International Journal of Structural Integrity

Crack detection in powder compacts using machine learning models

Sameen Mustafa
Faculty of Engineering, Libera Università di Bolzano, Bolzano, Italy
Elias Ganthaler
GKN Powder Metallurgy, Brunico, Italy
Attaullah Buriro
School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK and
Faculty of Engineering, Libera Università di Bolzano, Bolzano, Italy
Anton Dignös
Faculty of Engineering, Libera Università di Bolzano, Bolzano, Italy
Thomas Villgrattner
GKN Powder Metallurgy, Brunico, Italy, and
Franco Concli and Angelika Peer
Faculty of Engineering, Libera Università di Bolzano, Bolzano, Italy

Received 22 May 2025
Revised 23 July 2025
Accepted 4 August 2025

Abstract

Purpose – Cracks in powder metallurgy (PM) components, being a common problem, pose significant manufacturing challenges but are detrimental to be detected as they affect the material's mechanical properties. To detect these cracks, non-destructive testing (NDTs) methods are often used, but they come with high costs and time delays, as samples need to be extracted from production at given intervals. To overcome these limitations, an indirect method of crack detection, that is, modelling it as a binary classification, is explored in this work.

Design/methodology/approach – This study introduces a supervised machine learning (ML) approach using force signal feature extraction to detect cracks. More specifically, a supervised learning algorithm is developed and validated for the classification of samples into samples with or without cracks based on the sensory data of the hydraulic press used for production. We compare different ensemble classifiers, including random forest (RF), AdaBoost (ADA), bagging, gradient boosting (GB) and extra trees (ET), in terms of their ability to classify workpieces using a dataset from real production.

Findings – To this end, the present study deals with experimental workpieces of a specific type produced by manually adjusting the press parameters to artificially induce cracks in parts of the workpieces. The best-performing model resulted in a classification accuracy as high as 99% offering a cost-effective and efficient alternative to traditional NDT methods.

Originality/value – This study provides a novel and indirect method for detecting cracks in PM components using ML models trained on press sensor data, which can significantly reduce the need for costly and time-consuming NDT techniques.

Keywords Crack detection, Feature extraction, Powder metallurgy, Supervised learning, Ensemble classifiers, Powder compaction

Paper type Research article

1. Introduction

In powder metallurgy (PM), cracks are usually defined as fractures or discontinuities in the powder compact (green/sintered) that are detrimental to be detected, as they affect the

© Sameen Mustafa, Elias Ganthaler, Attaullah Buriro, Anton Dignös, Thomas Villgrattner, Franco Concli and Angelika Peer. Published by Emerald Publishing Limited. This article is published under the Creative Commons Attribution (CC BY 4.0) licence. Anyone may reproduce, distribute, translate and create derivative works of this article (for both commercial and non-commercial purposes), subject to full attribution to the original publication and authors. The full terms of this licence may be seen at Link to the terms of the CC BY 4.0 licence.

Disclosure statement: No potential conflict of interest is reported by the authors

International Journal of Structural Integrity
Emerald Publishing Limited
e-ISSN: 1757-9872
p-ISSN: 1757-9864
DOI 10.1108/IJSI-05-2025-0131

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

material's mechanical properties. There can be different causes of crack occurrence, such as mechanical loading, improper compaction, etc. (Tweed, 2008; Mustafa *et al.*, 2023).

Detection of cracks is critical to ensure the quality and longevity of the parts produced. Non-destructive testing (NDT) methods, such as ultrasonic testing, radiography and magnetic particle inspection, are traditionally employed to identify such defects without damaging the specimens (Sharma, 2023; Kumpati *et al.*, 2021; Gandhi *et al.*, 2022).

Generally, NDT methods are complex and involve high operational costs (Sharma, 2023; Kumpati *et al.*, 2021). The equipment required for techniques such as ultrasonic testing or radiography is not only expensive but also demands regular calibration and maintenance (Gandhi *et al.*, 2022). Moreover, these methods typically require skilled operators. In addition, methods such as X-ray tomography involve the use of hazardous materials, necessitating stringent safety protocols and specialised handling. Furthermore, the application of NDT methods can be time-consuming. The setup of the equipment, the preparation of specimens, the execution of the tests and the interpretation of the results all contribute to delays in the quality assurance process (Achenbach, 2000; Balayssac and Garnier, 2017). The need for potentially halting operations to conduct quality assurance measures can lead to production losses, which is particularly detrimental in high-demand sectors such as the manufacturing sector (Hellier, 2003).

Given the limitations of NDT methods, there has been a recent growth in the development of cost-effective and efficient methods for crack detection. Machine learning (ML), particularly supervised learning, is emerging as a promising alternative. Using labelled data and sophisticated algorithms, ML models can be trained to identify crack-indicating patterns, potentially serving as an indirect method for crack detection (Zhao *et al.*, 2019). Furthermore, recent studies have emphasized the importance of accounting for signal variability and small-sample uncertainty in mechanical failure analysis, which further motivates the application of data-driven classification methods for crack detection (Liu *et al.*, 2022a, b). Supervised learning involves training a model on a labelled dataset, where the model learns to associate specific features with given outcomes. In the context of crack detection, inputs to the models can include surface images, acoustic emissions, or vibration signals, the results being "crack" or "no-crack" (Hellier, 2003). Once trained, these models can rapidly analyse new samples and accurately classify them, and thus overcome the shortcomings of traditional NDT methods.

The adoption of supervised learning models for crack detection offers several advantages: Firstly, once developed, these models can be deployed at a relatively low cost compared to traditional NDT equipment. Primary expenses involve initial data collection and model training (LeCun *et al.*, 2015). Secondly, ML models can provide real-time analysis, enhancing efficiency and reducing downtime. By integrating these models with existing monitoring systems, industries can continuously assess the quality and structural integrity of their assets without interrupting operations (Goodfellow, 2016). This capability is particularly valuable in sectors where continuous operation is critical, such as in manufacturing lines (Bishop and Nasrabadi, 2006). Supervised learning models can be trained on diverse datasets, improving their ability to detect even small defects (Ronneberger *et al.*, 2015). This study aims to develop and validate a supervised learning model for the detection of cracks in specimens. Using a comprehensive data set and a unique feature extraction method, the proposed model seeks to offer a cost-effective and efficient alternative to traditional NDT methods. The key contributions of this study are the introduction of an accurate and lightweight ML-based pipeline for classifying the produced workpieces into workpieces with and without cracks, and identification of the most important features that significantly impact the crack detection capability of the proposed model.

The remainder of the article is structured as follows: Section 2 outlines the materials and methods, the experimental design to produce cracks in green workpieces using a hydraulic press, the processing of data and feature extraction, the classification protocol, and the methods of classification by employing different ML models. Section 3 details various plots of

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

the processed data, the plots of decision boundaries, and results of the models used in the study for classification. The article concludes with Section 4, which presents a summary of the contributions and potential future research directions.

