Deliverable 3.2

Procedures for events detection and implementation/suggestion of corrective actions

	Name	Date
Prepared by:	Marco Vannucci Stefano Pede	12.11.2025
Checked by:	Giovanni Bavestrelli	14.11.2025
Approved by:	Renato Girelli	18.11.2025

Project co-funded by the Research Fund for Coal and Steel (RFCS)

G.A. 101112102

Project details

Project Name	**Remote expert virtual system enhancing human management capabilities and favors preservation, transfer and continuous evolution of knowledge for steelmaking operation**		
Project Acronym	iSteel-Expert		
Project No.	101112102	Duration	36 months
Project Start Date	01.07.2023	Project End Date	30.06.2026

Document details

WP:	**3**	**WP Leader:**	**SSSA**
WP Title:	Detailed procedures for KPI calculation, events identification and solutions implementation		
Task:	3.3	Task Leader:	SSSA
Task Title:	Implementation of advanced logics for event detection, implementation / suggestion of corrective actions and further incorporation of presonnel know-how		
Deliverable No.	3.2		
Deliverable Title	Procedures for events detection and implementation/suggestion of corrective actions		
Dissemination level	PU		
Written By	SSSA		
Contributing beneficiary(ies)	TENOVA, SIDER		
Approved by	Renato Girelli		
Status	Draft		
Date	18.11.2025		

Document history

Vers	DATE	AUTHOR / REVIEWER	NOTES
0.1	12.11.2025	Marco Vannucci, Stefano Pede	Preparation of the document
0.2	14.11.2025	Giovanni Bavestrelli	Review and update
1.0	18.11.2025	Renato Girelli	Review and update
1.1			
1.2			
2.0			
2.1			

Table of Contents

1. Introduction

This deliverable presents the work carried out within the iSteel-Expert project, with specific focus on Task T3.3 "*Implementation of advanced logics for event detection, implementation/suggestion of corrective actions and further incorporation of personnel know-how.*" The main objective of this task is to design and deploy advanced artificial intelligence (AI)-based strategies capable of detecting process anomalies during Electric Arc Furnace (EAF) operations in a more systematic and automated way, compared to the current practice where such anomalies are primarily identified and reported by human operators.

The increasing availability of data collected from various sources within the project framework prompted the implementation of these advanced logics.

The sensor network installed on the EAF includes microphones, accelerometers, temperature sensors, energy consumption meters, and other devices capable of providing continuous monitoring of operational conditions. Together with operator-reported alarms and the set of production Key Performance Indicators (KPIs) already defined in the project, these data streams create a unique opportunity to develop an intelligent monitoring layer that can detect, anticipate, and react to abnormal events.

The core ambition of Task T3.3 is not only to enable anomaly detection, but also to suggest corrective measures or alternative settings, leveraging both historical data and knowledge acquired during test campaigns. By exploiting large datasets that cover past production cycles, it becomes possible to adjust internal system parameters — such as decision thresholds in rule-based modules — and to identify correlations between process variables and the onset of anomalies. This integration of AI models, statistical reasoning, and human validation enables the iSteel-Expert system to more closely reproduce expert operator reasoning, while maintaining adaptability to unforeseen operational contexts.

Machine learning and artificial intelligence models play a central role in this task. They are used to analyze multidimensional inputs, capture hidden patterns, and link them with process deviations or performance drops. Supervised learning methods trained on operator-labeled events, as well as unsupervised approaches that can recognize deviations from normal behavior without prior labels, are explored to support robust event detection. Overall, the activities reported in this deliverable aim at equipping the iSteel-Expert platform with sophisticated and adaptive monitoring and decision-support functions. By combining multi-source data, operator knowledge, and AI-based reasoning, Task T3.3 contributes to the reduction of human workload in anomaly detection, enhances process safety and efficiency, and fosters the continuous optimization of EAF operations.

The activities in Task T3.3 were carried out in strong collaboration between industrial and research partners, each contributing according to their expertise and role in the project:

- **TENOVA**, as plant constructor and technology provider, supported the project by ensuring the collection and availability of high-quality process data from the installed monitoring systems.
- **SIDER**, as a steel producer, employed the iSteel-Expert framework in real production environments, collected sensor and process data during operations, and supplied the anomaly alarms reported by plant personnel that served as ground truth for model validation.
- **SSSA** developed the predictive models for event detection and corrective action suggestions, leveraging AI and machine learning techniques to analyze multidimensional datasets and embedding operator knowledge into the decision-making framework.

2. Data sources

For the development of the predictive models within Task T3.3, a dedicated dataset was collected during the normal production operations of the EAF at SIDER Potenza. The dataset comprises some selected heats performed in a time interval that goes from 2025/01/01 to 2025/09/30 and covers a total of 1804 heats, thereby ensuring sufficient variability in process conditions and operating scenarios to support robust model training and validation.

The dataset is composed of heterogeneous sources, reflecting the diverse sensor networks and monitoring systems installed on the EAF. The major data sources integrated for this activity include:

- Chemical data
- EAF Data
- Electrical data

In parallel with sensor and process data, plant personnel registered alarms and anomalous events that occurred during the same observation period. These events, considered of high relevance for anomaly detection, were annotated by experienced operators with precise timestamps. Each annotation was therefore associated with a specific heat and with the corresponding plant configuration, allowing a detailed linkage between recorded anomalies and the operational context in which they emerged. After being cleaned during the pre-processing phase, the dataset is composed of 1804 heats, in 149 of which one or more anomalies occurred.

