, PROC_F: Features from the galvanizing line.

In some cases, the FS-LR models are capable of delivering performances below the acceptance criteria; however, it is important to remember that they are not submitted to a cross-validation framework, meaning that these initial results may be inflated. Nevertheless, these models are intended to provide a performance baseline, in terms of FNR and FPR scores, that is to say, they are used only as an initial exploration. By having this in mind, some conclusions are taken from these models. For instance, as it was mentioned above, the ROC curve presents a trade-off between both metrics as expected, where the points in the left side of the graph have low FNR but high FPR, while the points in the right side have the opposite behavior. On the other hand, talking about the PRM values of the variables selected by the FS-LR models, a combination of the coils' chemistry (CHEM), the time series statistics from the galvanizing line (PROC_TS) along with punctual features (PROC_F) from the same source are involved in the defects' occurrence. In the graphs depicted in Figure 5B and 6B, the variables are sorted in a descending manner according to the impact, whether positive or negative, they have over the appearance of defects. For instance,

to have a PRM of 0.5 (PROC_TS_3 range in Figure 6B) means that it can half the base defect rate, in this case of 1.2% $(40/3{,}323)$, so special attention to the range of this variable must be paid.

5.2 Random Forest Classifier (RFC) Results

In the RFC models, the database for each defect is segmented into clusters defined according to product segmentation characteristics, as defined in Table 6. The resulting distribution of total number of coils and total number of defective products for each cluster is presented in Table 5. For these models, besides the model per se, two new techniques are tested: the flexible cutoff value to modify the final decision of the RFC models, and the PCA as a dimensionality reduction strategy. Each technique is tested separately to evaluate their effect over the model's performance.

The cutoff value is analyzed by fixing the rest of parameters and techniques employed, such as the number of trees (400) and the PCA parameters (designed to explain 90% of variables' variability). A grid of different cutoff values is tested covering the range between 0 and 0.5. In order to evaluate the effect of this technique, a zoom-in in the first five cutoff values is illustrated for each of the clusters of bare spots in Figure 7, and for dross-derived defects in Figure 8. Table 10 puts together the FNR and FPR scores resulting from the tested cutoff values for all clusters. From these results it can be stated that the cutoff value has an effect on the trade-off between the FNR and FPR metrics, where the lower the cutoff, the lower the FNR, while the higher the cutoff, the lower the FPR. Therefore, the use of flexible cutoff values is a good asset to reach a balance between the two metrics to comply with the acceptance criteria defined by the manufacturer.

Table 10. FNR and FPR scores achieved with flexible cutoff values in RFC models for all clusters.

Bare Spots					**Dross-Derived defects**				
Cluster	**#**	**Cutoff**	**FNR**	**FPR**	**Cluster**	**#**	**Cutoff**	**FNR**	**FPR**
CR-HSLA I	1	0.025	25.7%	33.8%	CR-HSLA	1	0.025	37.5%	24.9%
	2	0.05	42.9%	20%		2	0.05	57.5%	11.1%
	3	0.075	57.1%	13.2%		3	0.075	67.5%	5.9%
	4	0.1	65.7%	10%		4	0.1	85.0%	3.2%
	5	0.125	71.4%	6.7%		5	0.125	90.0%	1.9%
CQ I	1	0.025	21.7%	34.4%	CQ, EDDS & REFOS	1	0.025	16.7%	16.7%
	2	0.05	26.1%	17.8%		2	0.05	33.3%	7.8%
	3	0.075	30.4%	10.9%		3	0.075	33.3%	4.7%
	4	0.1	43.5%	5.9%		4	0.1	33.3%	2.9%
	5	0.125	47.8%	4.1%		5	0.125	38.9%	1.4%
CR-HSLA II, CQ II, EDDS & DS	1	0.025	47.4%	20.9%					
	2	0.05	68.4%	8.7%					
	3	0.075	73.7%	4.0%					
	4	0.1	78.9%	1.7%					
	5	0.125	78.9%	1.2%					

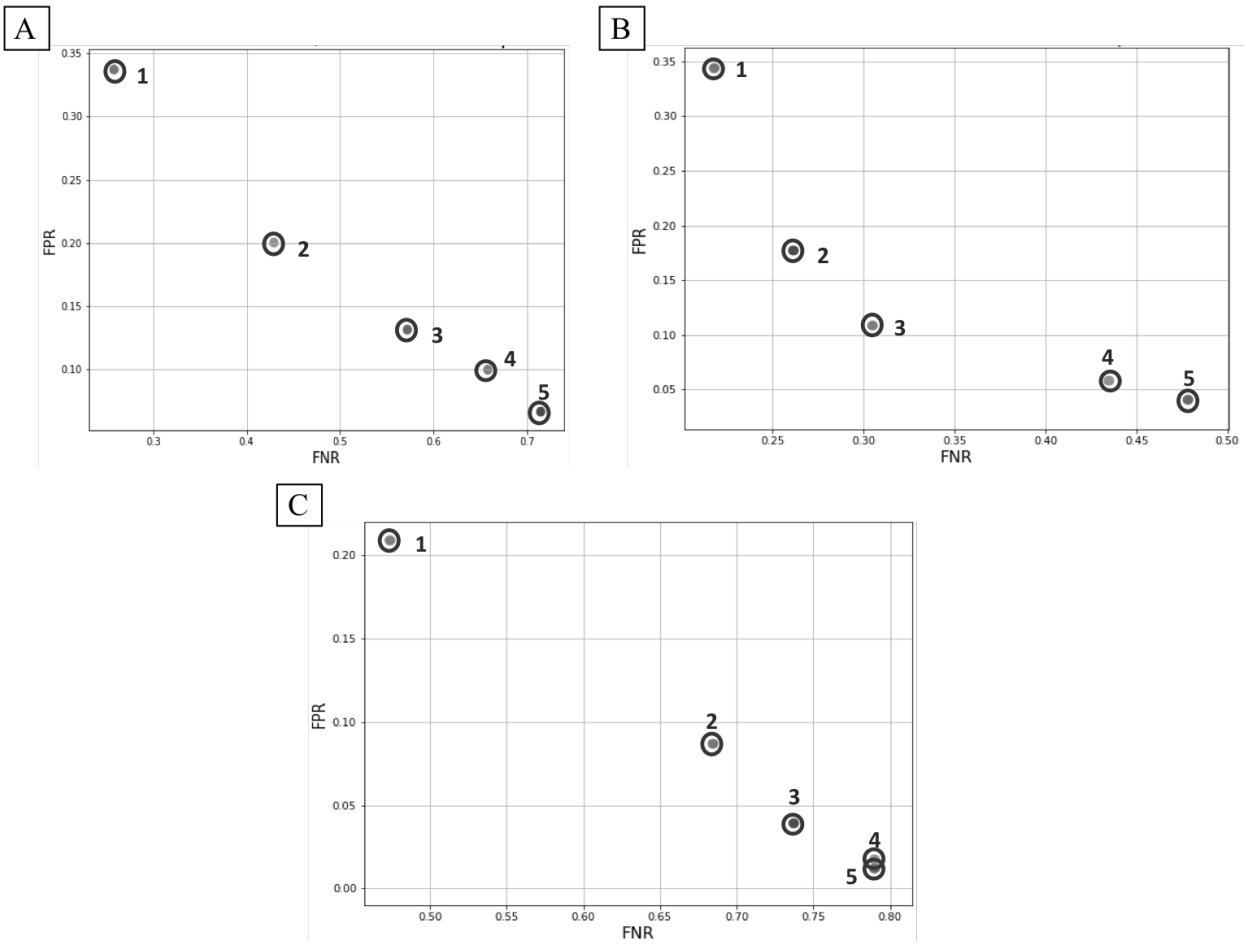

Figure 7. FNR and FPR scores resulting from modifying the cutoff values for the Bare Spots clusters. A) CR-HSLA I cluster, B) CQ I cluster, C) CR-HSLA II, CQ II, EDDS & DS cluster. The cutoff values presented in ascending order are: 0.025, 0.05, 0.075, 0.1 and 0.125.

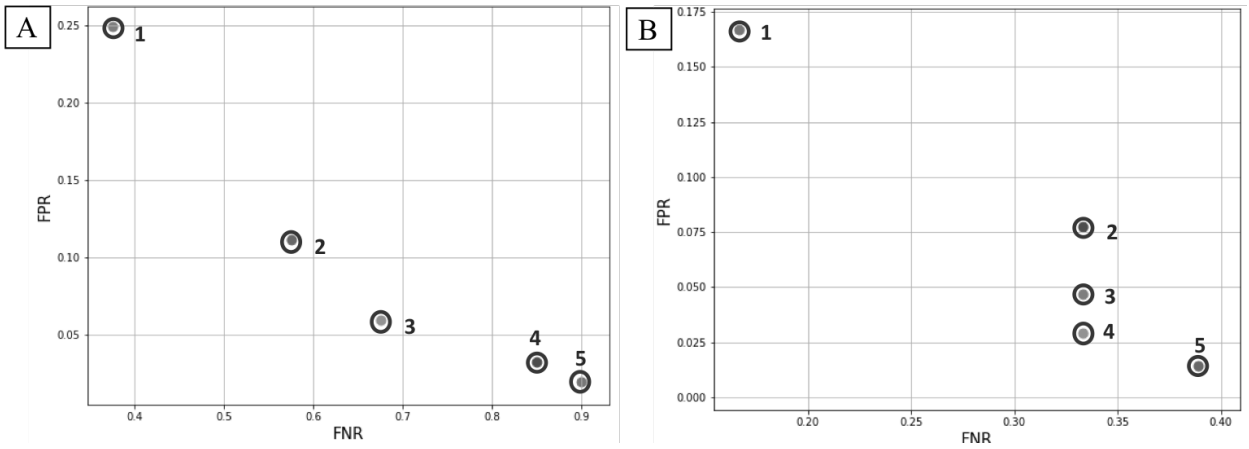

Figure 8. FNR and FPR scores resulting from modifying the cutoff values for the Dross-Derived defects clusters. A) CR-HSLA cluster, B) CQ, EDDS & REFOS. The cutoff values presented in ascending order are: 0.025, 0.05, 0.075, 0.1 and 0.125.