2. Experimentation and methodology

Green compacts were produced using a hydraulic powder press under various constant settings. The part being examined is illustrated in Figure 1. It is a 4-level compact, manufactured using a press equipped with two upper punches (TR and UL2) and three lower punches (LL1, LL2, and LL3) active during production, see Figure 2, which presents a schematic of the hydraulic powder press used to produce the part (for further details, see Ganthaler *et al.* (2023)). A typical press cycle for part production, along with the forces of the different punches (levels) in operation, is depicted in Figure 3. For more in-depth explanations, interested readers are directed to the study by MoradiMaryamnegari *et al.* (2023). Within each cycle, the compaction and ejection phases are particularly critical, as cracks are known to occur during the transition from the compaction to the ejection phase. In a hydraulic press, the press position (PP) refers to the position of the punch where the press exerts maximum force by moving the ram or piston downward to compress or shape the material. During this stage, the hydraulic fluid is actively pressurising the system to perform the required work. The relieve position (RP) corresponds to the position of the punches when the press releases the applied pressure by retracting the piston, allowing the material to be removed and preparing the press for the next operation. This also accounts for the elastic effect of the tool attached to the punch.

2.1 Experimental design

Cracks were artificially induced at different positions in the specimen (as indicated in Table 1) by changing the RPs of different punch levels of the hydraulic press. In total, four experiments were performed. Experiment 1 served as the baseline, featuring samples without cracks. The remaining three experiments involved cracks at different locations on the workpiece. Table 1 also outlines the experimental setups for each trial, where cracks were introduced by adjusting the RPs of the press. This adjustment involved either maintaining alignment of the RPs at their original status (oRP) or modifying the RPs of different punch levels to generate cracks.

In the experimental design for this study, two factors were considered: The relieve position (RP) of the levels and the expected crack position (CP). As independent variables, the respective RP were considered for each of the punches: LL1, LL2, LL3, TR and UL2. The dependent variable CP corresponds to the position of the crack occurrence in the specimen: no-crack, TR-UL2, LL1-LL2, and LL2-LL3. CP is the outcome variable, indicating whether a crack is expected at a specific position based on the change in the RP. The RP for each level was gradually changed in small steps to achieve a dataset with very close settings for results of cracks and no-cracks, to make the classification problem more challenging, and ensure that the dataset also contains non-trivial cases for classification. Initially, the step-size was chosen as 0.1 mm in order to limit the number of experimental settings. The RP of the different punches

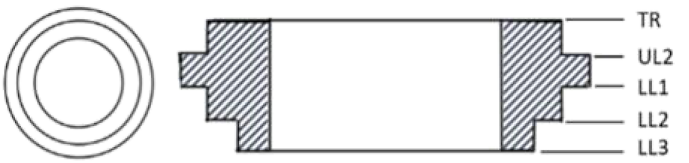

Figure 1. The experimental workpiece. Top view (left) and cross-sectional view (right). Source: Mustafa *et al.* (2024)

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

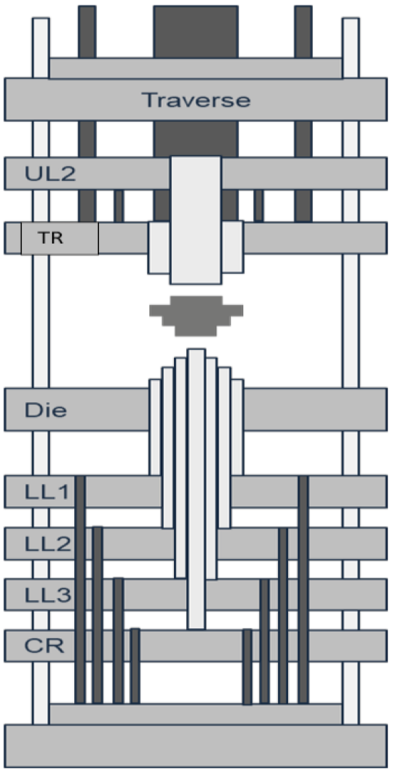

Figure 2. Schematic of the hydraulic press. Source: Mustafa *et al.* (2024)

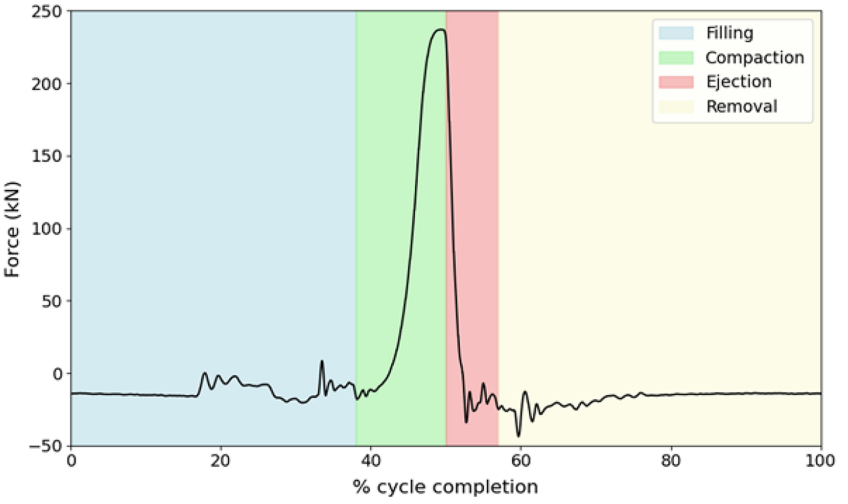

Figure 3. The press cycle with different phases: filling, compaction, ejection and removal. Source: Authors' own work

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

Table 1. Relieve position and expected crack position

Exp.	Relieve position (mm) LL1	LL2	LL3	TR	UL2	Expected crack position
1	= oRP	= oRP	= oRP	= oRP	= oRP	No-crack (base line)
2	= oRP	= oRP	= oRP	Up	= oRP	TR-UL2
3	Down	Down	Down	Up	Up	LL1-LL2
4	Up	= oRP	= oRP	Up	= oRP	LL2-LL3

Source(s): Authors' own work

was changed gradually until a crack was seen, and this particular value of RP was taken as a reference. The level was then moved back by half the step size (0.05 mm) to check for cracks. If there was still a crack, the level was moved back again by half the previous step size (0.025 mm). This iterative process was applied to Experiments 2–4 in Table 1. For example, for collecting the data for cracks at the location TR-UL2, the iterative process of change in the RPs can be seen in the settings 2 a–2 g in Table 2, where "x" stands for no-crack and "TR-UL2" denotes the presence of cracks at that particular position. For this location, the experiment started with an initial RP of 0.3 mm, moving by 0.1 mm to check for crack occurrence. At 0.6 mm the presence of a crack was observed, and the RP was reduced by half step to 0.55 mm. Since the crack was still present at 0.55 mm, again the RP was moved by the same step size to a value of 0.5 mm. At this point, no-crack was observed. The RP was now moved back by half the previous step size to a value of 0.525 mm, and the cracks occurred again. Data from these last two settings were used for classification by labelling them as 0 and 1 for no-crack and