The abnormal events were grouped into distinct classes, corresponding to different categories of undesired or noteworthy process behavior. The following table summarizes the distribution of such events across the recognized classes:

Table 1 - Anomaly occurrences in the dataset

Event	Occurrences
Further decarburization before tapping	39
Roof closing trouble due to bucket scrap	25
Strong reaction inside furnace	32
Electrodes breakage	67
Explosion inside furnace	4
Liquid/oxidized slag	9
Slag reaction	8

The total amount of abnormal events is 184, greater than the total of the abnormal heats, since for some heats anomalies belonging to more than one category occurred.

This structured dataset — combining high-resolution process data, equipment signals, and domain expert annotations — has provided the foundation for both supervised and unsupervised modeling activities. It ensures that the predictive models can be trained not only to identify patterns linked to known anomalies but also to highlight emerging deviations from normal operating conditions.

3. Predictive models development

Data preparation

Chemical data comprises 42 numeric signals; EAF data comprises 42 signals, of which 28 are numeric ones; Electrical data comprises 20 signals, 19 of which are numeric. Non-numeric signals are discarded at this stage of the study. For each numeric signal of each heat, aggregate statistics over the whole time series of the heat were calculated. In particular, the following statistics were considered:

- Mean
- Standard deviation
- Minimum
- 25^{th} percentile
- Median
- 75^{th} percentile
- Maximum

So, the dataset has a total of 89 * 7 = 623 features for each of the 1804 heats. However, columns having over 70% of NaN values or having exactly the same value for each heat (i.e., non-variant columns) were dropped out, so finally yielding a dataset containing 368 features for 1804 heats (1804 rows and 368 columns). The cleaning operation therefore discarded 39% of the features as not statistically useful for the considered dataset, since they were almost entirely composed of NaNs or invariant values.

The cleaned dataset was then divided into a training dataset and a test dataset with a proportion of 70%-30%. The division has been done so as to retain the same percent distribution of the anomalies in both datasets. Then, several supervised AI models were trained and then tested, and their results compared. The study was conducted in two phases, with two different objectives:

1) Train the models to detect heats with a generic anomaly, without distinguishing between the different classes: the target was a Boolean value indicating if in the heat any anomalies occurred;
2) Train the models to detect heats in which a specific anomaly occurred, therefore distinguishing between the different classes: in this case, for each heat there are as many models as the anomaly classes, for each one the target is a Boolean value indicating if an anomaly belonging to the specific class occurred in the heat.

It must be noted that it was not possible to train a model for the “Explosion inside furnace” anomaly because too few events belong to this class.

Experimental set-up

Four different algorithms were tested:

- Random Forest (RF)
- Histogram Gradient Boost (HGB)
- Support Vector Machine (SVM)
- Adaboost

For each algorithm, multiple training and test runs were performed:

1) The model was trained over the training dataset and validated using the test dataset
2) The model was trained over the training dataset using cross-validation. The cross-validation scores were compared to the test scores obtained at point 1;
3) A hyperparameter optimization was performed using the grid search method: for each run of the grid search, a model was trained over the training dataset using 5-fold cross-validation, in order to find the best tuning of the hyperparameters to maximize the recall score. Then a model was re-trained over the training dataset using the tuned values for the hyperparameters, and tested over the test dataset, obtaining the “best model test scores”.

The choice of the recall as the target score to be maximized is justified by the fact that in anomaly detection is more important to identify as many anomalies as possible than obtaining a high precision, thus tolerating a somewhat high number of false positives as long as all or almost all the true anomalies are detected. On the other hand, accuracy is not so useful in a so unbalanced dataset, as it is evident that a high score in detecting the normal heats (very easy to obtain because they are over 90% of the dataset) is sufficient for the accuracy to be very high even with a poor detection of anomalies.

The following table summarizes the hyperparameters tuned in the extensive grid search.

Table 2 - Grid search parameters

Algorithm	Grid search parameters
SVM	"kernel": ["linear", "poly", "rbf", "sigmoid"] "degree": [2, 3, 4, 5, 10] "gamma": ["scale", "auto"] "coef0": [0.0, 0.5, 1.0] "shrinking": [true, false] "probability": [true, false]
RF	"n_estimators": [50, 100, 200, 300] "criterion": ["gini", "entropy", "log_loss"] "min_samples_split": [2, 5, 10] "min_samples_leaf": [1, 2, 3, 4, 5] "max_features": ["sqrt", "log2"]
HGB	"max_iter": [100, 200, 300, 500] "max_leaf_nodes": [31, 50, 100, null] "min_samples_leaf": [20, 40, 80, 100] "l2_regularization2: [0, 0.2, 0.5]
Adaboost	"n_estimators": [50, 100, 200, 300] "learning_rate": [1.0, 2.0, 3.5, 5.0]

Training data were standardized before training, and the test data were also standardized before validating the model using the statistics of the training dataset.

Although columns having more than 70% of NaNs were discarded during the pre-processing phase, some NaNs survived. This is an issue for SVM and Adaboost, which don't handle NaNs, so heats containing NaNs must be discarded, thus reducing the dataset. On the other hand, RF and HGB natively handle NaNs, so for these algorithms it was possible to use the entire dataset. The NaNs were not imputed because it was seen that using the simple imputer (i.e. substituting NaNs with a descriptive statistics such as the mean or the median) didn't improve the performance of the classifier considerably, and using the iterative imputer (i.e. estimating the missing values by modeling each feature as a function of the other ones) was too computationally heavy.

