Scientific Reports

https://doi.org/10.1038/s41598-025-29977-1

Article in Press

AI-powered predictive framework for crack detection in steel–copper laser welding

Received: 26 August 2025

Accepted: 20 November 2025

Published online: 25 November 2025

Cite this article as: Nagendra J., Prashanth K.S., Veerabhadrappa K. *et al.* AI-powered predictive framework for crack detection in steel–copper laser welding. *Sci Rep* (2025). https://doi.org/10.1038/s41598-025-29977-1

J. Nagendra, K. S. Prashanth, Kavadiki Veerabhadrappa, K. D. Bopanna, M. Vinutha, E. Naresh & Din Bandhu

AI-Powered Predictive Framework for Crack Detection in Steel–Copper Laser Welding

J Nagendra[1], K S Prashanth[2], Kavadiki Veerabhadrappa[3], K D Bopanna[4], M Vinutha[5], E Naresh[6*], Din Bandhu[7*]

[1]Department of Robotics and Artificial Intelligence, Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India

[2]Department of Physics, New Horizon College of Engineering, Bengaluru, Karnataka, India

[3]Department of Aerospace Engineering, B. M. S. College of Engineering, Bengaluru, Karnataka, India

[4]Senior Product and System Engineer, Engineering R&D services, PLM Technologies, Capgemini, Bengaluru, India

[5]Department of Artificial Intelligence and Machine Learning, Acharya Institute of Technology, Bengaluru, Karnataka, India

Department of Computer Science and Engineering, B. M. S. College of Engineering, Bengaluru, Karnataka, India

[7]Department of Mechanical Engineering, Galgotias University, Greater Noida, Uttar Pradesh, India

*Corresponding Authors: dinosingh@hotmail.co.uk; naresh.cse@bmsce.ac.in

Abstract: Laser welding of dissimilar materials, such as steel and copper, is highly susceptible to crack formation, which compromises joint integrity and service life. Traditional inspection techniques are slow, labor-intensive, and error-prone, underscoring the need for intelligent defect-prediction systems. This study utilizes a publicly available dataset of 360 weld cross-sections, generated using a definitive screening design (DSD) that encompasses six key process parameters: laser power, welding speed, angular orientation, focal position, gas flow rate, and sheet thickness. Multiple machine learning classifiers, including Decision Trees, Random Forests, Gradient Boosting, Support Vector Machines, and Neural Networks, were systematically evaluated using Orange data mining software with imbalance handling strategies. The novelty of this study lies in the application of the Orange data

mining tool to address data imbalance in welding defect prediction and its optimization through a neural network framework, thereby enhancing both model reliability and predictive performance. Among them, a Multilayer Perceptron (MLP) neural network achieved the best performance, attaining 94.9% accuracy, 86.3% sensitivity, and 96.8% specificity, with an AUC of 0.961. The results establish neural networks as a robust and scalable tool for defect classification in steel–copper welding, offering a practical pathway for intelligent process monitoring and predictive quality assurance in Industry 4.0 manufacturing.

Keywords: Laser welding; steel–copper joints; crack detection; machine learning; neural networks; Industry 4.0; classification

1 Introduction:

The precision and adaptability of laser welding have profoundly reshaped modern manufacturing, particularly in applications that require the joining of dissimilar materials. Among these, steel-copper joints pose significant challenges because of the distinct thermal properties, melting points, and thermal conductivities of the two metals. These differences often lead to the formation of weld defects, such as cracking, which can critically compromise the mechanical performance and service life of welded components. Conventional defect detection methods, which largely depend on visual inspections and destructive testing, are laborious and time-consuming and are inherently susceptible to human error and inconsistencies. As manufacturing processes advance toward the principles of Industry 4.0, the integration of data-driven solutions, particularly machine learning (ML) and artificial intelligence (AI), has become indispensable for intelligent defect prediction and adaptive process control [1] [2].

Machine learning-based classification models offer powerful tools for automating defect detection and improving predictive accuracy in complex welding processes. These models can unravel the nonlinear interactions between welding parameters that traditional statistical methods may overlook. The application of supervised machine learning algorithms, such as Decision Trees, Random Forests, Gradient Boosting, Support Vector Machines, and Neural Networks, gained considerable traction in welding research for classifying weld defects, optimizing parameters, and enhancing process reliability [3] [4]. Specifically, in the field of laser welding, ML models have enabled real-time monitoring, defect classification, and predictive quality assurance, facilitating the shift toward smart, self-correcting welding systems [5].

The adoption of ML and AI in welding has evolved from simple parameter estimation to sophisticated predictive systems that integrate in-process sensing data for real-time control. Researchers have demonstrated that ML models can effectively predict weld defects by learning from structured process data, improving the overall accuracy and efficiency of

welding operations [6] [7]. Additionally, these approaches significantly reduce the need for exhaustive post-weld inspections, offering a pathway toward more sustainable and cost-effective manufacturing solutions.

In this study, a definitive screening design was employed to systematically investigate the influence of six critical process parameters: laser beam power (W), welding speed (m/min), angular position in welding direction (°), focal position (mm), gas flow rate (l/min), and steel sheet thickness (mm) on the weld depth and the geometrical characteristics of the weld metal in steel-copper lap joints. The dataset used in this study is publicly available and was published by Jonas Rinne in the Mendeley Data Repository (Version 2, DOI: 10.17632/2s5m3crbkd.2) [8]. It was generated using 18 distinct parameter combinations across three levels, with each combination repeated five times. Each welded sheet section was then divided into four cross-sections, resulting in a total of 360 cross-sectional samples. Each data entry corresponds to a specific weld cross-section, detailing the weld geometry and categorizing the presence or absence of weld metal cracking as a binary classification target.

It is essential to note that the authors of this study did not conduct the experimental work or generate the dataset. Instead, the present research utilizes the publicly available dataset published by Jonas Rinne [8], which comprehensively documents the laser welding experiments of steel-copper joints. This dataset serves as the foundational input for developing the machine learning model and for the subsequent defect prediction analyses presented in this study. The laser welding experiments were conducted using a Trumpf TruDisc 16002 laser system with a 400 µm spot size. The lap joint configuration comprised a copper DHP bottom sheet (2 mm thickness, 45×45 mm) and a stainless steel 1.4301 top sheet with thicknesses ranging from 0.5 mm to 0.7 mm. Argon 5.0 shielding gas was employed to prevent oxidation and ensure process stability. This meticulously structured experimental design provided a reliable and diverse dataset for developing and validating machine learning classification models that can predict weld defects based on process parameters. Through the application of various supervised classification algorithms, this research aims to establish an intelligent, data-driven framework for defect prediction in laser-welded steel-copper joints. The systematic analysis and model comparison are expected to contribute to the advancement of predictive quality assurance in welding, supporting the broader objectives of smart manufacturing and Industry 4.0 integration.

Future work may consider advanced ensemble methods and user-oriented platforms. The subsequent sections of this work are organized as follows: Section 1.1 offers a comprehensive review of the current state of the art, Section 1.2 about the background, Section 2 details about the methodology, Section 3 articulates the results, Section 4 analyzes the outcomes in the form of Discussion, Section 5 elaborated the conclusion of the key findings, Section 6 culminates the scope for further investigation.

1.1 State of the art

Table 1: Neural Network Case studies in Friction Stir Weld quality Monitoring

AI Taxonomy	Depth Layer Sizes, Training time, Testing times	Dataset, Sensors and parameters	Framework, Core Language, interface	Algorithm Used	Advantages	Disadvantages	Paper
Supervised Learning	6 inputs and 4 outputs	Commonly used parameters are TRS (rpm), WS (mm/s)	NA	ANN and RSM	ANN with Levenberg–Marquardt shows better performance than ANN with gradient descent with momentum. The results of various authors' ANN, RSM, and SVM, as well as ANFIS, techniques for FSW and FSSW processes are discussed.	Much more scope available to apply ANFIS and SVM to FSW/FSSW processes, even though SVM gives 100% approximately in all present training and testing cases	[10]
Supervised	Back Propagation NN with 1 hidden layer	Sensors used in this study included 136 Hall sensors, which were equipped on an MFL pig, along with a ring coil-type sensor. Based on the data from the sensor, the ANN functions like a specialist in classifying the data as either	MATLAB R2013b	The signal features in FSW are obtained through ' wavelet analysis'. These features are fed to two ANNs – one with a 'multi-layer feed-forward network trained with backpropagation (BPNN) and the other with radial basis function (RBFNN) to predict the yield and	A novel method for finding the mother wavelet is presented here, which aligns with existing methods. PCA has also been used to find the most useful features after wavelet analysis. The BPNN model performs better in predicting both	Both the ANNs predict the yield strength better than the ultimate tensile strength, which may be further investigated to achieve a better prediction for tensile strength.	[11]

		Defect or Non-Defect.		tensile strength of the weld.	outputs than the RBFNN.		
Supervised Learning	NN with 1 hidden layer with 4–8 neurons	Every time an FSW was done, the welding data was recorded. Four causative variables were used to predict the void formation: temperature, strain, torque and maximum shear stress	NA	'Bayesian Neural Network' and 'Decision Trees' are used here to predict the conditions for the formation of voids in the FSW process	Using a well-tested computational model to calculate the four causative variables, a 93.3% accuracy with the NN and 90% with the decision tree was found over the initial 83.3%	The calculation of the causative variables required a lot of computational work, and hence, other ways to improve the NN and DT models could be looked at	[12]

Table 2: Neural Network Cases in GTAW

AI Taxonomy	Depth Layer Sizes, Training time, Testing times	Dataset, Sensors and parameters	Framework, Core Language, interface	Algorithm Used	Advantages	Disadvantages	Paper
Supervised Learning	CNN architecture: 3 convolutional layers, 3 pooling layers, and 3 fully connected layers	Parameters such as weld pool dimension, weld pool oscillation frequency, and temperature distribution are used to predict backside bead width.	OpenCV PyTorch, Python	CNNs that predict the back-side bead width from top-side images are demonstrated in a Gas Tungsten Arc Welding Process	The CNN gives a testing loss of 0.264 and an MSE of 0.096 mm^2	If the RNN model can be integrated with the CNN, then the sequential reasoning of the RNN will help improve the performance of the model	[13]
Supervised Learning	CNN has 13 layers - 5 convolutional layers, 5 pooling	The welding parameters considered are current, time, thickness, number of passes,	Pytorch library Python	The dynamic nature of the weld pool is observed in the GTAW process, and a CNN is implemented to	Conventional methods analyze images and extract features that aid in classification, but such manual	The back-side width of the weld bead is not predicted here, which could be done to quantify	[14]

	layers, and 3 fully connected layers	arc length, and gas flow rate.		classify the weld states.	engineering is time-consuming and tedious. In contrast, the CNN model addresses this issue and achieves an accuracy of 97.55%.	the weld penetration state.	
Supervised	FCN with 2 fully connected layers. CNN with 3 convolutional layers, 3 pooling layers, and a fully connected layer	Images were captured during trial runs and have 30,008 images. A SOTA HDR sensor, the Xiris XVC-1000, with a dynamic range of 140 dB, was used to capture image data. The parameters considered were Gas flow rate(l/min), Travelling speed(cm/min), Voltage(V) and current(A)	Pytorch Python	A fully connected Neural Network (FCN) and CNN were used to classify the welds in Tungsten Inert Gas Welding.	A novel system for monitoring the TIG process was introduced, utilizing an HDR camera to capture images of the weld pool. This differs from previous methods in that it adapts to the ambiguous essence of a good weld. Hence, giving 93.4% accuracy for the CNN and 89.5% for the FCN	A significant amount of data is required to train the FCN and CNN models, and such large datasets might be challenging to obtain in this specific case, as it utilizes a specialized HDR camera.	[15]

Table 3: Neural Network Case studies in weld defect identification and classification

AI Taxonomy	Depth Layer Sizes, Training time, Testing times	Dataset, Sensors and parameters	Framework, Core Language, interface	Algorithm Used	Advantages	Disadvantages	Paper
NA	NA	NA	NA	Manufacturing processes of groups DIN 8580 and handling operations in	ML has a significant scope in production, and the structured overview of ML use	Missing industry-specific guidelines and	[16]

				accordance with VDI 2860. ANN in Milling Processes, SVM to characterize the cutting edge	cases provided in this paper will help guide production groups in identifying their use cases within the manufacturing industry.	an unstructured representation of uses may discourage people from applying ML to their problems.	
Supervised	ANN with two layers – 1 hidden layer and 1 output layer. 15 neurons in the hidden layer	NA	NA	The classification of leakage of magnetic flux from weld joints as defective or non-defective was done using an ANN in API 5L-X65. The signal patterns were then used to classify the defects into 'External Corrosion (EC), Internal Corrosion (IC) and Lack of Penetration (LP)'	The ANN classifies D and ND with a high efficiency of 94.2% and the defect types with an efficiency of 71.7% on the validation set. The pre-processing used – 'Fourier analysis', 'Wavelet analysis', 'Moving-average filter' and 'Savitzky-Golay filter' helps reduce the ANN training time.	The pre-processing used does not result in a significant improvement in results, and hence, other pre-processing methods could	[17]
Supervised	Region Proposal Network with two sibling fully connected layers. ResNet-101 is used as the backbone.	GDXray dataset. Based on the X-ray images, ResNet-101 is utilized as a feature extractor to achieve improved bounding-box accuracy.	NA	X-ray images are used to identify casting flaws using the 'Mask-Region-based CNN' network.	This system simultaneously detects defects and segments them, enabling the identification of a wide range of defects and resulting in higher defect detection accuracy. Transfer learning is also used to reduce the input demands and increase the prediction accuracy.	The training process for the models is computationally expensive and complex. These models could further be represented in a standardized way, making it easier to distribute the training models.	[18]

Supervised	Single Hidden layer with 64 neurons, 10 layers of Gaussian kernel	A dataset of 120 images is used for visual inspection by a CNN, which analyzes the image data. A Gaussian kernel is being used to extract features of images	NA	Welding defects were classified into four types using a CNN, which consists of two steps: image convolution to extract the image and classification of the extracted image using the CNN. A Gaussian Kernel was used to extract images without loss of information.	The training data achieves nearly 100% accuracy, and the testing (validation) data yields 95.83% accuracy, indicating the model's good performance.	The test data consists of only 24 images, which is relatively few for testing. Since the training data achieved 100% accuracy, having more data for testing might reveal a case of overfitting. Hence, testing the model with some more data would be advisable.	[19]
Supervised	Single hidden-layer FNN	Data collected by all sensors were synchronized, and 25%(3482) samples were used to train, and 75%(10,443) were used for testing Photodiode sensor of industrial scale and spectrometer were used in	NA	A fully connected neural network prediction model and SVM classification accurately estimate the welding status and classify the weld defect, respectively.	The study demonstrates that a less expensive sensor with a simple data structure can be utilized after the model has been established using multivariate statistics.	The prediction of keyhole positions is inaccurate due to the presence of weld defects.	[20]

		this experiment Discrete wavelet transform was used to extract features Spatter features, plume features, keyhole size, coordinate parameters, keyhole length and molten pool depth were used to predict the weld status.					
Supervised	CNN with 11 layers was used	The dataset comprises weld images for both training and testing purposes. A three-way weld image was used, including top front, top back, and back seam. The non-zero pixel method was used to visualize these features. Welding current and feeding speed were used as parameters to cover a wider range of welding conditions.	Python	CNN is employed in 'Robotic Arc Welding' using weld images to detect defects in aluminum alloy	The model gives outstanding performance with 99.38% accuracy and a standard deviation of 0.018	The proposed model can only detect surface defects, not internal defects such as porosity or cracks. The robustness of the model can also be improved by adding more macro weld data.	[21]
Unsupervised	Autoencoder with 2	GD X-ray weld database X-ray	NA	Weld defects are classified using an	Autoencoder with three hidden layers	Even though SURF is faster	[22]

	and 3 hidden layers	SURF is used to extract features from input weld images.		Autoencoder Classifier, and features are extracted using Speeded-Up Robust Features (SURF).	performs better with 98% accuracy	than SIFT, it is not as robust as SIFT and using SIFT may yield us better results	
Supervised	A single hidden-layer ANN is used	Training data: test data::3:1 Fiber laser YSL-6000 was used here, along with a KUKA six-axis robot KR-60HA. An optical fiber probe equipped with a COL-UV/VIS colimator was used to capture spectral data. Laser power, minimum welding velocity, beam focus shift and argon gas flow rate were used as parameters.	NA	The welding defects were detected and classified using ANN and SVM, with the welding experiments done on galvanized steel.	The method is very easy to implement because of its low computational cost, flexibility, and non-intrusiveness	Increasing the dimensions of the data may lead to a local minimum in the ANN, whereas having more data could result in a more accurate model with SVM.	[23]
Supervised	CNN with 2 convolutional layers, a transposed convolutional layer, and a max-	Radiographic image data has been used here. Image segmentation has been performed to obtain parameters such as total intensity and centroid information	MATLAB	A CNN-U-Net architecture is used to classify weld defects into six types in steel plates.	The Unet makes the process more efficient and achieves a perfect accuracy of 94.3%.	Image segmentation plays a crucial role here, and accurately partitioning the images is a challenging task. If we can further improve the image	[24]

	pooling layer	regarding its orientation.				segmentation, the model's accuracy may also improve.	