The other technique employed, PCA, is evaluated in the same manner as the cutoff values, i.e. by fixing the other parameters and techniques (cutoff values and number of trees) used in the RFC models. PCA is designed to obtain the principal components of a group of variables to an extent of being able to explain 90% of their variability. The variables, before being submitted to this technique, are grouped according to the zones from the galvanizing line to where they belong. This approach is followed to avoid the mixture of variables and to ease their interpretation in the variable importance analysis, since the PCA technique can be reverted to obtain the original variables that were compressed to be further analyzed. The effect of PCA over the performance of the RFC models is evaluated by comparing the FNR and FPR scores obtained by using and not using the PCA on the input data. Prior to this comparison, a grid of cutoff values is tested in the model to find the best solutions in terms of FNR and FPR. Then, the same parameters are maintained, but this time with PCA activated, to assess its effect. Figure 9 presents the results obtained for all clusters in both defects. The use of PCA replicates the performance of the RFC model without PCA in the FNR metric for three clusters (CQ I from Bare Spots, and CR-HSLA and CQ, EDDS & REFOS from Dross-Derived defects), while for the other two clusters, which belong to the bare spots' ones, for one cluster it improves the FNR score and for the other it worsens it. On the other hand, for the FPR metric, PCA's effect tends to increase it for four clusters (all except the CQ, EDDS & REFOS cluster from the Dross-Derived defect); however, the increment is not substantial. Apparently, PCA has no positive effect over the performance of the models, but, since a low dimensionality is desired for the next objective of the project (variables' importance), PCA is maintained.

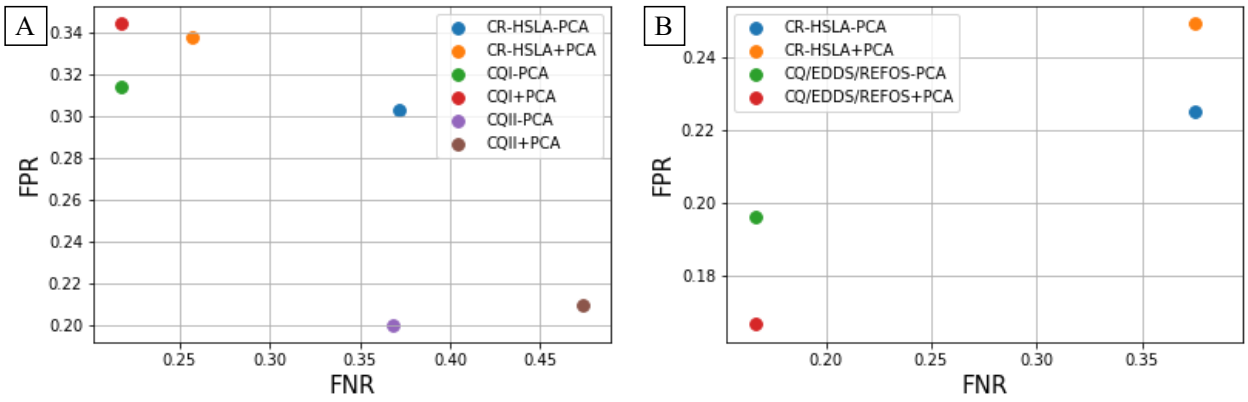

Figure 9. FNR and FPR scores for Bare Spots and Dross-Derived defects clusters by using and not using PCA.
A) Bare Spots clusters, B) Dross-Derived defects clusters. CQ II cluster in A accounts for the cluster CR-HSLA II, CQ II, EDDS & DS.

The best results obtained, by varying the cutoff values used, in FNR and FPR scores is presented in Table 11. At this point of the project, recalling the acceptance criteria ($FNR \leq 15\%$ and $FPR \leq 25\%$), in bare spots, none of the clusters could comply with the FPR criterion, while only CQ I is the only one below FNR criterion. Differently, dross-derived defects clusters do not comply with the FNR criterion, while only the cluster CQ, EDDS & REFOS presents a FPR score below 25%. Despite RFC models' characteristics suit properly with the attributes of our problem, so far, the performance results are not enough. Then, another yet similar approach has to be considered to enhance the model performance, in this case, the GBC model.

Table 11. Bare Spots and Dross-Derived defects clusters best results with RFC models.

Bare Spots			Dross-Derived defects		
Cluster	**FNR %**	**FPR %**	**Cluster**	**FNR %**	**FPR %**
CR-HSLA I	17%	34.4%	CR-HSLA	28%	19.3%
CQ I	13%	34.2%	CQ, EDDS & REFOS	17%	13%
CR-HSLA II, CQ II, EDDS & DS	21%	40.5%			

5.3 Gradient Boosting Classifier (GBC) Results

As it was mentioned in Chapter 3, GBC consists in a combination of decision trees ensembled differently than RFC does; however, both models possess the capacity to manage high-dimensional and imbalanced datasets. GBC is not tested in the first place due to the more extensive parameter tuning needed (maximum tree depth and shrinking rate), but, since RFC has failed to classify properly the clusters' data, it is chosen. For GBC models, the clusters' data subsets are tested again with flexible cutoff values and PCA as a dimensionality reduction strategy to explain the 90% of the variables' variability; additionally, different GBC's hyperparameters are analyzed: 200 estimators (B), a grid of maximum tree depth (d) between 3 and 7, and a grid of shrinking rates (ϵ) between 0.02 and 0.16 with a 0.02 step size.

These parameters and hyperparameters are iterated together to obtain the combination of them that could deliver the best results in term of the FNR and FPR scores. With this combination in hand, the next technique is introduced to analyze its effect over the performance metrics trade-off: Class Weights. As for the other techniques studied in the RFC models, the set of parameters and hyperparameters, besides the class weights, is fixed to test the class weights' effect. A grid of weights between 1 and 20 is used for the non-dominant class (defective materials' class), whereas the dominant class (non-defective materials' class) is maintained with a class weight of 1. The scatterplots with the results obtained for each cluster in both defects, bare spots and dross-derived defects, are presented in Figure 10 and 11 respectively. The FNR and FPR scores reached with these weight classes for all clusters is presented in Table 12. From the table and the scatter plots it can be concluded that class weights have a clear effect over the performance results of the models, where the higher the class weight for the non-dominant class, the lower the FNR, at expenses of sacrifice the FPR metric. Therefore, this technique stays as a part of the models to aid in the coping of the FNR and FPR scores.

Table 12. FNR and FPR scores achieved with class weights values in GBC models for all clusters.

Bare Spots					Dross-Derived defects				
Cluster	**#**	**Class Weights**	**FNR**	**FPR**	**Cluster**	**#**	**Class Weights**	**FNR**	**FPR**
CR-HSLA I	1	1:1	51.4%	12.9%	CR-HSLA	1	1:1	80%	5.6%
	2	1:5	17.1%	36.8%		2	1:5	30%	55.1%
	3	1:10	5.7%	53.2%		3	1:10	12.5%	90.9%
	4	1:15	5.7%	61.4%		4	1:15	5%	93.9%
	5	1:20	5.7%	69.6%		5	1:20	5%	95.3%

CQ I	1	1:1	39.1%	11.6%	CQ, EDDS & REFOS	1	1:1	33.3%	3.2%
	2	1:5	30.4%	27.8%		2	1:5	22.2%	9.3%
	3	1:10	21.7%	38.4%		3	1:10	22.2%	15.6%
	4	1:15	21.7%	41.8%		4	1:15	11.1%	17.9%
	5	1:20	21.7%	47.3%		5	1:20	5.5%	21.2%
CR-HSLA II, CQ II, EDDS & DS	1	1:1	84.2%	3.7%					
	2	1:5	36.8%	15%					
	3	1:10	26.3%	26.6%					
	4	1:15	21%	34.7%					
	5	1:20	21%	37.6%					

