



Baština Akademije nauka i umjetnosti Bosne i Hercegovine

SMART MANUFACTURING ON THE HORIZON: A Conceptual Shift from Industry 4.0 to Industry 5.0: Industrija

Karabegović, Isak

2026

Academy of Sciences and Arts of Bosnia and Herzegovina

<https://bastina.anubih.ba/handle/123456789/875>

Preuzeto s Baštine Akademije nauka i umjetnosti Bosne i Hercegovine

<https://bastina.anubih.ba/>

Explainable and Trustworthy AI for Steel Plates Fault Detection Using Shap and Lime

Nerma Mezetović^{*1}, Lejla Banjanović-Mehmedović¹, Isak Karabegović²

Abstract: *This paper investigates the potential of Explainable Artificial Intelligence (XAI) methods to improve the interpretation of AI-based models in smart manufacturing applications. The study focuses on SHAP and LIME techniques, analyzing their ability to explain model decisions, identify the most influential input features, and provide insights into both individual predictions and overall model behaviour.*

A practical case study based on the Steel Plates Faults dataset is presented using a Random Forest classification model to investigate the application of explainable AI techniques for steel plate fault detection. The results demonstrate that SHAP and LIME methods provide valuable insights into model decision-making by identifying the key features influencing fault classification. These explainability approaches enhance model transparency and trustworthiness, supporting the development of more reliable AI-based solutions for industrial quality inspection and smart manufacturing applications.

Keywords: *explainable artificial intelligence (XAI), model interpretability, machine learning, SHAP, LIME, transparency*

1. Introduction

Artificial intelligence (AI) has become a key technology in the transformation of modern industrial systems, enabling machines to perform complex tasks such as learning, reasoning, and decision-making. Advances in machine learning algorithms, computational capabilities, and data availability have accelerated the adoption of AI in various domains, including smart manufacturing, robotics, and autonomous systems [1]. In industrial environments, AI-based solutions are increasingly applied for quality inspection, predictive maintenance, process monitoring, and production optimization, contributing to improved efficiency and more adaptive decision-making.

Despite their high predictive performance, many advanced machine learning models remain difficult to interpret due to their complex internal structures. This limitation is particularly evident in deep learning and other nonlinear models, which are commonly considered black-box systems. Although the input data and final predictions are available, the reasoning process leading to a specific decision

^{*1}University of Tuzla, Faculty of Electrical Engineering, Tuzla, Bosnia and Herzegovina

²Academy of Science and Arts of Bosnia and Herzegovina, Sarajevo, Bosnia and Herzegovina

E-mail: nerma.mezetovic@fet.ba

is often unclear to human users [2]. This lack of transparency represents a significant barrier to the wider adoption of AI in industrial applications, where reliability, safety, and accountability are critical requirements.

In manufacturing environments, AI-driven decisions can directly influence product quality, equipment operation, and production efficiency. Therefore, achieving high prediction accuracy alone is not sufficient; it is also necessary to understand which factors contribute to a model decision and whether the model relies on meaningful patterns. Without such insight, detecting errors, validating model behaviour, and ensuring reliable operation become considerably more challenging [1].

To overcome these limitations, Explainable Artificial Intelligence (XAI) has emerged as an important research field focused on developing methods that improve the understanding of AI model decisions. XAI approaches aim to transform complex model outputs into meaningful explanations, enabling users to analyse model behaviour, verify predictions, and identify the factors influencing decision-making [2]. Consequently, explainability has become a fundamental component of trustworthy AI, together with transparency, fairness, robustness, accountability, and user confidence [3].

The importance of XAI is further emphasized by ethical and regulatory requirements for AI deployment. Modern AI systems are expected not only to achieve high performance but also to provide transparent and accountable decision-making processes. Frameworks such as the General Data Protection Regulation (GDPR) and the European Union Artificial Intelligence Act (EU AI Act) highlight the importance of transparency, documentation, human oversight, and risk management in AI applications [4], [5].

Among different XAI approaches, model-agnostic methods have attracted significant attention due to their ability to explain various machine learning models independently of their internal architecture. Two widely adopted methods are Local Interpretable Model-Agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP). These techniques provide insights into feature contributions and enable both local explanation of individual predictions and global analysis of overall model behaviour. Their flexibility makes them particularly suitable for analysing complex machine learning models used in industrial applications.

The objective of this paper is to investigate the contribution of XAI methods to the development of transparent and trustworthy AI systems for industrial fault detection. The study focuses on SHAP and LIME approaches and analyses their capabilities, advantages, and limitations. Furthermore, a practical case study based on the Steel Plates Faults dataset is presented using a Random Forest classification model to demonstrate how SHAP and LIME can identify the most influential features and provide deeper insights into the decision-making process of machine learning models.

2. Fundamentals of Explainable Artificial Intelligence

Two fundamental concepts are central to understanding transparent artificial intelligence systems: *interpretability* and *explainability*. Although these terms are often used interchangeably, they describe different aspects of understanding AI model behaviour. Interpretability refers to the inherent ability of a model to provide insight into its internal decision-making process, while explainability refers to the use of additional techniques to generate understandable explanations of model predictions.

Interpretability is an intrinsic property of models whose internal structure can be directly understood by humans. In interpretable models, the relationship between input variables and outputs is explicitly represented, allowing users to analyse how individual features contribute to a prediction [6]. Decision trees, linear regression models, and rule-based systems are typical examples of white-box models, where the reasoning process is directly accessible without additional explanation mechanisms [7].

The importance of interpretability is particularly evident in industrial applications, where AI decisions may influence product quality, equipment reliability, and operational safety. In such environments, understanding why a model generates a specific prediction is essential for model validation, error analysis, and informed decision-making. For example, identifying the reasons behind a detected manufacturing defect or a predicted equipment failure can support preventive actions and improve system reliability.

In contrast, explainability focuses on providing understandable interpretations of complex AI models whose internal mechanisms cannot be directly examined. Instead of modifying the original model, explainability methods generate additional representations, such as feature importance scores, surrogate models, or visual explanations, to reveal the factors influencing a prediction [8]. Therefore, explainability serves as a bridge between complex machine learning models and the human need for transparency, trust, and accountability, Figure 1.

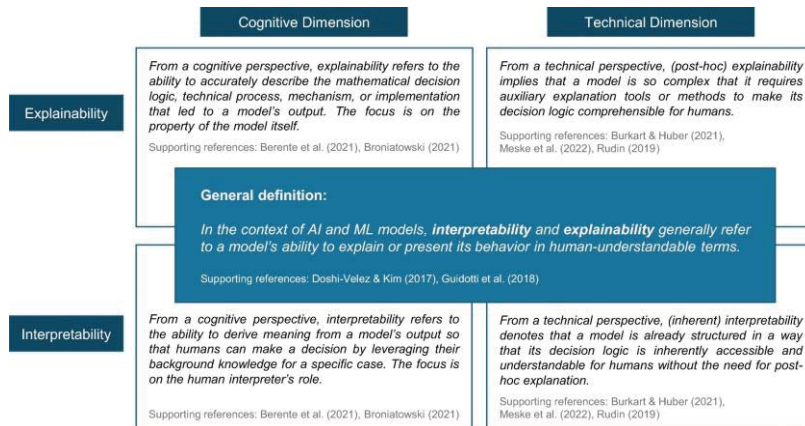


Figure 1. Conceptual comparison between interpretability and explainability in AI models [9]

2.1. Taxonomy of XAI Methods

Explainable artificial intelligence represents a broad collection of methods that differ according to how explanations are generated and presented. The most common classifications are based on whether explanations are integrated into the model itself or generated after training, whether they describe the entire model behaviour or individual predictions, and whether they depend on a specific model architecture.