As shown in the next section, the models were quite successful in detecting a generic anomaly (i.e. heats with an anomaly without concern about which anomaly occurred), but when they were trained to detect specific anomaly categories, their performances were poor, except for the Electrodes breakage anomaly. This result can be explained by the fact that the dataset is very unbalanced: when considering 149 abnormal heats out of 1804 (8.2%), it was possible to obtain somewhat good results despite the strong unbalancing, but when considering a few dozen occurrences out of 1804 (see Table 1 in Section 2), the unbalancing resulted excessive. Therefore, some experiments were performed using the undersampling of the training dataset: several normal heats were discarded to rebalance the dataset, so as to obtain training datasets having 20%, 30%, 40%, and 50% of anomalies. Then the models trained over the undersampled dataset were tested over the test dataset, whose

percentage of anomalies was untouched. Even with the models detecting a generic anomaly, an undersampling experiment was done in order to see if performances improve.

Results

The following tables show the scores of the models trained to detect a generic anomaly. The tables are ordered by the undersampling rate. A 0.1 rate means that a certain number of normal heats randomly chosen have been discarded in order to obtain 10% of abnormal heats. The 0.1 rate is very close to the original anomaly prevalence (0.087).

Adaboost (training undersampling 0.1)

Set	Training Anomaly Prevalence	Prevalence	Accuracy	Recall	Precision	Balanced Accuracy
Training Scores	0.1	0.1	0.975258	0.793814	0.950617	0.894616
Test Scores	0.1	0.087049	0.946921	0.487805	0.833333	0.739251
Cross Val Scores	0.1		0.34021	0.516316	0.734872	0.747843
Best Model Test Scores	0.1	0.087049	0.95966	0.634146	0.866667	0.812422

HGB (training undersampling 0.1)

Set	Training Anomaly Prevalence	Prevalence	Accuracy	Recall	Precision	Balanced Accuracy
Training Scores	0.1	0.1	1	1	1	1
Test Scores	0.1	0.083026	0.950185	0.511111	0.821429	0.750525
Cross Val Scores	0.1		0.952885	0.615238	0.8866	0.802806
Best Model Test Scores	0.1	0.083026	0.95203	0.555556	0.806452	0.771742

RF (training undersampling 0.1):

Set	Training Anomaly Prevalence	Prevalence	Accuracy	Recall	Precision	Balanced Accuracy
Training Scores	0.1	0.1	1	1	1	1
Test Scores	0.1	0.083026	0.953875	0.533333	0.857143	0.762643
Cross Val Scores	0.1		0.942308	0.567143	0.842138	0.767642
Best Model Test Scores	0.1	0.083026	0.95203	0.533333	0.827586	0.761636

SVM (training undersampling 0.1)

Set	Training Anomaly Prevalence	Prevalence	Accuracy	Recall	Precision	Balanced Accuracy
Training Scores	0.1	0.1	0.958763	0.608247	0.967213	0.802978
Test Scores	0.1	0.087049	0.936306	0.317073	0.866667	0.656211
Cross Val Scores	0.1		0.938144	0.432632	0.901414	0.713452
Best Model Test Scores	0.1	0.087049	0.955414	0.707317	0.763158	0.843193

Adaboost (training undersampling 0.2)

Set	Training Anomaly Prevalence	Prevalence	Accuracy	Recall	Precision	Balanced Accuracy
Training Scores	0.2	0.2	0.995876	0.989691	0.989691	0.993557
Test Scores	0.2	0.087049	0.917197	0.560976	0.522727	0.756069
Cross Val Scores	0.2		0.903093	0.671053	0.815298	0.816212
Best Model Test Scores	0.2	0.087049	0.910828	0.731707	0.491803	0.829807

HGB (training undersampling 0.2)

Set	Training Anomaly Prevalence	Prevalence	Accuracy	Recall	Precision	Balanced Accuracy
Training Scores	0.2	0.2	1	1	1	1
Test Scores	0.2	0.083026	0.961255	0.755556	0.772727	0.867717
Cross Val Scores	0.2		0.901923	0.60619	0.862876	0.79109
Best Model Test Scores	0.2	0.083026	0.953875	0.644444	0.763158	0.813168

RF (training undersampling 0.2)

Set	Training Anomaly Prevalence	Prevalence	Accuracy	Recall	Precision	Balanced Accuracy
Training Scores	0.2	0.2	1	1	1	1
Test Scores	0.2	0.083026	0.9631	0.8	0.765957	0.888934
Cross Val Scores	0.2		0.896154	0.645714	0.812564	0.804633
Best Model Test Scores	0.2	0.083026	0.942804	0.622222	0.666667	0.797027

SVM (training undersampling 0.2)

Set	Training Anomaly Prevalence	Prevalence	Accuracy	Recall	Precision	Balanced Accuracy
Training Scores	0.2	0.2	0.917526	0.597938	0.983051	0.79768
Test Scores	0.2	0.087049	0.949045	0.634146	0.742857	0.806608
Cross Val Scores	0.2		0.872165	0.411053	0.877922	0.699099
Best Model Test Scores	0.2	0.087049	0.906582	0.634146	0.472727	0.783352

Adaboost (training undersampling 0.3)

Set	Training Anomaly Prevalence	Prevalence	Accuracy	Recall	Precision	Balanced Accuracy
Training Scores	0.3	0.30031	0.996904	1	0.989796	0.997788
Test Scores	0.3	0.087049	0.940552	0.731707	0.638298	0.846086
Cross Val Scores	0.3		0.879279	0.794737	0.806401	0.855388
Best Model Test Scores	0.3	0.087049	0.893843	0.658537	0.428571	0.787408

HGB (training undersampling 0.3)