Table 2. Different relieve positions and resulting crack positions

Exp. No.	Relieve position (mm) LL1	LL2	LL3	TR	UL2	Real crack position
1 (baseline)	−0.48	−0.23	0	0.2	0.56	x
2 a	−0.48	−0.23	0	0.3	0.56	x
2 b	−0.48	−0.23	0	0.4	0.56	x
2 c	−0.48	−0.23	0	0.5	0.56	x
2 d	−0.48	−0.23	0	0.6	0.56	TR-UL2
2 e	−0.48	−0.23	0	0.55	0.56	TR-UL2
2 f	−0.48	−0.23	0	0.5	0.56	x
2 g	−0.48	−0.23	0	0.525	0.56	TR-UL2
3 a	0	−0.23	0	0.3	0.56	x
3 b	0	−0.23	−0.1	0.3	0.56	x
3 c	0	−0.23	−0.2	0.3	0.56	LL2-LL3 (inside)
3 d	0	−0.23	−0.15	0.3	0.56	LL2-LL3 (inside)
3 e	0	−0.23	−0.125	0.3	0.56	LL2-LL3 (inside)
3 f	0	−0.23	−0.11	0.3	0.56	LL2-LL3 (inside)
3 g	0	−0.23	0.105	0.3	0.56	x
4 a	0	−0.23	0.05	0.4	0.5	x
4 b	0	−0.23	0.05	0.4	0.4	x
4 c	0	−0.23	0.05	0.4	0.3	TR-UL2
4 d	0	−0.23	0.05	0.4	0.35	TR-UL2
4 e	0	−0.23	0.05	0.4	0.375	x

Source(s): Authors' own work

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

crack, respectively, with a balanced dataset of 200/200 samples. This table also shows the different relieve positions and the corresponding occurrence of cracks at different locations. During the experiment, it was found that in contrast to expectations, cracks could only be induced at two positions, as detailed in the table.

2.2 Data processing and feature extraction

The dataset consists of various time-dependent values recorded using different sensors installed on the press. The hydraulic press implements built-in low-level closed-loop controllers, which use a sampling rate of 2 kHz. The press also has dedicated pressure sensors installed in the double-acting hydraulic cylinders for each level of the press (TR, UL2, etc.) from which the force acting on each level is calculated. This is done by dividing the difference in the pressures in the cylinders by the respective cross-sectional area of the tool attached to the level in action. This force calculation is pre-programmed and recorded in real time. These recordings also define the completion of the press cycle for producing a workpiece. Data is recorded at different instances of the full press cycle on a percent basis.

Position signals were not considered, as the tool position has to be calibrated each time a new experiment is run and a new tool is mounted (change of specimen shape, remounting the tools, etc). Further, as the press uses a fixed die, the powder can be pressed at different heights, and thus the punch positions cannot be considered for this method, as it will be different after every new setup of the machine.

In order to train the model for consistent performance over a large dataset recorded under different setup conditions and to overcome calibration issues, only signals that are independent from calibration could be considered, and thus only the force signals were chosen for this study.

The recorded data needed to be synchronised and converted into an organised form through data preprocessing since classifiers are presumed to work better on the pre-processed data rather than raw data, leading to higher accuracies and better reliability (Massimo *et al.*, 2023). In this study, the variables considered are the forces of each active level during production. Here, the feature extraction process involves calculating the difference between forces of each of the five active levels and then performing statistical operations to calculate the mean, minimum, maximum, and standard deviation of these differences. These are denoted as *min_forc_TR_UL2* or *std_forc_TR_UL2* to denote, respectively, the minimum force difference between TR and UL2 and the standard deviation of the force differences between TR and UL2. These features were calculated by using the formulae below:

$$\mu = \frac{1}{n}\sum_{k=1}^{n}\Delta F_k, \quad \sigma = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(\Delta F_k - \mu)^2}, \quad \Delta F_k = F_k^{(i)} - F_k^{(i+1)}$$

where μ is the mean, σ is the standard deviation, and ΔF_k is the difference between the force of the adjacent levels.

Altogether, there are 16 extracted features that are converted into a data frame to be used as a feature matrix for ML algorithms. While more complex time-domain or mutation-based features could potentially enrich the input space, our objective in this study was to develop a method that is practical for real-world deployment on industrial production lines. Thus, we limited the feature set to simple, interpretable statistical measures, as explained earlier, to ensure computational efficiency, ease of implementation, and robustness across varying process conditions.

The press cycle can be divided into 4 phases (see Figure 3), whereby relevant for cracks are the compaction and the ejection phases. Thus, it was decided to evaluate the ML algorithms on sectors corresponding to the compaction and ejection phases only.

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

2.3 Classifier selection and hyperparameter optimization

Classifiers serve as fundamental components in ML, enabling models to learn patterns from data and make informed decisions by assigning labels to new, unseen samples. In this study, we employed several tree-based classifiers as our classification models, specifically random forest (RF) (Breiman, 2001), AdaBoost (ADA) (Schapire, 2013), gradient boosting (GB) (Konstantinov and Utkin, 2021), bagging (Opitz and Maclin, 1999) and extra trees (ET) (Kharwar and Thakor, 2022) classifiers. The choice was informed by (1) the fact that our data was not linear, (2) their demonstrated effectiveness in prior research work (Massimo *et al.*, 2023), (3) their equal effectiveness on small/large datasets, and (4) their robustness to overfitting.

Additionally, hyperparameter tuning plays a critical role in maximising the performance of these ensemble models (Bergstra and Bengio, 2012). Properly tuning parameters such as the number of trees in RF or the learning rate in ADA can significantly influence the model's accuracy and efficiency. Grid search was performed for each of the models to achieve optimal performance by finding the best combination of parameters. The hyperparameter ranges explored are listed in Table 3.

The selected hyperparameter ranges were informed by commonly recommended values in the literature for tree-based ensemble methods and refined through preliminary experiments. For example, learning rates in the range of 0.01–1 balance learning stability and convergence speed in boosting-based models. Similarly, maximum depth values ranging from 10 to 30 allow the trees to capture sufficient data complexity while avoiding overfitting, which is particularly important given our structured and low-dimensional feature space. These ranges ensure coverage of both conservative and more expressive model configurations without making the search space excessively large.

The process of choosing the optimal model is detailed in algorithm 1. The algorithm outlines a process for hyperparameter tuning and evaluation of the classifiers used in this study. It begins by defining parameter grids for the classifiers, each with various hyperparameters to be tuned. These classifiers are initialised with a fixed random state for reproducibility. For each classifier, a grid search is performed to find the best hyperparameters based on accuracy. The best classifiers are then evaluated through additional cross-validation to calculate accuracy scores and confusion matrices. The results, including mean accuracy, standard deviation and general classification reports, are printed and visualised with heatmaps (see Figure 11) of the confusion matrix for each classifier.

Table 3. Hyperparameter tuning for different models

Classifier	Hyperparameter	Range
RF	Number of estimators	25 to 200 in steps of 25
	Maximum depth	None to 30 in steps of 10
ADA	Number of estimators	25 to 200 in steps of 25
	Learning rate	0.01, 0.1, 1
GB	Number of estimators	25 to 200 in steps of 25
	Learning rate	0.01, 0.1, 0.2
	Maximum depth	3, 4, 5
Bagging	Number of estimators	25 to 200 in steps of 25
	Maximum samples	0.5, 0.7, 1
ET	Number of estimators	25 to 200 in steps of 25
	Maximum depth	None to 30 in steps of 10
SVM	C	0.1, 1, 10,100
	Kernel	linear, rbf, poly
	Gamma	scale, auto

Source(s): Authors' own work

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

Algorithm 1. Hyperparameter tuning and evaluation of classifiers

Define Parameter Grids and Initialize Classifiers:

Define parameter grids and initialize classifiers: *param_grid_rf*, *rf_classifier*, etc.