2.1.1. Intrinsic Interpretability and Post-hoc Explainability

XAI methods can first be classified into *intrinsic interpretability* and *post-hoc explainability approaches*. As previously defined, intrinsically interpretable models are understandable by construction, with their decision-making process directly represented through mathematical relationships or logical rules. However, a major challenge of these transparent approaches in industrial environments is the trade-off between interpretability and performance [9]. Although they provide high transparency, their predictive performance may be severely limited when dealing with highly nonlinear and complex industrial problems, which often require more sophisticated, non-linear models. To better understand these differences, Figure 2 summarizes and compares the characteristics of intrinsic and post-hoc interpretability across different stages of the machine learning process, from data input to model assessment, with the post-hoc approach discussed in more detail in the following section.

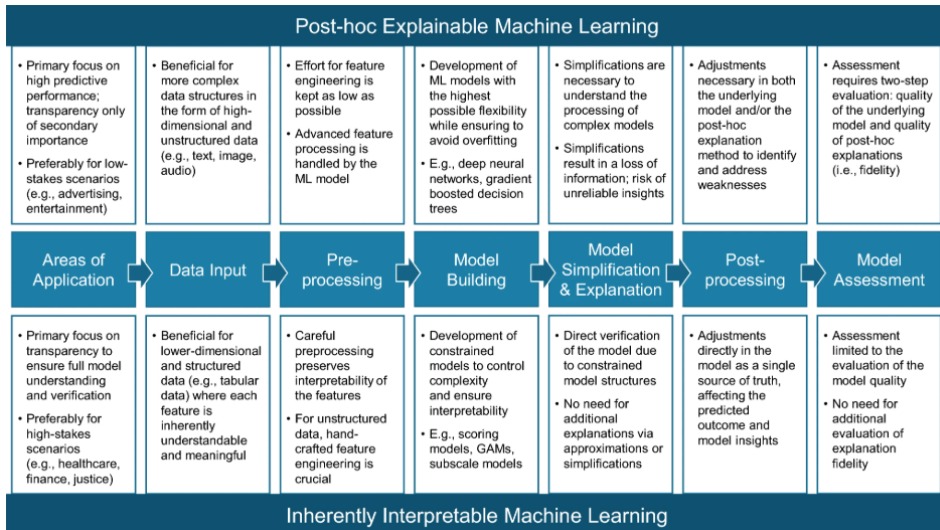


Figure 2. Comparison between intrinsically interpretable models and post-hoc explanation approaches [9]

Post-hoc explainability methods are applied after a model has been trained and are particularly important for complex black-box models such as ensemble methods and deep neural networks. These techniques analyse the relationship between model inputs and outputs without modifying the original model, providing information about feature influence and decision-making patterns [9]. Post-hoc methods enable the use of high-performance models while still providing a certain level of transparency required for trustworthy AI deployment. In this study, we adopt this post-hoc perspective, applying SHAP and LIME to interpret the predictions of a Random Forest classifier trained on the steel plates faults dataset.

2.1.2. Global and Local Explanations

Another important distinction in XAI concerns the scope of explanations. Global explanations describe the overall behaviour of a model by analysing general decision patterns and the importance of input features across the entire dataset. They provide insight into how the model operates as a whole.

In contrast, local explanations focus on individual predictions and identify the specific features that influenced a particular decision. This approach is especially valuable for analysing individual cases, such as explaining why a specific industrial product was classified as defective or why a particular anomaly was detected.

Global and local explanations provide complementary perspectives, Figure 3. While global explanations improve the understanding of general model behaviour, local explanations provide detailed reasoning behind individual predictions, which is often crucial for practical decision support [7].

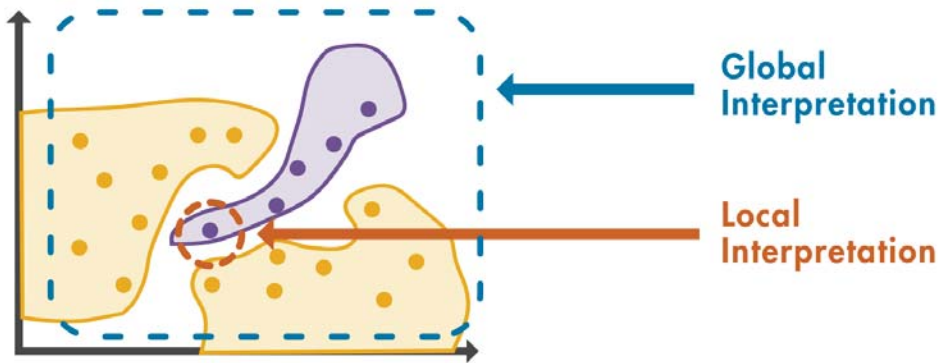


Figure 3. Difference between global and local explanations in XAI [10]

2.1.3. Model-Specific and Model-Agnostic Approaches

Because machine learning algorithms differ in their architectures, complexity, and internal mechanisms, XAI methods are commonly divided into model-specific and model-agnostic approaches.

Model-specific methods are designed for a particular class of models and use information from their internal structure, such as parameters, weights, or activation patterns. These approaches can provide detailed explanations but are limited to specific model architectures [10]. Examples include gradient-based methods such as Integrated Gradients, activation-based methods such as Grad-CAM, and relevance-propagation approaches such as Layer-wise Relevance Propagation (LRP) [11], [12].

In contrast, model-agnostic methods explain model behaviour independently of the underlying algorithm. They treat the model as a black box and analyse the relationship between input features and predictions. This flexibility enables their application to different machine learning models without requiring modifications to the original architecture [13].

Model-agnostic methods are particularly valuable in industrial applications because they allow engineers to select the most accurate model for a given task while maintaining the possibility of generating meaningful explanations. Among the most widely adopted approaches are Local Interpretable Model-Agnostic

Explanations (LIME) and SHapley Additive exPlanations (SHAP), which are further analysed in the following section.

Table 1 summarizes the main differences between model-specific and model-agnostic XAI approaches.

Table 1. Comparative overview of model-agnostic and model-specific approaches

Aspect	Model-agnostic approach	Model-specific approach
Applicability	Independent of internal architecture; usable across model families	Tied to a particular architecture; limited to a specific class of models
Operating mechanism	Perturbing inputs and observing changes in the output	Accessing internal weights, gradients, and layers directly
Computational cost	Often high (thousands of repeated predictions)	Usually lower (computed directly from existing parameters)
Fidelity	May be less precise, since it relies on approximating model behaviour	Often more precise, since it reflects the model's actual internal logic
Advantages	Universality, flexibility, applicability across different models	Higher precision and deeper insight into the model's internal structure
Limitations	May yield approximate explanations that overlook model-specific detail	Restricted to particular models and often demands deeper technical expertise
Representative examples	LIME, SHAP, Partial Dependence Plots	Integrated Gradients, CAM, Grad-CAM, LRP

3. Model-Agnostic Explainable AI Methods: Lime And Shap

Model-agnostic XAI methods have become widely adopted because they enable the interpretation of different machine learning models without requiring access to their internal structures. Instead of modifying the original model, these approaches analyse the relationship between input features and model outputs to generate understandable explanations [14]. Among the most widely used model-agnostic techniques are *Local Interpretable Model-Agnostic Explanations (LIME)* [13] and *SHapley Additive exPlanations (SHAP)* [15].