Set	Training Anomaly Prevalence	Prevalence	Accuracy	Recall	Precision	Balanced Accuracy
Training Scores	0.3	0.299712	1	1	1	1
Test Scores	0.3	0.083026	0.929889	0.555556	0.581395	0.759669
Cross Val Scores	0.3		0.890518	0.721429	0.899165	0.842092
Best Model Test Scores	0.3	0.083026	0.889299	0.8	0.413793	0.848692

RF (training undersampling 0.3)

Set	Training Anomaly Prevalence	Prevalence	Accuracy	Recall	Precision	Balanced Accuracy
Training Scores	0.3	0.299712	1	1	1	1
Test Scores	0.3	0.083026	0.961255	0.733333	0.785714	0.857612
Cross Val Scores	0.3		0.879048	0.645714	0.912632	0.815842
Best Model Test Scores	0.3	0.083026	0.940959	0.644444	0.644444	0.806126

SVM (training undersampling 0.3)

Set	Training Anomaly Prevalence	Prevalence	Accuracy	Recall	Precision	Balanced Accuracy
Training Scores	0.3	0.30031	0.941176	0.814433	0.9875	0.905004
Test Scores	0.3	0.087049	0.912951	0.512195	0.5	0.731679
Cross Val Scores	0.3		0.848173	0.608421	0.851067	0.779863
Best Model Test Scores	0.3	0.087049	0.895966	0.682927	0.4375	0.799603

Adaboost (training undersampling 0.5)

Set	Training Anomaly Prevalence	Prevalence	Accuracy	Recall	Precision	Balanced Accuracy
Training Scores	0.5	0.5	1	1	1	1
Test Scores	0.5	0.087049	0.823779	0.804878	0.305556	0.81523
Cross Val Scores	0.5		0.840216	0.834737	0.848571	0.84
Best Model Test Scores	0.5	0.087049	0.828025	0.829268	0.314815	0.828588

HGB (training undersampling 0.5)

Set	Training Anomaly Prevalence	Prevalence	Accuracy	Recall	Precision	Balanced Accuracy
Training Scores	0.5	0.5	1	1	1	1
Test Scores	0.5	0.083026	0.776753	0.888889	0.25641	0.827744
Cross Val Scores	0.5		0.807317	0.798095	0.829091	0.807619
Best Model Test Scores	0.5	0.083026	0.857934	0.955556	0.364407	0.902325

RF (training undersampling 0.5)

Set	Training Anomaly Prevalence	Prevalence	Accuracy	Recall	Precision	Balanced Accuracy
Training Scores	0.5	0.5	1	1	1	1
Test Scores	0.5	0.083026	0.863469	0.8	0.356436	0.834608
Cross Val Scores	0.5		0.836469	0.778571	0.875326	0.79881
Best Model Test Scores	0.5	0.083026	0.880074	0.755556	0.386364	0.823452

SVM (training undersampling 0.5)

Set	Training Anomaly Prevalence	Prevalence	Accuracy	Recall	Precision	Balanced Accuracy
Training Scores	0.5	0.5	0.896907	0.85567	0.932584	0.896907
Test Scores	0.5	0.087049	0.791932	0.707317	0.252174	0.753659
Cross Val Scores	0.5		0.844939	0.793158	0.89	0.844211
Best Model Test Scores	0.5	0.087049	0.828025	0.756098	0.303922	0.795491

As can be seen, generally the higher the training anomaly prevalence the higher the recall (i.e. the detection of actual positives), but the lower the precision (i.e. false positives increase). However, it is not necessary to do a dramatic undersampling, as the scores are pretty good even with an anomaly prevalence close to the natural one (downsampling 0.1). Good algorithms seem to be the Adaboost and the SVM, that perform well without downsampling.

When trying to detect the specific anomalies, the performances are not so good.

Adaboost (original anomaly prevalence)

Anomaly	test_prev	test_recall	test_precision	best_model_recall	best_model_precision
High Carbon melt	0.019	0.222	0.4	0.333	0.75
Roof closing trouble	0.013	0.0	0.0	0.167	1.0
Slag reaction	0.004	0.0	0.0	0.0	0.0
Strong reaction inside furnace	0.017	0.0	0.0	0.125	0.25
Electrodes breakage	0.036	0.588	0.909	0.824	0.583
Liquid slag	0.006	0.0	0.0	0.0	0.0

HGB (original anomaly prevalence)

Anomaly	test_prev	test_recall	test_precision	best_model_recall	best_model_precision
High Carbon melt	0.018	0.5	1.0	0.1	1.0
Roof closing trouble	0.013	0.0	0.0	0.0	0.0
Slag reaction	0.004	0.0	0.0	0.0	0.0
Strong reaction inside furnace	0.015	0.0	0.0	0.0	0.0
Electrodes breakage	0.035	0.684	0.867	0.474	0.9
Liquid slag	0.006	0.0	0.0	0.0	0.0

Rf (original anomaly prevalence)

Anomaly	test_prev	test_recall	test_precision	best_model_recall	best_model_precision
High Carbon melt	0.018	0.1	0.5	0.1	0.5
Roof closing trouble	0.013	0.0	0.0	0.0	0.0
Slag reaction	0.004	0.0	0.0	0.0	0.0
Strong reaction inside furnace	0.015	0.125	1.0	0.0	0.0
Electrodes breakage	0.035	0.526	0.833	0.474	0.75
Liquid slag	0.006	0.0	0.0	0.0	0.0

SVM (original anomaly prevalence)

Anomaly	test_prev	test_recall	test_precision	best_model_recall	best_model_precision
High Carbon melt	0.019	0.0	0.0	0.1	0.5
Roof closing trouble	0.013	0.0	0.0	0.0	0.0
Slag reaction	0.004	0.0	0.0	0.0	0.0
Strong reaction inside furnace	0.017	0.0	0.0	0.0	0.0
Electrodes breakage	0.036	0.529	0.9	0.474	0.75
Liquid slag	0.006	0.0	0.0	0.0	0.0

Only for the Electrodes breakage it was possible to obtain a somewhat good result (recall for Adaboost is 0.824). However, it is the anomaly with most occurrences, and after a discussion with the plant experts, it emerged that this anomaly is quite distinguishable from a typical trend of the electric signals so that it's even recognizable without the aid of AI models.