Create Classifiers Dictionary:

Store classifiers and parameter grids in a dictionary: *classifiers*

Initialize StratifiedKFold:

kf ← StratifiedKFold(n_splits = 5, shuffle = True, random_state = 42)

Perform Grid Search:

for each *name*, (*classifier*, *param_grid*) in *classifiers* **do**

grid_search ← GridSearchCV(*classifier*, *param_grid*, cv = *kf*)

grid_search.fit(X, y)

best_classifiers[name] ← *grid_search.best_estimator_*

end for

Evaluate Best Models:

for each *name*, *model* in *best_classifiers* **do**

accuracy_scores ← []

for each *train_index*, *test_index* in *kf.split*(X, y) **do**

X_train, *X_test* ← X[train_index], X[test_index]

y_train, *y_test* ← y[train_index], y[test_index]

model.fit(X_train, y_train)

y_pred ← *model.predict*(X_test)

accuracy_scores.append(*accuracy_score*(y_test, y_pred))

end for

Calculate mean, std of accuracy: *mean_accuracy*, *std_accuracy*

y_pred_all ← *cross_val_predict*(model, X, y, cv = *kf*)

overall_conf_matrix ← *confusion_matrix*(y, *y_pred_all*)

Print results: classification reports, accuracies, confusion matrices.

end for

2.4 Evaluation using decision trees and support vector machines

To visualise the decision boundaries between the classes “crack” and “no-crack” for each experiment, decision tree (DT) and support vector machine (SVM) classifiers were used. Plotting decision boundaries using DTs is a valuable technique for visualising how these models classify data in a multi-dimensional space. By graphically representing the regions where different classes are predicted, insights can be gained into the model’s decision-making process and the complexity of the underlying data distribution. This visualisation helps

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

identify potential issues such as overfitting, where the model may create overly complex boundaries that do not generalise well to unseen data. Similarly, plotting decision boundaries for SVM models reveals how these models maximize the margin and separate classes in transformed feature spaces, helping to diagnose underfitting or overfitting (Shahbudin, 2010). Recent studies emphasise that visualising decision boundaries not only aids in model evaluation but also fosters trust in ML applications by making the decision-making process more accessible to non-experts (Blockeel *et al*., 2023; Costa and Pedreira, 2023).

Algorithm 2 explains the process to plot the decision boundaries by using the extracted features in the feature matrix. Firstly, a Principal Component Analysis (PCA) is applied to reduce the feature matrix X to two principal components for easier 2D visualisation. A DT classifier is then initialised, and a 5-fold cross-validation is performed to maintain balanced class distribution. For each fold, the model is trained on the training set and evaluated on the test set, with accuracy scores recorded. Finally, the decision boundaries of the trained classifier are plotted on a mesh grid, with the PCA-transformed data overlaid as a scatter plot, providing a visual representation of how the classifier separates different classes in the reduced dimensional space. The SVM classifier was trained and evaluated, and the decision boundaries were plotted in a similar way as defined in Algorithm 2.

Algorithm 2. Plotting Decision Boundaries using DTs

Apply PCA:

X_pca ← PCA(*n_components* = 2).fit_transform(X){Reducing feature matrix X to 2 Principal Components. }

Initialize Cross-Validation and Classifier:

kf ← StratifiedKFold(*n_splits* = 5, *shuffle* = *True*, *random_state* = 42)

dt_classifier ←DecisionTreeClassifier(*random_state* = 42)

Cross-Validation:

for each *train_index*, *test_index* in *kf.split*(*X_pca*, y) **do**

X_train, *X_test* ← *X_pca*[*train_index*], *X_pca*[*test_index*]

y_train, *y_test* ← y[*train_index*], y[*test_index*]

dt_classifier.fit(*X_train*, *y_train*)

accuracy_scores.append(accuracy_score(*y_test*, *dt_classifier*.predict(X_test)))

end for

Print Results:

mean_accuracy ←np.mean(accuracy_scores)

std_accuracy ←np.std(accuracy_scores)

Print mean_accuracy ± std_accuracy

Plot Decision Boundary:

Create mesh grid for plotting decision surface.

Use DecisionBoundaryDisplay.from_estimator to plot the decision boundary.

Scatter plot PCA-transformed data.

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

2.5 Classification protocol

For each experimental setting (each row in Table 2), data for 200 workpieces was collected. The stroke rate of the press was kept constant at 35 strokes per minute for every setting, where each stroke corresponds to the production of one workpiece. For each of the experiments with cracks in Table 1, features from the two closest settings were extracted, where the difference in the relief positions was minimum.

For each of the three crack locations in Table 2 balanced dataset of 200/200 samples was used for the crack/no-crack conditions corresponding to these two closest settings. The reason behind the selection of only the closest settings is that during the experiments the positions of the punches were changed gradually according to the design of the experiment. Thus, we were interested in detecting only these small, subtle changes between crack and no-crack conditions. After feature extraction, the dataset was labelled as 0 for samples without cracks and 1 for those with cracks. For classification, the feature matrix of dimension $n \times 16$ was used, where n is the number of samples, each with 16 features. The target vector contained binary values of 0 and 1. In addition to classifying cracked versus non-cracked workpieces within each experiment, a combined classification task was performed to identify the crack location in samples belonging to different experiments.

A 5-fold cross-validation (see Figure 4) was performed due to its advantages in terms of model evaluation and generalisation performance (Yadav and Shukla, 2016). By dividing the dataset into five subsets and training the model multiple times, cross-validation reduces variance in performance estimates, leading to more stable and reliable results. This iterative process helps mitigate overfitting risks, ensuring that the chosen model generalises well to unseen data, making x-fold cross-validation a preferred choice in many ML applications. To evaluate the performance of the classifiers using 5-fold cross-validation, 10-fold cross-validation was also performed, and the accuracies of both methods were compared.

3. Results and discussions

To provide insights into the difficulty level of the classification problem, the difference between the forces of each level in action for the samples with cracks and no-crack are shown in Figure 5–7. As can be observed from these plots, the force curves for the samples with and without cracks are not much distinct and cannot be easily distinguished visually, and hence the problem is non-trivial for a human decision maker. Thus, it requires further analysis and evaluation using different ML models to detect cracks.

3.1 Training of different models and hyperparameter tuning

The best hyperparameters for the different classifiers across experiments are summarised in Table 4. These models, with their respective tuned hyperparameters, were utilised for training

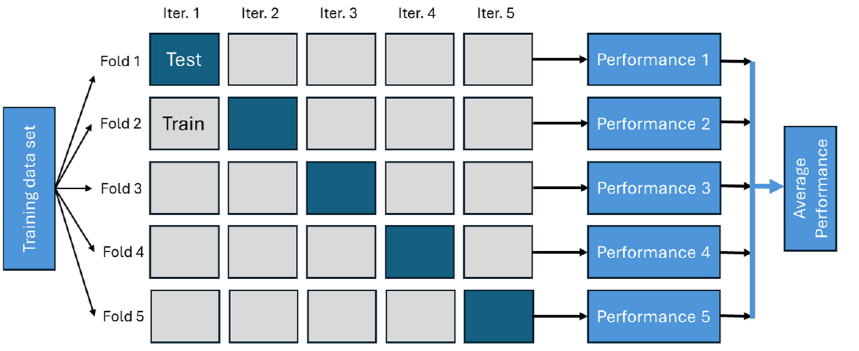

Figure 4. 5-fold cross validation. Source: Authors' own work

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

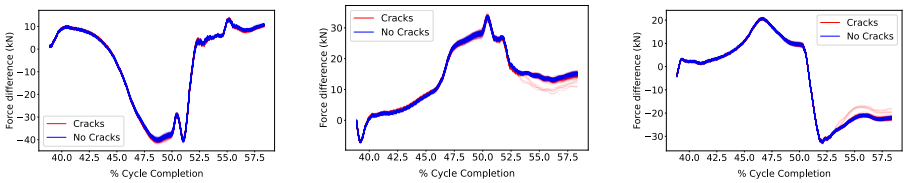

Figure 5. Difference in forces between LL1-UL2, LL1-LL2, and LL2-LL3, respectively, during compaction and ejection for Exp. 2. Source: Authors' own work