3.1. LIME: Local Interpretable Model-Agnostic Explanations

LIME is a model-agnostic explanation technique designed to interpret individual predictions of complex machine learning models. The main idea behind LIME is that although a model may be highly nonlinear and difficult to understand globally, its behaviour around a specific instance can often be approximated using a simpler interpretable model [13].

The method generates perturbed samples around the instance of interest and obtains predictions from the original model. These samples are weighted according to their similarity to the original instance and are then used to train a simple surrogate model, typically linear regression or a decision tree. The parameters of this local model indicate the contribution of individual features to the prediction, providing insight into why a specific decision was made [13].

The main steps of LIME can be summarized as follows:

- selecting an instance to be explained;
- generating perturbed samples around the selected instance;
- obtaining predictions from the original model;
- assigning weights according to the distance from the original sample;
- training an interpretable surrogate model;
- analysing feature contributions to generate the explanation.

LIME improves the transparency of black-box models by providing human-understandable local explanations. In addition, the quality of the explanation strongly depends on the selection of the local neighbourhood [13].

3.1.1. Strengths and Limitations of LIME

The main advantage of LIME is its flexibility and applicability to different machine learning algorithms, including Random Forest, support vector machines, and neural networks. Since it does not require knowledge of the model architecture, it can be applied to a wide range of classification and regression problems [7].

However, LIME has several limitations. The stochastic nature of perturbation sampling may reduce explanation stability, while unrealistic generated samples can negatively influence the quality of the surrogate model. Furthermore, LIME focuses only on individual predictions and does not provide a direct global understanding of the overall model behaviour [7].

3.2. SHAP: SHapley Additive exPlanations

SHAP, introduced by Lundberg and Lee, is one of the most influential XAI methods for interpreting machine learning predictions. It is based on Shapley values from cooperative game theory, where each input feature is considered as a contributor to the final model prediction [15].

The main principle of SHAP is to distribute the difference between the model prediction and the average prediction among the input features according to their individual contributions. Positive SHAP values indicate features that increase the prediction value, while negative values represent features that decrease it. The magnitude of the SHAP value reflects the strength of the feature influence [15].

SHAP supports both local and global explanations [15]. Local explanations describe why a specific prediction was generated, while global explanations aggregate SHAP values across multiple samples to identify the most influential features in the entire dataset.

Several SHAP variants have been developed depending on the underlying model structure. KernelSHAP provides a model-independent approximation suitable for different machine learning models, while TreeSHAP is optimized for tree-based algorithms such as Random Forest and gradient boosting models, providing faster and more efficient explanations [15].

3.2.1. Visualization of SHAP Explanations

SHAP visualizations provide an intuitive representation of feature contributions and are essential for interpreting model behaviour. Common visualization techniques include:

- *Summary plots*, which show global feature importance and the direction of feature influence across the dataset.
- *Waterfall plots*, which explain individual predictions by showing how each feature contributes to increasing or decreasing the final output.
- *Force plots*, which illustrate the contribution of individual features relative to the model baseline value.

These visualizations enable engineers and domain experts to analyse model decisions without requiring detailed knowledge of the underlying machine learning algorithm [15].

3.2.2. Strengths and Limitations of SHAP

The main advantage of SHAP is its strong theoretical foundation and ability to provide consistent feature attribution. Unlike LIME, SHAP explanations can be aggregated to obtain a reliable global interpretation of model behaviour. This

property makes SHAP particularly suitable for industrial applications where identifying the most influential process variables is important [7].

However, the original SHAP formulation can be computationally demanding, especially for models with a large number of features. In addition, correlated input variables may affect the interpretation of feature importance. Therefore, selecting an appropriate SHAP variant and considering the characteristics of the dataset are important for practical deployment [7].

3.3. Comparison of LIME and SHAP Methods

LIME and SHAP provide complementary approaches to explaining machine learning models. LIME focuses primarily on local interpretation by approximating the behaviour of a model around a specific instance using an interpretable surrogate model. In contrast, SHAP provides feature contributions based on Shapley values, enabling both local and global analysis of model behaviour.

The main difference between the two methods lies in their explanation principles. LIME offers simplicity and flexibility but may suffer from instability due to random sampling. SHAP provides more theoretically consistent explanations but can require higher computational resources.

For industrial AI applications, the choice between LIME and SHAP depends on the specific requirements of the problem. While LIME is useful for understanding individual predictions, SHAP is particularly suitable for analysing feature importance and overall model behaviour. In this work, both methods are applied to a Random Forest classification model for steel plate fault detection to evaluate their ability to improve model transparency and trustworthiness.

4. Application of XAI Techniques: Case Study

This chapter presents a case study applying the XAI methods discussed earlier to the problem of classifying surface defects on steel plates. The dataset used brings together measurements of the physical and geometric properties of individual plates alongside the type of defect each one exhibits. The aim here is not merely to build a working classifier, but to examine how that classifier arrives at its decisions.

A conventional multi-class machine learning model is developed first, after which SHAP and LIME are applied to explain its predictions. Particular attention is given to both global and local explanations, and to how the two XAI techniques compare in terms of interpretability and practical relevance for industrial deployment.

4.1. Description of the Classification Problem and Dataset

4.1.1. Problem description and industrial motivation

Quality control is one of the most important steps in any modern production line. During the manufacturing of steel plates and sheets, various surface flaws inevitably appear. Left unaddressed, such defects can compromise the mechanical properties of the material, reduce its market value, and ultimately lead to the finished product being rejected outright. Manual inspection, the traditional response to this problem, is slow, costly, and inherently subjective, leaving it vulnerable to human error. Automating the process with machine learning allows defects to be classified quickly and consistently, in something close to real time.

Conventional predictive models used for this purpose, however, tend to behave as black boxes. A model may flag a defect with high accuracy and still leave production engineers with no idea why it reached that conclusion. Without insight into which factors actually drove a decision, it becomes difficult to adjust production parameters in time to prevent the same defect from recurring. Overcoming exactly this kind of opacity is the underlying motivation for the case study presented in this chapter.

4.1.2. Dataset characteristics

The analysis relies on the publicly available Steel Plates Faults dataset, obtained from the UCI Machine Learning Repository [16]. It contains 1,941 instances, each corresponding to a steel plate sample, described by 27 input features that capture the geometric and visual characteristics of a surface defect and labelled with one of seven possible fault types.

The input variables are derived from image analysis of industrial steel plates and represent a numerical encoding of the irregularities detected. They fall into three broad groups [16]:

- *Geometric and spatial properties*: the coordinates of the flaw on the plate (X_Minimum, X_Maximum, Y_Minimum, Y_Maximum), the area of the defect (Pixels_Areas), its perimeter (X_Perimeter, Y_Perimeter), and the thickness of the plate itself (Steel_Plate_Thickness).
- *Optical properties*: pixel luminosity within the defect region, which helps distinguish darker flaws such as stains from lighter surface anomalies (Sum_of_Luminosity, Minimum_of_Luminosity, Maximum_of_Luminosity).
- *Material type*: categorical variables encoded in binary form indicating the grade of steel used (TypeOfSteel_A300 and TypeOfSteel_A400).