Table 4: Neural Network Case studies in weld quality – width, deformation, and withdraw forces

AI Taxonomy	Depth Layer Sizes, Training time, Testing times	Dataset, Sensors and parameters	Framework, Core Language, interface	Algorithm Used	Advantages	Disadvantages	Paper
Supervised	CNN for assessing weld quality using image data -1 set of convolutional and fully connected layers for every defect	A CCD camera captures polychrome images of the front, back, and top of the weld seam after welding. 21 parameters were used, including but not limited to laser power, electrical resistance and feed rate and also image data taken from three angles	Scikit learn Keras Python.	SVM, KNN, Rf, and ANN are used in conjunction with machine parameters to predict weld quality. A CNN is used on image data to assess weld quality, and based on the detected defects.	The combination of all the best sub-modules into a big quality monitoring system shows the best results and seems well-suited for applications other than the welding of hairpins, with a recall of 92%	The formation of burr during cutting and residual insulation during stripping should also be taken into account as parameters that can lead to better performance of the model.	[25]
Supervised	ANN with an input layer, 2 hidden layers, each with 50	NA	MATLAB	First, PCA is used to eliminate redundancy through dimensionality reduction, and then	The ANN used produces good results, especially when PCA is applied beforehand. The	The mean processing time was 25 ms in MATLAB for the ANN,	[26]

	neurons, and an output layer with 5 neurons			an ANN is used to identify faults in welding.	temperature profiles also align with the ANN results.	which could be reduced by implementing the ANN in a different language, such as C. Gas flow reduction and low welding current can be identified as separate defects to enhance the model.	
Supervised	CNN with 3 hidden convolutional layers and 2 fully connected hidden layers	Intra-logistic assembly system measures the magnetic and geometric properties of all incoming magnets. Parameters such as length, width, height, magnitude of magnetism, direction of error angle, position, shift, and rotation are considered	SciKit learn Tensorflow Python	An SVR with an RBF kernel was used to estimate the withdrawal force. SVM and CNN were used for quality classification of models using visual features, with SVM having 88.2% classification accuracy compared to CNN's 86.6% An RBF-SVM was also used to quality classify models using acoustic features	Multiple models are used, and the best ones are identified for ultrasonic crimping and withdrawal force.	Dynamic features have to be incorporated into the models. The usefulness of the visual and acoustic features also has to be evaluated.	[27]
Supervised	Extreme learning machine model with 1	A Rogowski coil and a line voltage sensor were	MATLAB nntraintool2017B	A regression model was developed using the PCA technique, along	The improved behavior of the regression model based on PCA is	The effects of welding parameters on weld quality	[28]

	hidden layer, feedforward NN	used to measure the voltage between the upper and lower electrodes. Welding current and welding time, electrode pressure were taken as parameters for dynamic resistance signals		with manually extracted features. The diameter of the welded joints was also predicted using an ANN with extracted features.	attributed to the removal of subjective factors that result from manual extraction methods.	are not revealed due to the sensitive nature of dynamic resistance to process parameters.	
Supervised	ELM: 'Single hidden layer feedforward network'	NA	NA	A method based on "Complete ensemble empirical mode decomposition with adaptive noise," and an extreme learning machine is proposed to evaluate weld quality	Elm is fast and very suitable for multi-classification with fewer samples	Though Elm's have a fast training time, they have considerable evaluation time. They also cannot capture complex relationships due to having only 1 hidden layer, so we'd have to have a large number of hidden neurons	[29]
Unsupervised	DBN with two layers of Restricted Boltzmann	A current sensor, a voltage sensor were used here.	NA	Local Mean Decomposition(LMD) and Deep Belief Network (DBN) are	This method proves to be better than PCA and even ELM classification. Thus, it	Only 2 hidden layers are used in the DBN, which can be	[30]

	Machine(RBM)	Shielding gas percentage, wire diameter, and arc droplet were used as parameters to determine the weld seam		used to classify weld quality into four types	is more effective to classify welds	increased to hence increasing the performance to greater than 95%	
Supervised	The ANN has an architecture of 5–2-1	90 data points were used for training and 30 for validation Laser sensors and acoustic emission sensors were used in this study The input impedance of the system was used as a monitoring signature	MATLAB	The classification of weld quality in resistance spot welding is proposed here by using an ANN and electrical impedance as input	This study provides us with a solution to non-destructively and economically find the quality of spot welding in the production line	Even though a small dataset gives us acceptable accuracy, indeed, with more data, the ANN's performance can be improved	[31]
Supervised	The NN has the architecture 11–20-1	A CCD camera captures the image of the stable keyhole Keyhole length, width and area are the	NA	The weld width in "Variable Polarity Plasma Arc Welding" is predicted using ELM	The ELM is better than BPNN since it has faster computation time and better prediction accuracy	Though Elms have a fast training time, they have considerable evaluation time. They also cannot capture complex relationships due to having	[32]

		parameters used to predict the backside bead width				only 1 hidden layer, so we'd have to have a large number of hidden neurons	
Supervised	DNN with 3 hidden layers are used in this study Shallow ANN with 1 hidden layer	RSW data from major American automotive manufacturers : Current, force, time, material type, sheet thickness, are the parameters	NA	A deep neural network is proposed to predict the nugget width in resistance spot welding joints and compared to previous models.	The RSW data between highly complex and non-linear systems is modeled well by the DNN.	The prediction performance of Predictive models can be further improved by studying the effect of ensemble learning. The prediction capabilities can also be extended to classification scenarios	[33]
Supervised	NN with 3–10-1 architecture trained with BackPropagation	A temperature sensor, a hall sensor, an acceleration sensor, and a photoelectric diode are used. Current, voltage, wire feeding speed, and welding speed are used as parameters.	NA	NN is used to predict the process quality of ship group products, I.e Evaluate the welding angular deformation	The deformation values of the actual and the predicted ones are compared, and it is observed that the model agrees with actual values	Weld quality is not just deformation, and is affected by various factors that need to be studied.	[34]

Table 5: Neural Network case studies in finding the relation between parameters in different welding processes

AI Taxonomy	Depth, Layer sizes, Training time, Testing time	Dataset, Sensors and Parameters	Framework, Core Language, Interface	Algorithm Used	Advantages	Disadvantages	Paper
Supervised	Multilayer Feed Forward network with 3 input neurons, 1 hidden layer, and 1 output neuron	The Telesonic M4000 was used to conduct the experiments Weld time, weld pressure, and vibration amplitude are parameters being used here	NA	The joint strength in Ultrasonic Spot Welding of Al-Cu dissimilar metals is modeled by using ANN and ANFIS.	The ANFIS proves to be better than the ANN here, with a 99.98% Rˆ2 value compared to the ANN's 99.30%. Since the techniques work for a continuous domain, the results are precise and within the bounds of the input parameters.	The ANFIS system	[35]
Supervised	BPNN and RBFNN with a single hidden layer were used	Data is obtained from an experimental setup with a high-speed visual image-capturing system	MATLAB 2016a	PCA is used to decrease the dimensionality of the features.3 machine learning algorithms: RBFNN, BPNN, and SVR is used to model the relationship between chosen features and weld width	The use of PCA enhances the predictive power of all the models. The comparison of various models also helps in identifying the best one, which is BPNN	The use of multi-sensor fusion technology, along with deep learning, can be looked into to increase the efficiency and reliability in monitoring the welding process	[36]
Supervised	CNN with 4	Two cameras	Keras, Python	CNN was used to	Considering burnout as	The	[37]

	convolutional layers and 4 max-pooling layers, and a fully connected layer	were used for capturing images. Excessive penetration and, shape of the specimen were used as parameters to predict burn through.		classify the burn-through, and a regression model was used to predict the penetration depth	a continuous phenomenon and using a regression model for it resulted in good results for predicting penetration depth	generalization performance can be improved by introducing disturbance data into the training data	
Supervised	A Feedforward network with 1 hidden layer with 36 hidden neurons was used	70% of the data was used for training, and 15% each for testing and validation	NA	Machine learning techniques—ANN are used to find the relation between "weld depth signal and the weld seam surface quality" and predict the surface quality using OCT	The use of LVM over Bayesian Regularization BP is good because BR BP has a strong tendency to overfit, as observed	The inclusion of objective criteria for the classification of weld seam quality can improve the model's performance. The given model can also be optimized further to give better than 81.8% accuracy	[38]

Table 6 Neural Network case studies in real-time monitoring of welding processes

AI Taxonomy	Depth, Layer sizes, Training time, Testing time	Dataset, Sensors and Parameters	Framework, Core Language, Interface	Algorithm Used	Advantages	Disadvantages	Refs
Supervised	A neural network with 3–8-1 Architecture is	NA	MATLAB	The monitoring of the welding process is done by microphone array	The optimization algorithm's accuracy has been improved to 99.6%	Since microphone array technology is	[39]

	used			technology, logistic regression, binomial logistic regression, and neural networks to solve 'the imbalance problem between positive and negative samples'		being used, the arc sound signal can be researched to identify different weld defects	
Supervised	Modification of ResNet with 3 convolutional layers	The large dataset is built using a Tachyon 1024 FPA camera The dataset was built with laser power and process speeds, which are the parameters Enthalpy and mass inputs determine clad bead geometry	Keras, Tensorflow Python	A convolutional neural network, ConvLBM, is used here to supervise the manufacturing based on the laser in real-time by extracting features from infrared coaxial images	The ConvLBM is very good since it can be used for three different materials, highlighting its flexibility and adaptability to new materials	The CNN must be validated with new indicators for cladding quality as well, and more tests should be performed for different materials.	[40]
Supervised	ANN with 8 hidden layers was used for UTS, and 10 hidden layers for modified weld parameters prediction	Data from multiple sensors was acquired. Force, torque and power of the FSW process are sensed, Acoustic	MATLAB	A quality monitoring approach is proposed here through cloud and ANNs to predict the UTS and modified weld parameters	The computation is done in the cloud, which makes it much more efficient and cost-effective	Additional sensors for temperature and vibration can further enhance the model and improve process control.	[41]

		emission and current signatures were also used as parameters					

T**able 7:** Neural Network Case studies in other miscellaneous weld activities

AI Taxonomy	Depth, Layer sizes, Training time, Testing time	Dataset, Sensors and Parameters	Framework, Core Language, Interface	Algorithm Used	Advantages	Disadvantages	Refs
Transfer Learning	DN121 with 4 dense blocks	Pre-trained model: Material in Context Dataset. A single-camera setup with a telecentric lens and a custom-made ring LED illuminator is used to obtain images	Pytorch, python	DenseNet is chosen as the reference SOTA over which transfer learning is done	The best performing model is DN1211-G32-M3-B16 with a test set accuracy of 90.28% Transfer learning also helps us train a huge number of parameters with just 306 images, and that too with a good accuracy of 97.22%	Transfer learning may hurt us if the defects require negative learning. We also have to make sure the initial problem is similar to ours when using transfer learning.	[42]
Supervised Learning	CNN with 6 convolutional layers, 6 pooling layers, and 2 fully	The training and testing data sets had 300 and 100 patterns, respectively.	Pytorch 0.4.6 Python 3.6, c#	The momentary events in laser welding that affect laser quality were classified using a CNN.	The high efficiency of structuring statistical data of the CNN has been fully exploited here. GPUs have also been utilized to	The accuracy of classification could be increased by using a larger dataset.	[43]

	connected layers	A high-speed X-ray radiography sensor was used along with a fiber laser source. AE sensing was also performed using a piezo sensor, PICO HF-1.2. The repetition rate(Hz), pulse durations, and laser powers were the parameters chosen.			reduce computation time.		
Supervised	NA	The training data is 80% and the rest is test data. Weld current, voltage, and weld speed are the parameters considered in this study	NA	'Naive Bayes', 'SVM', and 'ANN' were used to classify the sound signals during the shielded metal arc welding process	The ANN performed the best among all the algorithms used, with an 82.76% efficiency	The efficiencies of all the algorithms are generally low, lower than 90%, and can be improved by filtering out the noise during the welding process.	[44]
Supervised	A NN with 5 inputs, 1 output, and 2 hidden layers, each with 7 neurons, trained with	A 6-axis FANUC welding robot was used with a Fronius TPS500I as the power	NN train tool MATLAB	A neural network is trained to avoid voids in junctions while deriving Eulerian paths to weld rib-web geometries.	The suggested approach can be used as an alternative to other tool path preparation strategies for rib-web systems, potentially reducing the	The network is only trained to avoid voids injunctions but not for peaks. This can be further explored	[45]

	Bayesian regularization	source. The ordinal number of layers, the orientation of the welding torch, and a Boolean variable that controls the weld function were used as parameters.			likelihood of welding defects and enhancing welding movement performance.	by incorporating the height into the training.	
Reinforcement Learning	Auto-encoder architecture: 1024–2048-1024–512-256–16	A camera-based system, combined with photo diodes, serves as a sensor. The camera captures the geometrical parameters of the keyhole. The welding seam depth and seam width are also used as parameters	NA	Deep NN is used to extract data, which is then used to find the important aspects of sensor data. This is then used to accurately predict welding power.	The process has demonstrated its ability to learn about welding process activity and accurately predict sensor signals. The DNN also extracts more meaningful features than PCA.	New, specific process needs can be found to extend the process	[46]
Supervised	A single hidden-layer ANN is used	Data has been generated from the experiment. The parameters in this study are voltage, current, gas flow rate,	MATLAB	The best weld of two different metals is predicted using an ANN	The different numbers of hidden neurons are tested, and the ANN with the 4–11-1 architecture and 7 epochs has the least error, and the output has a high correlation between the best weld and welding	The Taguchi method gives us comparable results and does not indicate which parameter has the highest effect on the performance	[47]

		travel speed, inclination and filler wire diameter			parameters		
Supervised learning	IT2-RBF-NN with 6 layers, MLP-NN with 2 hidden layers	Obtained from TWI, Ltd, South Yorkshire, UK, Weld tool rotational speed, welding speed, traverse force, welding speed, and downward force are the parameters of the FSW process here	NA	'Granular computing' and 'computational intelligence' are used to create a data-driven model that predicts in real-time. The approach is evaluated based on 'IT2-RBF-NN, MLP-NN, and T1-RBF-NN'	The IT type 2 neural fuzzy system gives good generalization performance and is tolerant to bad input. It also has a low computation cost and provides feedback in a linguistic format	Further validation of these results with different materials and tools can be tried	[48]

Table 8: Regression Case Studies in Weld Quality Monitoring

AI Taxonomy	Depth, Layer sizes, Training time, Testing time	Dataset, Sensors and Parameters	Framework, Core Language, Interface	Algorithm Used	Advantages	Disadvantages	Refs
Supervised Learning	NA	"AE data was recorded using a PCI-8 data acquisition board manufactured by MistrasGroup Inc." Time-driven data and hit-driven data are recorded. Bead width, weld-affected width,	Sci-kit learn, Python.	The general classification method of Logistic Regression is used. The different weld states are predicted using Adversarial Sequence Tagging.	The results of the logistic regression yield a prediction accuracy of 82.35%, while sequence tagging achieves 91.18%, both of which are better than the metallographic analysis.	A different sequence tagging method, such as structured SVM, can be explored, which has not been done here.	[49]

		and heat input are used as parameters to determine the weld quality.					
Supervised	NA	The data from the gas tungsten arc welding experiment was used. Welding pool length, width and convexity are the chosen Characteristic Performance parameters	NA	'Gaussian Process Regression (GPR) is used to model the real-time welding process. The 'Bayesian Optimization algorithm', along with GPR, is used to provide the prediction result	This paper enables a robot with industrial skills to respond in real-time to observations	The process parameters can be optimized to improve system performance	[50]
Supervised	NA	The experiment was performed on three different tubes from the Tubes Division of Tata Steel Limited.	NA	Logistic regression is used to predict failure on a real-time dataset during the manufacturing of electrical-welded tubes.	The algorithm gives the same variables as important variables as the literature reveals	Weak-welded tubes were only predicted with an accuracy of 82.55%, which should be improved by trying out other classification models.	[51]
Supervised	NA	Parameters used in this study were Current. Wait time and number of segments.	NA	The relationship between weld quality and weld parameters in gas tungsten arc welding is found using the "Gaussian process regression" and "Bayesian optimization algorithm."	Engineering efforts for experimentation are reduced due to the proposed study. Since the model also explores a wider range of parameter values and the entire parameter space, it is more effective and efficient.	The process parameters can be optimized to improve system performance	[52]

Table 9: Decision trees and random forests implemented in weld quality monitoring