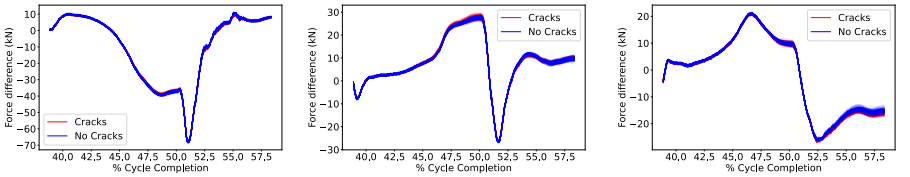

Figure 6. Difference in forces between LL1-UL2, LL1-LL2, and LL2-LL3, respectively, during compaction and ejection for Exp. 3. Source: Authors' own work

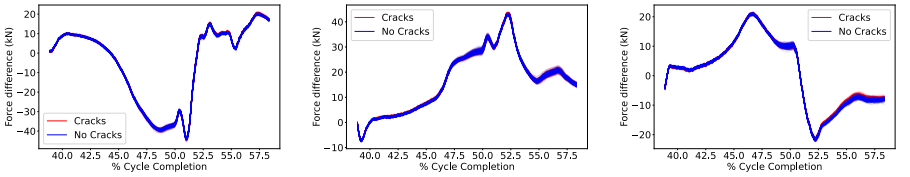

Figure 7. Difference in forces between LL1-UL2, LL1-LL2, and LL2-LL3, respectively, during compaction and ejection for Exp. 4. Source: Authors' own work

Table 4. Best hyperparameters for different models across experiments

Classifier	Hyperparameter	Exp. 2	Exp. 3	Exp. 4
RF	Number of Estimators	100	200	125
	Maximum Depth	None	None	None
ADA	Number of Estimators	200	200	50
	Learning Rate	1	1	1
GB	Number of Estimators	200	125	200
	Learning Rate	0.2	0.2	0.2
	Maximum Depth	3	3	4
Bagging	Number of Estimators	100	50	150
	Maximum Samples	0.7	0.5	0.7
ET	Number of Estimators	25	200	125
	Maximum Depth	None	10	None
SVM	C	1	0.1	100
	kernel	rbf	rbf	rbf
	gamma	scale	scale	scale

Source(s): Authors' own work

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

and evaluation in each individual experiment as well as for combined analysis across experiments. Specifically, the learning rate for ADA and GB classifiers remained unchanged, suggesting that these models exhibit robustness and generalisability in their hyperparameter configurations when applied to different datasets. Conversely, the hyperparameters for RF, bagging and ET classifiers varied across experiments. The RF, bagging, and ET classifiers did not use the same hyperparameters across all experiments. While ADA and GB ended up with stable (or consistent) hyperparameters in each experiment.

While this study employed individually tuned hyperparameters for each experiment, future research could explore the feasibility of identifying average or generalised hyperparameter values for RF, bagging, and ET classifiers. Such an approach could streamline model deployment across diverse datasets, reducing computational costs associated with repeated hyperparameter optimisation. Additionally, exploring advanced tuning techniques such as Bayesian optimization (Snoek *et al.*, 2012) or hyperband (Li, 2018) could further refine these findings and enhance model reliability across varying experimental setups.

3.2 Visualisation using decision boundaries

The decision boundaries generated by the DT and SVM classifiers (Figures 8 and 9) illustrate the separation between the two classes of samples with cracks and without cracks based on the two principal components. While the simplicity of DT classifiers makes them interpretable, the performance varies significantly across the experiments due to the level of overfitting, as reflected in the accuracy.

3.2.1 DT. For Exp. 2, the decision boundaries are relatively simple, resulting in an accuracy of 78%. The model successfully captured the general patterns in the data without much overfitting. However, some misclassification was evident in regions where the two classes overlap significantly. On the other hand, for Exp. 3 and Exp. 4, the decision boundaries became more complex, reflecting the DT classifier's attempt to model fine-grained details in the training data. This results in overfitting, where the model focuses on noise rather than meaningful trends. Consequently, the accuracy dropped to 54%, indicating poor generalisation.

3.2.2 SVM. Exp. 2 demonstrates a relatively clear separation between the two classes of crack/no-crack, reflected in the highest observed accuracy of 85% among all the three

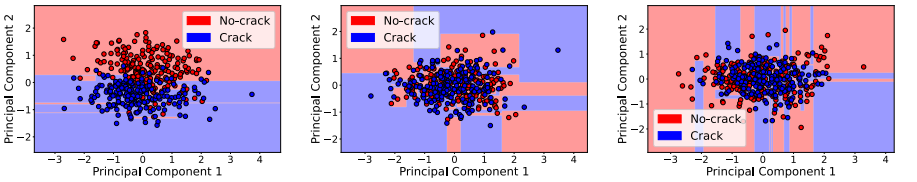

Figure 8. Decision boundaries from DT for: Exp. 2, accuracy = 78% (left), Exp. 3, accuracy = 54% (middle) and Exp. 4, accuracy = 55% (right). Source: Authors' own work

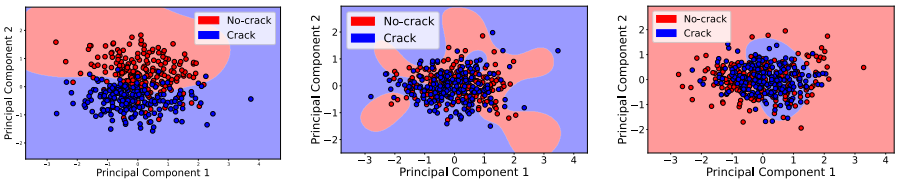

Figure 9. Decision boundaries from SVM for: Exp. 2, accuracy = 85% (left), Exp. 3, accuracy = 59% (middle) and Exp. 4, accuracy = 58% (right). Source: Authors' own work

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

experiments. The SVM boundary here is well aligned with the distribution of the data points, indicating that the features extracted in this experiment provide good discriminatory power between cracked and non-cracked samples. The classifier successfully created a margin that generalises well, with minimal misclassification along the boundary.

In contrast, plots of Exp. 3 (59% accuracy) and Exp. 4 (58% accuracy) show significantly less distinct class separation. The decision boundaries in these cases are noticeably more complex and irregular, failing to capture the true underlying distributions. The overlap between classes is substantial, resulting in higher misclassification rates and much lower predictive performance. This might suggest that the signal patterns or feature sets for these experiments do not contain sufficient information for the SVM to effectively distinguish between crack and no-crack samples, or that the class distributions themselves are less separable in the chosen feature space.