The output variable identifies the fault type, with the dataset covering seven mutually exclusive classes of surface damage [16]:

- *Pastry*: a surface irregularity resembling a thickened or clumped layer of material on the plate.
- *Z_Scratch*: scratches produced during processing, handling, or transport of the material.
- *K_Scratch*: a distinct, typically deeper and more pronounced type of surface scratch.
- *Stains*: discolouration on the surface, most often caused by liquids, oils, or production processes.
- *Dirtiness*: contamination of the surface by dust, dirt, or other foreign particles.
- *Bumps*: physical protrusions on the surface caused by mechanical stress.
- *Other_Faults*: a catch-all category for rarer or less clearly defined defects that do not fit the categories above; it is also the most frequent class in the dataset.

Table 2. Sample distribution across defect classes

Defect class	Number of samples
Other_Faults	673
Bumps	402
K_Scratch	391
Z_Scratch	190
Pastry	158
Stains	72
Dirtiness	55



As Table 2 shows, the classification problem is defined over a markedly imbalanced dataset. Other_Faults is far better represented than minority classes such as Dirtiness and Stains, a distribution that is entirely typical of real industrial processes, where certain specific faults simply occur far less often than others. Because the Steel Plates Faults dataset consists exclusively of numerical measurements extracted through automated image analysis, rather than actual photographs of the defects, a visual sense of the raw data is instead given through annotated sample tables. For clarity, and given the practical limits on space, only six representative attributes are shown: Pixels Area, Plate Thickness, Luminosity Index, Conveyor Length, and the two binary steel-type indicators (A300 and A400), for five representative samples drawn from two classes, Bumps and Other_Faults (Table 3 and Table 4).

Table 3. Annotated samples from the Bumps class

Sample	Pixels Area (px)	Plate Thickness (mm)	Luminosity Index	Conveyor Length (mm)	Steel A300	Steel A400
1	3234	80	-0.157	1692	1	0
2	178	80	-0.228	1694	1	0
3	38	60	-0.115	1690	1	0
4	147	50	-0.213	1358	0	1
5	74	70	-0.116	1688	1	0

Table 4. Annotated samples from the Other Faults class

Sample	Pixels Area (px)	Plate Thickness (mm)	Luminosity Index	Conveyor Length (mm)	Steel A300	Steel A400
1	1695	300	-0.263	1692	0	1
2	949	80	-0.371	1687	1	0
3	169	80	-0.165	1687	1	0
4	836	150	-0.189	1360	1	0
5	381	143	-0.005	1694	1	0

4.2. Machine Learning Model Development

This section covers the preparation of the data and the construction of a classifier for steel-plate defects. The task is defined as multi-class classification, with every sample assigned to exactly one of the seven fault categories.

4.2.1. Data preparation

The dataset was first loaded and its columns assigned appropriate names, then converted to a numerical format suitable for machine learning algorithms; missing values and the overall structural integrity of the data were also checked at this stage. One characteristic of this particular dataset is that the seven defect classes are originally encoded as seven separate binary columns, where a value of 1 marks the presence of a given fault and 0 its absence. To enable multi-class classification, these columns were collapsed into a single target variable by identifying, for each row, the column carrying the maximum (i.e., positive) value.

The resulting class labels were then converted into numerical values using a LabelEncoder, completing the preparation of the dataset for the learning algorithm. To obtain an honest evaluation of the model on data it had not seen during training, the full set of 1,941 samples was split into a training set (80%) and a test set (20%). The training set is used to fit the model's parameters, while the test set is reserved exclusively for evaluating performance on previously unseen data, mimicking how the model would actually be used in production.

Given the pronounced class imbalance noted before, the split was stratified by class label, so that the proportion of each class remained approximately the same in both the training and the test set. This reduces the risk that minority classes such as Dirtiness or Stains end up under-represented in either subset, and, in turn, the risk of a model biased toward the dominant classes. A fixed random seed was used throughout to keep the split and the experiment as a whole - reproducible.

4.2.2. *Building the classification model*

Random Forest was selected as the classification algorithm. It is one of the best-known machine learning methods for classification and regression problems, built on the principle of ensemble learning - combining a large number of individual models to obtain predictions that are both more accurate and more stable than any single model could produce on its own [17].

In a Random Forest, each individual model is a decision tree. Rather than relying on a single tree, which tends to overfit the training data, the algorithm grows a large number of trees that together form the "forest"; the model's final decision is obtained by combining the outputs of every tree, improving generalisation and curbing overfitting.

Two mechanisms keep the individual trees sufficiently different from one another [17]:

- *Bagging (Bootstrap Aggregating)*: each tree is trained on a random subset of the data sampled with replacement from the original dataset, so that some observations appear in a given tree's training sample multiple times while others do not appear at all.
- *Feature randomness*: when splitting a node, each tree considers only a randomly selected subset of the available features rather than all of them, which keeps one or two dominant features from shaping every tree in the same way and keeps correlation between trees low.

Once all the trees have been trained, their outputs are aggregated to reach a final decision. For classification, each tree casts a vote for a class and the final prediction follows a majority vote; for regression, each tree instead predicts a numerical value and the final prediction is the average across all trees [17].

Random Forest is known for high accuracy, resistance to overfitting, and the ability to handle large datasets, including ones with missing values; it also supports feature-importance estimates that help clarify how the model reaches its decisions. Its main drawbacks are reduced interpretability, given the sheer number of trees involved, and higher memory and processing requirements compared with simpler models.

The algorithm was chosen here for its strong generalisation ability and its wide use in classification problems. As a black-box model, it is also well suited to demonstrating what explainable AI methods can add on top of a conventional classifier.

The model was configured with 300 decision trees to improve prediction stability and reduce variance, and a minimum leaf size of two samples to keep individual trees from over-fitting to isolated observations. A class-weighting scheme (balanced subsampling) was also introduced to counter the class imbalance identified earlier, giving smaller, under-represented classes more weight during training and reducing the model's bias toward the dominant classes.

Configured this way, the model achieves stable performance while remaining suitable for the post-hoc interpretation carried out using SHAP and LIME.

4.3. Model Performance Evaluation

Evaluating a model's ability to generalise to unseen data is a critical step in any machine learning pipeline. Here, the model was tested on the held-out test set, which played no role in training, in order to approximate how it would behave in a real deployment.

Several standard classification metrics were used to quantify performance:

- *Accuracy*: the proportion of correctly classified samples, positive and negative alike, out of the total number of samples.
- *Precision*: the proportion of predicted-positive samples that are actually positive, indicating how reliable the model's positive predictions are.
- *Recall*: the proportion of actual positive samples the model correctly identifies, a critical metric wherever missing a positive case is unacceptable.
- *F1-score*: the harmonic mean of precision and recall, used as a balanced measure, particularly valuable when classes are unevenly represented.
- *Support*: the number of actual samples belonging to a given class in the test set - a property of the data itself rather than of the model's performance.

The Random Forest model achieved an overall accuracy of 78.66%. A per-class classification report, shown in Table 5, offers a more detailed picture.