AI Taxonomy	Depth Layer Sizes, Training time, Testing times	Dataset, Sensors and parameters	Framework, Core Language, interface	Algorithm Used	Advantages	Disadvantages	Refs
Supervised Learning	Decision tree, J48 and Random Forest	Recording of the arc sound signal, then converted to a wave format. Spectroscopy sensors have been used to capture the radiation emitted by plasma. A high-quality microphone is used to capture the sound of the arc during welding. Current, voltage and travel speed are the parameters considered in this study.	Matlab	J48 and Random forest classification algorithms were used to classify the weld into three types: “good weld, weld with lack of fusion, and burn through”. The raw data from the arc sound was converted into amplitude signals to use as input.	The J48 algorithm achieves an efficiency value of 70.78%, identifying the attributes that distinguish the instance most clearly and classifying it accordingly. The random forest is an ensemble of decision trees, which achieves an efficiency of 88.69%. These classifications help us realize that too high heat or too high current results in burn-through, while too low of either results in incomplete penetration.	The unnecessary sounds, except the arc sound generated while welding, can be filtered out to give us a better algorithm	[53]
Supervised the C4.5 and Naive Bayes algorithms	NA	Three hundred sample images were captured by keeping the samples normal to a digital	MATLAB R2008b	The C4.5 decision tree algorithm is used for defect classification, with its input being the set of histogram	The decision tree is superior to the Naive Bayes approach, as it yields a misclassification	Since we are dealing with images and can obtain more data, an ANN approach could	[54]

		camera (Canon DIGITAL IXUS 75). The classifier algorithm is used to extract the features from the images, which are the parameters in this study		features. Naive Bayes is also used to detect defects classified into good, minor scratches, and deep scratches.	error of 9.6% compared to 17.3% for Bayes.	be used to achieve better classification results.	
Supervised classification	NA	Data was captured using two photogrammetric cameras. The key parameters are identified to be the neighbourhood size and relevant 3d features	NA	3 different decision trees were tested for classification – J48, Reptree, and Hoeffding Tree using the 3d point cloud of the welds	The proposed model allows us to automate the classification of weld beads in a noninvasive manner	The precision and recall of the ML models can be improved by utilizing the RGB values obtained through photogrammetry. The classification can be done by including more classes along with a Deep Neural Net	[55]
Supervised	NA	Data was acquired using a DAQ (NI 9234) device. Real-time vibration data was collected using an accelerometer sensor.	NA	Feature classification of the vibration signals is performed using various machine learning approaches, including decision trees, Hoeffding trees, logistic model trees, and random forests.	The random forest gives the best classification accuracy of 93.51% at both speeds – 1400 rpm and 1800 rpm.	The models can be improved to perform better at the slower speed of 1400 rpm.	[56]

Supervised	NA	Data from experiments no. 1, 4, and 6 were used for training, while numbers 2, 3, 5, and 7 were used for testing. Topside weld pool and backside weld pool image information was obtained using three light path sensors. The topside weld pool, length, width, and area are considered as parameters here.	NA	Random forests are proposed here to monitor the penetration during welding	With no. of trees of 80, the random forest gives fast processing and high penetration accuracy with a 2.11% error rate	Some penetration states are ignored in this study due to simplified experimental conditions, which can be included in further work	[57]
Supervised	NA	The TVC data acquisition system is used to capture data. Wire feed rate, sickout distance, welding speed, gas flow rate and wire diameter are chosen as weld parameters in this study.	NA	Current and Voltage signals are used in Decision trees with split criteria such as “Gini index,” “towing,” and “deviance” to classify weld defects.	The decision tree gave a better classification accuracy of 96.83% on training data with voltage signal as input and “deviance”	This classification process could be taken online by integrating the decision tree with a suitable controller.	[58]

Table 10: K-Nearest Neighbours Implemented in Weld Quality Monitoring

AI Taxonomy	Depth, Layer sizes, Training time, Testing time	Dataset, Sensors and Parameters	Framework, Core Language, Interface	Algorithm Used	Advantages	Disadvantages	Refs

Supervised	NA	Signals were obtained for each weld experiment and made into datasets. Pin speed ratio and Empirical force index were used as the parameters in this study	NA	'KNN', 'Fuzzy KNN', and 'Artificial Bee Colony (ABC)' were used to predict weld quality in FSW	A 1-NN was found to be the best model with three features – rotation speed, EFI, and feed rate, with a 93.16% accuracy. The feature pool was expanded by adding 'wavelet signals of X-force, Y-Force, rotational speed, feed rate, and plunge force, and on applying ABC, several cases had 100% accuracy	To better classify the classification accuracy without wavelet signals, the points at the boundary between classes were removed, which is not advisable, but it should include additional information to classify the misclassified points correctly	[59]
Supervised Learning	KNN and PCA were used	The PB–Sn camera was used to observe the occurrence of different weld defects. The camera was used to determine the dimensions of the keyhole and weld pool area.	NA	PCA was used to reduce the dimensions of the data, and then a classification was done into four classes using KNN	The model suggested here could be applied in other experimental setups that utilize multiple sensors or to process high-dimensional data from sensors in camera-based monitoring.	The 840 nm camera could be avoided, as it leads to a decrease in fault detection accuracy.	[60]

1.2 Background

A variety of supervised machine learning algorithms were employed in this study to perform classification tasks with high precision and reliability. Each model brings unique learning principles, computational strengths, and practical applicability. This section provides an in-depth explanation of each algorithm, along with relevant real-world case studies that illustrate their effectiveness.

1.2.1 **k-Nearest Neighbors (kNN):**

This algorithm is a straightforward yet potent non-parametric technique that categorizes data points according to the classes of their nearest neighbors in the feature space. The classification decision is determined by computing the distance, usually employing the Euclidean metric, between a query point and its 'k' nearest data points in the training set. The algorithm's efficacy is rooted in its straightforward design and its performance with smaller, low-dimensional datasets [24]. kNN has been widely applied in handwriting recognition tasks, such as classifying handwritten postal codes, where the shape and structure of digits can be accurately compared based on their spatial proximity [58,59]. The case studies that adopted various AI Taxonomies and parameters, followed by the KNN algorithms, are represented in Table 10.

1.2.2 **CN2 Rule Induction:**

This algorithm is a rule-based classifier that constructs a set of human-interpretable “if-then” rules for decision-making. It iteratively identifies significant rules that differentiate between classes with minimal error. This model is particularly useful when interpretability and transparency are required in the classification process. A notable application of CN2 is in medical diagnosis [61], where it is used to develop clear, rule-based systems for classifying diseases based on patient symptoms and laboratory test results, enabling healthcare professionals to easily understand and apply the decision rules [62].

1.2.3 **Decision Tree (DT):**

The DT algorithm builds a tree-structured model by recursively splitting the dataset into branches based on feature thresholds that maximize class purity at each node. Decision Trees are widely appreciated for their interpretability and ease of visualization [63]. They can handle both categorical and continuous variables and are capable of modeling nonlinear relationships. An exemplary application is in credit risk analysis, where decision trees classify loan applicants based on features such as income, credit history, and employment stability, providing transparent decision paths for loan approvals.

1.2.4 **Random Forest** (RF):

Random Forest is an ensemble learning technique that generates several decision trees during the training phase and amalgamates their outputs to yield a more precise and consistent categorization. Random Forest mitigates overfitting and improves generalization by including randomization in feature selection and data sampling [63]. This algorithm has proven highly effective in defect detection within manufacturing industries, where it classifies complex surface anomalies based on sensor data with high precision and robustness. A detailed outline of the case studies that have adopted DT and RF is presented in Table 9.

1.2.5 AdaBoost (Adaptive Boosting-AB):

AB is another ensemble method that combines multiple weak classifiers, typically decision stumps, into a single strong classifier. AdaBoost operates by assigning greater focus to misclassified instances during each iteration, thereby improving the model's ability to handle difficult cases [64]. This approach has found significant success in real-time face detection systems, such as the Viola-Jones framework [65], where AdaBoost rapidly identifies facial features from large-scale image datasets with minimal false detections.

1.2.6 Gradient Boosting (GB)

Gradient Boosting (GB) is a sequential ensemble method that constructs models incrementally, with each subsequent model aimed at reducing the residual errors of its predecessor. It employs gradient descent to improve the loss function and is exceptionally successful at modeling intricate, nonlinear interactions. Gradient Boosting is widely used in web search ranking systems, where it learns to prioritize and rank websites based on user interactions and click-through statistics, thereby enhancing the relevance of search results [10].

1.2.7 Support Vector Machine (SVM)

The SVM method creates an ideal hyperplane that optimizes the margin between distinct classes in the feature space. Support Vector Machines are highly adept at handling high-dimensional datasets and can effectively address both linear and nonlinear classification challenges by employing kernel functions [54], [60], [63], [65], [67]. SVM has demonstrated strong performance in text categorization, notably in spam email detection, where it classifies high-dimensional, sparse text data with excellent precision and computational efficiency [69], [82], [87].

1.2.8 Logistic Regression (LR)

LR is a widely used statistical method that models the probability of a binary outcome based on a linear combination of input features [66]. The logistic function maps predicted values to probabilities, enabling clear threshold-based classification [48], [49]. Logistic Regression is particularly valued for its simplicity, computational efficiency, and interpretability. It is

extensively used in medical research for predicting the likelihood of diseases, such as cardiovascular conditions, based on patient-specific factors including age, cholesterol levels, and lifestyle variables [61]. Table 8 discusses the case studies that have adopted LR in weld quality monitoring.

1.2.9 Neural Networks (NN)

NN are inspired by the structure and function of biological neurons, comprising interconnected layers of nodes that can model highly complex, nonlinear relationships in data [11]. Neural Networks are flexible and capable of learning intricate patterns through layered feature transformations [12]. A well-known application is in image classification tasks, such as the MNIST dataset, where neural networks have achieved outstanding accuracy in recognizing handwritten digits, contributing significantly to advancements in computer vision [28], [32]. The case studies that have adopted NN are described in Tables 1, 2, 3, 4, 5, 6, and 7 for a real-time weld monitoring system for various weld techniques.

1.2.10 Naïve Bayes (NB)

The Naïve Bayes classifier is a probabilistic model derived from Bayes' Theorem, predicated on the premise of conditional independence of features. Notwithstanding this simplistic assumption, Naïve Bayes frequently exhibits exceptional performance in high-dimensional domains and with quite limited training datasets [44]. It is extensively used in natural language processing tasks, such as document classification and sentiment analysis, where it efficiently categorizes textual data based on word frequencies and class-conditional probabilities [54].

Each of these machine learning models contributes unique advantages to the classification framework employed in this study. The diverse selection ensures a holistic evaluation by covering distance-based, rule-based, probabilistic, tree-based, ensemble, and neural network-based approaches. This comprehensive approach supports the reliable identification of the most suitable algorithm for the classification problem under investigation [44].

This study presents a novel application of Orange software for addressing class imbalance in weld defect classification [67]. Unlike traditional code-centric approaches, Orange offers a visual, interactive environment that enables efficient implementation of resampling techniques, model training, and performance evaluation without the need for extensive programming. Its real-time model comparison and seamless integration of balancing strategies provide a transparent and reproducible workflow. The use of Orange for imbalanced data handling, particularly in the context of industrial weld defect classification, is scarcely reported in existing literature, marking a unique contribution of this research [67]. However, coding is also implemented in certain aspects of the current investigation.

2. Methodology

A structured, multi-phase machine learning (ML) workflow (Figure 1) was employed in this study to systematically analyze the dataset and evaluate the classification performance of various algorithms. The methodology comprised four key stages: data acquisition, exploratory data analysis (EDA), model development, and performance evaluation [68].

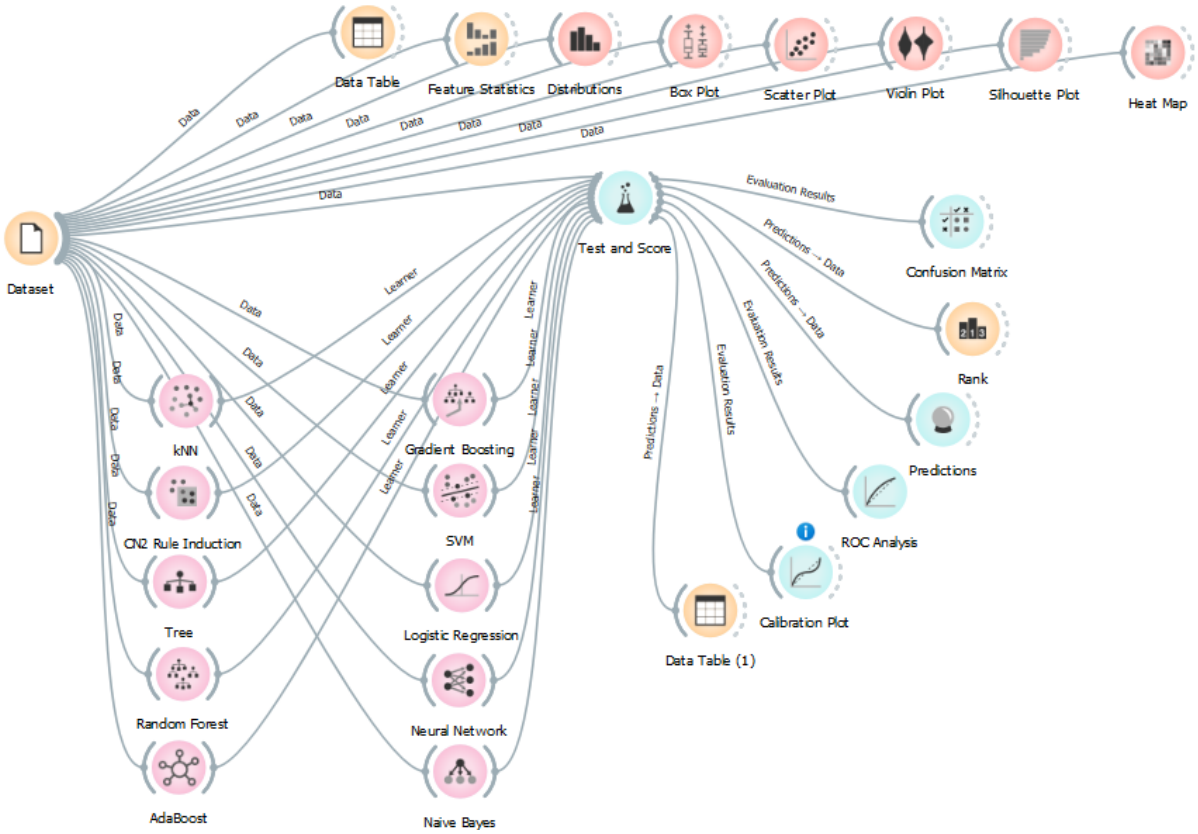


Figure 1: Methodology adopted in Orange Data Mining

2.1. Data Acquisition and Pre-processing

The dataset consists of 360 instances obtained from the "Screening datasets for laser-welded steel-copper lap joints" datasets. In this investigation, a definitive screening design was utilized to systematically explore the effects of six independent variables: laser beam power (W), welding speed (m/min), angular orientation along the welding path (°), focal position (mm), shielding gas flow rate (l/min), and the steel sheet's material thickness (mm) as shown in Table 11. Each factor was tested at three distinct levels, forming 18 unique combinations of process parameters. The experimental trials were performed on laser-welded steel-copper lap joints, with the steel sheet consistently placed on the upper side. To ensure statistical reliability, each parameter combination was repeated five times, and each welded sheet was segmented into four cross-sections, producing a

total of 360 individual cross-sectional samples. Every record in the dataset represents a single cross-section, meticulously measured for its weld metal geometry. Moreover, an additional binary variable was incorporated to indicate whether cracking was present within the weld metal (yes/no).

Table 11: Attributes and experimental details of the x

con	Factors	Levels of factors	Data type
B1	Power (W)	900, 1050, 1200	Numerical
B2	Welding speed (m/min)	0.8, 1.0, 1.2	Numerical
B3	Gas flow rate (l/min)	10, 15, 20	Numerical
B4	Focal position (mm)	2, 0, -2	Numerical
B5	Angular position (°)	15, 0, -15	Numerical
B6	Material strength steel (mm)	0.5, 0.6, 0.7	Numerical
B7	Weld number	1, 2, 3, 4, 5	Numerical
B8	Position on the weld path (mm)	8, 16, 24, 32	Numerical
B9	Cracking in the weld metal (yes/no)	Yes / No	Text
B10	Weld width steel (µm)	1191 to 3637	Numerical
B11	Weld width copper (µm)	53 to 629	Numerical
B12	Weld depth copper (µm)	3 to 705	Numerical
B13	Gap (µm)	10 to 502	Numerical
B14	Count of cracks	0 to 10	Numerical
B15	Average crack length (µm)	146.79 to 647.63	Numerical

The dataset included relevant predictor features and target class labels essential for the classification task. An initial data inspection was conducted to ensure the structural integrity, completeness, and consistency of the data. Any anomalies, such as missing values, outliers, or formatting discrepancies, were addressed to guarantee the quality of the input data and to enable reliable model development. The data set was further processed under 5-fold cross-validation with a train-test split ratio of 70:30

2.2 Exploratory Data Analysis (EDA)

A comprehensive exploratory analysis was conducted to identify the underlying patterns and statistical properties of the dataset. Descriptive statistics were computed to assess the central tendencies, spread, and variability of the features. Graphical visualizations were employed extensively to facilitate deeper insights. These included distribution plots, box plots, scatter plots, violin plots, silhouette plots, and heat maps, which collectively provided a multi-faceted view of the data. The EDA phase was

crucial in identifying potential outliers, class imbalances, and correlations among features, thereby guiding model selection and optimization strategies.