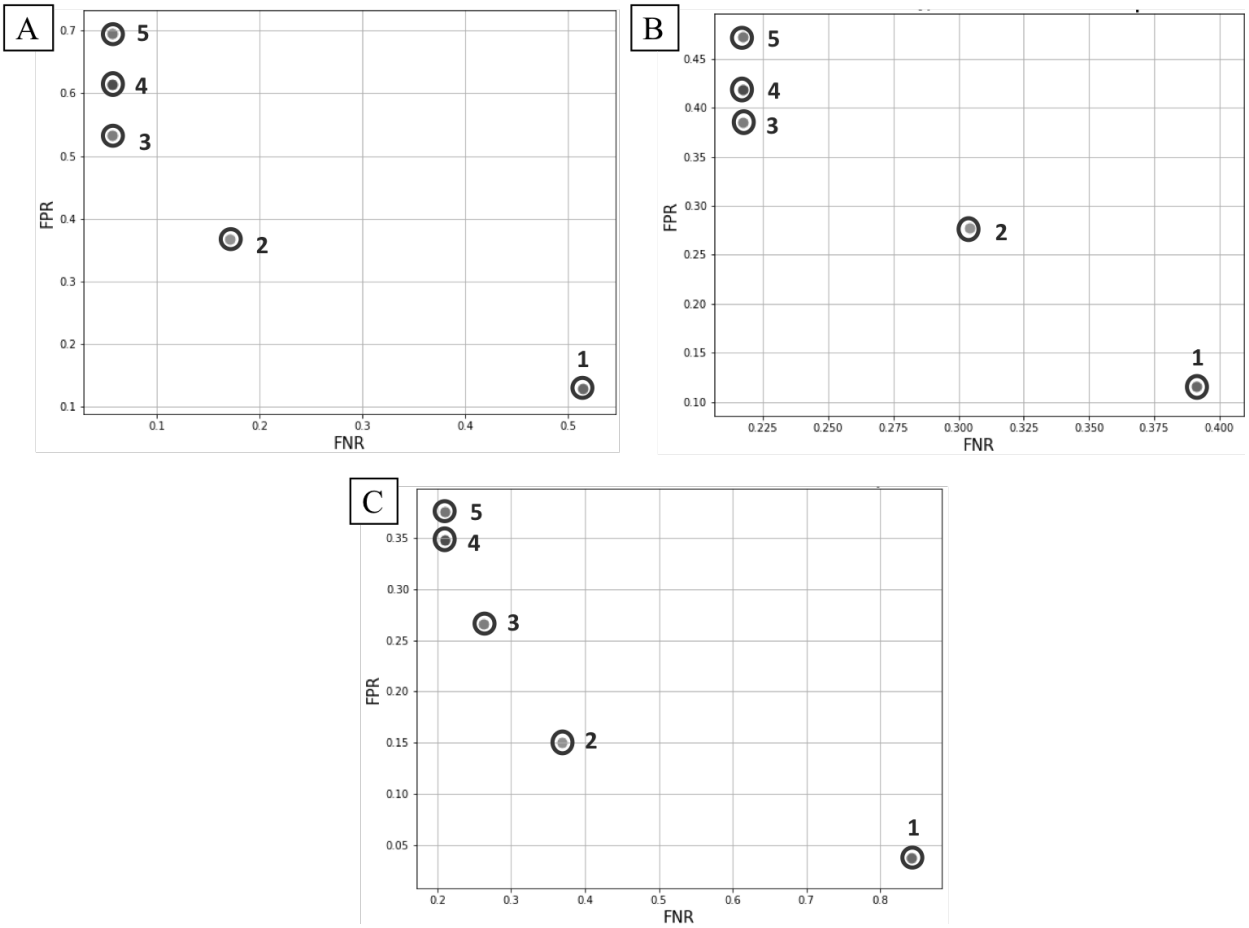

Figure 10. Class weights effect over FNR and FPR scores for the bare spots' clusters with the GBC model. A) CR-HSLA I cluster, B) CQ I cluster, C) CR-HSLA II, CQ II, EDDS & DS cluster.

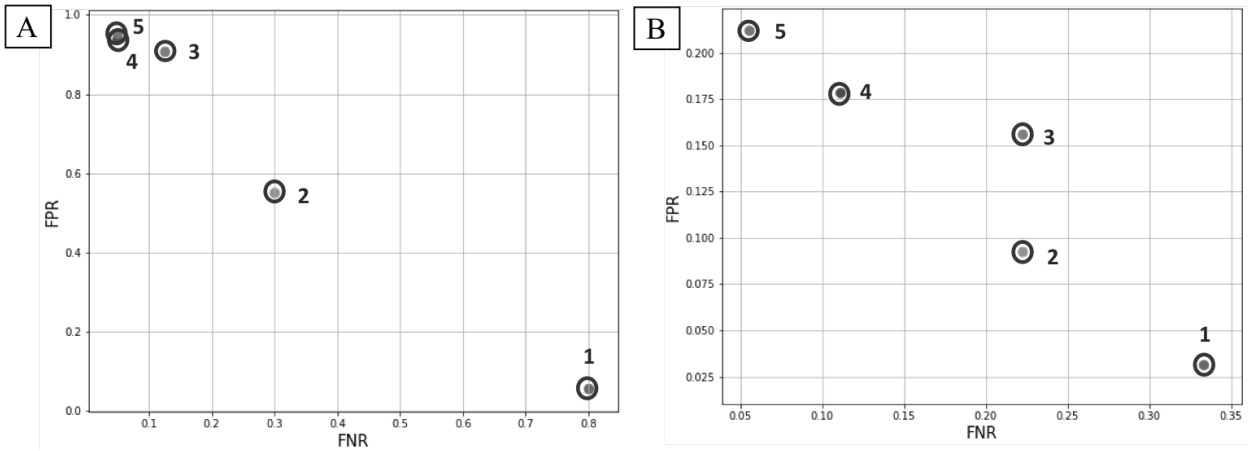

Figure 11. Class weights effect over FNR and FPR scores for the dross-derived defects' clusters with the GBC model. A) CR-HSLA cluster, B) CQ, EDDS & REFOS cluster.

The best results reached with the GBC models, in conjunction with the aforementioned techniques and parameters, for all clusters are portrayed in Table 13. For this model, GBC is able to obtain lower results in the FNR score than RFC for most of the clusters (CR-HSLA I from Bare Spots, and both clusters of Dross-Derived defects), inclusive, below the acceptance criterion of this metric. However, this decrement in FNR requires a considerable sacrifice in FPR, where only for the CQ, EDDS & REFOS cluster, the FPR metric is below the acceptance criterion. Therefore, the obtained results at this point are not enough, and another approach, this time oriented to deal with the trade-off between the FNR and FPR scores, may work better. Moreover, at that moment, the galvanizing manufacturer comment that the PCA strategy for dimensionality reduction will not be effective, since the number of selected variables by the models, after reverting the PCA process, are still too high for the next step of the project, the variables' importance for preventive actions designed. Consequently, along with the new model approach, a different variable selection process has to be explored to reduce the high dimensionality of the problem instead of PCA.

Table 13. Bare Spots and Dross-Derived defects clusters best results with GBC models.

Bare Spots			Dross-Derived defects		
Cluster	**FNR %**	**FPR %**	**Cluster**	**FNR %**	**FPR %**
CR-HSLA I	5.7%	53.2%	CR-HSLA	5%	93.9%
CQ I	21.7%	38.4%	CQ, EDDS & REFOS	5.6%	21.2%
CR-HSLA II, CQ II, EDDS & DS	21%	34.7%			

5.4 Low FNR and Low FPR Random Forest Classifier (LFNR-LFPR-RFC) ensemble Results

The LFNR-LFPR RFC ensemble is a multi-objective model whose goals are to minimize the FNR score as a primary goal, and to minimize the FPR score as a secondary goal. RFCs are chosen as base learners due to their ease of configuration, as they require much less hyperparameter tuning

than GBCs, where one is in charge of the first goal, and the other to the second one. Both RFC units communicate using a stacking approach where the results obtained by the first RFC are boosted by the second RFC, namely, the first model delivers a low FNR score, while the second, without permanently altering the FNR results achieved, tries to lower the FPR score. At the end, the decisions made by both learners are combined using weighted averaging as a combination rule. For this ensemble, a set of parameters are studied. For instance, a grid of cutoff values between 0 and 0.5, a grid of class weights for the first RFC model and a different grid for the second one between 1 and 1000 with 10 base steps, and ensemble weights for the weighted averaging between 0.5 and 1, prioritizing the effect of the first model, are used. In addition, instead of using PCA as a dimensionality reduction strategy, a custom-made, supervised variable preselection strategy oriented to the goals of the project is proposed. For such variable preselection scheme to be applied, a grid of normalization bins between 4 and 12, a grid of FNR's goal between 5% and 10%, and a grid for the number of variables for the first RFC model between 2 and 40, and a different grid for the second model between 2 and 100, are analyzed.

This set of parameters is varied for all clusters, and for each of them, the Pareto frontier is identified. Scatterplots with results for each cluster are presented in Figure 12 for Bare Spots' clusters, and in Figure 13 for Dross-Derived defects' clusters. Table 14 contains the FNR and FPR scores for all clusters. These results show that this ensemble in combination with the variable preselection strategy are capable of classifying the defective materials with a performance below the acceptance criteria. This is fulfilled for three clusters: CQ I and CR-HSLA II, CQ II, EDDS & DS for bare spots, and CQ, EDDS & REFOS for dross-derived defects. On the CR-HSLA I (Bare Spots) and CR-HSLA (Dross-Derived) clusters, the FNR criterion is met, which is a priority over the second goal. Besides this, they produce the lowest FPRs for an acceptable FNR observed throughout our experiments. The best results obtained with this model are presented in the next section of this Chapter, where a comparison between models is made.

Table 14. FNR and FPR scores achieved with the LFNR-LFPR RFC ensemble for all clusters.