Being evident that the original anomaly prevalence is too low for obtaining good detection, the dataset was undersampled. The results are summarized in the following tables.

Adaboost (undersampling 0.3)

Anomaly	test_prev	test_recall	test_precision	best_model_recall	best_model_precision
High Carbon melt	0.019	0.556	0.139	0.667	0.222
Roof closing trouble	0.013	1.0	0.076	0.833	0.081
Slag reaction	0.004	0.5	0.025	1.0	0.036
Strong reaction inside furnace	0.017	0.625	0.079	0.25	0.018
Electrodes breakage	0.036	0.765	0.277	0.882	0.429
Liquid slag	0.006	0.667	0.022	0.667	0.016

HGB (undersampling 0.3)

Anomaly	test_prev	test_recall	test_precision	best_model_recall	best_model_precision
High Carbon melt	0.018	0.8	0.145	0.7	0.104
Roof closing trouble	0.013	0.429	0.059	0.857	0.067
Slag reaction	0.004	0.0	0.0	0.0	0.0
Strong reaction inside furnace	0.015	0.75	0.107	0.875	0.076
Electrodes breakage	0.035	0.789	0.268	0.579	0.333
Liquid slag	0.006	0.0	0.0	0.0	0.0

RF (undersampling 0.3)

Anomaly	test_prev	test_recall	test_precision	best_model_recall	best_model_precision
High Carbon melt	0.018	0.6	0.214	0.9	0.158
Roof closing trouble	0.013	0.429	0.073	1.0	0.111
Slag reaction	0.004	0.5	0.023	0.0	0.0
Strong reaction inside furnace	0.015	0.375	0.057	0.5	0.148
Electrodes breakage	0.035	0.737	0.275	0.842	0.296
Liquid slag	0.006	0.0	0.0	0.0	0.0

SVM (undersampling 0.3)

Anomaly	test_prev	test_recall	test_precision	best_model_recall	best_model_precision
High Carbon melt	0.019	0.667	0.1	0.9	0.158
Roof closing trouble	0.013	0.667	0.154	1.0	0.111
Slag reaction	0.004	0.0	0.0	0.0	0.0
Strong reaction inside furnace	0.017	0.125	0.111	0.5	0.148
Electrodes breakage	0.036	0.765	0.565	0.842	0.296
Liquid slag	0.006	0.0	0.0	0.0	0.0

Adaboost (undersampling 0.5)

Anomaly	test_prev	test_recall	test_precision	best_model_recall	best_model_precision
High Carbon melt	0.019	1.0	0.125	0.778	0.064
Roof closing trouble	0.013	1.0	0.076	0.667	0.038
Slag reaction	0.004	1.0	0.01	1.0	0.01
Strong reaction inside furnace	0.017	0.875	0.06	0.75	0.058
Electrodes breakage	0.036	0.824	0.226	0.882	0.214
Liquid slag	0.006	1.0	0.011	0.0	0.0

HGB (undersampling 0.5)

Anomaly	test_prev	test_recall	test_precision	best_model_recall	best_model_precision
High Carbon melt	0.018	0.8	0.071	0.7	0.046
Roof closing trouble	0.013	0.0	0.0	0.0	0.0
Slag reaction	0.004	0.0	0.0	0.0	0.0
Strong reaction inside furnace	0.015	0.875	0.048	0.875	0.042
Electrodes breakage	0.035	0.895	0.144	0.842	0.18
Liquid slag	0.006	0.0	0.0	0.0	0.0

RF (undersampling 0.5)

Anomaly	test_prev	test_recall	test_precision	best_model_recall	best_model_precision
High Carbon melt	0.018	0.9	0.089	0.8	0.113
Roof closing trouble	0.013	0.714	0.045	0.571	0.062
Slag reaction	0.004	1.0	0.014	1.0	0.008
Strong reaction inside furnace	0.015	0.625	0.028	0.75	0.047
Electrodes breakage	0.035	0.842	0.145	0.842	0.162
Liquid slag	0.006	1.0	0.012	0.667	0.014

SVM (undersampling 0.5)

Anomaly	test_prev	test_recall	test_precision	best_model_recall	best_model_precision
High Carbon melt	0.019	0.889	0.093	0.8	0.113
Roof closing trouble	0.013	0.833	0.041	0.571	0.062
Slag reaction	0.004	0.0	0.0	1.0	0.008
Strong reaction inside furnace	0.017	0.75	0.061	0.75	0.047
Electrodes breakage	0.036	0.941	0.235	0.842	0.162
Liquid slag	0.006	0.333	0.007	0.667	0.014

With undersampling, specific anomaly detection improves considerably. The algorithms are able to detect even infrequent anomalies (except for Slag reaction and Liquid slag, with very few occurrences). The downside is a dramatic reduction of precision: the models learn to catch as many detections as possible, but a lot of normal heats are wrongly classified as abnormal, thus increasing the false positives rate, as shown in the following figures.