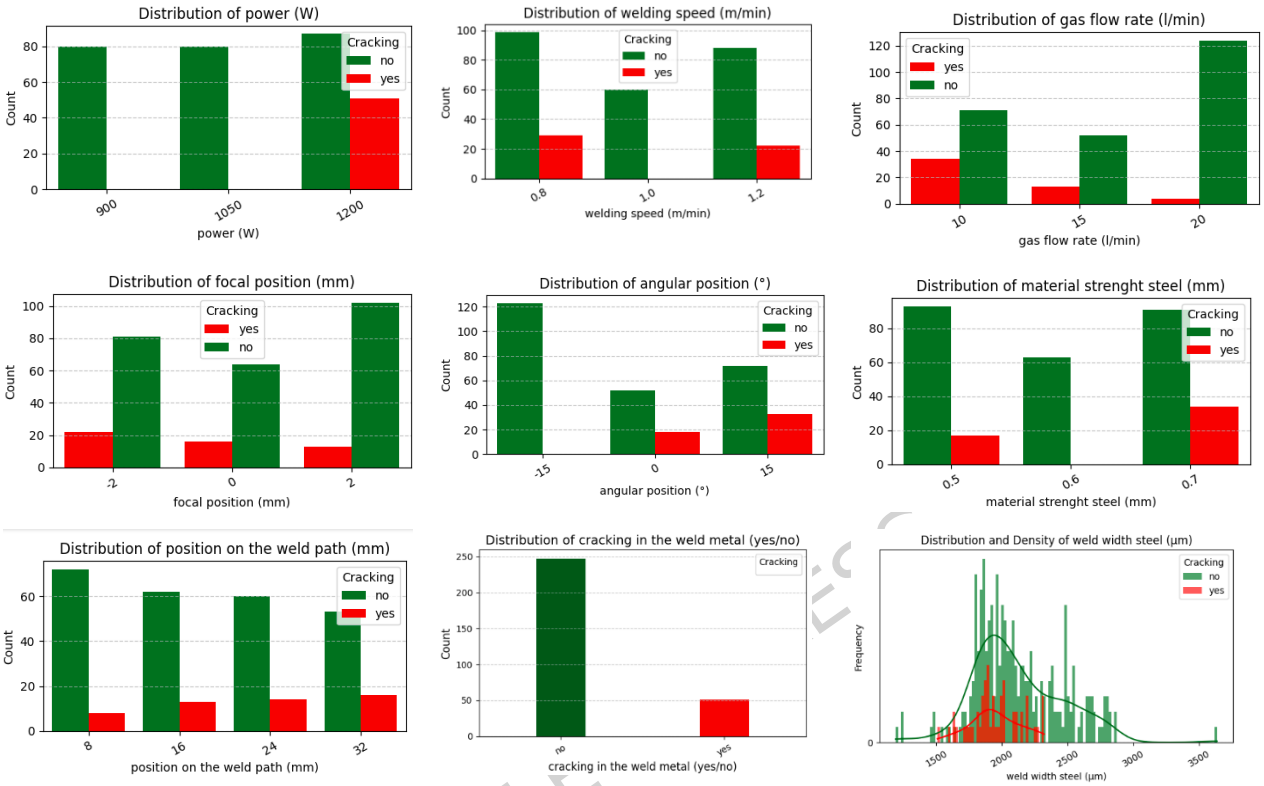


Figure 2: Frequency plot

Specific process and material parameters notably influence the formation of cracks. Cracks tend to occur most frequently when the power input is around 1200 W, and the walking speed is either low (approximately 0.8 m/s) or high (approximately 1.2 m/s), indicating a nonlinear relationship, as shown in Figure 2. Powder flow rates near 10 g/min and 14 g/min show higher crack occurrence, while lower or higher values are associated with fewer defects. Focal positions around 2.5 mm to 2 mm and 1.5 mm are critical zones for crack formation, indicating sensitivity to beam positioning. Similarly, an angular position of approximately 15° is correlated with a high incidence of cracks. Material strength around 0.69 ksi also appears to be a contributing factor, with lower strengths being comparatively safer. Along the weld path, positions near 17 mm, 25 mm, and 33 mm are more prone to cracks. The optimal weld width for avoiding cracks lies outside the 1900–2100 µm range, which otherwise exhibits a high crack frequency. These observations highlight key operating windows where crack mitigation strategies should be focused. Table 12 provides the operating range, and its correlation is represented in Figure 3.

Table 12: Process parameters adopted for data generation

Parameter	Condition Favoring Cracks (Yes)
Power (W)	Around 1200 W
Walking Speed (m/s)	Near 0.8 m/s and 1.2 m/s
Powder Flow Rate (g/min)	Peaks at 10 g/min and 14 g/min
Focal position (mm)	Near 2.5 to 2 mm and around 1.5 mm
Angular Position (°)	Highest crack frequency at 15°
Material Strength Steel (ksi)	Around 0.69 ksi
Position on Weld Path (mm)	Around 17 mm, 25 mm, and 33 mm
Weld Width Steel (µm)	Between 1900 µm and 2100 µm

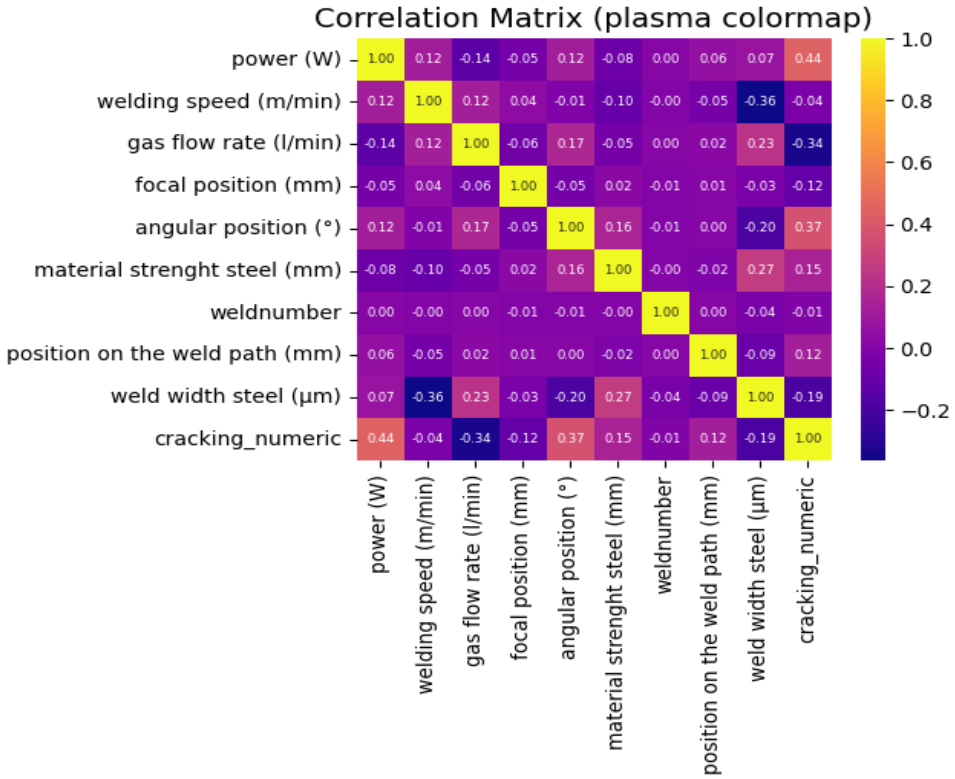


Figure 3: Correlation matrix

2.3 Data balancing

The dataset initially showed significant class imbalance, with far more 'no' (no cracking) samples than 'yes' (cracking) samples. To address this, SMOTE was applied to oversample the minority class, and class weights were adjusted to ensure a balanced learning environment. The orange data mining tool has an inbuilt function to handle the unbalanced dataset [67]. Model performance was evaluated using precision, recall, F1-score, and ROC-AUC to provide a reliable assessment of its accuracy. These techniques enabled the model to effectively detect both classes and enhance its sensitivity to cracking cases.

2.4 Model Development and Training

In this study, a diverse set of supervised machine learning algorithms was developed to perform classification and enable robust comparative analysis. The classifiers included k-Nearest Neighbors (kNN), CN2 Rule Induction, Decision Tree, Random Forest, AdaBoost, Gradient Boosting, Support Vector Machine (SVM), Logistic Regression, Neural Network, and Naïve Bayes. The inclusion of a broad spectrum of algorithms was intended to capture different learning paradigms, from distance-based and rule-based approaches to ensemble methods and neural networks.

Each machine learning model was trained using the same dataset to ensure uniformity and fairness in performance evaluation. To promote the reliability of the results and minimize the risk of overfitting, cross-validation techniques were applied. Cross-validation systematically partitions the data into training and testing subsets, allowing the models to be repeatedly evaluated on unseen data and providing more stable performance estimates.

The diversity of the selected models offered the advantage of exploring both simple and complex classification boundaries. While tree-based ensemble methods, such as Random Forest and Gradient Boosting, are well-suited for capturing nonlinear feature interactions, models like Logistic Regression and Naïve Bayes provide valuable baseline comparisons and interpretability. Additionally, the coefficients of the Logistic Regression model were analyzed to quantify the relative importance and directionality of each feature's influence on the target classification outcome.

This multi-model development framework ensured that the classification problem was approached from varied algorithmic perspectives, thereby enhancing the depth and robustness of the subsequent evaluation phase.

2.4.1 K-Nearest Neighbors (KNN)

The K-Nearest Neighbors (KNN) algorithm was applied with a fixed value of k = 5, using Euclidean distance as the similarity metric. Each of the five nearest neighbors contributes equally to the prediction due to the uniform weighting scheme. For a given input x, the Euclidean distance to a training sample x_i is computed as:

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n} (x_j - x_{ij})^2} \quad (1)$$

The majority vote among the neighbors determines the predicted class:

$$\hat{y} = \text{mode}\{y_i \mid x_i \in N_k(x)\}$$

This configuration helps capture local structure without introducing model bias.

2.4.2 CN2 Rule Induction

The CN2 Rule Induction algorithm was used to generate interpretable classification rules. It was configured to use an ordered rule induction strategy with the Exclusive Covering Algorithm and entropy as the evaluation criterion. The entropy of a rule's coverage is calculated as:

$$H(s) = -\sum_{c \in C} p(c) \log_2 p(c) \quad (2)$$

The algorithm was restricted to generate rules with a minimum coverage of one instance and a maximum rule length of five conditions. These constraints ensured both statistical relevance and human interpretability of the rule set.

2.4.3 Decision Tree

A binary decision tree classifier was built with a maximum depth of 100 and a minimum of two instances per leaf node. Tree growth was halted early if the class purity at a node reached or exceeded 95%. The model utilized classical impurity measures, such as the Gini index or entropy, to determine optimal splits. For example, Gini impurity is defined as:

$$G(s) = 1 - \sum_{i=1}^{C} p_i^2 \quad (3)$$

where p_i is the proportion of samples of class i at node S. This setting allows deep yet controlled tree growth, maintaining balance between complexity and generalization.

2.4.4 Random Forest

The Random Forest ensemble model consisted of 10 decision trees, each trained on bootstrap samples and feature subsets limited to fewer than five variables per split. Each tree makes a prediction independently, and the final output is obtained by majority voting:

$$\hat{y} = \text{mode}\,(T_1(x), T_2(x), \ldots\ldots, T_{10}(x) \quad (4)$$

This configuration promotes model diversity and reduces overfitting by reducing variance.

2.4.5 AdaBoost

The AdaBoost ensemble employed 50 decision stumps (trees of depth one) as weak learners, with a learning rate of 1.0. It used the SAMME.R algorithm for multiclass boosting, which incorporates confidence scores. The weight of each learner is computed by:

$$a_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \quad (5)$$

The final classifier is:

$$H(x) = \sin\left(\sum_{t=1}^{50} \alpha_t h_t(x)\right) \quad (6)$$

Here, a linear regression loss function was used to guide updates, making the classifier sensitive to hard-to-classify samples.

2.4.6 Gradient Boosting

Gradient Boosting was configured with 100 trees, each limited to a maximum depth of 3. The learning rate was 0.1, and subsampling was set to 1.0, meaning the full training data was used for each iteration. The prediction function updates iteratively as:

$$\hat{y}_m(x) = \hat{y}_{m-1}(x) + \nu f_m(x) \quad (7)$$

where $f_m(x)$ is the tree fitted to the negative Gradient of the loss function at step mmm, and ν is the learning rate. Feature splits were constrained to subsets with fewer than two instances, promoting generalizability.

2.4.7 Support Vector Machine (SVM)

The SVM classifier was configured with a sigmoid kernel, a cost parameter C = 1.0, a regression loss tolerance ε = 0.1, and an iteration limit of 100. The optimization problem is defined as:

$$\min_{w,\, b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \xi_i \quad (8)$$

Subject to:

$$y_i(w^T \varnothing(x_i) + b) \geq 1 - \xi_i \quad (9)$$

The sigmoid kernel used for transformation is:

$$K(x,x^i) = \tanh(\propto x^T x^i + c) \quad (10)$$

A numerical tolerance of 0.001 was enforced for convergence precision.

2.4.8 Logistic Regression

The Logistic Regression model utilized L2 regularization (Ridge) and assumed a balanced class distribution. The logistic (sigmoid) function gives the class probability:

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(w^T x + b)}} \quad (11)$$

The model minimizes the following objective:

$$L(w) = -\sum_{i=1}^{n}[y_i \log p_i + (1 - y_i)\log(1 - p_i)] + \lambda ||w||_2^2 \quad (12)$$

This setting enables robust and interpretable linear classification.

2.4.9 Neural Network (MLP)

The Neural Network model was implemented as a multilayer perceptron (MLP) with 100 hidden neurons and the ReLU activation function:

$$h_j = \max(0, w_j^T x + b_j) \quad (13)$$

It was optimized using the Adam solver with a regularization parameter of α = 0.05 and trained for up to 200 iterations. The weights were updated iteratively via Adam's rule:

$$\theta_{t+1} = \theta_t - \eta . \frac{\hat{m}_t}{\sqrt{\hat{\gamma}_t} + \varepsilon} \quad (14)$$

This configuration provided the flexibility of a deep learner with controlled complexity and convergence stability.

2.4.10 Naïve Bayes

The **Naïve Bayes** classifier operated under the assumption of feature independence. The posterior probability is calculated using Bayes' Theorem:

$$P(y \mid x) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y) \quad (15)$$

Assuming features are normally distributed, the Gaussian Naïve Bayes likelihood is given by:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(- \frac{(x_i - \mu_y)^2}{2\sigma_y^2} \right) \quad (16)$$

This method served as a lightweight baseline, particularly effective on high-dimensional data. The consolidated hyperparameter-tuned ML model configuration is represented in Table 13.

Table 13: ML model configuration after hyperparameter tuning

Model	Configuration Details
K-Nearest Neighbors (KNN)	Number of Neighbors = 5, Distance Metric = Euclidean,

Table 13: ML model configuration after hyperparameter tuning

Model	Configuration Details
	Weight = Uniform
CN2 Rule Induction	Rule Ordering = Ordered Algorithm = Exclusive Covering Evaluation = Entropy Rule Filtering: Min Coverage = 1, Max Rule Length = 5
Decision Tree	Tree Type = Binary Min Instances in Leaves = 2 Max Tree Depth = 100 Halt Condition: 95% Majority class
Random Forest	Number of Trees = 10 Subset Split Limit = < 5
Adaboost	Base Estimator = Decision Tree Number of Estimators = 50 Learning Rate = 1.00- Boosting method = SAMME.R Loss Function = Linear Regression
Gradient Boosting	Number of Trees = 100 Learning Rate = 0.10 Tree Depth = 3 Subset Split Limit = < 2 Subsampling Rate = 1.00
Support Vector Machine (SVM)	Cost Parameter (C) = 1.00 Loss Epsilon = 0.10 Kernel = Sigmoid Numerical Tolerance = 0.0010 Max Iterations = 100
Logistic Regression	Regularization = Ridge (L2) Class Distribution = Balanced
Neural Network (MLP)	Hidden Layer Neurons = 100 Activation = ReLU Solver = Adam Regularization α = 0.05 Max Iterations = 200
Naïve Bayes	No specific parameters mentioned (default Gaussian NB assumed)

2.5 Model Evaluation and Validation

The performance of all developed models was rigorously assessed using a comprehensive suite of evaluation metrics. The primary performance

indicators included accuracy, precision, recall, F1-score, and the area under the Receiver Operating Characteristic (ROC) curve (AUC), which collectively offered a balanced assessment of classification effectiveness across multiple dimensions. The confusion matrix was used to examine detailed class-wise performance, highlighting both correct classifications and misclassification patterns [61].

Further validation was conducted using ROC analysis to evaluate the discriminative power of each model across varying classification thresholds. The predicted class labels were analyzed to ensure consistency and accuracy. At the same time, model ranking tables were generated to objectively compare the performance of all classifiers and identify the most effective algorithm for the given classification task.

This study employed a systematic and data-centric methodology, integrating detailed exploratory analysis, the development of diverse machine learning models, and rigorous multi-metric evaluation. The approach ensured a comprehensive, transparent, and reproducible framework for classifier comparison, supporting the selection of the most appropriate model for the classification problem under investigation.

3 Results:

3.1 Exploratory Data Analysis:

The comprehensive feature selection analysis presented evaluates the relative significance of nine process parameters governing the welding operation. Multiple statistical and machine learning-based selection techniques, namely Information Gain, Gain Ratio, Gini Index, ANOVA, Chi-Square (χ^2), ReliefF, and Fast Correlation-Based Filter (FCBF), were employed to ensure a robust, multi-perspective assessment of variable importance. The ranking position of the respective parameters is shown in Figure 4.

Power (W) consistently emerged as the most critical parameter across all selection metrics. It exhibited the highest Information Gain (0.220), Gain Ratio (0.144), Gini Index (0.068), ANOVA score (70.822), Chi-Square statistic (33.408), ReliefF weight (0.488), and FCBF value (0.251). This unequivocally establishes power as the dominant factor influencing welding performance.