Table 5. Classification report on the test set

Class	Precision	Recall	F1-score	Support
Bumps	0.7162	0.6543	0.6839	81
Dirtiness	0.8333	0.9091	0.8696	11
K_Scratch	0.9726	0.9103	0.9404	78
Other_Faults	0.7200	0.8000	0.7579	135
Pastry	0.5758	0.5938	0.5846	32
Stains	0.9231	0.8571	0.8889	14
Z_Scratch	0.9706	0.8684	0.9167	38

The support column confirms the imbalance already visible in the full dataset, with Other_Faults dominating the test set and Dirtiness represented by only a handful of samples. The model performs strongly on K_Scratch and Z_Scratch, identifying these fault types with very few errors, while performance is weakest on Pastry, where the model frequently confuses this class with other types of imperfection. Bumps also show only moderate results, with roughly a third of genuine defects missed. Notably, Dirtiness stands out despite its tiny sample size, as the model successfully catches most real contamination cases even with very little training data to learn from.

A confusion matrix (Figure 4) was also used to examine the model's errors in more detail, tracking the four possible outcomes of a classification decision: true positives, true negatives, false positives (Type I errors), and false negatives (Type II errors) [18].

The confusion matrix confirms that the model performs best on classes with more training examples and struggles more with under-represented ones - an expected consequence of the imbalance discussed earlier, since dominant classes exert a stronger pull on what the model learns. Other_Faults is recognised most reliably, unsurprisingly given its size, and K_Scratch and Z_Scratch are also classified fairly well; Dirtiness and Stains show high precision, though their small sample sizes warrant some caution in that reading. The most pronounced confusion occurs between Bumps and Other_Faults, with a substantial number of samples crossing from one class into the other - a sign that the two share visually similar characteristics. Pastry is also frequently misclassified, chiefly as Other_Faults or Bumps, suggesting its defining patterns are either weaker or under-represented in the data.

These evaluation metrics indicate reasonably solid overall performance, but they are inherently limited when it comes to interpretability: they quantify how well the model performs without ever answering the more useful question of why a particular decision was made, or which input characteristics actually drove it.

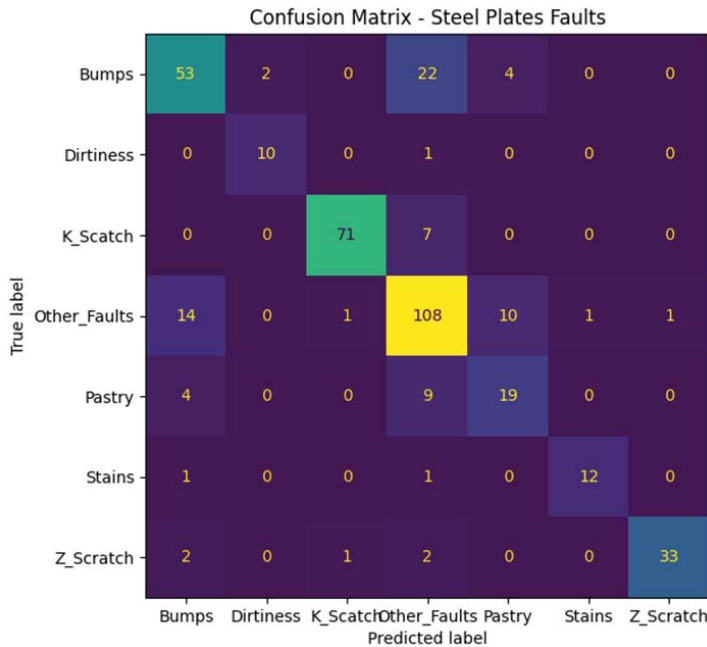


Figure 4. Confusion matrix [18]

Knowing that a prediction is right or wrong says nothing about which attributes led the model there. It is precisely this gap that motivates the explainable AI analysis that follows, which aims to expose the model's internal reasoning and identify the features most responsible for each decision.

4.4. Model Explanation Using SHAP and LIME

This section examines, in detail, how the trained model actually uses its input features when classifying steel-plate defects. Where the previous section focused on overall accuracy and aggregate performance, the analysis here turns to the model's internal logic - the contribution each individual attribute makes to a given decision. A single representative case from the test set is used throughout for the local analyses, since walking through one concrete example in detail gives a far more intuitive sense of the model's reasoning than trying to summarise many cases at once.

4.4.1. Global model analysis using SHAP

To understand how the model behaves across the dataset as a whole, a global SHAP (SHapley Additive Explanations) analysis was carried out. Rather than explaining one prediction at a time, this analysis surfaces the general patterns and

priorities the model relies on. Because the underlying classifier is a Random Forest, SHAP values were computed using the TreeExplainer implementation from the SHAP library, which is specifically optimised for efficient and exact computation on tree-based models.

The first step in interpreting the model globally is identifying its most influential inputs. Figure 5 ranks the 15 most important variables the Random Forest relies on when classifying steel-plate defects, with bar length representing each feature's mean absolute SHAP value across the test set.

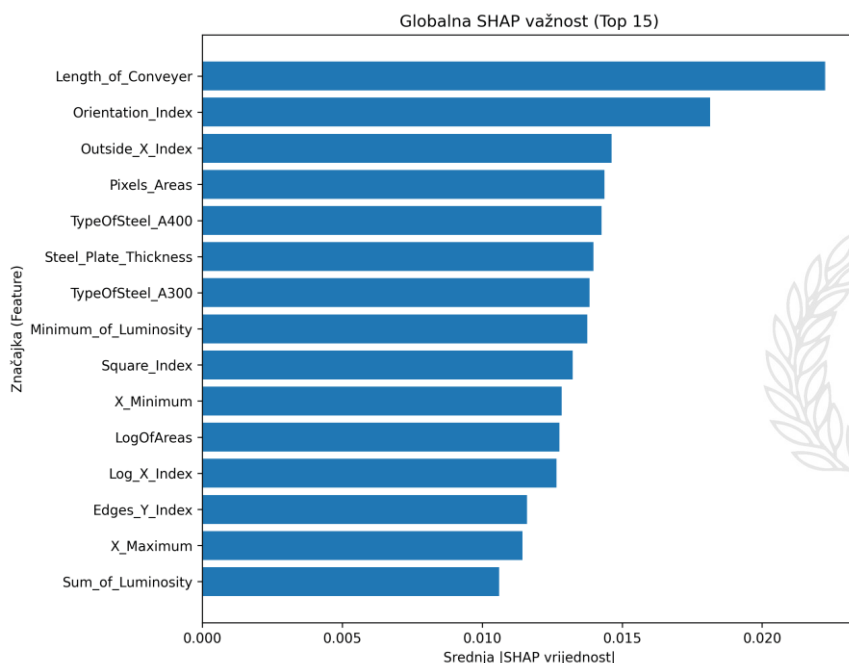


Figure 5. Global feature importance derived from SHAP [18]

Identifying the most influential input characteristics:

- **Production and operational parameters dominate.** Length_of_Conveyer stands out as the single most important feature by a clear margin. From an industrial standpoint this is a genuinely actionable insight: it suggests that the occurrence of specific defects is tied directly to the length of the production line, or the particular zone in which a plate is processed and transported.
- **Defect geometry and orientation matter.** The next most influential variables are Orientation_Index and Outside_X_Index, closely followed by total defect area (Pixels_Areas). This tracks with basic engineering

intuition: different defect types have recognisably different physical shapes - scratches are elongated and oriented in a particular way, while stains occupy a different kind of area and shape altogether - and the model leans heavily on these spatial cues when making its decisions.