Bare Spots				**Dross-Derived defects**			
Cluster	**#**	**FNR**	**FPR**	**Cluster**	**#**	**FNR**	**FPR**
CR-HSLA I	1	8.6%	30.5%	CR-HSLA	1	12.5%	35.4%
	2	14.3%	28.6%		2	17.5%	33.6%
	3	20%	23.5%		3	20%	31.1%
	4	22.9%	21.3%		4	22.5%	29%
CQ I	1	8.7%	15.7%	CQ, EDDS & REFOS	1	5.6%	6.4%
	2	17.4%	14.5%		2	11.1%	6.3%
	3	21.7%	13.9%		3	16.7%	4%
	4	34.8%	9.2%		4	38.9%	0.2%
CR-HSLA II, CQ II, EDDS & DS	1	5.2%	12.1%				
	2	10.5%	11.2%				
	3	15.8%	10.6%				
	4	21%	9.4%				

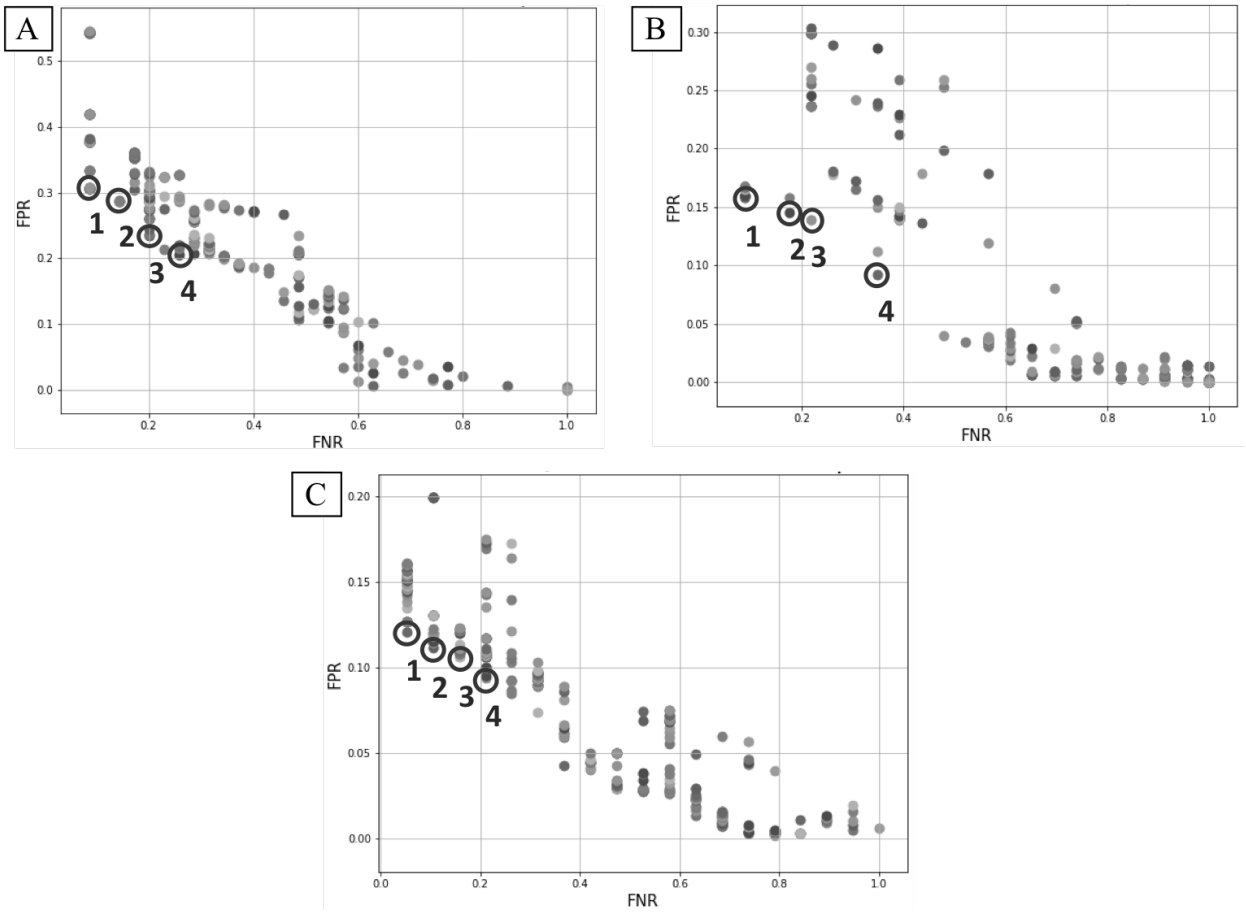

Figure 12. FNR and FPR scores for the bare spots' clusters with the LFNR-LFPR-RFC model.
The points encircled represent the Pareto front for each cluster. A) CR-HSLA I cluster, B) CQ I cluster, C) CR-HSLA II, CQ II, EDDS & DS cluster.

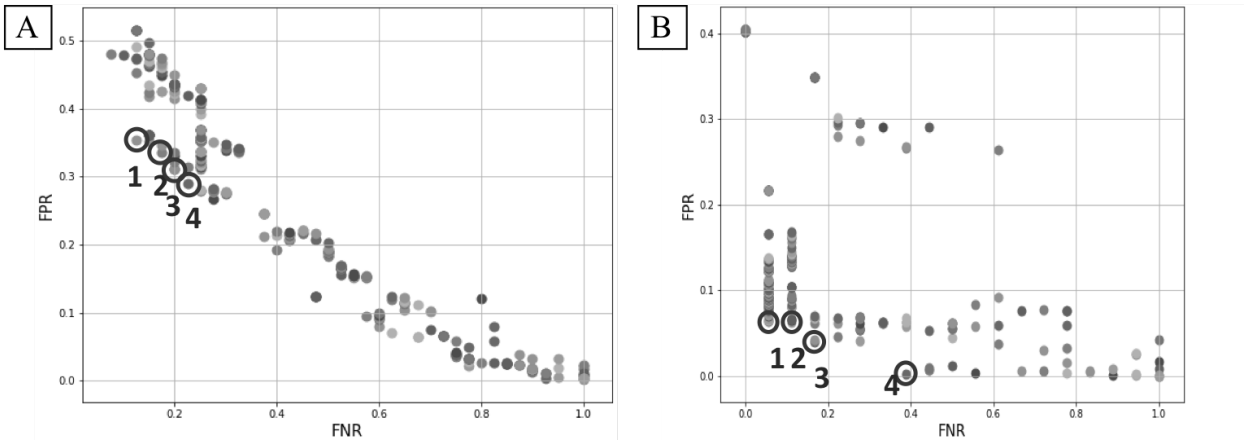

Figure 13. FNR and FPR scores for the dross-derived defects' clusters with the LFNR-LFPR-RFC model.
The points encircled represent the Pareto front for each cluster. A) CR-HSLA cluster, B) CQ, EDDS & REFOS cluster.

5.5 Models' Results Comparison

The best results for the last three models tested, RFC, GBC and LFNR-LFPR RFC, for each surface defect are presented in Table 15 and 16. The results written with a red font are the ones who could not meet the manufacturer's criteria, black font are for the ones who could meet the criteria, and the highlighted cells represent the best results of the cluster who also meet the criteria. For all clusters, the LFNR-LFPR RFC ensemble is able to outperform the others, achieving acceptable results, according to the acceptance criteria, in three out of five clusters: CQ I with 8.7% (FNR) and 15.7% (FPR), and CR-HSLA II, CQ II, EDDS & REFOS with 5.3% (FNR) and 12.1% (FPR) for bare spots, and CQ, EDDS & REFOS with 5.6% (FNR) and 6.4% (FPR) for dross-derived defects. For the remained two clusters, the LFNR-LFPR RFC ensemble is able to provide FNR scores below the acceptance criteria, and the lowest FPR possible, in comparison with the results of the other models who could provide and acceptable FNR score. Such clusters' results are: CR-HSLA I with 8.6% (FNR) and 30.6% (FPR) for bare spots, and CR-HSLA with 12.5% (FNR) and 35.4% (FPR) for dross-derived defects.

In addition, the LFNR-LFPR RFC ensemble not only outperform the off-the-shelf model tested, but also is capable of delivering these results at early stages in the galvanizing line. As it can be seen in Table 15, the scenario used to select the set of variables, which will compete to belong to the winner set used by the ensemble, is the number zero that comprises the values of the zone 1 and 2, along with the setpoint values of the rest of the zones (zones 3-10). In other words, the ensemble can predict the occurrence of a surface defect as soon as the setpoint values in the galvanizing line are decided depending on the products' characteristics, i.e., before the coil enters the galvanizing line per se.

Table 15. Models' best results for Bare Spots' clusters.

Cluster	# Samples	# Defects	Model	Scenario	FNR %	FPR %
CR-HSLA I	1,128	35	RFC	1	17%	34.4%
			GBC	6	5.7%	53.2%
			LFNR-LFPR RFC	0	8.6%	30.6%
CQ I	772	23	RFC	1	13%	34.2%
			GBC	1	21.7%	38.4%
			LFNR-LFPR RFC	0	8.7%	15.7%
CR-HSLA II, CQ II, EDDS & DS	1,333	19	RFC	1	21%	40.5%
			GBC	1	21%	34.7%
			LFNR-LFPR RFC	0	5.3%	12.1%

Table 16. Models' best results for Dross-Derived defects' clusters.