		#	Info. gain	Gain ratio	Gini	ANOVA	χ^2	ReliefF	FCBF
1	power (W)		0.220	0.144	0.068	70.822	33.408	0.488	0.251
2	angular position (°)		0.151	0.097	0.042	46.481	32.776	0.378	0.158
3	gas flow rate (l/min)		0.097	0.063	0.034	39.535	25.292	0.422	0.000
4	material strenght steel (mm)		0.077	0.050	0.021	6.361	4.692	0.295	0.000
5	welding speed (m/min)		0.062	0.041	0.015	0.455	0.387	0.231	0.000
6	weld width steel (µm)		0.034	0.017	0.012	11.514	7.491	0.066	0.000
7	position on the weld path (mm)		0.012	0.006	0.005	4.544	3.893	-0.027	0.009
8	focal position (mm)		0.011	0.007	0.004	3.986	2.779	0.382	0.000
9	weldnumber		0.001	0.000	0.000	0.021	0.000	-0.083	0.001

Figure 4: Ranking of the contributing factors

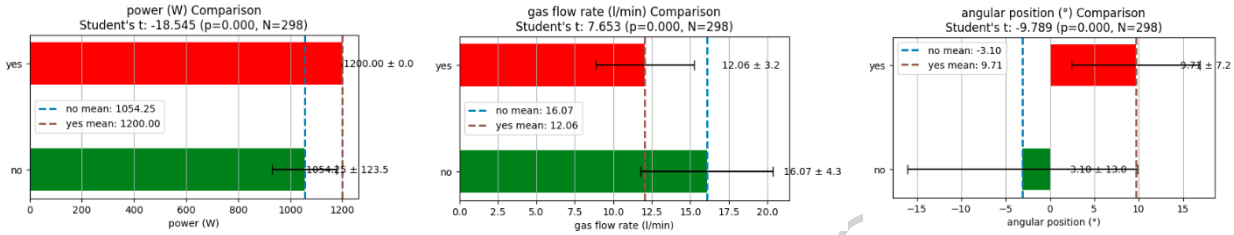
Angular Position (°) was identified as the second most influential feature, particularly evident from its substantial ANOVA (46.481) and Chi-Square (32.776) scores, indicating a strong effect on process variability and defect formation. The gas flow rate (l/min) consistently demonstrated relevance, especially with high ReliefF (0.422) and ANOVA (39.535) values, underscoring its importance in stabilizing the welding process and minimizing defects. Material Strength (mm) and Welding Speed (m/min) showed moderate relevance, with material strength registering moderate ReliefF (0.295) and welding speed contributing marginally across selection methods. Weld Width (µm) and Position on the Weld Path (mm) consistently received lower importance scores, suggesting their limited influence on welding quality within the studied process window. Focal position (mm), despite lower significance in traditional statistical measures, displayed a relatively higher ReliefF weight (0.382), indicating a potential local effect that may warrant further investigation. Weld Number exhibited negligible relevance across all feature selection techniques, reinforcing its minimal impact on the process outcomes within the current experimental configuration.

The convergence of these results across diverse methodologies underscores the criticality of these parameters in determining welding performance.

3.1.1 Bar Plots with Mean ± Standard Deviation

The bar plots represent the mean values, accompanied by either standard deviation or 95% confidence intervals as error bars, illustrating the variability and precision of the measurements. Statistical comparisons between groups were performed using a t-test, with the t-statistic (t) indicating the magnitude of difference and the p-value (p) representing the significance of the observed differences. A smaller p-value (commonly < 0.05)

indicates a statistically significant difference between groups. At the same time, the confidence interval provides a range within which the true population mean is likely to lie, giving additional insight into the reliability of the estimates. The comparison plots presented illustrate the Distribution of key process parameters—Power (W), Gas Flow Rate (l/min), and Angular Position (°)—between two welding outcome classes ("yes" for successful welds and "no" for defective ones). Each plot depicts the mean values with standard deviation error bars for both groups, alongside statistical annotations from Student's t-tests. These visualizations effectively highlight significant differences in parameter settings between classes, providing critical insights into the process factors that influence weld quality.



(a) (b) (c)

Figure 5: Bar plot showing the mean values of power (w) as (a), gas flow rate (/pm) as (b), and angular position as (c)

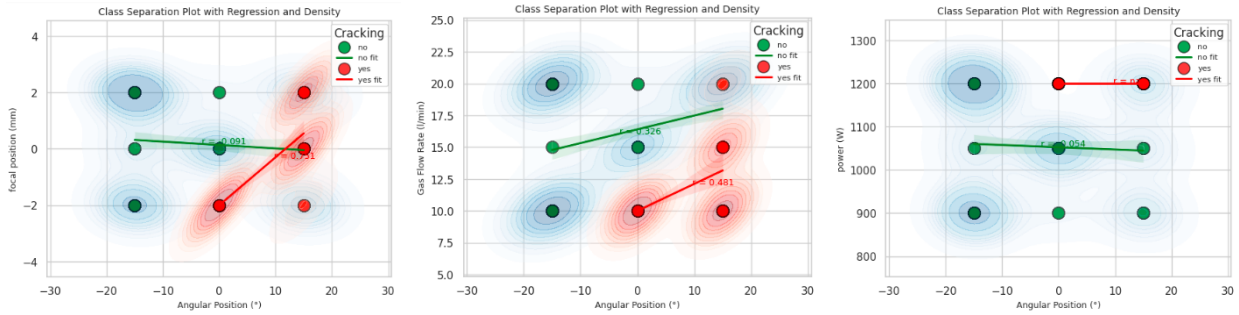
For power (Figure 5a), successful welds operated at a fixed setting of 1200.00 ± 0.0 W, while defective welds averaged 1054.25 ± 123.5 W (t = 18.545, p = 0.000), highlighting the need for stable power levels. In gas flow rate (Figure 5b), the "no" class recorded a higher mean of 16.07 ± 4.3 l/min compared to 12.06 ± 3.2 l/min in the "yes" class (t = 7.653, p = 0.000), indicating that controlled shielding gas delivery is crucial. For angular position (Figure 5c), the "yes" class averaged 9.71° ± 7.2°, while the "no" class showed –3.10° ± 13.0° (t = –9.789, p = 0.000), emphasizing the importance of accurate torch alignment.

Overall, the statistical evidence identifies power, gas flow rate, and angular position as decisive parameters for weld quality, as also seen in Figure 4. Their significant differences validate their inclusion in predictive models and reinforce the necessity of strict process control in advanced welding systems.

3.1.2 Scatter Plot / Interaction Plot:

The interaction plots generated between angular position and key process parameters—focal position, gas flow rate, and power—provide critical insights into how these variables influence the occurrence of cracking in weld metal. The analysis distinctly categorizes the observations into "yes" (cracking present) and "no" (cracking absent) and overlays regression trends

along with Pearson correlation coefficients (r-values) to quantify the strength and direction of these relationships.



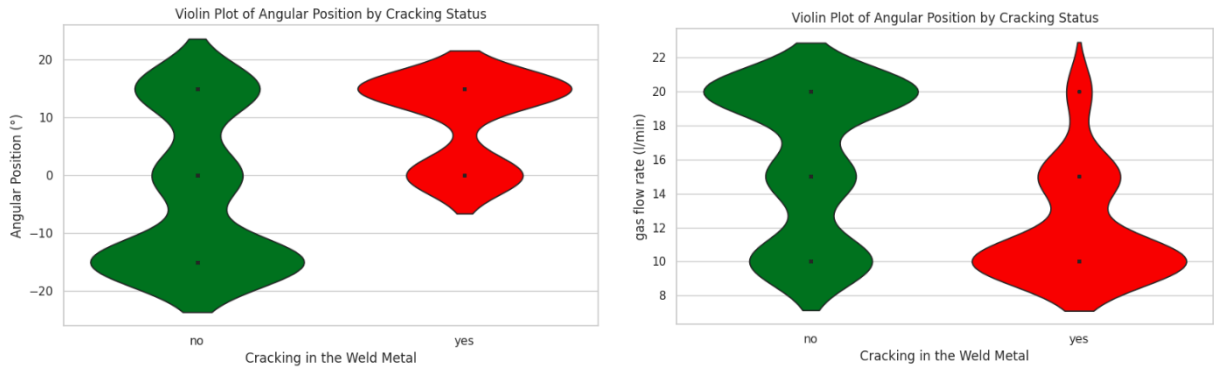
(a) (b) (c)

Figure 6: Interaction plot, angular position vs focal position (a), Vs gas flow rate (b), Vs power (c)

Figure 6a illustrates the interaction between angular position and focal position, revealing distinct trends within each category. The "no" category shows a weak negative correlation (r = -0.091), indicating minimal change in focal position with increasing angular position. In contrast, the "yes" category exhibits a strong positive correlation (r = 0.731), suggesting that higher angular positions significantly increase focal position and may contribute to cracking. Figure 6b shows angular position versus gas flow rate. Both categories display positive correlations: moderate for "no" (r = 0.326) and stronger for "yes" (r = 0.481). The steeper trend in cracked welds implies that elevated gas flow at larger angular positions may promote cracking, highlighting the need for careful regulation. Figure 6c presents angular position versus power, where the "no" category has a very weak negative correlation (r = -0.054) and the "yes" category is flat (r = 0.00), indicating power is largely unaffected by angular changes and is not a primary cracking factor.

Overall, focal position and gas flow rate are strongly category-dependent, whereas power remains relatively stable, emphasizing the need for context-specific parameter optimization to reduce weld defects.

3.1.3 Violin plot



(a) (b)

Figure 7: The violin plot demonstrating the contribution of angular position (a) and gas flow rate in lpm (b)

Figure 7a shows the Distribution of angular position (°) with respect to weld cracking. In the "no crack" group (blue), angular positions are symmetrically distributed around -15°, 0°, and 15°, indicating low crack likelihood at these values. In contrast, the "yes crack" group (red) is narrower and skewed toward positive angles, mainly around 15°, with minimal representation at negative angles. This clustering suggests that higher angular positions increase the risk of cracking, while reduced spread indicates that limited angular conditions contribute to cracking. Figure 7b presents the distributions of gas flow rates (L/min). The "no crack" group shows modes at 10, 15, and 20 L/min, reflecting stable flow conditions. The "yes crack" group also exhibits a tri-modal distribution, with higher density at the extremes (10 and 20 L/min) and a narrow mid-range (~15 L/min), indicating that deviations from intermediate flow rates promote cracking.

Together, these plots reveal that cracks are associated with higher angular positions (~15°) and extreme gas flows. In contrast, balanced angles and mid-range flow (~15 L/min) favor crack-free welds, highlighting the importance of precise process control.

3.2 Evaluation metrics:

The performance of classification models is typically assessed using a suite of evaluation metrics that capture various aspects of model accuracy, discrimination, and error balance. These include the area under the Curve (AUC), Classification Accuracy (CA), F1-score, Precision, Recall, Matthews Correlation Coefficient (MCC), Specificity, and Logarithmic Loss (LogLoss). Each metric provides a unique insight into model behavior, helping to select an optimal model based on the nature of the dataset and the problem context.

The Area Under the Curve (AUC) represents the area under the Receiver Operating characteristic (ROC) curve, which plots the True Positive Rate

(TPR) against the False Positive Rate (FPR) at various threshold settings. A higher AUC indicates better model discrimination capability between positive and negative classes. It is especially useful in imbalanced datasets and probabilistic classifiers, providing a threshold-independent performance measure [1].

Classification Accuracy (CA) is defined as the ratio of correctly classified instances (both positives and negatives) to the total number of samples. It is calculated using the formula:

$$\text{Accuarcy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (17)$$

where TP is true positives, TN is true negatives, FP is false positives, and FN is false negatives. While accuracy gives a quick overview, it can be misleading when the dataset is imbalanced. Although accuracy is easy to interpret and widely used, it can be misleading when the dataset is imbalanced [2]. The F1-score is the harmonic mean of Precision and Recall, providing a single score that balances the trade-off between false positives and false negatives. It is particularly useful in scenarios involving imbalanced classification. The formula is:

$$F1 = 2 \text{ X } \frac{\text{Precision X Recall}}{\text{Precision+Recall}} \quad (18)$$

Precision, also known as Positive Predictive Value, measures the proportion of identifications that were actually correct. It is computed using the formula:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (19)$$

Precision becomes particularly important when the cost of false positives is high, such as in email spam filters or fraud detection. Recall, also known as Sensitivity or True Positive Rate (TPR), measures the proportion of actual positives that the model correctly identifies. It is calculated as:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (20)$$

Recall is crucial in applications where missing a positive case (false negative) is unacceptable, such as in medical diagnoses and other critical situations. The **Matthews Correlation Coefficient (MCC)** is a more balanced metric that accounts for all four categories of the confusion matrix. It is particularly useful for evaluating performance on imbalanced datasets. The formula is:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (21)$$

MCC values range from -1 to +1, where +1 indicates perfect prediction, 0 indicates random prediction, and -1 indicates total disagreement between prediction and observation.

Specificity, also known as the True Negative Rate (TNR), measures the proportion of actual negatives that are correctly identified by the model. It complements recall and is useful when both types of classification errors are significant. It is computed as:

$$Specificity = \frac{TN}{TN+FP} \quad (22)$$

High specificity is desirable when minimizing false positives is crucial, such as in quality control systems. **Logarithmic Loss (LogLoss)** is a metric that measures the performance of a classification model where the output is a probability value between 0 and 1. It penalizes false classifications, especially those made with high confidence. The formula is:

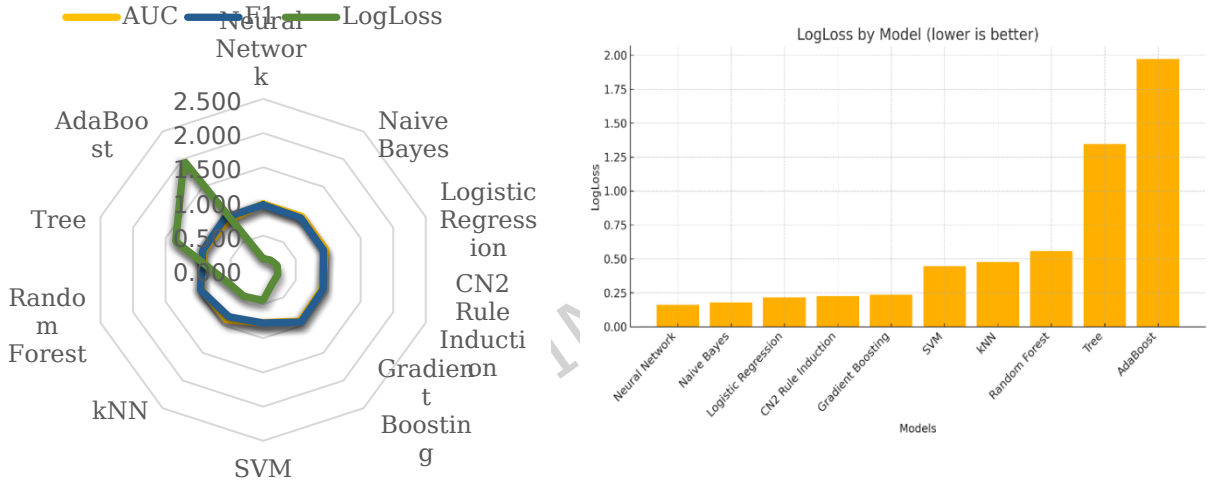
$$LogLoss = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)] \quad (23)$$

where y_i is the true binary label, p_i is the predicted probability of the positive class, and NNN is the number of samples. Lower LogLoss values indicate better-calibrated probability predictions [7]. Together, these metrics form a comprehensive evaluation framework for classification models. The selection of the appropriate metric depends on the domain-specific costs of errors, class distribution, and the objective of the prediction task. The consolidated scores of all the adopted ML models are shown in Table 14.

Table 14: Evaluation results of all the models

Models	AUC	CA	F1	Prec	Recall	MCC	Spec	LogLoss
Neural Network	0.985	0.964	0.964	0.965	0.964	0.856	0.912	0.099
AdaBoost	0.936	0.961	0.962	0.963	0.961	0.846	0.912	0.205
Gradient Boosting	0.981	0.956	0.956	0.957	0.956	0.824	0.894	0.139
Random Forest	0.986	0.953	0.953	0.954	0.953	0.811	0.878	0.100
Tree	0.951	0.953	0.952	0.952	0.953	0.801	0.828	0.756

Logistic Regression	0.979	0.944	0.944	0.944	0.944	0.772	0.827	0.134
Naïve Bayes	0.972	0.942	0.942	0.942	0.942	0.762	0.827	0.152
CN2 Rule Induction	0.950	0.928	0.926	0.925	0.928	0.689	0.726	0.198
kNN	0.915	0.872	0.872	0.872	0.872	0.475	0.602	0.410
SVM	0.693	0.769	0.774	0.779	0.769	0.089	0.323	0.392



(a) (b)

Figure 8: Rador plot of AUC, F1, LogLoss (a) and Bar plot (b) of the LogLoss distribution

The radar plot, as shown in Figure 8, presents a comparative evaluation of various machine learning models, with a particular emphasis on Logarithmic Loss (LogLoss), a critical metric for assessing the quality of probabilistic predictions. Unlike metrics such as F1 score or AUC, which focus on classification performance, LogLoss penalizes incorrect predictions made with high confidence, making it especially valuable for applications requiring well-calibrated probability outputs. From the visualization, models such as CN2 Rule Induction, Decision Tree, and k-Nearest Neighbors (kNN) exhibit relatively high deviations in LogLoss, indicating inferior probability estimation and potential overfitting or poor calibration. These models, therefore, are less suited for tasks where the reliability of predicted

probabilities is crucial. Conversely, the Neural Network model consistently demonstrates the lowest LogLoss value among all models evaluated, suggesting robust generalization and better confidence calibration despite moderate compromises in F1 score or AUC. While ensemble methods like Gradient Boosting and Random Forest also show competitive LogLoss performance, the Neural Network remains the most reliable and recommendable model when prioritizing predictive certainty, highlighting its suitability for deployment in scenarios where misclassification risks must be minimized.