- **Material characteristics carry real weight.** Variables describing the material itself - steel type (TypeOfSteel_A400, TypeOfSteel_A300) and plate thickness (Steel_Plate_Thickness) - rank high among the most important features, confirming the industrial intuition that particular alloys or thicknesses are more prone to specific fault types (thinner plates, for instance, deform more easily and are more likely to develop Bumps).
- **Optical characteristics play a supporting role.** Variables describing pixel brightness, such as Minimum_of_Luminosity, also carry meaningful weight, confirming that the model successfully exploits contrast and brightness variation to separate darker defects such as dirtiness from lighter surface anomalies.

Taken together, the global SHAP analysis shows that the black-box model is not basing its decisions on noise, but on physically and operationally meaningful geometric, material, and process-related characteristics. For engineers and quality-control staff on the factory floor, the decisive role played by Length_of_Conveyer points to conveyor-mechanism inspection as a priority for reducing defect rates.

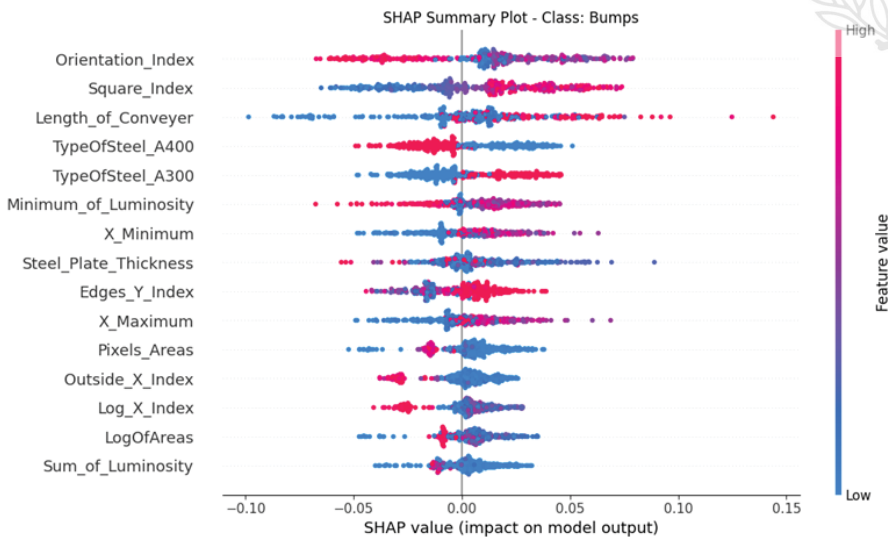


Figure 6. SHAP summary plot for the Bumps defect class [18]

To see not just which features matter but how their actual values push a prediction up or down, a SHAP summary plot was generated for the Bumps class, shown in Figure 6, chosen as one of the most heavily represented fault types among the seven the model distinguishes.

Each point on this plot represents one test-set sample; its position along the horizontal axis (the SHAP value) shows whether that feature pushed the probability of a Bumps defect up or down for that particular sample. A close reading of the point distribution yields several concrete rules the model appears to be applying:

- **Geometry (Orientation_Index and Square_Index):** low values of the orientation index (a cluster of blue points on the right) push strongly toward a positive Bumps prediction, while high values of the square index (red points, also on the right) likewise raise the predicted probability.
- **Operational parameters (Length_of_Conveyer):** unusually high conveyor lengths (red points shifted sharply to the right) substantially raise the likelihood of this defect - a critical, production-relevant finding, since it suggests longer conveyor sections are more prone to generating physical bumps in the material.
- **Material type (TypeOfSteel_A300 versus A400):** the two steel grades pull in opposite directions - high values (presence) of A300 sit on the positive side of the SHAP axis, while high values of A400 push the prediction the other way, indicating that A300 steel is considerably more prone to developing Bumps than A400.

This level of granularity, and this kind of insight into how individual parameters interact, is simply unavailable from conventional evaluation metrics alone - underscoring why an explainability layer is not just a nice-to-have but a genuine necessity for tackling complex industrial classification problems.

4.4.2. *Local analysis of individual decisions using SHAP and LIME*

While the global SHAP analysis reveals broad patterns across the dataset, practical deployment on a production line often calls for something narrower: understanding why the model flagged one specific plate as defective. An engineer working in real time needs to know exactly which parameters drove the classification of the piece of steel currently in front of them, and that diagnostic need is what local interpretation is built to satisfy. To illustrate this, one test-set sample flagged as defective by the model was isolated for detailed analysis. The result, visualised as a waterfall plot, is shown in Figure 7.

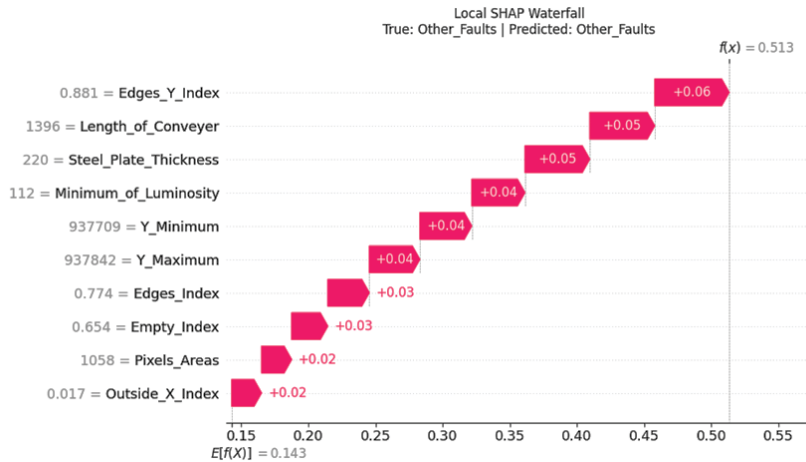


Figure 7. Local SHAP interpretation of a single model decision [18]

As indicated in the figure, this sample's true class is Other_Faults, and the model classified it correctly. The plot traces the model's decision path from a baseline value - the expected prediction $E[f(X)] = 0.143$, the average probability the model would assign to the Other_Faults class for any random sample before considering its specific characteristics - up to the final prediction $f(x) = 0.513$ (51.3%) once the actual features of this particular plate were taken into account, at which point the sample was classified as Other_Faults.

Every one of the key variables for this sample pushed the prediction in a positive direction (shown as red segments moving right). The model's reasoning breaks down as follows:

- **Edge geometry (Edges_Y_Index):** this was the single strongest driver of the positive classification, contributing +0.06 to the predicted probability.
- **Operational parameters (Length_of_Conveyer):** consistent with the global analysis, conveyor length again contributed meaningfully to the classification (+0.05).
- **Material characteristics (Steel_Plate_Thickness):** the comparatively large plate thickness further reinforced the model's decision (+0.05).
- **Spatial and optical characteristics:** additional, smaller contributions came from optical features such as minimum luminosity and the precise spatial coordinates of the defect on the plate (Y_Minimum and Y_Maximum).