Cluster	# Samples	# Defects	Model	Scenario	FNR %	FPR %
CR-HSLA	2,414	40	RFC	6	28%	19.3%
			GBC	1	5%	93.9%
			LFNR-LFPR RFC	0	12.5%	35.4%
CQ, EDDS & REFOS	1,282	18	RFC	6	17%	13%
			GBC	1	5.6%	21.2%
			LFNR-LFPR RFC	0	5.6%	6.4%

5.6 FNR and FPR Confidence Intervals

The Confidence Intervals (CI) for the FNR and FPR scores obtained for each cluster by the LFNR-LFPR RFC ensemble, are calculated by applying one-parameter Neyman's construction and the Agresti-Coull interval. Both strategies are chosen since they are appropriate in the CI calculation for binomial distributions, which is the case of this project (Agresti & Coull, 1998; Efron & Hastie, 2016). One-parameter Neyman's construction fits, in the case of a binomial distribution, two binomial distributions ($f_{\hat{\theta}_{lo}}, f_{\hat{\theta}_{up}}$) at each side of the estimator ($\hat{\theta}$) that puts an $\alpha/2$ probability to the left and to the right of the $\hat{\theta}$ to find its confidence intervals. Said binomial distributions must comply with the equations

$$\int_{\hat{\theta}}^{1} f_{\hat{\theta}_{lo}}(r)dr = \frac{\alpha}{2}$$

and

$$\int_{-1}^{\hat{\theta}} f_{\hat{\theta}_{up}}(r)dr = \frac{\alpha}{2},$$

where $f_{\hat{\theta}_{lo}}$ is the distribution whose expected value is the lower bound of the confidence interval, $f_{\hat{\theta}_{up}}$ is the distribution whose expected value is the upper bound of the confidence interval, r is a dummy variable, α is the confidence level, and $\hat{\theta}$ is the estimator value. Such Neyman's intervals are given by the expected values ($\hat{\theta}_{lo}$, $\hat{\theta}_{up}$) of the left and right binomial distributions (Efron & Hastie, 2016). In the case of Agresti-Coull intervals, they are a modification of the standard CI that, instead of using the Wald CI

$$\hat{p} \pm Z_{\alpha/2}\frac{\sqrt{\hat{p}(1-\hat{p})}}{n},$$

they use the score confidence interval

$$\frac{\left(\hat{p} + \frac{Z_{\alpha/2}^2}{2n} \pm Z_{\alpha/2}\sqrt{\frac{\left[\hat{p}(1-\hat{p}) + \frac{Z_{\alpha/2}^2}{4n}\right]}{n}}\right)}{1 + \frac{Z_{\alpha/2}^2}{n}},$$

where $\hat{p}$ is the estimator, n is the sample size, and $Z_{\alpha/2}$ is the standard score give a α confidence level. The Agresti-Coull CI are calculated with

$$CI_{AC} = \tilde{p} \pm Z_{\alpha/2}\sqrt{\frac{\tilde{p}}{\tilde{n}}(1-\tilde{p})},$$

where

$$\tilde{p} = \frac{1}{\tilde{n}}\left(\hat{p} + \frac{Z_{\alpha/2}^2}{2}\right)$$

and

$$\tilde{n} = n + Z_{\alpha/2}^2 .$$

The confidence intervals with a $\alpha = 5\%$ for FNR and FPR scores for all clusters are presented in Table 17 and 18. For the FNR CIs, the interval is broad and asymmetrical, different from the FPR CIs. This effect happens since the n resulting for the FNR CIs calculation (false negatives + true positives) is of low magnitude, in comparison with the FPR CIs (false positives + true negatives) (Efron & Hastie, 2016).

Table 17. FNR and FPR Confidence Intervals for Bare Spots' clusters.

Cluster	FNR %	Neyman's CI	Agresti-Coull CI	FPR %	Neyman's CI	Agresti-Coull CI
CR-HSLA I	8.6%	[3.2, 23.1]	[2.2, 23.1]	30.6%	[27.9, 33.4]	[27.9, 33.3]
CQ I	8.7%	[2.8, 28]	[1.2, 28]	15.7%	[13.3, 18.6]	[13.3, 18.5]
CR-HSLA II, CQ II, EDDS & DS	5.3%	[1.3, 26]	[0, 26.5]	12.1%	[10.4, 14]	[10.4, 14]

Table 18. FNR and FPR Confidence Intervals for Dross-Derived defects' clusters.

Cluster	FNR %	Neyman's CI	Agresti-Coull CI	FPR %	Neyman's CI	Agresti-Coull CI
CR-HSLA	12.5%	[5.7, 26.8]	[5, 26.6]	35.4%	[33.5, 37.4]	[33.5, 37.3]
CQ, EDDS & REFOS	5.6%	[1.4, 27.2]	[0, 27.6]	6.4%	[5.2, 7.9]	[5.2, 7.9]

5.7 LFNR-LFPR RFC Variable Importance

As mentioned in Chapter 3, the variable importance is obtained with the Gini Importance, which is already included in the RFC Scikit-learn Python's library. The list of variables presented in this section corresponds only to the winner model in each cluster, which is the LFNR-LFPR RFC for all cases. This ensemble generates two lists of variables since it is composed of two RFC as basic learners. For each cluster, the larger subset always contains the smaller one, since both of them are selected with the same preselection strategy. However, the importance given to each variable differs depending on the RFC model used by the ensemble. To exemplify the list of variables and their corresponding histograms, only the high sensitivity (the RFC in charge of minimizing the FNR score) ones are presented.

The list of variables contains the name of the variables with their corresponding importance, the accumulated sum of importance, and the source of each variable relative to its localization in the galvanizing process. The name of the variables is coded for confidentiality issues of the company. As for the histograms, only a varied sample of three to four histograms is included. The histograms contained counts of defective and non-defective coils, where the non-defective coils are presented in orange, and the defective ones in gray. The left vertical corresponds to the number of non-

defective coils, while the right one corresponds to the non-defective. The right vertical is scaled up to represent the effect the variable has in a certain bin over the base defect rate. Then, where the orange bar is higher than the gray one, the variable in that range of values (given by the bin edges) presents a lower base defect rate. On the other hand, where the gray bar is higher than the orange one, the variable in that range of values presents a higher base defect rate.

Tables 19-23 and Figures 14-18 portray the list of variables with their importance and the sample of histograms for each of the clusters. The majority of the variables belong to setpoint variables' time series from the galvanizing process that are generated before the coil enters the line according to the coil's product characteristics. The second most common variable source is the chemistry of the coil, which is known also before the coil enters the galvanizing line. In summary, the variables selected by the variables' preselection strategy are not only capable of classifying the defective materials on their respective clusters, but also occur before the coil enters the galvanizing line. Therefore, it can be stated that the surface defect prediction for all clusters happens at an early stage of the galvanizing process. Then, there is a relatively large window of time to make preventive actions such as adjustments in the line processes to avoid the defect occurrence. Furthermore, these lists of variables provide insights on possible causes of the appearance of both surface defects, hence, future preventive actions can be proposed by experts in the galvanizing line to reduce their incidence and thus improve their products' quality.

Moreover, as the third specific objective states, these variables have to be not only important for the models but also actionable by the engineers in the galvanizing line. For that reason, at all time, these results are consulted and validated by experts in the line to be sure that these variables are capable to predict with a good performance the surface defects' occurrence, and provide sufficient insights in the decision-making process related to the design of preventive actions to decrease their incidences.

Table 19. List of variables with their importance for the CR-HSLA I Bare Spots' cluster.
PROC_TS: Time series statistic from the galvanizing process. PROC: Variables from the galvanizing process.

#	Variable	RF Importance	Accumulated Importance	Variable Type
1	PROC_TS_1_media	37.8%	37.8%	PROC
2	PROC_TS_2_media	28.9%	66.7%	PROC
3	PROC_TS_3_prc05	16.8%	83.5%	PROC
4	PROC_TS_4_prc05	16.5%	100%	PROC

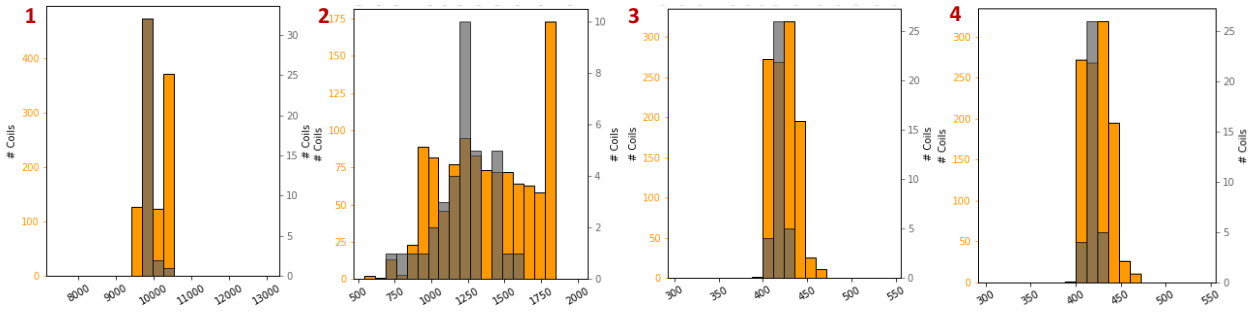

Figure 14. Histograms of a selection of variables used in the CR-HSLA I Bare Spots' cluster.
The numbers correspond to the order they have on the variables' importance list.

Table 20. List of variables with their importance for the CQ I Bare Spots' cluster. PROC_TS: Time series statistic from the galvanizing process. CHEM: Variable related to the chemical characteristics of the coil. PROC: Variables from the galvanizing process.