Overall, these results highlight the sensitivity of SVM classifiers to the quality and discriminatory power of the extracted features. While SVM is capable of drawing complex, non-linear boundaries, its effectiveness is ultimately constrained by the inherent separability of the data.

3.2.3 Decision boundaries for different locations of cracks. Figure 10 compares the decision boundaries for different locations of cracks in the workpiece using DT and SVM. The DT boundaries in this case are simple, leading to a high accuracy of 93%. Similarly, SVM also performs well with a slightly higher accuracy of 94%. For both the models, it can be observed that the clusters corresponding to each location of cracks are well-separated in the PCA-transformed space. Here, the decision boundaries split the feature space into distinct regions, effectively classifying most samples with minimal overlap. One notable observation is that in the case of DT, the no-crack region is confined to a narrow vertical strip, suggesting that the

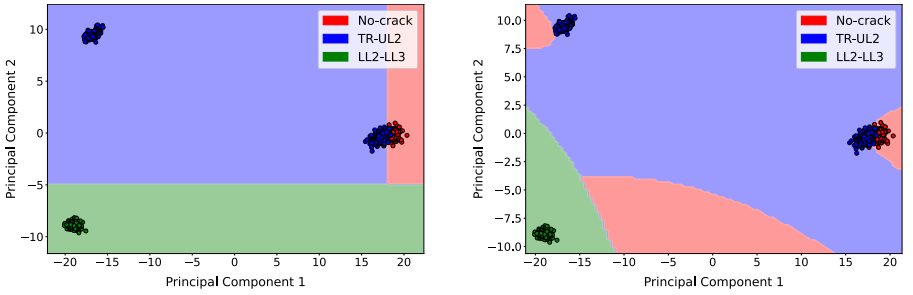

Figure 10. Decision boundaries for different locations of cracks: DT, accuracy = 93% (left) and SVM, accuracy = 94% (right). Source: Authors' own work

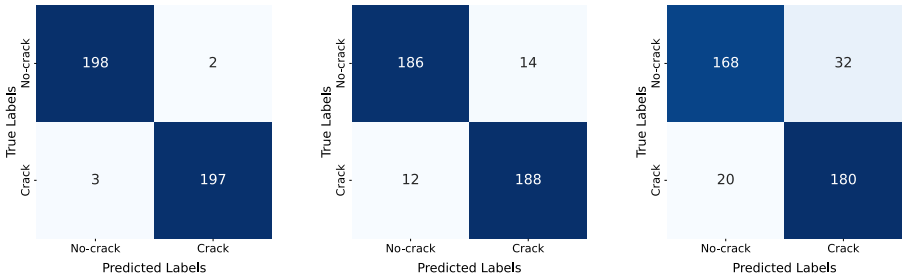

Figure 11. Confusion Matrices of the best-performing model for Exp. 2: ADA (left), Exp. 3: GB (middle) and Exp. 4: ET (right). Source: Authors' own work

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

principal components provide a clear separation for this class. The TR-UL2 and LL2-LL3 regions are primarily divided by a horizontal boundary, reflecting a strong distinction along the second principal component. The clear separation of clusters indicates that the principal components effectively capture the variance among the three classes, allowing the DT to make relatively straightforward and interpretable splits. However, there is a little overlap between some samples of no-crack and TR-UL2 regions (samples belonging to Exp. 2), which might result from similar settings of the RPs as shown in Table 2, where the only difference lies between the RPs of the TR.

For SVM, the data points for each class are well separated in the PCA-transformed feature space, resulting in distinct and largely non-overlapping clusters. The SVM has generated clear, smoothly curved boundaries that partition the space according to the natural grouping of the data. This clustering suggests that the features extracted for these classes are highly discriminative, allowing the SVM to achieve a high classification accuracy with minimal ambiguity along the boundaries. The effectiveness of the SVM in this scenario demonstrates the benefit of strong class separability in the feature space, both for model interpretability and for predictive accuracy. The clear margins between classes indicate that the chosen features, when combined with dimensionality reduction, provide robust information for distinguishing between different crack locations. This result contrasts with Exp. 2–4, where the classes overlap substantially, highlighting the importance of both feature engineering and visualisation for assessing classification feasibility in practical applications.

Altogether, the complex decision boundaries for Experiments 2–4 observed for DT and the low accuracy found for the SVM classifier indicate that the problem of visually classifying the samples in classes of cracks and no-cracks within single experiments is not trivial, as discussed previously. Thus, the choice of ensemble classifiers in the classification analyses is justified.

3.3 Performance analyses

Table 5 summarises the performance of the five classifiers across the three experiments using two metrics: F1-score (for crack and no-crack classes) and 5-fold cross-validation accuracy (expressed as x% ± y%, where x is the mean accuracy across five folds and y is the percentage variation). The results indicate that classifier performance varies depending on the experimental dataset, with notable differences in both metrics across classifiers and experiments.

Table 5. Comparison of classifier performance across experiments

Experiment	Metric (Crack/ No-Crack)	RF	ADA	GB	Bagging	ET
Exp. 2	Precision	0.98/0.96	0.98/0.98	0.97/0.97	0.97/0.97	0.99/0.99
	Recall	0.96/0.98	0.97/0.98	0.97/0.97	0.96/0.97	0.98/0.99
	F1-Score	0.97/0.97	0.98/0.98	0.97/0.97	0.97/0.97	0.99/0.99
	5-Fold Accuracy	97 ± 2%	98 ± 1%	97 ± 1%	97 ± 2%	99 ± 1%
Exp. 3	Precision	0.93/0.92	0.93/0.93	0.93/0.94	0.91/0.91	0.94/0.92
	Recall	0.92/0.93	0.93/0.93	0.94/0.93	0.91/0.92	0.92/0.94
	F1-Score	0.92/0.93	0.93/0.93	0.94/0.93	0.91/0.91	0.93/0.93
	5-Fold Accuracy	93 ± 3%	93 ± 3%	94 ± 2%	91 ± 3%	93 ± 3%
Exp. 4	Precision	0.85/0.83	0.83/0.86	0.84/0.85	0.85/0.84	0.85/0.89
	Recall	0.82/0.85	0.83/0.86	0.85/0.83	0.84/0.85	0.90/0.84
	F1-Score	0.84/0.84	0.84/0.85	0.85/0.84	0.84/0.85	0.87/0.87
	5-Fold Accuracy	84 ± 4%	84 ± 2%	84 ± 2%	85 ± 3%	87 ± 2%

Source(s): Authors' own work

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

In Experiment 2, ET achieved the highest F1-scores (0.99 for both classes) and 5-fold accuracy (99% ± 1%), outperforming other classifiers. ADA closely followed with an F1-score of 0.98 and 98% ± 1% accuracy. RF, GB, and Bagging demonstrated slightly lower, but comparable performance, with consistent F1-scores of 0.97 and accuracies around 97% ± 1–2%. These results suggest that the classifiers are highly effective at distinguishing between crack and no-crack cases for this dataset, with ET being particularly robust.

In Experiment 3, performance across all classifiers declined slightly compared to Experiment 2. This might stem from the fact that the crack is present between LL2 and LL3, and the RP of these levels has higher positioning resolution. Further, the lower levels correspond to greater relative movement of the punches and more powder volume in the die, potentially leading to noise in the dataset. GB demonstrated the highest F1-score (0.93 for crack and 0.94 for no-crack) and accuracy (94% ± 2%), indicating its ability to generalise better under these conditions. RF, ADA, and ET showed comparable F1-scores (around 0.93) and accuracies (93% ± 3%), while Bagging had the lowest performance (F1-score of 0.91 and accuracy of 91% ± 3%).