Table 15: Compare models by: Area under ROC curve

	kNN	Naïve Bayes	SVM	Tree	AdaBoost	Random Forest	Gradient	Logistic Regressi	Neural Network	CN2 Rule
kNN	1.00	0.12	0.99	0.46	0.59	0.33	0.46	0.13	0.18	0.48
Naïve Bayes	0.89	1.00	0.97	0.86	0.91	0.64	0.70	0.44	0.57	0.73
SVM	0.01	0.03	1.00	0.11	0.17	0.09	0.19	0.03	0.04	0.18
Tree	0.54	0.14	0.89	1.00	0.96	0.09	0.45	0.10	0.07	0.50
AdaBoost	0.41	0.09	0.83	0.05	1.00	0.00	0.24	0.06	0.03	0.25
Random Forest	0.68	0.37	0.91	0.91	1.00	1.00	0.79	0.34	0.33	0.88
Gradient Boosting	0.55	0.30	0.82	0.55	0.76	0.21	1.00	0.28	0.27	0.24
Logistic Regression	0.87	0.56	0.97	0.90	0.94	0.66	0.72	1.00	0.65	0.76
Neural Network	0.82	0.43	0.96	0.93	0.97	0.67	0.73	0.35	1.00	0.77
CN2 Rule Induction	0.53	0.27	0.82	0.50	0.75	0.13	0.24	0.25	0.23	1.00

The correlation heatmap, as shown in Table 15, illustrates the pairwise similarity in AUC (Area Under the Curve) performance across different machine learning models. High correlation values (closer to 1.00) between models suggest that they rank instances similarly in terms of predicted probabilities. For instance, models such as Logistic Regression, AdaBoost, Random Forest, and SVM exhibit strong mutual correlations (values exceeding 0.90), indicating they behave consistently in terms of discriminative ability. Similarly, Neural Networks show a notable correlation with AdaBoost (0.76), CN2 Rule Induction (0.77), and Naïve Bayes (0.83), suggesting that these models share a common prediction trend in terms of effectively ranking positive and negative instances. In contrast, SVM demonstrates very low correlation with models such as kNN (0.01), Naïve Bayes (0.03), and tree (0.11), highlighting distinct prediction patterns likely arising from its margin-based optimization strategy.

From a practical perspective, models with high intercorrelations may offer redundant information; thus, ensembling such models might yield limited performance gains. Instead, leveraging less correlated yet reasonably performing models (e.g., combining SVM with Logistic Regression or kNN with Random Forest) could enhance diversity and result in more robust ensemble learning. However, the consistency among the top-performing models, Logistic Regression, AdaBoost, and Random Forest, suggests they can be reliably used for stable AUC outcomes. In conclusion, Logistic Regression emerges as a consistently strong performer with high correlation to most other models, affirming its robustness. Neural Networks also present a favorable profile due to their moderate-to-high correlation with several models, reinforcing earlier observations from the LogLoss perspective. Therefore, Neural Networks and Logistic Regression can be recommended as primary modeling choices, with ensemble approaches carefully constructed to maximize diversity by incorporating decorrelated models, such as SVM.

3.3 Computational time

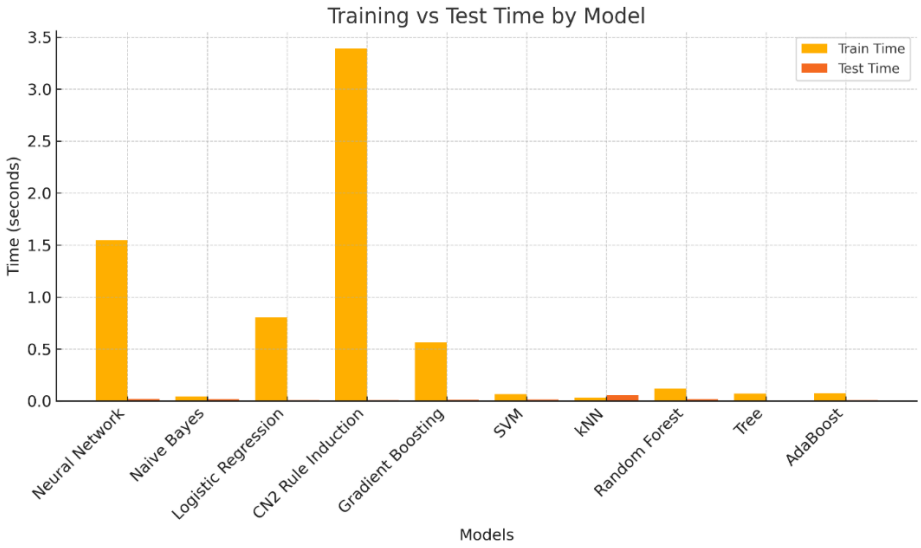


Figure 9: Computation time comparison of the ML models

The bar graph titled *"Training vs Test Time by Model"* presents a comparative analysis of the computational time required by various machine learning algorithms during the training and testing phases. The data highlights significant disparities in time consumption, reflecting the varying levels of algorithmic complexity and computational demands among the evaluated models. Notably, the CN2 Rule Induction algorithm demonstrates the highest training time, exceeding 3.4 seconds, as shown in Figure 9. This elevated time can be attributed to the model's reliance on exhaustive rule searching and heuristic evaluations, which are inherently resource-intensive. Similarly, the Neural Network model exhibits substantial training duration, approximately 1.5 seconds, owing to the iterative gradient-based optimization processes and the handling of multiple layers of parameters.

In contrast, algorithms such as Naïve Bayes, Decision Trees, and AdaBoost maintain consistently low training and testing times. These models benefit from relatively straightforward mathematical formulations and efficient fitting procedures, making them highly suitable for real-time systems or applications constrained by computational resources. Logistic Regression and Gradient Boosting models exhibit moderate training times, but their test times remain impressively low. This suggests that, although the model fitting process may be computationally demanding due to parameter tuning and ensemble strategies, the resulting models are efficient at inference. The Support Vector Machine (SVM), k-Nearest Neighbors (kNN), and Random Forest algorithms also demonstrate acceptable training times with negligible

test times, affirming their potential suitability in balanced workloads where both model building and inference are of concern.

A common observation across all models is that test time remains minimal, typically well below 0.05 seconds. This underscores that once the models are trained, their deployment for inference is generally computationally inexpensive. Such an outcome is particularly relevant in applications like embedded systems, mobile environments, and real-time analytics, where inference latency must be tightly controlled.

Despite its comparatively high training time, the Neural Network demonstrates one of the lowest test times among all models. This makes it highly advantageous for deployment scenarios where fast inference is critical, and the training phase can be performed offline or on dedicated infrastructure. Moreover, Neural Networks offer superior representational capacity and are particularly effective for modeling complex, high-dimensional, and nonlinear relationships in data domains where simpler models may underperform. In contemporary machine learning practice, the initial training cost is often justifiable when paired with robust generalization, adaptability, and scalability. Additionally, the application of optimization techniques such as early stopping, dropout, mini-batch learning, and architecture tuning can help mitigate training overhead. Therefore, in environments where predictive accuracy is prioritized and computational resources for training are available, the Neural Network stands out as a viable and recommended model, balancing complexity with high performance and inference efficiency. Given its capacity to learn complex data patterns and its negligible inference latency, the Neural Network is recommended as the most suitable model for applications demanding both high accuracy and operational scalability, particularly when supported by adequate computational infrastructure.

3.4 Confusion Matrix:

The comparative analysis of confusion matrices across the nine machine learning models, as presented in Figure 10, highlights notable differences in performance. Among these, the Neural Network stands out as the most effective classifier, achieving an overall accuracy of 96.4% by correctly classifying 347 out of 360 cases. It records 46 true positives out of 51 actual positive cases, corresponding to a sensitivity of 90.2%, and incurs only 5 false negatives, demonstrating excellent ability to capture positive outcomes. On

the negative side, it yields 8 false positives against 301 true negatives, resulting in a specificity of 97.4%. This combination of high sensitivity and specificity underscores the Neural Network's robustness and its suitability in critical scenarios where both accurate detection of positives and minimization of false alarms are essential.

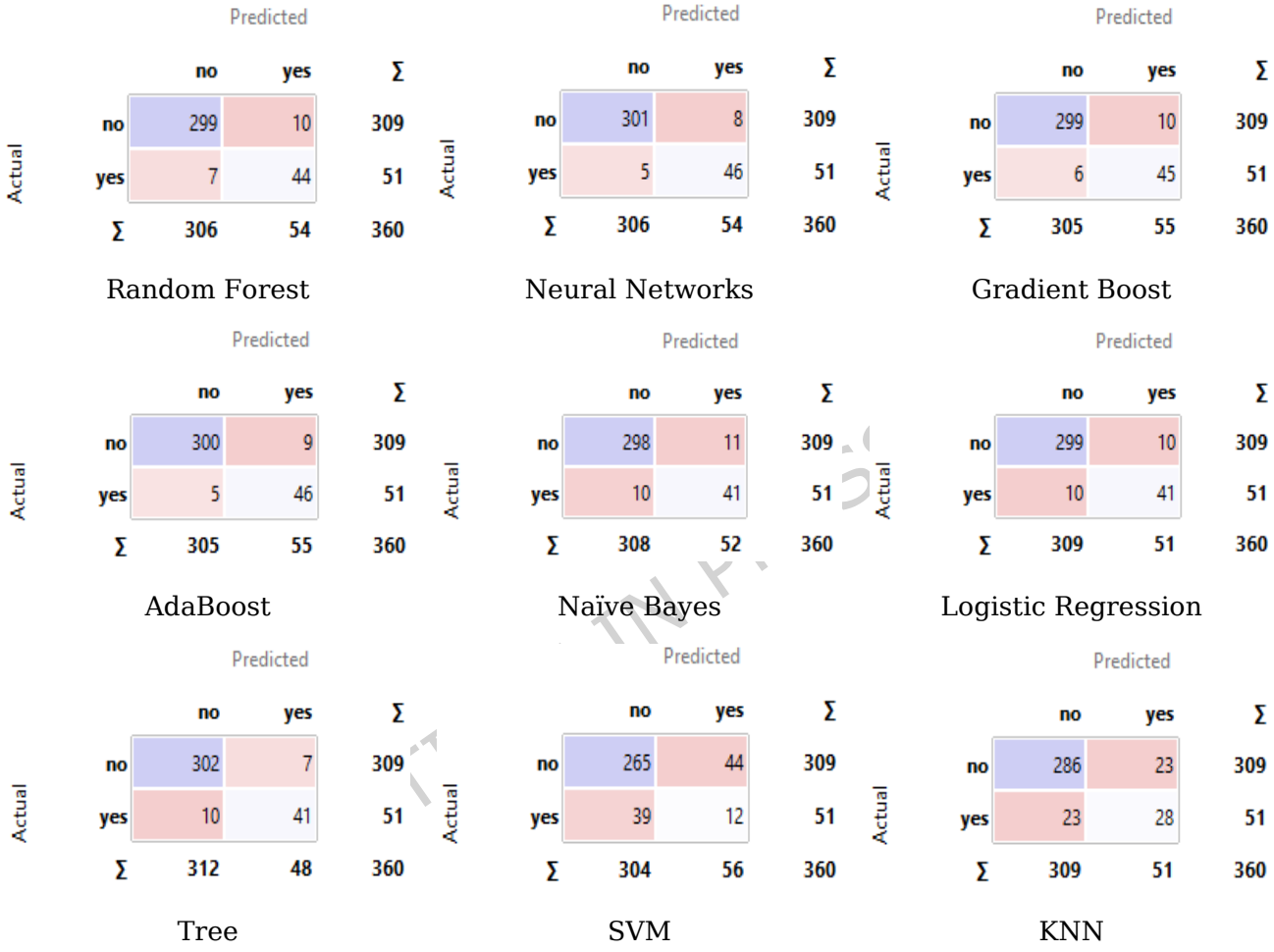


Figure 10: Confusion matrix of the top nine ML models

Ensemble methods such as Random Forest, Gradient Boosting, and AdaBoost also deliver strong performance, with accuracies of 95.2%, 95.6%, and 95.8%, respectively. For example, Random Forest correctly identifies 44 positive cases (sensitivity 86.3%) with only 7 false negatives, while maintaining 299 true negatives (specificity 96.8%). Similarly, Gradient Boosting and AdaBoost both detect 45 positives, corresponding to a sensitivity of 88.2%, with high specificities of 96.8% and 97.1%, respectively. Although these models demonstrate reliability, their positive case detection rates are marginally

lower than those of the Neural Network, which consistently leads in recall while maintaining an excellent overall balance.

Traditional classifiers, such as Logistic Regression and Naïve Bayes, perform moderately well, with accuracies of 94.7% and 94.2%, respectively. Both models correctly identify 41 positive cases (80.4% sensitivity) but miss 10 positives each, resulting in a slightly weaker recall compared to ensemble methods. The Decision Tree achieves 95.3% accuracy, detecting 41 positives again, and offers the highest specificity (97.7%) due to only 7 false positives. However, it too falls short in sensitivity relative to the Neural Network. In contrast, Support Vector Machine (SVM) and k-Nearest Neighbors (KNN) show considerably weaker outcomes. SVM achieves an accuracy of 84.4% with 39 positives detected (76.5% sensitivity) and 44 false positives, while KNN performs slightly better at 87.2% accuracy but with a notably lower sensitivity of 54.9% (28 positives detected, 23 missed).

Overall, these results confirm that while ensemble-based methods provide strong and balanced classification capabilities, the Neural Network consistently outperforms them by achieving the highest true positive count with minimal false negatives, all while maintaining very strong negative class detection. This establishes Neural Networks as the most effective and dependable model in this comparative evaluation.

4 Discussion:

Out of all the ML models adopted for determining the correct prediction, the Neural Network is ranked best according to the evaluation metrics. The Neural Network eliminates the features with the minimum coefficients, resulting in negligible contributions to the prediction. The network structure adopted in this investigation is illustrated in Figure 11 and applied to the unseen data, with the results presented in the current section.

The neural network model employed in this investigation is a Multilayer Perceptron (MLP) architecture, meticulously structured to facilitate both feature abstraction and binary classification. The model consists of an input layer that accepts 11 independent variables, followed by two hidden layers that progressively reduce the network’s complexity, first to 9 neurons and then to 5, culminating in a single-node output layer responsible for delivering a binary recommendation (“Yes” or “No”). This gradual dimensionality reduction reflects the model’s inherent ability to eliminate non-performing or redundant input features, enhancing the interpretability and efficiency of the

learning process. The hidden layers thus act not merely as computational bridges, but as functional filters, compressing the input data into a concise and high-impact representation.

In terms of computational configuration, the network was trained using 100 neurons in the hidden layer with the ReLU (Rectified Linear Unit) activation function, known for its ability to introduce non-linearity while maintaining computational simplicity. The optimization of weights was carried out using the Adam optimizer, which adaptively adjusts learning rates for each parameter, thereby expediting convergence and improving stability. To prevent overfitting and ensure generalization across unseen data, L2 regularization ($\alpha = 0.05$) was applied, penalizing overly complex weight configurations. The training was executed over a maximum of 200 iterations, providing sufficient epochs for convergence without inducing excessive training overhead.