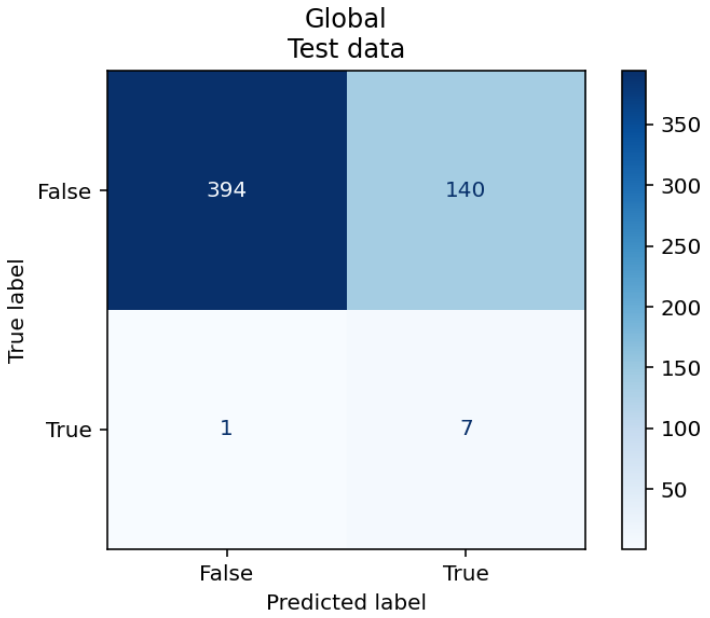

Figure 1 - Confusion matrix for HGB for detecting Strong reaction inside furnace (undersampling 0.5)

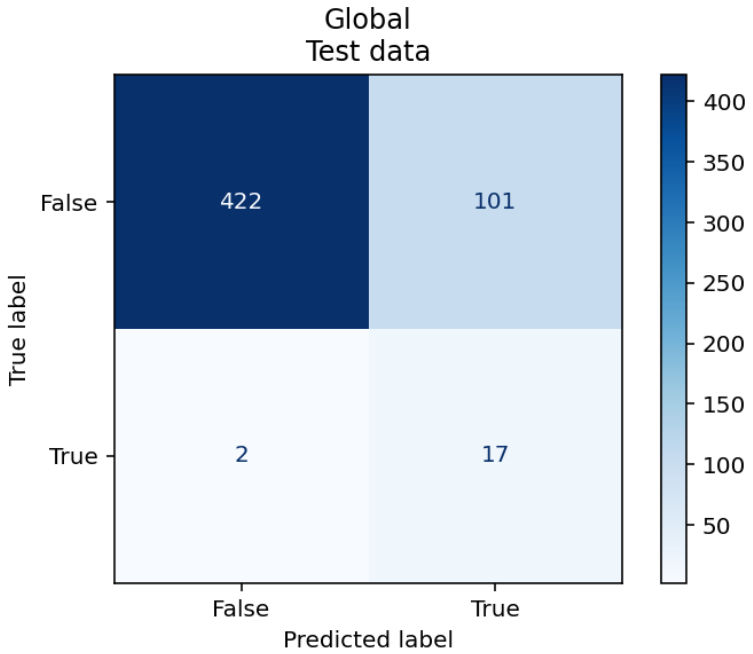

Figure 2 - Confusion matrix for HGB for detecting Electrodes breakage (undersampling 0.5)

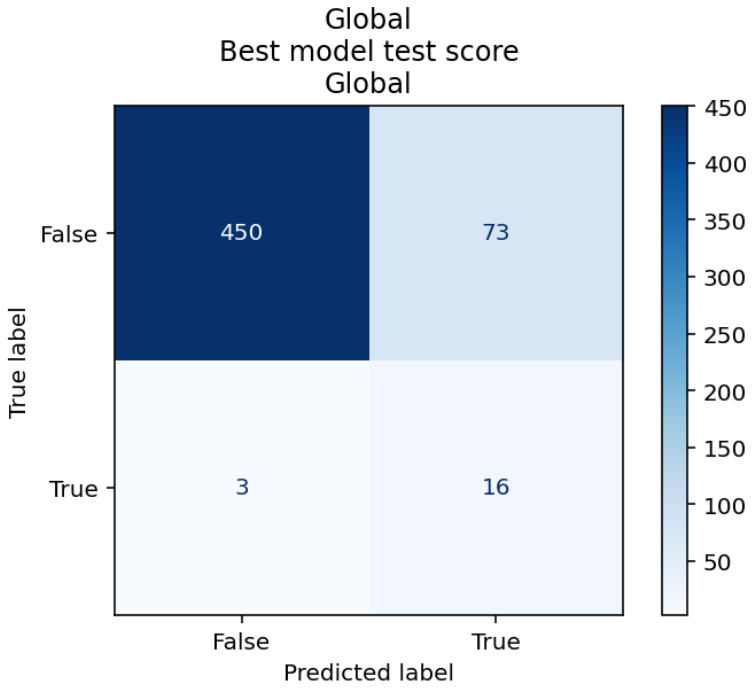

Figure 3 - Confusion matrix for HGB for detecting Electrodes breakage (undersampling 0.5)