#	Variable	RF Importance	Accumulated Importance	Variable Type
1	CHEM_1	51.7%	51.7%	CHEM
2	CHEM_2	26.7%	78.4%	CHEM
3	PROC_TS_1_media	10.9%	89.3%	PROC
4	PROC_TS_1_prc05	10.7%	100%	PROC

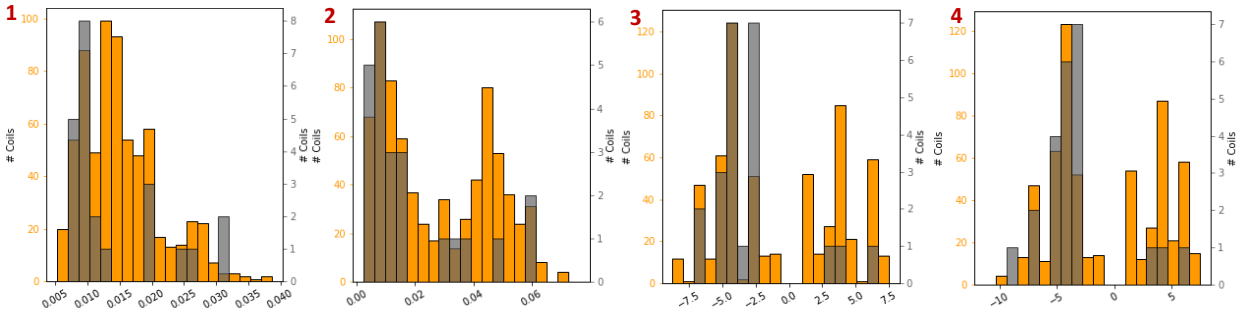

Figure 15. Histograms of a selection of variables used in the CQ I Bare Spots' cluster. The numbers correspond to the order they have on the variables' importance list.

Table 21. List of variables with their importance for the CR-HSLA II, CQ II, EDDS & DS Bare Spots' cluster. PROC_TS: Time series statistic from the galvanizing process. U-PROC: Variable related to upstream processes. PROC: Variables from the galvanizing process. SEGM: Variables related to the coil's product characteristics.

#	Variable	RF Importance	Accumulated Importance	Variable Type
1	PROC_TS_1_prc95	36.8%	36.8%	PROC
2	U-PROC_1	19%	55.8%	U-PROC
3	SEGM_1	16.8%	72.6%	SEGM
4	PROC_TS_2_prc95	11.5%	84.1%	PROC
5	U-PROC_2	8.6%	92.7%	U-PROC
6	SEGM_2	7.3%	100%	SEGM

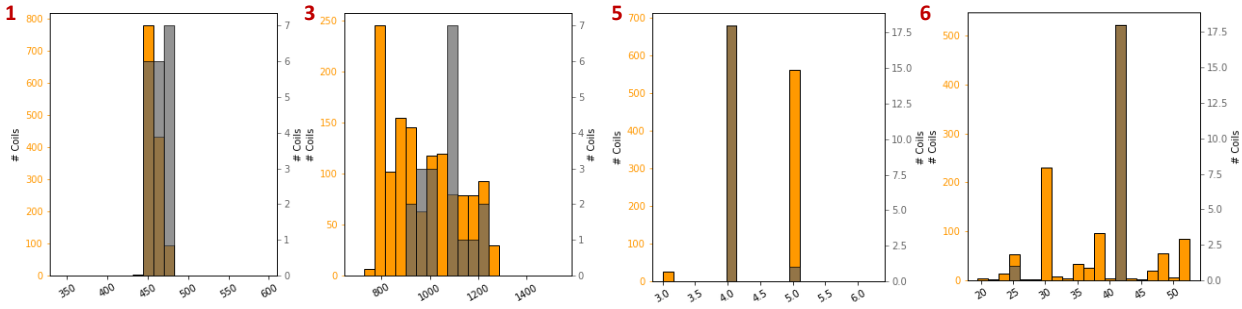

Figure 16. Histograms of a selection of variables used in the CR-HSLA II, CQ II, EDDS & DS Bare Spots' cluster. The numbers correspond to the order they have on the variables' importance list.

Table 22. List of variables with their importance for the CR-HSLA Dross-Derived defects' cluster. PROC_TS: Time series statistic from the galvanizing process. CHEM: Variable related to the chemical characteristics of the coil. PROC: Variables from the galvanizing process.

#	Variable	RF Importance	Accumulated Importance	Variable Type
1	PROC_TS_1_prc95	15.9%	15.9%	PROC
2	CHEM_1	14.2%	30.1%	CHEM
3	PROC_TS_2_prc95	14.1%	44.2%	PROC
4	CHEM_2	13.7%	57.9%	CHEM
5	PROC_TS_3_prc05	10.9%	68.8%	PROC
6	PROC_TS_4_prc05	8.6%	77.4%	PROC
7	PROC_TS_3_prc95	7.4%	84.8%	PROC
8	PROC_TS_4_media	7%	91.8%	PROC
9	CHEM_3	5.2%	97%	CHEM
10	CHEM_4	3%	100%	CHEM

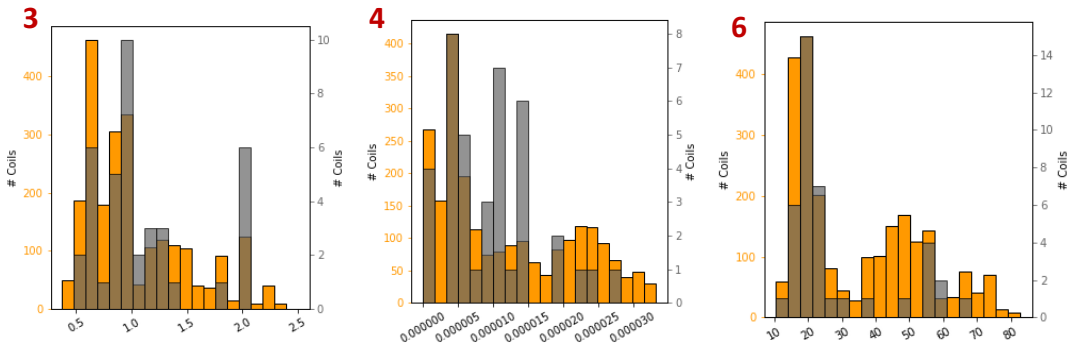

Figure 17. Histograms of a selection of variables used in the CR-HSLA Dross-Derived defects' cluster. The numbers correspond to the order they have on the variables' importance list.

Table 23. List of variables with their importance for the CQ, EDDS & REFOS Dross-Derived defects' cluster. PROC_TS: Time series statistic from the galvanizing process. CHEM: Variable related to the chemical characteristics of the coil. PROC: Variables from the galvanizing process.

#	Variable	RF Importance	Accumulated Importance	Variable Type
1	PROC_TS_1_media	25%	25%	PROC
2	PROC_TS_1_prc05	15.8%	40.8%	PROC
3	CHEM_1	11%	51.8%	CHEM
4	PROC_TS_2_prc95	10.3%	62.1%	PROC
5	PROC_TS_2_media	10.1%	72.2%	PROC
6	PROC_TS_3_prc95	9.9%	82.1%	PROC
7	PROC_TS_3_media	9.3%	91.4%	PROC
8	PROC_TS_3_prc05	8.6%	100%	PROC

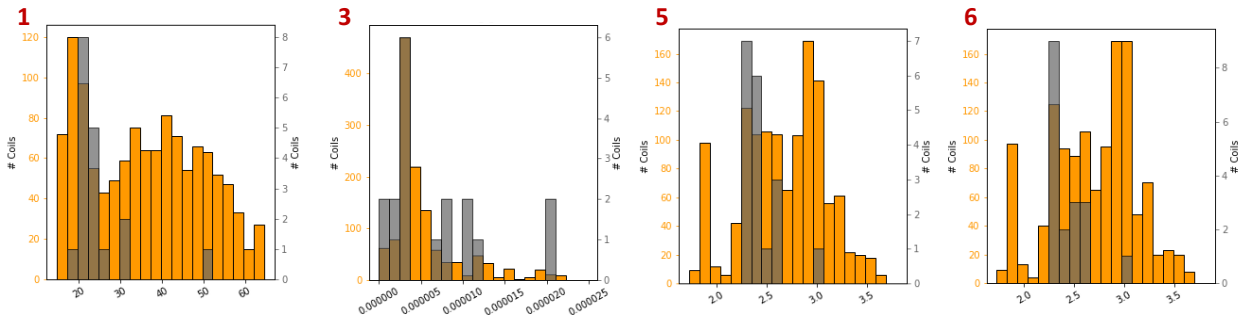

Figure 18. Histograms of a selection of variables used in the CQ, EDDS & REFOS Dross-Derived defects' cluster. The numbers correspond to the order they have on the variables' importance list.

5.8 Economic Impact

In Chapter 4, the quality control process executed in the presence of surface defects was described, where two possible outputs are possible: retaining the defective coil due to the severity of the damage or forwarding it to the packaging department before their shipment to the final client. Depending on the decision made for the coil, there are three possible scenarios:

1. To trim the defective sections. This happens when the affected section of the coil is isolated and small.
2. To reprocess the coil. This means that the coil is washed and re-galvanized.
3. To degrade the coil to scrap.

Despite the final destination of the coil, the three scenarios generate undesired costs, being the most alarming of them all the zinc waste derived from the galvanized sections affected.