Table 6 presents the SHAP values for the analyzed sample. The feature_value column shows the actual value of each feature, shap_contribution indicates whether the feature increased (positive) or decreased (negative) the predicted

probability, and abs_shap gives the absolute contribution used to rank feature importance.

Table 6. SHAP values for the selected sample

Feature	Feature value	SHAP contribution	SHAP
Edges_Y_Index	0.8808	0.055441	0.055441
Length_of_Conveyer	1396	0.048365	0.048365
Steel_Plate_Thickness	220	0.048276	0.048276
Minimum_of_Luminosity	112	0.039601	0.039601
Y_Minimum	937709	0.038666	0.038666
Y_Maximum	937842	0.037634	0.037634
Edges_Index	0.7736	0.030908	0.030908
Empty_Index	0.6541	0.026932	0.026932
Pixels_Areas	1058	0.022451	0.022451

These figures are consistent with, and confirm, the reading given by the waterfall plot.

LIME was applied as a complementary local method. Where SHAP offers global consistency grounded in cooperative game theory, LIME instead targets local fidelity, explaining a single prediction by fitting a simple, interpretable surrogate model in its immediate neighbourhood. A tabular LIME explainer was fitted on the training data, discretising continuous features into interpretable ranges, and an explanation was then generated for the same test-set sample used above.

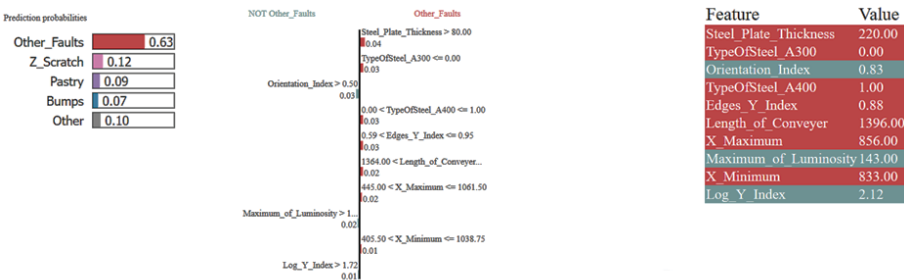


Figure 8. Local explanation of the model's prediction using LIME [18]

The LIME output splits into three parts: predicted class probabilities on the left, the local decision rules and their weights in the middle, and the actual feature values for this specific sample on the right, Figure 8. Consistent with the underlying Random Forest, the LIME surrogate model identifies Other_Faults as the most likely class, at a probability of 0.63.

The middle panel, where red bars raise the probability of Other_Faults and blue bars lower it, points to several key drivers behind this classification: plate thickness above 80.00 (actual value 220.00) is among the strongest contributing factors, while the steel-grade indicators pull in different directions, with certain grades (e.g., A400) pushing the probability up and others pulling it down. Edge geometry (Edges_Y_Index in the 0.59–0.95 range, actual value 0.88) and conveyor length (1,364.00–1,650.00 mm, actual value 1,396.00) further reinforce the decision, while the defect's X-axis position contributes a smaller, but still non-trivial, share.

LIME also surfaced a couple of counter-factors: a high orientation index (Orientation_Index > 0.50) and maximum luminosity above 140.00 both pointed away from Other_Faults, suggesting the sample might belong to a different class. Their combined negative influence, however, was not large enough to overturn the final classification.

Taken together, the global and local explanations produced by SHAP and LIME converge on a consistent set of influential parameters, rather than pointing to conflicting or arbitrary signals. Across all three analyses, conveyor length (Length_of_Conveyer), defect geometry (particularly edge- and orientation-related indices), steel type, and plate thickness repeatedly emerge as the strongest drivers of classification, while optical features such as luminosity play a smaller, supporting role.

The fact that an entirely different explanation method (LIME, based on local surrogate modelling) independently highlights largely the same features as SHAP (grounded in cooperative game theory) strengthens confidence that these parameters reflect genuine, physically meaningful patterns in the data rather than artefacts of a single explanation technique. From a practical standpoint, this convergence gives engineers and quality-control staff a reliable, cross-validated basis for prioritising interventions - for instance, inspecting conveyor mechanisms and monitoring plate thickness and steel grade during production - rather than relying on the interpretation of a single method alone.

5. Discussion: XAI in Smart Manufacturing Systems

5.1. Contribution of XAI to AI-Based Industrial Applications

Explainable artificial intelligence plays an important role in improving the transparency, reliability, and practical adoption of AI-based solutions in industrial environments. One of the key contributions of XAI is supporting decision-making on the shop floor by revealing the factors that influence model predictions. Instead of providing only black-box outputs, XAI methods enable operators and engineers to better understand system behaviour and make more informed interventions [19].

Furthermore, XAI supports the validation and verification of AI models before and during deployment. By identifying the most influential features, methods such as SHAP help engineers assess whether models rely on meaningful physical relationships or unwanted data correlations. For example, SHAP-based analysis in predictive maintenance applications has demonstrated the ability to identify critical variables related to equipment degradation, improving the reliability of maintenance decisions [20].

Another important contribution of XAI is its ability to reveal data-related issues, including spurious correlations and inconsistencies within training datasets. By analysing feature contributions and model behaviour, XAI techniques can support dataset refinement and continuous improvement of AI models [21]. Therefore, XAI represents an essential component for developing trustworthy, transparent, and reliable AI systems in modern industrial applications.

5.2. Challenges in Practical Deployment of XAI

Despite the significant benefits of XAI, its practical deployment in industrial environments remains challenging. One of the main limitations is computational complexity, particularly for model-agnostic methods such as SHAP and LIME. Generating accurate explanations can require substantial computational resources, especially for complex models and large-scale industrial datasets, which limits their application in real-time scenarios [22], [23].

Another important challenge is adapting explanations to the needs of different users. Although XAI methods can provide detailed insights into model behaviour, their outputs are often designed for data scientists rather than industrial operators and maintenance experts. Therefore, developing domain-specific explanations that are intuitive and actionable for end users remains an open research challenge [19], [24].

Furthermore, scalability represents a critical issue in real-world manufacturing systems. Industrial environments typically involve multiple AI models, heterogeneous data sources, and resource-constrained edge devices, requiring efficient and adaptable explanation frameworks. Consequently, future XAI solutions must address not only model transparency but also robustness, scalability, and integration across complete industrial workflows [25], [26].

5.3. Future Perspectives

Future development of XAI in smart manufacturing is expected to focus on deeper integration with industrial infrastructures, particularly digital twins. Combining XAI with digital twins can provide not only predictive insights but also transparent explanations and uncertainty information, enabling operators to better understand and trust AI-driven decisions [27], [28]. Integrating explanation mechanisms directly into digital-twin platforms could make interpretability a natural component of industrial monitoring and decision-support systems.

Another important direction is the development of inherently transparent and trustworthy AI models that reduce dependence on post-hoc explanation techniques. Future industrial AI systems should incorporate context-aware explanations, self-explainability, and standardized evaluation methods to ensure reliable operation in dynamic manufacturing environments [24], [29].

Overall, the future of XAI in smart manufacturing will depend not only on developing new explanation algorithms but also on their effective integration into real industrial workflows, where transparency, reliability, and human understanding are essential requirements for successful AI adoption.