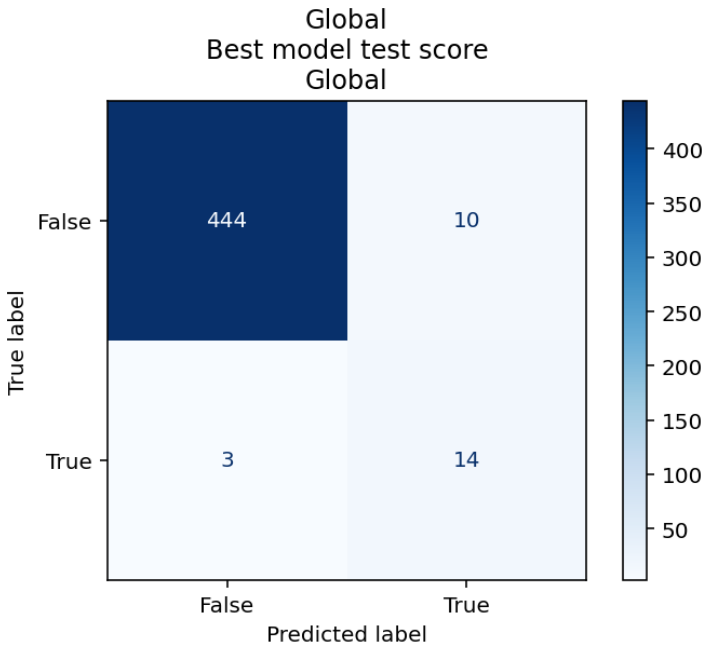

Figure 4 - Confusion matrix for Adaboost for detecting Electrodes breakage (original anomaly prevalence)

To conclude this section, we can summarize as follows:

- good performances are obtained in generic anomaly detection, without concern about which category anomalies belong to;
- poor performances are obtained in specific anomaly detection, because the prevalence of each anomaly is very low. Only in detecting Electrodes breakage the models succeed;
- however, with undersampling of the training dataset, models detect even specific anomalies very well, at cost of a high false positive rate.

It is now to decide how acceptable is to have such a high false positive rate, as long as the models succeed in detecting as much true anomalies as possible. For very serious anomalies, a high false positive rate may be forgiven if most real anomalies are detected. For less serious anomalies, a good compromise must be found.

Further analyses will include XGB algorithm and unequal misclassification cost instead of the undersampling.

4. Support System Development

This section describes a proof-of-concept methodology aimed at recommending process countermeasures in EAF steelmaking, triggered by machine learning (ML) predictions of problematic heats. To date, this approach has not yet been deployed on live operational data—the framework for the method is ready, but practical implementation awaits the finalization and validation of the underlying predictive models.

The methodology is activated when the ML model, using process data inputs, predicts that the current heat may be problematic. At this point, the system attempts to suggest alternative process parameter settings—such as temperature setpoints, energy input, tap-to-tap time, or charge mix—that would avoid triggering the predicted alarm, while remaining as close as possible to the current operating state. These suggestions must respect operational and physical constraints set by safety, equipment limits, and process logic.

A genetic algorithm (GA) serves as the optimization engine for this task. The choice of GA is justified by the complex, multi-dimensional, and highly constrained nature of steelmaking processes. GAs are particularly well-suited for searching large solution spaces, handling non-linear relationships, and escaping local optima, problems commonly encountered in industrial optimization. GAs maintain a population of candidate solutions, which are evolved across generations through selection, crossover, and mutation, enabling robust search for practical process adjustments.

The core of the approach lies in the definition of the objective (fitness) function. For each candidate process condition, the function considers two aspects:

1. **Alarm Avoidance:** The ML model is used to predict whether the new condition causes any alarm. Candidates that avoid all alarms are favored.
2. **Proximity to Current State:** To prioritize feasible interventions, the objective function incorporates a penalty term that increases as parameter values deviate from the current operating state. This ensures suggested solutions are achievable and minimally disruptive.

Mathematically, the fitness for a candidate solution x can be expressed as:

$$J(x) = \alpha \cdot \Delta(x_{\text{current}}, x) + \beta \cdot \text{AlarmPenalty}(x)$$

where $\Delta(x_{\text{current}}, x)$ quantifies the overall distance from the current state, $\text{AlarmPenalty}(x)$ is zero for alarm-free candidates and large for alarm-triggering candidates, and α, β are weighting coefficients regulating the trade-off between proximity and safety. Only candidate solutions that remain within operational constraints are considered valid.

A scheme of the operation of the suggestion system is provided in figure X where the main components and their interactions are put into evidence.

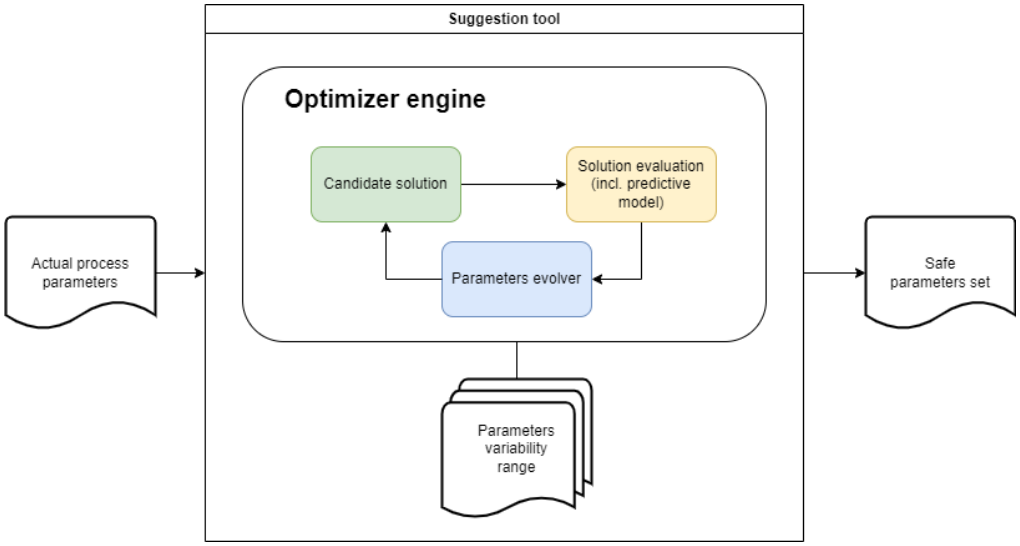

Figure 5 Flow-diagram of the suggestion tool based on GA for alarm avoidance.