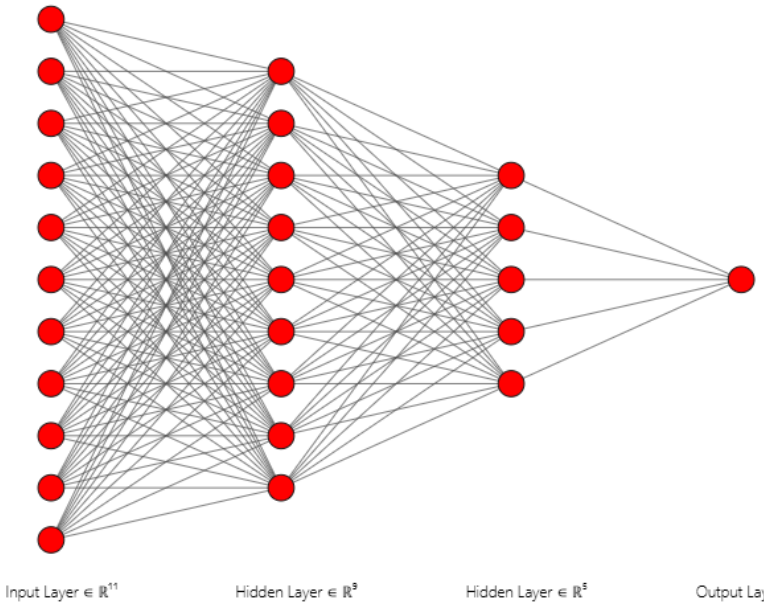
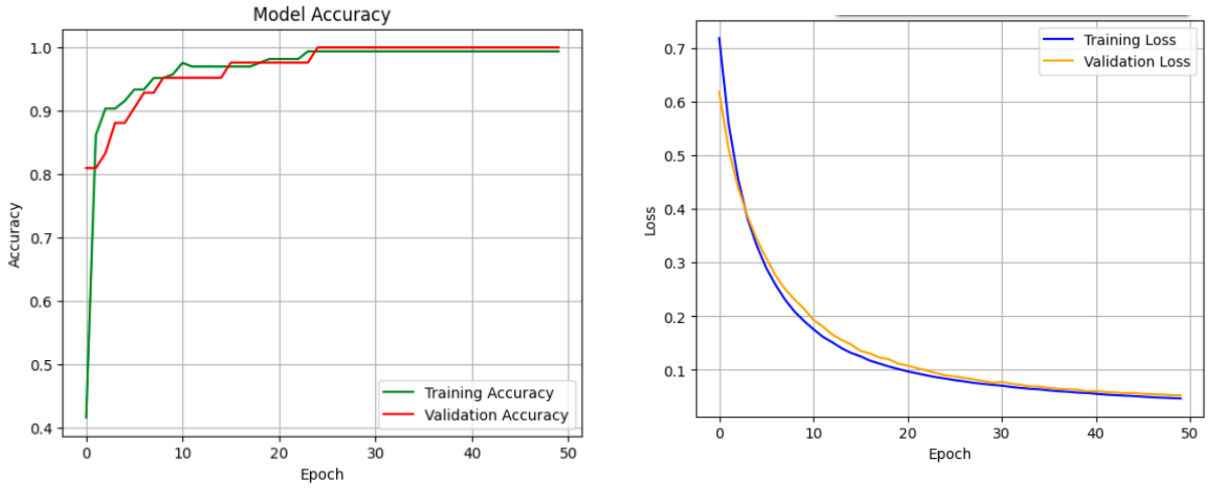


Figure 11: Adopted Neural Network model

This combination of structured dimensional reduction via hidden layers and a carefully tuned learning configuration contributes to the model's robust performance in classification tasks, as reflected in its high true positive rate and low false negative count. By leveraging the ReLU activation and Adam solver in conjunction with regularization, the network maintains both expressive capacity and generalizability. In summary, the adopted neural network framework not only supports accurate decision-making but also demonstrates methodological soundness, rendering it a highly suitable model

for deployment in predictive applications with high-dimensional input features. Figure 12 illustrates the trend of model accuracy enhancement and loss reduction over the epochs.

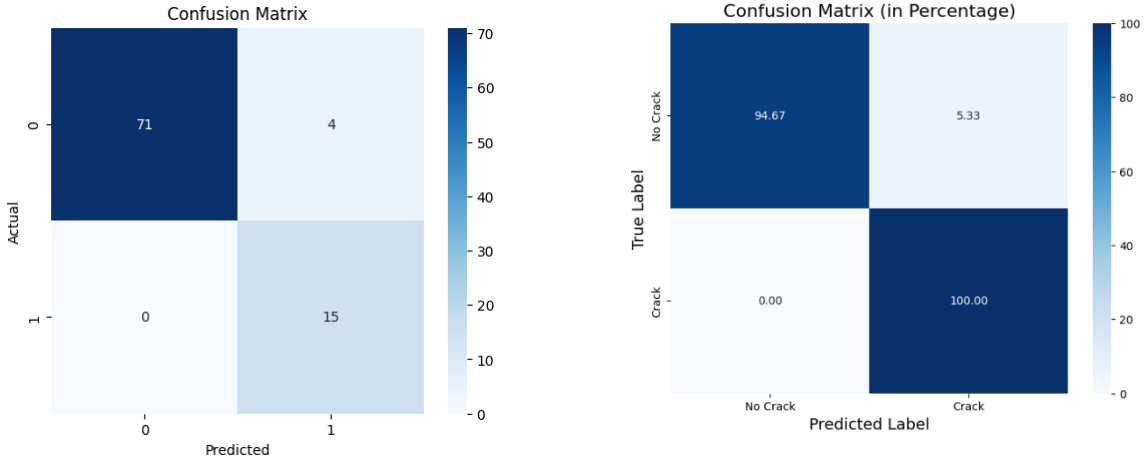


(a) (b)

Figure 12: Model accuracy (a) and loss (b) against Epoch

4.1 Confusion Matrix

The confusion matrix is a fundamental tool for evaluating the classification performance of machine learning models. It provides a detailed account of prediction outcomes by comparing actual labels with predicted outputs, offering counts of true positives, false positives, true negatives, and false negatives. These components enable the calculation of key metrics such as accuracy, sensitivity, specificity, and precision, which are essential for a rigorous assessment of model effectiveness.



Number of Instances Prediction Proportion

Figure 13: Confusion Matrix of test model based on number of instances (a), percentage of prediction (b)

Figure 13 shows the confusion matrix for the neural-network classifier (90 total observations), attests to a consistently strong and well-balanced predictive performance. Of the 75 instances belonging to the “No” class, 71 were correctly identified (true negatives), leaving only 4 false positives. Conversely, of the 15 “Yes/crack” cases, the network correctly detected 15 true positives, with 0 false negatives. These raw counts translate into an overall accuracy of 94.67% (71/75), underpinned by a specificity of 100% (71/71) and a sensitivity of 100% (15/15). The positive-class precision is high at 78.9% (15 / 19), yielding an F_1-score of approximately 0.882, while the negative-class precision (negative predictive value) reaches 100% (71 / 71). Taken together, these figures highlight a model that rarely raises false alarms (low FP rate) and, more importantly for many binary-classification tasks, misses comparatively few true positives (low FN rate). Such a balance is difficult to achieve in practice, especially when the minority class (“Yes”) carries greater operational risk. Yet, the neural network manages to sustain high recall without sacrificing overall precision. In conjunction with the near-ideal ROC profiles and favorable computational characteristics reported earlier, the present confusion-matrix evidence reinforces the conclusion that the neural network offers the most reliable and application-ready solution among the algorithms assessed in this study.

4.2 ROC Analysis

The Receiver Operating Characteristic (ROC) curve obtained from the neural network classifier provides a strong visual and quantitative assessment of the model’s discriminative performance. Importantly, this ROC curve is derived from the predictions made on the test set, ensuring that the evaluation reflects the model’s ability to generalize to unseen data.

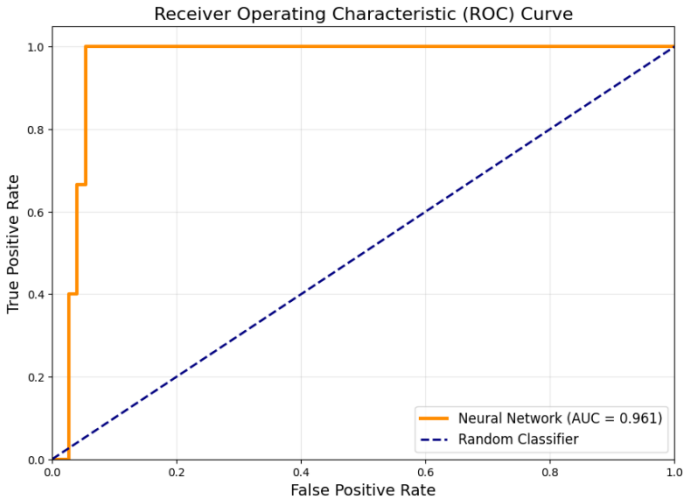


Figure 14: ROC analysis curve

The ROC curve (Figure 14) demonstrates an impressive Area Under the Curve (AUC) of 0.961, which signifies excellent classification capability. The curve exhibits a steep ascent near the y-axis with a minimal false positive rate, closely hugging the top-left corner of the plot. This pattern indicates that the model effectively identifies true positives (crack cases) while minimizing false alarms.

The smooth and sharply rising trajectory of the curve reflects both high sensitivity and high specificity across a wide range of thresholds. Additionally, the substantial distance between the neural network's ROC curve and the diagonal reference line (representing a random classifier) confirms the model's robust performance well beyond chance level.

The AUC value of 0.961, as observed on the test set, further validates the model's ability to achieve a strong balance between sensitivity and specificity. This balance is crucial for practical applications such as weld defect detection, where both false negatives (missed cracks) and false positives (false alarms) can lead to significant operational or safety consequences.

These findings are further supported by the confusion matrix results on the test data, which show perfect sensitivity (100%) and high specificity (94.67%), reinforcing the neural network's effectiveness in identifying defects accurately without sacrificing the precision of negative classifications.

The model's configuration, utilizing the ReLU activation function, Adam solver, and L2 regularization, contributes to its generalization strength

and computational efficiency, making it a reliable and scalable solution for this classification task.

5 Conclusions

This study systematically evaluated multiple machine learning models for weld crack detection, identifying the Neural Network, specifically the Multilayer Perceptron (MLP), as the most accurate and dependable classifier. Performance evaluation was conducted using various metrics, including accuracy, loss, precision, recall, and F1 score, providing a comprehensive assessment of the model's classification capabilities. The confusion matrix confirmed the neural network's high accuracy (94.67%) with perfect sensitivity and specificity, effectively detecting all defect cases without false negatives. The model also maintained strong precision (78.95%) and recall, ensuring low false alarms while reliably identifying critical defects. ROC curve analysis further validated the model's robustness, with an AUC of 0.961, reflecting its outstanding ability to distinguish between cracked and non-cracked welds. Overall, the neural network demonstrated balanced, consistent, and reliable performance across both defect and non-defect classes, making it particularly effective for imbalanced datasets. Based on these findings, the neural network model is strongly recommended for deployment in real-world weld quality assessment and similar predictive applications where safety, detection accuracy, and operational reliability are critical. The model was optimized with ReLU activation, Adam solver, L2 regularization ($\alpha = 0.05$), and 200 iterations, resulting in efficient learning, rapid convergence, and robust generalization on unseen data.

To address the inherent class imbalance in the dataset, the study effectively utilized the Orange data mining tool, which offers powerful pre-processing workflows and supports resampling strategies to improve minority class representation, making it a suitable choice for such investigations. Orange enabled efficient feature extraction and transformation, including selection, ranking, and scaling of key features, ensuring optimal model training. The tool's class balancing and transformation techniques significantly improved the neural network's learning efficiency, while its hidden layers further refined features, enhancing predictive accuracy. The combined use of Orange's imbalance handling techniques and the neural network's optimized configuration resulted in a highly reliable, balanced, and application-ready solution for weld quality assessment.

A user interface (UI) was developed based on the trained neural network model to facilitate ease of use and practical applicability. The interface

(Figure 15(a), Annexure I) enables users to input key welding parameters, including power, welding speed, gas flow rate, focal position, angular position, material strength, and weld path details. Upon submission, the model predicts the likelihood of crack occurrence in the weld and provides the corresponding probability (Figure 15 (b), Annexure I). The UI was implemented using the Flask framework in Python, with front-end development supported by HTML, thereby enabling seamless interaction between the predictive model and end-users.

For deployment, the application is currently hosted on a local server for demonstration purposes. However, the framework is adaptable for deployment on internal servers or cloud platforms, ensuring scalability and accessibility in industrial environments. Such an approach enables real-time monitoring and decision support in manufacturing processes, thereby bridging the gap between predictive modeling and practical implementation within Industry 4.0 ecosystems.

6 Scope for further investigation.

As low-code and no-code platforms gain traction in Industry 4.0, it becomes increasingly important to establish their credibility through rigorous validation. Future research should focus on a comparative assessment of data mining tools, such as Orange and KNIME, against conventional coding-based approaches. Such benchmarking would not only highlight their strengths and limitations but also ensure the reliability and robustness of predictive analytics outcomes when applied in industrial contexts.

Beyond methodological validation, further extensions of this work could include the real-time deployment of predictive models within manufacturing systems, enabling dynamic monitoring and decision-making. The framework can also be broadened to encompass multi-material welding scenarios and other complex manufacturing processes, thereby enhancing its generalizability. Additionally, integrating predictive modeling with digital twin frameworks offers a promising pathway to improve scalability, adaptability, and practical relevance, ultimately strengthening the role of advanced analytics in Industry 4.0.

Declarations

- **Authors Contribution:**

J. Nagendra: Conceptualization, Methodology, Writing – Original Draft; **K. S. Prashanth:** Software, , Formal Analysis, Validation, Visualization; **Kavadiki Veerabhadrappa:** Resources, Data Acquisition, Review & Editing; **K. D. Bopanna:** Methodology, Validation, Writing – Review & Editing; **M. Vinutha:** Data Curation, Formal Analysis, Literature Review, Visualization; **E. Naresh:** Resources, Software, Formal Analysis; **Din Bandhu:** Conceptualization, Formal Analysis, Writing – Review & Editing. All authors have read and approved the final manuscript.

Funding:

The present research was conducted independently and did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Competing interests

The author(s) declare no competing interests.

Data availability

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

[1] M. Norouzian, M. Khakpour, M. Orosnjak, A.A. Kumar, S. Kedziora, Prediction of weld quality in laser welding of hardmetal and steel using high-speed imaging and machine learning methods, Journal of Advanced Joining Processes 11 (2025). https://doi.org/10.1016/j.jajp.2025.100318.

[2] V. Vasan, N.V. Sridharan, R.J. Balasundaram, S. Vaithiyanathan, Ensemble-based deep learning model for welding defect detection and classification, Eng Appl Artif Intell 136 (2024). https://doi.org/10.1016/j.engappai.2024.108961.

[3] H. Yang, H. Wang, H. Li, X. Song, Weld Defect Cascaded Detection Model Based on Bidirectional Multi-scale Feature Fusion and Shape Pre-classification, ISIJ International 62 (2022) ISIJINT-2022-035. https://doi.org/10.2355/isijinternational.ISIJINT-2022-035.

[4] K. Asif, L. Zhang, S. Derrible, J.E. Indacochea, D. Ozevin, B. Ziebart, Machine learning model to predict welding quality using air-coupled

acoustic emission and weld inputs, J Intell Manuf 33 (2022) 881–895. https://doi.org/10.1007/s10845-020-01667-x.

[5] A. Raj, U. Chadha, A. Chadha, R.R. Mahadevan, B.R. Sai, D. Chaudhary, S.K. Selvaraj, R. Lokeshkumar, S. Das, B. Karthikeyan, R. Nagalakshmi, V. Chandramohan, H. Hadidi, Weld quality monitoring via machine learning-enabled approaches, International Journal on Interactive Design and Manufacturing (2023). https://doi.org/10.1007/s12008-022-01165-9.

[6] C. Hake, M. Omlor, A. Breitbarth, G. Notni, K. Dilger, Artificial intelligence methods for in-process high-speed image analysis in laser beam welding of hairpins, IOP Conf Ser Mater Sci Eng 1296 (2023) 012007. https://doi.org/10.1088/1757-899X/1296/1/012007.

[7] K.C. Laxman, N. Tabassum, L. Ai, C. Cole, P. Ziehl, Automated crack detection and crack depth prediction for reinforced concrete structures using deep learning, Constr Build Mater 370 (2023). https://doi.org/10.1016/j.conbuildmat.2023.130709.

[8] J. Rinne, Screening datasets for laser welded steel-copper lap joints, (2021). https://doi.org/10.17632/2S5M3CRBKD.2.

[9] A. Daza, J. Bobadilla, J.C. Herrera, A. Medina, N. Saboya, K. Zavaleta, S. Siguenas, Stacking ensemble based hyperparameters to diagnosing of heart disease: Future works, Results in Engineering 21 (2024). https://doi.org/10.1016/j.rineng.2024.101894.

[10] M. Asmael, Q. Zeeshan, D. Solyali, T. Nasir, Applications of Machine Learning to Friction Stir Welding Process Optimization, 32 (2020) 171–186. https://doi.org/10.17576/jkukm-2020-32(2)-01.

[11] B. Das, S. Pal, S. Bag, Weld quality prediction in friction stir welding using wavelet analysis, The International Journal of Advanced Manufacturing Technology 89 (2017) 711–725. https://doi.org/10.1007/s00170-016-9140-0.

[12] Y. Du, T. Mukherjee, T. DebRoy, Conditions for void formation in friction stir welding from machine learning, NPJ Comput Mater 5 (2019) 68. https://doi.org/10.1038/s41524-019-0207-y.

[13] Q. Wang, W. Jiao, P. Wang, Y. Zhang, A tutorial on deep learning-based data analytics in manufacturing through a welding case study, J Manuf Process 63 (2021) 2–13. https://doi.org/10.1016/j.jmapro.2020.04.044.

[14] Y. Cheng, Q. Wang, W. Jiao, R. Yu, S. Chen, Y. Zhang, J. Xiao, Detecting dynamic development of weld pool using machine learning from innovative composite images for adaptive welding, J Manuf Process 56 (2020) 908–915. https://doi.org/https://doi.org/10.1016/j.jmapro.2020.04.059.

[15] D. Bacioiu, G. Melton, M. Papaelias, R. Shaw, Automated defect classification of SS304 TIG welding process using visible spectrum camera and machine learning, NDT & E International 107 (2019) 102139. https://doi.org/10.1016/j.ndteint.2019.102139.

[16] A. Mayr, D. Kißkalt, M. Meiners, B. Lutz, F. Schäfer, R. Seidel, A. Selmaier, J. Fuchs, M. Metzner, A. Blank, J. Franke, Machine Learning in Production – Potentials, Challenges and Exemplary Applications, Procedia CIRP 86 (2019) 49–54. https://doi.org/10.1016/j.procir.2020.01.035.