From the comparison between the costs of a false negative (FN) and a false positive (FP), the FN cost is much more important since it means the coil will pass as non-defective, and it will arrive to the final client. If this event happens, more than the cost of replacing the defective coil, the image of the manufacturer along with the relationship with the client is at risk, meaning that sales may decrease in the near future.

A specific amount of the costs related to whichever of these events could not be calculated by the manufacturer, hence, at this point, it is not possible to calculate the future savings these predictive models may bring to the manufacturer.

Chapter 6

Conclusions & Future Work

A set of classification models based on decision trees approaches, such as the off-the-shelf RFC and GBC, along with the custom-made LFNR-LFPR RFC ensemble, were trained on occurrences of the bare spots and dross-derived surface defects in the company's galvanizing line. After analyzing their performances regarding FNR and FPR scores, we were able to confirm the hypotheses stated in this thesis. Specifically, the LFNR-LFPR RFC ensemble outperformed the single-objective models tested, it was capable of predicting defects at early stages in the line, and it was able of providing a list of important and actionable variables.

First, from the four models tested in this work, LFNR-LFPR RFC was able to outperform the others and comply, for the majority of the clusters, with the acceptance criteria for FNR and FPR scores defined by the galvanized products manufacturer. For Bare Spots' clusters, FNR scores below 9% were achieved for all of them (6 percentage points below the FNR acceptance criterion), while the FPR scores of two of them were below 16% (9 percentage points below the FPR acceptance criterion) and 30.6% in the remaining cluster. On the other hand, for Dross-Derived defects' clusters, the FNR scores obtained were of 12.5% and 5.6% (2 and 9 percentage points below the FNR acceptance criterion respectively), and FPR scores of 35.4% and 6.4% (18 percentage points below the FPR acceptance criterion), where the former one could not met the FPR acceptance criterion. Even though two FPR scores were out of the acceptance range, they were the smallest ones achieved by any of the modeling strategies for a manufacturer-accepted FNR threshold. Therefore, from the models tested, the custom-made LFNR-LFPR RFC ensemble presents the best performance in this problem for all the analyzed clusters. It is true that in two clusters the FPR criterion could not be met; however, these results, including the others that could met the specifications, are likely to improve with time since the ensemble could be retrained with datasets that comprises a larger defective sample.

Secondly, the LFNR-LFPR RFC ensemble showed the ability deliver acceptable predictions at early stages of the galvanizing line. For all the clusters created, the LFNR-LFPR RFC ensemble made their predictions with the scenario 0, which corresponds to setpoint variables from all the zones in the galvanizing line, along with variables form upstream processes (zone 1), that permits to know the steel coil status, related to the surface defects, before it even enters the galvanizing line. Therefore, this prediction gives a sufficient time gap to make adjustments or line modifications to avoid the surface defects' occurrence.

Lastly, the variables used by the LFNR-LFPR RFC ensemble demonstrate that they are not only capable of predicting the surface defects' occurrence, but also provide with insights of possible ranges of values that, when present, could increase the surface defects occurrence probability. Moreover, said list of variables, generated by the variables' preselection strategy and the LFNR-LFPR RFC ensemble model, comes from various sources that occur before the coil enters to the galvanizing line, such as setpoints from the galvanizing process, chemistry of the coils and variables from upstream processes. Since defect prediction happens at an early stage of the galvanizing line, a wide window of time to implement preventive actions according to the variables' importance is given to the galvanizing line engineers.

Therefore, despite the two clusters were the FPR criterion was not met, it is concluded that the objectives stated for this project are satisfied by the LFNR-LFPR RFC ensemble performance, and the hypothesis declared is accepted.

Finally, although not requested by the company, an economic impact assessment was carried out to evaluate the real impact the LFNR-LFPR RFC ensemble will represent to the company. From the company, an exact number could not be provided since the cost of the corrective actions employed currently has not being quantified by them. However, based on the predictive models' performance of reducing both the FNR and FPR, to save money is feasible.

For future work, both the LFNR-LFPR RFC ensemble model algorithm and the study of its economic impact can be improved. Of the ensemble, the current LFNR-LFPR RFC parameter tuning is slow and tedious since it has to be done manually. For the future, a more automated parameter selection system, such as a light-weight genetic algorithm, could be developed and adjusted to search for the optimal set of parameters for the ensemble model. In addition, this ensemble model could be generalized to apply it in whichever binary classification system, and, at the same time, test the robustness of the ensemble in different situations. On the other hand, the economic impact of this model specific to the surface defects problem could certainly be improved by making a thoroughly economic evaluation of the current costs derived from these defects, and the long-term economic impact the model represents for the company in terms of the false negative and false positives it generates.

References

Abbass, H. (2001). A Memetic Pareto Evolutionary Approach to Artificial Neural Networks. AI 2001: Advances In Artificial Intelligence, 2256, 1-12. doi:10.1007/3-540-45656-2_1

Agresti, A., & Coull, B. A. (1998). Approximate is Better than "Exact" for Interval Estimation of Binomial Proportions. The American Statistician,52(2), 119-126. doi:10.1080/00031305.1998.10480550

Ajersch, F., Ilinca, F., & Hétu, J. -. (2004). Simulation of flow in a continuous galvanizing bath: Part II. transient aluminum distribution resulting from ingot addition. Metallurgical and Materials Transactions B: Process Metallurgy and Materials Processing Science, 35(1), 171-178. doi:10.1007/s11663-004-0107-4

Ajersch, F., Ilinca, F., & Goodwin, F. E. (2013). Numerical simulation of flow, temperature and composition variation in the snout and sink roll region of continuous galvanizing baths. Paper presented at the Materials Science and Technology Conference and Exhibition 2013, MS and T 2013, , 2 1145-1152.

Ajersch, F., & Ilinca, F. (2018). Review of modeling and simulation of galvanizing operations. Steel Research International, 89(1) doi:10.1002/srin.201700074

Azimi, A., Ashrafizadeh, F., Toroghinejad, M. R., & Shahriari, F. (2012a). Metallurgical analysis of pimples and their influence on the properties of hot dip galvanized steel sheet. Engineering Failure Analysis, 26, 81-88. doi:10.1016/j.engfailanal.2012.05.026

Azimi, A., Ashrafizadeh, F., Toroghinejad, M. R., & Shahriari, F. (2012b). Metallurgical assessment of critical defects in continuous hot dip galvanized steel sheets. Surface and Coatings Technology, 206(21), 4376-4383. doi:10.1016/j.surfcoat.2012.04.062

Bellhouse, E. M., & McDermid, J. R. (2008). Analysis of the fe-zn interface of galvanized high al-low si TRIP steels. Materials Science and Engineering A, 491(1-2), 39-46. doi:10.1016/j.msea.2007.12.033

Bennett, K., & Campbell, C. (2000). Support vector machines. ACM SIGKDD Explorations Newsletter, 2(2), 1-13. doi:10.1145/380995.380999

Breiman, L. (2001). Random Forests. Machine Learning,45(1), 5-32. doi:10.1023/a:1010933404324

Castin, L., & Frénay, B. (2018). Clustering with Decision Trees: Divisive and Agglomerative Approach. ESANN.

Chandra, A., & Yao, X. (2004). DIVACE: Diverse and Accurate Ensemble Learning Algorithm. Lecture Notes In Computer Science, 3177, 619-625. doi:10.1007/978-3-540-28651-6_91

Chen, H., & Yao, X. (2006). Evolutionary Multiobjective Ensemble Learning Based on Bayesian Feature Selection. 2006 IEEE International Conference On Evolutionary Computation, 267-274. doi:10.1109/cec.2006.1688318

Chen, H., & Yao, X. (2010). Multiobjective Neural Network Ensembles Based on Regularized Negative Correlation Learning. IEEE Transactions on Knowledge and Data Engineering, 22(12), 1738-1751. doi:10.1109/tkde.2010.26

Chen, W., Tseng, L., & Wu, C. (2014). A unified evolutionary training scheme for single and ensemble of feedforward neural network. Neurocomputing,143, 347-361. doi:10.1016/j.neucom.2014.05.057

Cohen, J., Cohen, P., West, S. G., & Aiken, L. S. (2015). Applied multiple regression/correlation analysis for the behavioral sciences. New York: Routledge.

D'Agostino, R. B., Belanger, A., & D'Agostino, R. B., Jr. (1990). A suggestion for using powerful and informative tests of normality. American Statistician, 44(4), 316-321. doi:10.1080/00031305.1990.10475751

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions On Evolutionary Computation, 6(2), 182-197. doi:10.1109/4235.996017

Dietterich, T. G. (2000). Ensemble methods in machine learning doi:10.1007/3-540-45014-9_1

Dixon, W. (1960). Simplified Estimation from Censored Normal Samples. The Annals of Mathematical Statistics,31(2), 385-391.

Dubois, M., & Bordignon, L. (2018). Process window for pre-oxidation for a full radiant tube furnace. Paper presented at the AISTech - Iron and Steel Technology Conference Proceedings, , 2018-May 2201-2212.

Efron, B., & Hastie, T. (2016). Computer age statistical inference. New York, NY, USA: Cambridge University Press.