6. Conclusion

The increasing adoption of artificial intelligence in industrial applications has significantly improved predictive capabilities; however, the growing complexity of machine learning models has created new challenges related to transparency and trust. In safety-critical manufacturing environments, accurate predictions alone are not sufficient, as engineers and operators also require an understanding of the factors influencing AI-driven decisions.

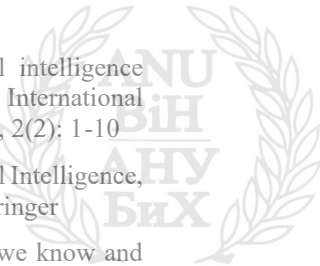
This paper investigated the role of XAI in industrial fault detection through a case study based on the Steel Plates Faults dataset using a Random Forest classification model. The application of model-agnostic explanation methods, specifically SHAP and LIME, demonstrated their ability to reveal the most influential features

and provide insights into both global model behaviour and individual predictions. These explanations improve model transparency and support better understanding, validation, and interpretation of AI-based decisions in manufacturing applications.

Although XAI methods provide significant benefits, their practical adoption still faces challenges related to computational complexity, explanation usability, and integration into real industrial workflows. Achieving an effective balance between predictive performance, interpretability, and computational efficiency remains an important research direction.

Overall, by transforming complex machine learning predictions into understandable and actionable information, techniques such as SHAP and LIME represent a key step toward trustworthy, human-centred AI systems for smart manufacturing.

References

- 
- [1] Desai, P. (2021), A comprehensive survey on explainable artificial intelligence (XAI): challenges, opportunities, and future research directions, *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, 2(2): 1-10
 - [2] Guidotti, R., Schmid, U., Longo, L. (Eds.) (2024), *Explainable Artificial Intelligence, Communications in Computer and Information Science*, vol. 2580, Springer
 - [3] Ali, S. et al. (2023), Explainable Artificial Intelligence (XAI): what we know and what is left to attain trustworthy artificial intelligence, *Information Fusion*, 99: 101805
 - [4] European Union (2016), Article 22 GDPR – Automated individual decision-making <https://gdpr-info.eu/art-22-gdpr/>
 - [5] Artificial Intelligence Act EU (2024), “EU AI Act Overview,” <https://artificialintelligenceact.eu/>
 - [6] Doshi-Velez, F., Kim, B. (2017), Towards a rigorous science of interpretable machine learning, *arXiv preprint arXiv:1702.08608*
 - [7] Molnar, C. (2022), *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable* (2nd ed.)
 - [8] Wilkinson, C., Yawney, J., Gadsden, S.A. (2026), Explaining explainability: a comprehensive survey on explainable artificial intelligence and relevant industry applications, *Intelligent Systems with Applications*, 30: 200647
 - [9] Zschech, P., Weinzierl, S., and Kraus, M. (2025), Inherently Interpretable Machine Learning: A Contrasting Paradigm to Post-hoc Explainable AI, *Business & Information Systems Engineering*, 68(2): 445-463

- [10] MathWorks (2026), Interpretability, MathWorks Inc., available at: <https://fr.mathworks.com/discovery/interpretability.html>
- [11] Mersha, M., Lam, K., Wood, J., AlShami, A. (2022), Explainable artificial intelligence: a survey of needs, techniques, applications, and future directions, IEEE Access, 10, arXiv preprint arXiv: 2409.00265
- [12] Devireddy, K. (2023), Comparative study of explainable AI methods: model-agnostic vs. model-specific approaches, arXiv preprint arXiv:2504.04276
- [13] Ribeiro, M.T., Singh, S., Guestrin, C. (2016), Model-agnostic interpretability of machine learning, ICML Workshop on Human Interpretability in Machine Learning
- [14] Adadi, A., Berrada, M. (2018), Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI), IEEE, 52138-52160
- [15] Lundberg, S.M., Lee, S.-I. (2017), A unified approach to interpreting model predictions, Advances in Neural Information Processing Systems (NeurIPS), 30
- [16] Buscema, M., Terzi, S., Tastle, W. (2010), Steel Plates Faults, UCI Machine Learning Repository <https://archive.ics.uci.edu/dataset/198/steel+plates+faults>
- [17] Breiman, L. (2001), Random Forests, Machine Learning, 45(1), 5-32
- [18] Mezetović, N. (2026), *Explainable Artificial Intelligence (XAI): Methods, Models and Application in Reliable Decision-Making*, Bachelor's thesis, Faculty of Electrical Engineering, University of Tuzla, Tuzla, Bosnia and Herzegovina
- [19] Ahangar, M.N., Farhat, Z.A., Sivanathan, A., Ketheesram, N., Kaur, S. (2026), Explainable AI-driven quality and condition monitoring in smart manufacturing, Sensors (MDPI), 26(3): 911
- [20] Tzionis et al. (2026), A review of explainable AI methods and their application in manufacturing systems, Discover Applied Sciences, Springer Nature <https://link.springer.com/article/10.1007/s42452-025-07908-z>
- [21] Xuan, X., Ono, J.P., Gou, L., Ma, K.-L., Ren, L. (2024), AttributionScanner: a visual analytics system for model validation with metadata-free slice finding, arXiv preprint arXiv:2401.06462
- [22] Dehghani, Z. (2026), Addressing explainability of generative AI using SMILE (Statistical Model-agnostic Interpretability with Local Explanations), arXiv preprint arXiv:2602.01206
- [23] Ezeji, I. N., Adigun, M. O., Oki, O. (2025), Strategic selection of application area for optimizing computational complexity in explainable decision support system using multi-criteria decision analysis (MCDA), Journal of Systemics, Cybernetics and Informatics, 23(2), 36-47
- [24] Zemmouchi-Ghomari, L. (2026), Explainable AI for predictive maintenance: a review and standardized evaluation framework, Management Science Letters, 16(1), 15-36 <https://www.researchgate.net/publication/398123644>

- [25] Moss, J., Gordon, J., Duclos, W., Liu, Y., Wang, Q., Wang, J. (2025), Explainable AI in IoT: a survey of challenges, advancements, and pathways to trustworthy automation, *Electronics*, 14(23), 4622 <https://www.mdpi.com/2079-9292/14/23/4622>
- [26] Ahangar, M. N., Farhat, Z. A., Sivanathan, A. (2025), AI trustworthiness in manufacturing: challenges, toolkits, and the path to Industry 5.0, *Sensors (Basel)*, 25(14), 4357 <https://pmc.ncbi.nlm.nih.gov/articles/PMC12298069/>
- [27] Oladimeji, O. O., Taiwo, G. A., Adenusi, D. A. (2026), Unveiling the footprints of explainable AI in Industry 4.0/5.0: a systematic review and bibliometric exploration, *Data Science and Management*, 9(1), 100160 <https://www.sciencedirect.com/science/article/pii/S2666764925000359>
- [28] Cohen, J., Huan, X. (2024), Uncertainty-aware explainable AI as a foundational paradigm for digital twins, *Frontiers in Mechanical Engineering*, 9, 1329146
- [29] Moosavi, S., Farajzadeh-Zanjani, M., Razavi-Far, R., Palade, V., Saif, M. (2024), Explainable AI in manufacturing and industrial cyber-physical systems: a survey, *Electronics*, 13(17), 3497 <https://www.mdpi.com/2079-9292/13/17/3497>