Experiment 4 presented the most challenging dataset, reflected in overall lower performance metrics for all classifiers. ET again outperformed others, achieving the highest F1-scores (0.87 for both classes) and accuracy (87% ± 2%). The remaining classifiers exhibited similar performance, with F1-scores around 0.84–0.85 and accuracies ranging from 84% ± 2%–85% ± 3%. This indicates that while all classifiers struggled with the complexity of this dataset, ET maintained a slight advantage in both precision and consistency.

These findings suggest that ET consistently provides superior performance across datasets of varying complexity, while GB and ADA also demonstrate reliable generalisation capabilities. The results further highlight the impact of dataset characteristics on model performance, emphasising the need for careful model selection and evaluation based on specific application requirements. Future work could explore ensembling these models or employing advanced techniques for feature engineering to improve classification outcomes further.

The outputs of the best-performing model for each of the experiments are shown as confusion matrices in Figure 11. In all three experiments, there are a few misclassified pieces, as can be seen in Figure 11, with the minimum accuracy of 85% for the best-performing model for Exp. 4. In contrast, in the case of all the crack locations combined, all the classifiers predicted cracks at each location with a perfect classification accuracy of 100%, irrespective of the values of the hyperparameters used. Finally, Figure 12 also shows the classifier performance in detecting the cracks at different locations in the workpiece. This indicates that

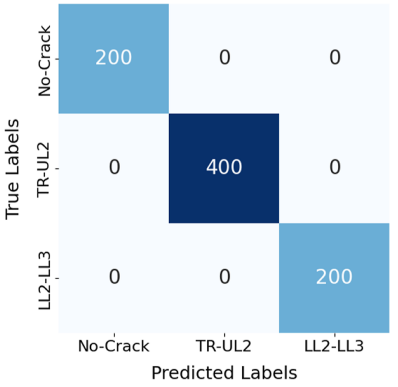

Figure 12. Classification based on the location of cracks. Source: Authors' own work

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

the models can identify the crack locations as different clusters far apart from each other. This further implies that in future applications, with a large data set containing labelled data based on cracks occurring at different levels of a complex workpiece, the crack locations can easily be distinguished using any of the five classifiers and with less computational cost.

We also evaluated the above models using 10-fold cross-validation and the results indicated the same mean accuracy as in 5-fold cross-validation.

Overall, the five ensemble classifiers used in this study overcome the limitations of single DT and SVM models by combining the predictions of multiple trees. These methods inherently reduce overfitting and improve generalisation due to reduced variance by aggregating the predictions of multiple decorrelated trees, and by focussing on correcting errors iteratively, leading to improved model performance. The ensemble models consistently achieved better performance metrics, suggesting their ability to capture complex patterns without overfitting. Although DT and SVM models suffer from overfitting when decision boundaries become overly complex, as shown in Experiments 3 and 4, the ensemble classifiers effectively addressed this issue, as exhibited by their high classification accuracies.

3.4 Feature importance

Additionally, the feature importance for each of the best performing models was studied. Figure 13 shows the feature importance scores for each of the three experiments. The values for the feature importance for a model sum up to 1, where larger values indicate a more important feature for classification in a particular dataset. For Exp. 2, the most important features involved the force differences between LL2 and LL3, and LL1 and UL2. A high importance to *max_forc_diff_LL1_UL2* aligns with the location of the crack present between TR and UL2, where TR and UL2 move relative to each other. On the other hand, higher importance to *min_forc_diff_LL2_LL3* signifies higher resultant forces occurring around these lower levels. This might occur due to the upward shift in the RPs of TR during part ejection.

For Exp. 3, the most important feature was *std_forc_diff_LL2_LL3*, accounting for about 50% importance among all features. This feature denotes the standard deviation of the force differences between LL2 and LL3, which aligns with the CP in Exp. 2, i.e. between LL2 and LL3. Since the cracks were induced by shifting the RP of LL3 downwards, a high importance given to features involving LL2 and LL2 confirms the presence of cracks at the expected position (LL2-LL3). In the case of Exp. 4, the cracks were induced by shifting the RP of UL2 upwards, which is also reflected in the third plot, where *max_forc_diff_LL1_UL2* is one of the most important features. In this study, all features were used to evaluate the models, as the total number of extracted features was feasible considering the total computational time.

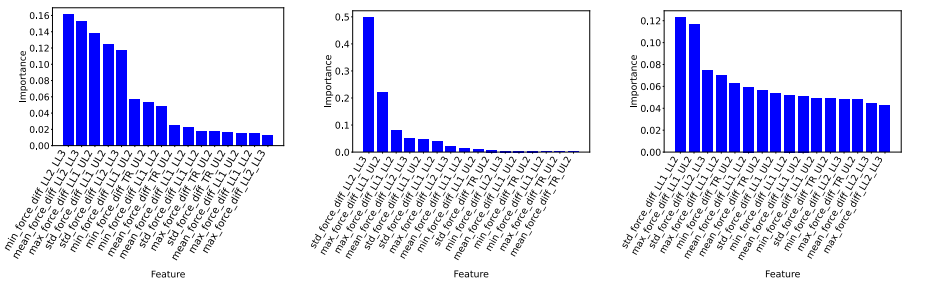

Figure 13. Feature importance for Exp. 2 (TR-UL2), Exp. 3 (LL2-LL3) and Exp. 4 (TR-UL2), respectively. Source: Authors' own work

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

4. Conclusion

This study presented a robust, indirect method of crack detection in metal powder compacts using supervised ML models. The results are promising, with high accuracy of the ML algorithms to classify the datasets into good and bad pieces based on the labels of crack and no-crack. Consequently, a robust and simple alternative to traditional NDT methods is the key finding of this study. While commonly used NDT methods in PM can be complex and cost-intensive, they are also often destructive in practice—such as when, during production, the samples inspected under a microscope need to be discarded, or when destructive cross-sectioning after sintering needs to be performed for observing cracks. In contrast, the present approach can detect cracks in production lines based on the extracted features from the real datasets, offering a more efficient and non-destructive alternative. The proposed method of crack detection seems to be promising for use online during production as future research, making this indirect method of testing a better and cheaper alternative.

However, a key drawback of supervised learning approaches is the substantial effort required to collect and label large, diverse datasets for training. This limitation underscores the potential of unsupervised learning techniques, which could reduce dependence on labelled data while still enabling effective defect detection. Future research could explore unsupervised learning models like isolation forest, etc., to improve adaptability across different types of parts produced by PM, further enhancing the feasibility of this method as a cost-effective alternative. Further research will focus on ensembling the individual classifiers used in this study to be able to develop a model that generalises over different settings and types of parts produced. Also, the study of crack propagation mechanisms presents a promising avenue for future research, building upon the modelling techniques described in this paper. Future work may consequently also focus on integrating continuous monitoring to capture crack evolution and developing modelling frameworks capable of predicting propagation paths under varying process conditions.

Acknowledgments

This work was partially supported by the European Union Next-Generation EU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 3.3 – Decreto del Ministero dell'Università e della Ricerca n.352 del 09/04/2022) within the Advanced-Systems Engineering Ph.D. Program at Free University of Bozen-Bolzano and the company GKN Powder Metallurgy.