[17] A.A. Carvalho, J.M.A. Rebello, L.V.S. Sagrilo, C.S. Camerini, I.V.J. Miranda, MFL signals and artificial neural networks applied to detection and classification of pipe weld defects, NDT & E International 39 (2006) 661–667. https://doi.org/10.1016/j.ndteint.2006.04.003.

[18] M. Ferguson, R. Ak, Y.-T.T. Lee, K.H. Law, Detection and Segmentation of Manufacturing Defects with Convolutional Neural Networks and Transfer Learning, Smart Sustain Manuf Syst 2 (2018) 137–164. https://doi.org/10.1520/SSMS20180033.

[19] A. Khumaidi, E.M. Yuniarno, M.H. Purnomo, Welding defect classification based on convolution neural network (CNN) and Gaussian kernel, in: 2017 International Seminar on Intelligent Technology and Its Applications (ISITIA), IEEE, 2017: pp. 261–265. https://doi.org/10.1109/ISITIA.2017.8124091.

[20] D. You, X. Gao, S. Katayama, WPD-PCA-Based Laser Welding Process Monitoring and Defects Diagnosis by Using FNN and SVM, IEEE Transactions on Industrial Electronics 62 (2015) 628–636. https://doi.org/10.1109/TIE.2014.2319216.

[21] Z. Zhang, G. Wen, S. Chen, Weld image deep learning-based on-line defects detection using convolutional neural networks for Al alloy in robotic arc welding, J Manuf Process 45 (2019) 208–216. https://doi.org/10.1016/j.jmapro.2019.06.023.

[22] K.S. V, J.A. D, AN INDUSTRIAL INSPECTION APPROACH FOR WELD DEFECTS USING MACHINE LEARNING ALGORITHM, International Journal of Advances in Signal and Image Sciences 5 (2019) 15. https://doi.org/10.29284/IJASIS.5.1.2019.15-21.

[23] Y. Chen, B. Chen, Y. Yao, C. Tan, J. Feng, A spectroscopic method based on support vector machine and artificial neural network for fiber laser welding defects detection and classification, NDT & E International 108 (2019) 102176. https://doi.org/10.1016/j.ndteint.2019.102176.

[24] J.D. Kothari, Detecting Welding Defects in Steel Plates using Machine learning and Computer Vision Algorithms, n.d. https://ssrn.com/abstract=3729754.

[25] A. Mayr, B. Lutz, M. Weigelt, T. Glabel, D. Kibkalt, M. Masuch, A. Riedel, J. Franke, Evaluation of Machine Learning for Quality Monitoring of Laser Welding Using the Example of the Contacting of Hairpin Windings, in: 2018 8th International Electric Drives Production Conference (EDPC), IEEE, 2018: pp. 1–7. https://doi.org/10.1109/EDPC.2018.8658346.

[26] N. Persaud, M. Davidson, B. Maniscalco, D. Mobbs, R.E. Passingham, A. Cowey, H. Lau, Awareness-related activity in prefrontal and parietal cortices in blindsight reflects more than superior visual performance, Neuroimage 58 (2011) 605–611. https://doi.org/10.1016/j.neuroimage.2011.06.081.

[27] A. Mayr, A. Meyer, J. Seefried, M. Weigelt, B. Lutz, D. Sultani, M. Hampl, J. Franke, Potentials of machine learning in electric drives production using the example of contacting processes and selective magnet assembly, in: 2017 7th International Electric Drives Production Conference (EDPC), IEEE, 2017: pp. 1–8. https://doi.org/10.1109/EDPC.2017.8328166.

[28] D. Zhao, M. Ivanov, Y. Wang, W. Du, Welding quality evaluation of resistance spot welding based on a hybrid approach, J Intell Manuf 32 (2021) 1819–1832. https://doi.org/10.1007/s10845-020-01627-5.

[29] S.K. Jui, A.B. Kamaraj, M.M. Sundaram, High aspect ratio micromachining of glass by electrochemical discharge machining (ECDM), J Manuf Process 15 (2013) 460–466. https://doi.org/10.1016/j.jmapro.2013.05.006.

[30] Y. Huang, X. Wang, D. Yang, L. Wang, J. Gu, X. Zhang, K. Wang, A Weld Quality Classification Approach Based on Local Mean Decomposition and Deep Belief Network, J Mater Eng Perform 30 (2021) 2229–2237. https://doi.org/10.1007/s11665-021-05495-9.

[31] S.-F. Ling, L.-X. Wan, Y.-R. Wong, D.-N. Li, Input electrical impedance as quality monitoring signature for characterizing resistance spot welding, NDT & E International 43 (2010) 200–205. https://doi.org/10.1016/j.ndteint.2009.11.003.

[32] D. Wu, H. Chen, Y. Huang, S. Chen, Online Monitoring and Model-Free Adaptive Control of Weld Penetration in VPPAW Based on Extreme Learning Machine, IEEE Trans Industr Inform 15 (2019) 2732–2740. https://doi.org/10.1109/TII.2018.2870933.

[33] S. Zamanzad Gavidel, S. Lu, J.L. Rickli, Performance analysis and comparison of machine learning algorithms for predicting nugget width of resistance spot welding joints, The International Journal of Advanced Manufacturing Technology 105 (2019) 3779–3796. https://doi.org/10.1007/s00170-019-03821-z.

[34] L. Li, D. Liu, J. Liu, H. Zhou, J. Zhou, Quality Prediction and Control of Assembly and Welding Process for Ship Group Product Based on Digital Twin, Scanning 2020 (2020) 1–13. https://doi.org/10.1155/2020/3758730.

[35] M.P. Satpathy, S.B. Mishra, S.K. Sahoo, Ultrasonic spot welding of aluminum-copper dissimilar metals: A study on joint strength by experimentation and machine learning techniques, J Manuf Process 33 (2018) 96–110. https://doi.org/10.1016/j.jmapro.2018.04.020.

[36] W. Cai, J. Wang, L. Cao, G. Mi, L. Shu, Q. Zhou, P. Jiang, Predicting the weld width from high-speed successive images of the weld zone using different machine learning algorithms during laser welding, Mathematical Biosciences and Engineering 16 (2019) 5595–5612. https://doi.org/10.3934/mbe.2019278.

[37] K. Nomura, K. Fukushima, T. Matsumura, S. Asai, Burn-through prediction and weld depth estimation by deep learning model monitoring the molten pool in gas metal arc welding with gap fluctuation, J Manuf Process 61 (2021) 590–600. https://doi.org/10.1016/j.jmapro.2020.10.019.

[38] C. Stadter, M. Schmoeller, L. von Rhein, M.F. Zaeh, Real-time prediction of quality characteristics in laser beam welding using optical coherence tomography and machine learning, J Laser Appl 32 (2020). https://doi.org/10.2351/7.0000077.

[39] L. Na, S. Chen, Q. Chen, W. Tao, H. Zhao, S. Chen, Dynamic welding process monitoring based on microphone array technology, J Manuf Process 64 (2021) 481–492. https://doi.org/10.1016/j.jmapro.2020.12.023.

[40] C. Gonzalez-Val, A. Pallas, V. Panadeiro, A. Rodriguez, A convolutional approach to quality monitoring for laser manufacturing, J Intell Manuf 31 (2020) 789–795. https://doi.org/10.1007/s10845-019-01495-8.

[41] D. Mishra, A. Gupta, P. Raj, A. Kumar, S. Anwer, S.K. Pal, D. Chakravarty, S. Pal, T. Chakravarty, A. Pal, P. Misra, S. Misra, Real time monitoring and control of friction stir welding process using multiple sensors, CIRP J Manuf Sci Technol 30 (2020) 1–11. https://doi.org/10.1016/j.cirpj.2020.03.004.

[42] P. Sassi, P. Tripicchio, C.A. Avizzano, A Smart Monitoring System for Automatic Welding Defect Detection, IEEE Transactions on Industrial Electronics 66 (2019) 9641–9650. https://doi.org/10.1109/TIE.2019.2896165.

[43] S. Shevchik, T. Le-Quang, B. Meylan, F.V. Farahani, M.P. Olbinado, A. Rack, G. Masinelli, C. Leinenbach, K. Wasmer, Supervised deep learning for real-time quality monitoring of laser welding with X-ray radiographic guidance, Sci Rep 10 (2020) 3389. https://doi.org/10.1038/s41598-020-60294-x.

[44] A. Sumesh, D.T. Thekkuden, B.B. Nair, K. Rameshkumar, K. Mohandas, Acoustic Signature Based Weld Quality Monitoring for SMAW Process Using Data Mining Algorithms, Applied Mechanics and Materials 813–814 (2015) 1104–1113. https://doi.org/10.4028/www.scientific.net/AMM.813-814.1104.

[45] L. Nguyen, J. Buhl, M. Bambach, Continuous Eulerian tool path strategies for wire-arc additive manufacturing of rib-web structures with machine-learning-based adaptive void filling, Addit Manuf 35 (2020) 101265. https://doi.org/10.1016/j.addma.2020.101265.

[46] J. Günther, P.M. Pilarski, G. Helfrich, H. Shen, K. Diepold, First Steps Towards an Intelligent Laser Welding Architecture Using Deep Neural

Networks and Reinforcement Learning, Procedia Technology 15 (2014) 474–483. https://doi.org/10.1016/j.protcy.2014.09.007.

[47] M.N. Khalid, V. Naranje, V.H. Gaidhane, Prediction of Best Weld Quality Using Artificial Neural Network, in: 2019 Amity International Conference on Artificial Intelligence (AICAI), IEEE, 2019: pp. 213–217. https://doi.org/10.1109/AICAI.2019.8701351.

[48] A. Baraka, G. Panoutsos, S. Cater, A real-time quality monitoring framework for steel friction stir welding using computational intelligence, J Manuf Process 20 (2015) 137–148. https://doi.org/10.1016/j.jmapro.2015.09.001.

[49] K. Asif, L. Zhang, S. Derrible, J.E. Indacochea, D. Ozevin, B. Ziebart, Machine learning model to predict welding quality using air-coupled acoustic emission and weld inputs, J Intell Manuf 33 (2022) 881–895. https://doi.org/10.1007/s10845-020-01667-x.

[50] H. Dong, M. Cong, Y. Zhang, Y. Liu, H. Chen, Modeling and real-time prediction for complex welding process based on weld pool, The International Journal of Advanced Manufacturing Technology 96 (2018) 2495–2508. https://doi.org/10.1007/s00170-018-1685-7.

[51] Y. Guo, X. Li, X. Fang, X. Lin, Y. Song, S. Jiang, B. Stanton, A comparison of four sampling methods among men having sex with men in China: implications for HIV/STD surveillance and prevention, AIDS Care 23 (2011) 1400–1409. https://doi.org/10.1080/09540121.2011.565029.

[52] T. Sterling, H. Chen, Robotic welding parameter optimization based on weld quality evaluation, in: 2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), IEEE, 2016: pp. 216–221. https://doi.org/10.1109/CYBER.2016.7574825.

[53] A. Sumesh, K. Rameshkumar, K. Mohandas, R.S. Babu, Use of Machine Learning Algorithms for Weld Quality Monitoring using Acoustic Signature, Procedia Comput Sci 50 (2015) 316–322. https://doi.org/10.1016/j.procs.2015.04.042.

[54] S. Ravikumar, K.I. Ramachandran, V. Sugumaran, Machine learning approach for automated visual inspection of machine components, Expert Syst Appl 38 (2011) 3260–3266. https://doi.org/10.1016/j.eswa.2010.09.012.

[55] P. Rodriguez-Gonzalvez, M. Rodriguez-Martin, Weld Bead Detection Based on 3D Geometric Features and Machine Learning Approaches, IEEE Access 7 (2019) 14714–14727. https://doi.org/10.1109/ACCESS.2019.2891367.

[56] K. Balachandar, R. Jegadeeshwaran, Friction stir welding tool condition monitoring using vibration signals and Random forest algorithm – A Machine learning approach, Mater Today Proc 46 (2021) 1174–1180. https://doi.org/10.1016/j.matpr.2021.02.061.

[57] C. Chen, N. Lv, S. Chen, Welding penetration monitoring for pulsed GTAW using visual sensor based on AAM and random forests, J Manuf Process 63 (2021) 152–162. https://doi.org/10.1016/j.jmapro.2020.04.005.

[58] A. Sumesh, B.B. Nair, K. Rameshkumar, A. Santhakumari, A. Raja, K. Mohandas, Decision tree based weld defect classification using current and voltage signatures in GMAW process, Mater Today Proc 5 (2018) 8354–8363. https://doi.org/10.1016/j.matpr.2017.11.528.

[59] D.J. Huggett, T.W. Liao, M.A. Wahab, A. Okeil, Prediction of friction stir weld quality without and with signal features, The International Journal of Advanced Manufacturing Technology 95 (2018) 1989–2003. https://doi.org/10.1007/s00170-017-1403-x.

[60] C. Knaak, U. Thombansen, P. Abels, M. Kröger, Machine learning as a comparative tool to determine the relevance of signal features in laser welding, Procedia CIRP 74 (2018) 623–627. https://doi.org/10.1016/j.procir.2018.08.073.

[61] A. Daza, J. Bobadilla, J.C. Herrera, A. Medina, N. Saboya, K. Zavaleta, S. Siguenas, Stacking ensemble based hyperparameters to diagnosing of heart disease: Future works, Results in Engineering 21 (2024) 101894. https://doi.org/10.1016/j.rineng.2024.101894.

[62] Y. Li, B. Yu, B. Wang, T.H. Lee, M. Banu, Online quality inspection of ultrasonic composite welding by combining artificial intelligence technologies with welding process signatures, Mater Des 194 (2020) 108912. https://doi.org/10.1016/j.matdes.2020.108912.

[63] A. Raj, U. Chadha, A. Chadha, R.R. Mahadevan, B.R. Sai, D. Chaudhary, S.K. Selvaraj, R. Lokeshkumar, S. Das, B. Karthikeyan, R. Nagalakshmi, V. Chandramohan, H. Hadidi, Weld quality monitoring via machine learning-enabled approaches, International Journal on Interactive

Design and Manufacturing (IJIDeM) (2023). https://doi.org/10.1007/s12008-022-01165-9.

[64] F. Duan, S. Yin, P. Song, W. Zhang, C. Zhu, H. Yokoi, Automatic Welding Defect Detection of X-Ray Images by Using Cascade AdaBoost With Penalty Term, IEEE Access 7 (2019) 125929–125938. https://doi.org/10.1109/ACCESS.2019.2927258.

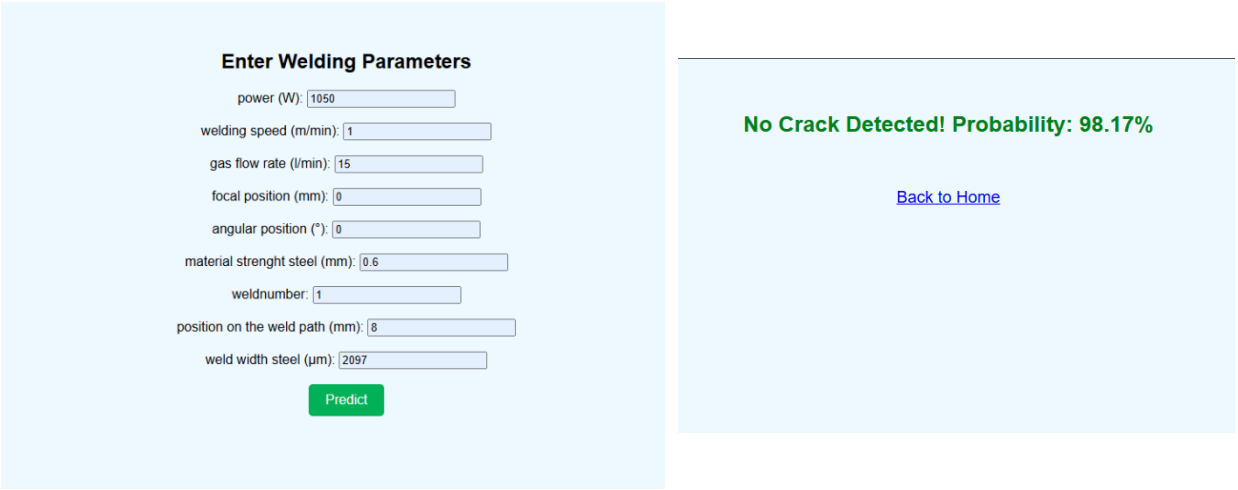
[65] M. Mohana, P. Subashini, Analysing the performance of Viola-Jones and multi-task convolution neural networks face detection algorithms using real-time video sequences, Int J Comput Vis Robot 15 (2025) 286–311. https://doi.org/10.1504/IJCVR.2025.146293.

[66] J. Stavridis, A. Papacharalampopoulos, P. Stavropoulos, A cognitive approach for quality assessment in laser welding, Procedia CIRP 72 (2018) 1542–1547. https://doi.org/10.1016/j.procir.2018.03.119.

[67] A. Ishak, K. Siregar, Aspriyati, R. Ginting, M. Afif, Orange Software Usage in Data Mining Classification Method on The Dataset Lenses, IOP Conf Ser Mater Sci Eng 1003 (2020) 012113. https://doi.org/10.1088/1757-899X/1003/1/012113.

[68] J. Demšar, B. Zupan, Orange: Data mining fruitful and fun - A historical perspective, Informatica (Slovenia) 37 (2013) 55–60.

Annexure A:



(a) (b)

Figure 15: User Interface for entering the parameters (a) and the predicted outcome (b)