Eiben A.E., Smith J.E. (2003) What is an Evolutionary Algorithm?. In: Introduction to Evolutionary Computing. Natural Computing Series. Springer, Berlin, Heidelberg. doi:10.1007/978-3-662-05094-1_2

Elsevier. (2020). How Scopus works. Retrieved September 10, 2020, from https://www.elsevier.com/solutions/scopus/how-scopus-works

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861-874. doi:10.1016/j.patrec.2005.10.010

Félix, G. S., Sellin, N., & Marangoni, C. (2013). Automatic control system for snout positioning in hot dip galvanization process. Chemical Engineering Transactions, 32, 1333-1338. doi:10.3303/CET1332223

Frank, E., Witten, I. H. & Hall, M. A. (2016). Data mining: Practical machine learning tools and techniques. Amsterdam: Elsevier.

Friedman, J. (2001). Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics, 29(5), 1189-1232.

Gelman, A., & Hill, J. (2018). Data analysis using regression and multilevel/hierarchical models. Cambridge: Cambridge Univ. Press.

Gu, S., & Jin, Y. (2014). Generating diverse and accurate classifier ensembles using multi-objective optimization. 2014 IEEE Symposium On Computational Intelligence In Multi-Criteria Decision-Making (MCDM), 9-15. doi:10.1109/mcdm.2014.7007182

Haimes, Y. Y., Li, D., & Tulsiani, V. (1990). Multiobjective Decision-Tree Analysis. Risk Analysis, 10(1), 111-127. doi:10.1111/j.1539-6924.1990.tb01026.x

Henrich, F.F., Jannasch, O., Daldrop, J., Arikan, S. & Fabina, M.. (2019). Classifying defects more reliably. Steel + Technology, 4, 64-67.

Hohil, M. E., Liu, D., & Smith, S. H. (1999). Solving the N-bit parity problem using neural networks. Neural Networks,12(9), 1321-1323. doi:10.1016/s0893-6080(99)00069-6

Hosmer, D. W., Wang, C., Lin, I., & Lemeshow, S. (1978). A computer program for stepwise logistic regression using maximum likelihood estimation. Computer Programs in Biomedicine,8(2), 121-134. doi:10.1016/0010-468x(78)90047-8

Ilinca, F., Hétu, J. -., & Ajersch, F. (2004). Numerical simulation of al and fe distribution during continous galvanizing operations. Paper presented at the Galvatech '04: 6th International Conference on Zinc and Zinc Alloy Coated Steel Sheet - Conference Proceedings, 1067-1078.

Jiang, H. M., Chen, X. P., Wu, H., & Li, C. H. (2004). Forming characteristics and mechanical parameter sensitivity study on pre-phosphated electro-galvanized sheet steel. Journal of Materials Processing Technology, 151(1-3), 248-254. doi:10.1016/j.jmatprotec.2004.04.069

Jiang, S. M., Feng, S. J., Li, Z. H., & Zhang, Q. F. (2014). Influence of oxide morphologies on the galvanizability of the third generation automotive steel doi:10.4028/www.scientific.net/AMR.887-888.233

Jolliffe, I. T. (1990). Principal Component Analysis: A beginners guide - I. Introduction and application. Weather,45(10), 375-382. doi:10.1002/j.1477-8696.1990.tb05558.x

Kocev, D., Vens, C., Struyf, J., & Džeroski, S. (2007). Ensembles of Multi-Objective Decision Trees. Machine Learning: ECML 2007 Lecture Notes in Computer Science, 624-631. doi:10.1007/978-3-540-74958-5_61

Kohavi, R.. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the 14th International Joint Conference on Artificial Intelligence, 2, 1137–1143.

Krogh, A. & Hertz, J. A. (1992). A Simple Weight Decay Can Improve Generalization. Advances in Neural Information Processing Systems, 4, 950-957.

Kuhn, M., & Johnson, K. (2020). Feature engineering and selection a practical approach for predictive models. Boca Raton: Chapman & Hall/CRC.

Li, Y., & Sun, J. L. (2014). Recognition and control of the influence factors on the surface defects of cold rolled strip with emulsion lubrication doi:10.4028/www.scientific.net/AMM.456.498

Liu, Y., & Yao, X. (1999). Ensemble learning via negative correlation. Neural Networks, 12(10), 1399-1404. doi:10.1016/s0893-6080(99)00073-8

Louppe, G., Wehenkel, L., Sutera, A. & Geurts, P. (2013). Understanding variable importances in forests of randomized trees. Proceedings of the 26th International Conference on Neural Information Processing Systems, 1, 431-439. doi:10.5555/2999611.2999660

Luo, Q., & He, Y. (2016). A cost-effective and automatic surface defect inspection system for hot-rolled flat steel. Robotics and Computer-Integrated Manufacturing, 38, 16-30. doi:10.1016/j.rcim.2015.09.008

Luo, Q., Fang, X., Liu, L., Yang, C., & Sun, Y. (2020). Automated visual defect detection for flat steel surface: A survey. IEEE Transactions on Instrumentation and Measurement, 69(3), 626-644. doi:10.1109/TIM.2019.2963555

Makhtar, M., Awang, M.K., Rahman, M.N., Fadzli, S.A., & Mohamad, M. (2015). Optimizing sensitivity and specificity of ensemble classifiers for diabetic patients. Journal of Theoretical and Applied Information Technology, 82(2), 230-236.

Menze, B. H., Kelm, B. M., Masuch, R., Himmelreich, U., Bachert, P., Petrich, W., & Hamprecht, F. A. (2009). A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC Bioinformatics, 10*(1), 213. doi:10.1186/1471-2105-10-213

Milligan, G. W., & Cooper, M. C. (1988). A study of standardization of variables in cluster analysis. Journal of Classification, 5(2), 181-204. doi:10.1007/BF01897163

Neal, R. (1996). Bayesian Learning for Neural Networks. Lecture Notes In Statistics, 118, 17-19; 126-139. doi:10.1007/978-1-4612-0745-0

Nembrini, S., König, I. R., & Wright, M. N. (2018). The revival of the Gini importance? Bioinformatics,34(21), 3711-3718. doi:10.1093/bioinformatics/bty373

Ren, Y., Zhang, L., & Suganthan, P. N. (2016). Ensemble classification and regression-recent developments, applications and future directions. IEEE Computational Intelligence Magazine, 11(1), 41-53. doi:10.1109/MCI.2015.2471235

Sahoo, P., Das, S. K., & Paulo Davim, J. (2017). Surface finish coatings. Comprehensive materials finishing (pp. 38-55) doi:10.1016/B978-0-12-803581-8.09167-0

Scikit-learn Community. (n.d.). Scikit-learn. Retrieved November 10, 2020, from https://scikit-learn.org/stable/index.html

Scipy Community. (2020). SciPy. Retrieved November 9, 2020, from https://docs.scipy.org/doc/scipy/reference/index.html

Sirin, S. Y. (2020). Drying of steel pipes at a hot-dip galvanizing plant by induction heating. Materialpruefung/Materials Testing, 62(3), 291-298. doi:10.3139/120.111485

Sun, S., Wei, Z., & Lu, H. (2017a). Fluid flow, heat and mass transfer in hot dip galvanizing bath. Xitong Fangzhen Xuebao / Journal of System Simulation, 29(7), 1538-1545. doi:10.16182/j.issn1004731x.joss.201707019

Sun, S., Wei, Z., Dai, X., & Lu, H. (2017b). Research on the characteristics of physical field of molten zinc in hot dip galvanizing pot. Jixie Qiangdu/Journal of Mechanical Strength, 39(2), 404-409. doi:10.16579/j.issn.1001.9669.2017.02.027

Tan, C., Lim, C., & Cheah, Y. (2013). A Modified micro Genetic Algorithm for undertaking Multi-Objective Optimization Problems. Journal Of Intelligent & Fuzzy Systems, 24(3), 483-495. doi:10.3233/ifs-2012-0568

Tan, C., Lim, C., & Cheah, Y. (2014). A multi-objective evolutionary algorithm-based ensemble optimizer for feature selection and classification with neural network models. Neurocomputing, 125, 217-228. doi:10.1016/j.neucom.2012.12.057

Tipping, M. (2000). Sparse bayesian learning and the relevance vector machine. The Journal Of Machine Learning Research, 1, 211-244. doi:10.1162/15324430152748236

Todorovski L., Blockeel H., Dzeroski S. (2002) Ranking with Predictive Clustering Trees. Machine Learning: ECML 2002. ECML 2002. Lecture Notes in Computer Science, 2430. doi:10.1007/3-540-36755-1_37

Tseng, L., & Chen, W. (2006). A Two-Phase Genetic Local Search Algorithm for Feedforward Neural Network Training. The 2006 IEEE International Joint Conference on Neural Network Proceedings, 2914-2918. doi:10.1109/ijcnn.2006.1716493

Tukey, J. (1962). The Future of Data Analysis. The Annals of Mathematical Statistics, 33(1), 1-67.

University of California at Irvine. (2020). UCI Machine Learning Repository. Retrieved 22 September 2020, from http://archive.ics.uci.edu/ml/index.php.

Weisstein, E. W. (n.d.). Statistical Correlation. Retrieved from https://mathworld.wolfram.com/StatisticalCorrelation.html

Wu, Q. W., Zhao, A. M., Yao, S., & Li, Z. (2018). Bare spot defect on a hot dip galvanized DP sheet strip doi:10.4028/www.scientific.net/MSF.913.294

Yang, X.S. (2014). Nature-inspired optimization algorithms. Nature-inspired optimization algorithms (pp. 1-263) doi:10.1016/C2013-0-01368-0

Zhou, Z. -. (2012). Ensemble methods: Foundations and algorithms. Ensemble methods: Foundations and algorithms (pp. 1-218) doi:10.1201/b12207