Although similar uses of GAs have been reported for multi-objective steelmaking tasks—such as optimizing steel composition and scheduling—no published literature describes the direct integration of ML-driven alarm prediction with GA-based search for actionable, real-time operational advice in EAF environments. This framework lays the groundwork for such a system. Once predictive models are finalized and validated, we plan to implement and test this methodology in real process scenarios.

4. Conclusions and future work

This deliverable has presented the work carried out within Task T3.3 of the ISteel-Expert project, focusing on the development of AI-based predictive models for anomaly detection in Electric Arc Furnace operations. The study leveraged a comprehensive dataset of 1804 heats collected between January and September 2025 at SIDER Potenza, encompassing heterogeneous data sources including chemical, EAF operational, and electrical measurements. The experimental campaign evaluated four distinct machine learning algorithms—Random Forest, Histogram Gradient Boost, Support Vector Machine, and Adaboost—across multiple training configurations and undersampling strategies. The results demonstrate a clear differentiation in model performance depending on the detection task complexity and data balancing approach.

For **generic anomaly detection**, where the objective was to identify heats experiencing any type of anomaly regardless of category, the models achieved satisfactory performance even with minimal data manipulation. Adaboost and SVM algorithms demonstrated particularly robust behavior, with balanced accuracy scores exceeding 0.80 and recall values ranging from 0.63 to 0.71 when using optimized hyperparameters. These results were obtained with undersampling rates close to the natural anomaly prevalence (approximately 8-10%), indicating that the models can effectively learn

to distinguish anomalous heats from normal operations without dramatic data rebalancing. The high precision values (above 0.75 in most cases) further confirm that false positive rates remain acceptably low, making these models suitable for deployment in production environments where operators can be alerted to potential issues with reasonable confidence.

The scenario changes considerably when attempting **specific anomaly detection**, where models must distinguish between different categories of abnormal events. With the original data distribution, only the "Electrodes breakage" anomaly—the most frequent class with 67 occurrences—yielded acceptable detection rates, achieving recall scores up to 0.82 with Adaboost. For other anomaly types with lower prevalence, such as "Roof closing trouble" (25 occurrences), "High-carbon melt" (39 occurrences), and "Strong reaction inside furnace" (32 occurrences), the models struggled to achieve meaningful detection capabilities, with recall values often at or near zero. This limitation is directly attributable to the severe class imbalance, where individual anomaly categories represent less than 2-3% of the total dataset. The application of aggressive undersampling strategies—increasing training set anomaly prevalence to 30% or 50%—dramatically improved recall for specific anomaly detection, with several models achieving recall scores above 0.80 and in some cases reaching perfect detection (1.0) for certain anomaly types. However, this improvement came at a significant cost in terms of precision, which dropped substantially, often below 0.20. This trade-off manifests as a high false positive rate, where numerous normal heats are incorrectly flagged as anomalous. The confusion matrices presented in Figures 1-4 clearly illustrate this phenomenon: while true positive detection increases markedly, false positives also surge, potentially overwhelming operators with excessive alarm notifications.

An important operational insight emerged from discussions with plant experts regarding the "Electrodes breakage" anomaly.
This event type exhibits distinctive electrical signal patterns, but operators immediately recognize a suddenly loss of normal operational conditions that require a prompt maintenance action.

The fact that ML models achieved good detection performance for this anomaly suggests they have successfully learned to identify these recognizable patterns, validating the modeling approach. However, it also indicates that the true value of AI-based detection systems will be most evident for subtler anomaly types that are difficult for human operators to consistently identify, particularly in the early stages before full manifestation.

The proof-of-concept **support system for corrective action suggestion**, based on genetic algorithms and integrated with ML predictions, represents a promising direction for transforming anomaly detection into actionable operational guidance. By formulating the problem as a constrained optimization task—balancing alarm avoidance with minimal deviation from current operating conditions—the proposed framework has the potential to provide operators with specific, implementable recommendations rather than mere alerts. While this system awaits deployment pending model finalization, its theoretical foundation addresses a critical gap in current practice: bridging the gap between detecting problems and solving them.

Looking forward, several ideas for improvement have been identified. These will be implemented in the next releases of the models and of the decision support system which are envisaged in the subsequent phases of the project. The **future work** will focus on:

- **Expanding the algorithmic portfolio** by incorporating additional state-of-the-art methods such as LightGBM, which has demonstrated superior performance in handling imbalanced datasets and large-scale industrial applications due to its efficient implementation and built-in support for categorical features.
- **Implementing specialized techniques for imbalanced classification**, including:
    - Class-specific algorithms designed explicitly for minority class detection, such as SMOTE (Synthetic Minority Over-sampling Technique) variants, ADASYN (Adaptive Synthetic Sampling), and cost-sensitive learning approaches.
    - Advanced resampling strategies that combine intelligent undersampling of the majority class with informed oversampling of minority classes, potentially using cluster-based or Tomek links methods to preserve decision boundary information while achieving better class balance.
    - Ensemble methods that combine multiple resampling strategies with diverse base classifiers to maximize both detection capability and precision across all anomaly types.

These enhancements aim to achieve the optimal balance between high recall for critical anomaly detection and acceptable precision to maintain operator confidence in the system, ultimately delivering a practical and reliable AI-powered monitoring solution for Electric Arc Furnace operations.