References

Achenbach, J.D. (2000), "Quantitative nondestructive evaluation", *International Journal of Solids and Structures*, Vol. 37 Nos 1-2, pp. 13-27, doi: 10.1016/s0020-7683(99)00074-8.

Balayssac, J.-P. and Garnier, V. (2017), *Non-destructive Testing and Evaluation of Civil Engineering Structures*, Elsevier.

Bergstra, J. and Bengio, Y. (2012), "Random search for hyper-parameter optimization", *Journal of Machine Learning Research*, Vol. 13 No. 2, pp. 281-305.

Bishop, C.M. and Nasrabadi, N.M. (2006), *Pattern Recognition and Machine Learning*, Springer, New York, Vol. 4.

Blockeel, H., Devos, L., Frénay, B., Nanfack, G. and Nijssen, S. (2023), "Decision trees: from efficient prediction to responsible AI", *Frontiers in Artificial Intelligence*, Vol. 6, 1124553, doi: 10.3389/frai.2023.1124553.

Breiman, L. (2001), "Random forests mach learn", Vol. 45 No. 1, pp. 5-32, doi: 10.1023/a:1010933404324.

Costa, V.G. and Pedreira, C.E. (2023), "Recent advances in decision trees: an updated survey", *Artificial Intelligence Review*, Vol. 56 No. 5, pp. 4765-4800, doi: 10.1007/s10462-022-10275-5.

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

Gandhi, N., Rose, R., Croxford, A.J. and Ward, C. (2022), "Understanding system complexity in the non-destructive testing of advanced composite products", *Journal of Manufacturing and Materials Processing*, Vol. 6 No. 4, p. 71, doi: 10.3390/jmmp6040071.

Ganthaler, E., MoradiMaryamnegari, H., Villgrattner, T. and Peer, A. (2023), "Automatic trajectory adaptation for the control of quality characteristics in a powder compaction process", *Journal of Manufacturing Processes*, Vol. 107, pp. 268-279, doi: 10.1016/j.jmapro.2023.09.060.

Goodfellow, I. (2016), *Deep Learning*, MIT Press.

Hellier, C. (2003), *Handbook of Nondestructive Evaluation*, Mcgraw-Hill, OH.

Kharwar, A.R. and Thakor, D.V. (2022), "An ensemble approach for feature selection and classification in intrusion detection using extra-tree algorithm", *International Journal of Information Security and Privacy (IJISP)*, Vol. 16 No. 1, pp. 1-21, doi: 10.4018/ijisp.2022010113.

Konstantinov, A.V. and Utkin, L.V. (2021), "Interpretable machine learning with an ensemble of gradient boosting machines", *Knowledge-Based Systems*, Vol. 222, 106993, doi: 10.1016/j.knosys.2021.106993.

Kumpati, R., Skarka, W. and Ontipuli, S.K. (2021), "Current trends in integration of nondestructive testing methods for engineered materials testing", *Sensors*, Vol. 21 No. 18, p. 6175, doi: 10.3390/s21186175.

LeCun, Y., Bengio, Y. and Hinton, G. (2015), "Deep learning", *Nature*, Vol. 521 No. 7553, pp. 436-444, doi: 10.1038/nature14539.

Li, L. (2018), "Hyperband: a novel bandit-based approach to hyperparameter optimization", *Journal of Machine Learning Research*, Vol. 18 No. 185, pp. 1-52.

Liu, X., Wu, Q., Su, S. and Wang, Y. (2022a), "Evaluation and prediction of material fatigue characteristics under impact loads: review and prospects", *International Journal of Structural Integrity*, Vol. 13 No. 2, pp. 251-277, doi: 10.1108/ijsi-10-2021-0112.

Liu, X., Liu, J., Wang, H. and Yang, X. (2022b), "Prediction and evaluation of fatigue life considering material parameters distribution characteristic", *International Journal of Structural Integrity*, Vol. 13 No. 2, pp. 309-326, doi: 10.1108/ijsi-11-2021-0118.

Massimo, D., Ganthaler, E., Buriro, A., Barile, F., Moraschini, M., Dignös, A., Villgrattner, T., Peer, A. and Ricci, F. (2023), "Estimation of mass and lengths of sintered workpieces using machine learning models", *IEEE Transactions on Instrumentation and Measurement*, Vol. 72, pp. 1-14, doi: 10.1109/tim.2023.3298413.

MoradiMaryamnegari, H., Hasseni, S.E.I., Ganthaler, E., Villgrattner, T. and Peer, A. (2023), "Neural-network-based automatic trajectory adaptation for quality characteristics control in powder compaction", *Journal of Intelligent Manufacturing*, Vol. 36 No. 2, pp. 1-21, doi: 10.1007/s10845-023-02274-2.

Mustafa, S., Peer, A. and Concli, F. (2023), "Non-destructive crack detection methodologies in green compacts: an overview", *International Conference on Flexible Automation and Intelligent Manufacturing*, Springer, pp. 836-847.

Mustafa, S., Ganthaler, E., Villgrattner, T., Concli, F. and Peer, A. (2024), "Determination of crack limits in sinter components by compaction process performance analysis", *Key Engineering Materials*, Vol. 996, pp. 87-96, doi: 10.4028/p-so4rks.

Opitz, D. and Maclin, R. (1999), "Popular ensemble methods: an empirical study", *Journal of Artificial Intelligence Research*, Vol. 11, pp. 169-198, doi: 10.1613/jair.614.

Ronneberger, O., Fischer, P. and Brox, T. (2015), "U-net: convolutional networks for biomedical image segmentation", *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference*, Munich, Germany, October 5-9, 2015, Springer, pp. 234-241, proceedings, part III 18.

Schapire, R.E. (2013), "Explaining adaboost", in *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, Springer, pp. 37-52.

Shahbudin, S. (2010), "Decision boundaries and classification performance of SVM and KNN classifiers for 2-dimensional datasets", *International Journal of Systems Applications, Engineering and Development*, NAUN.

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025

Sharma, K. (2023), "Analysis of non-destructive testing for improved inspection and maintenance Strategies", *The e-Journal of Nondestructive Testing*, Vol. 28 No. 7, doi: 10.58286/28287.

Snoek, J., Larochelle, H. and Adams, R.P. (2012), "Practical bayesian optimization of machine learning algorithms", *Advances in Neural Information Processing Systems*, Vol. 25.

Tweed, J. (2008), "Cracking in green compacts", *Euro PM2008-Tools For Improving PM*, Vol. 3, pp. 103-108.

Yadav, S. and Shukla, S. (2016), "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification", *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, IEEE, pp. 78-83.

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P. and Gao, R.X. (2019), "Deep learning and its applications to machine health monitoring", *Mechanical Systems and Signal Processing*, Vol. 115, pp. 213-237, doi: 10.1016/j.ymssp.2018.05.050.

Corresponding author
Sameen Mustafa can be contacted at: smustafa@unibz.it

For instructions on how to order reprints of this article, please visit our website:
www.emeraldgrouppublishing.com/licensing/reprints.htm
Or contact us for further details: **permissions@emeraldinsight.com**

Downloaded from http://www.emerald.com/ijsi/article-pdf/doi/10.1108/IJSI-05-2025-0131/10320585/ijsi-05-2025-0131en.pdf by University of Essex user on 20 October 2025