

IJAMEC
**International Journal of
Applied Methods in
Electronics and Computers**

**Vol. 14 (3)
2026**

Contents	Pages
<i>Research Articles</i>	
Indabot: A Simulation-Based Study of Low-Cost, ArUco Marker-Based Autonomous Navigation for Indoor Plant Phenotyping Fikret UYSAL, Kadir Sabancı	102-109
C-PPDSS: A Comprehensive Patient Personal Data Sovereignty System for Real-Time PII Redaction in Healthcare LLM Workflows Edoghogho Olaye, Williams O. Aigbe, David Onwuka, Daniel Obuh, Amarachi Madukwe	110-119
Sampling Strategies and Metric Sensitivity in Imbalanced Predictive Maintenance: A Comparative Study with Ensemble Methods Berk Mercan, Eren Ağar, Mert Çimentepe, Meltem Apaydın Üstün	120-129
YOLOv10 and RT-DETR for Weld Defect Detection: Experimental Evaluation Adem Dilbaz	130-142



Research Article

Indabot: A Simulation-Based Study of Low-Cost, ArUco Marker-Based Autonomous Navigation for Indoor Plant Phenotyping

Fikret Uysal ^{a,*} , Kadir Sabanci ^b 

^a Karamanoglu Mehmetbey University, Department of Electrical and Electronics Engineering, Karaman, Türkiye

^b Karamanoglu Mehmetbey University, Department of Electrical and Electronics Engineering, Karaman, Türkiye

ARTICLE INFO

Article history:

Received 15 May 2026

Accepted 21 June 2026

Keywords:

Autonomous mobile robot,
Indoor navigation,
PID control,
Webots simulation,
ArUco marker

ABSTRACT

Autonomous robotic platforms are of critical importance for high-throughput plant phenotyping in controlled agricultural environments. The vast majority of existing systems are designed for open-field deployment and rely on satellite-based navigation, which is not suitable for enclosed greenhouse settings. This study presents the design, simulation-based performance analysis of Indabot, an autonomous mobile robot developed for wheat phenotyping in indoor environments. Each wheat sample is uniquely tagged with an ArUco marker, enabling drift-free, plant-level localization without the need for any GPS infrastructure. A rotation-based random search navigation scenario was designed in the Webots R2023b environment, in which the OpenCV ArUco module was combined with discrete-time PID controllers and lidar distance feedback. The system achieved a 100% target detection rate across 13 independent simulation runs, with a mean mission completion time of 282.4 ± 6.2 s ($n=13$, $CV=2.2\%$), a marker centering error of 3.06 ± 0.51 pixels, and an average approach time per target of 19.4 ± 0.4 s. The findings demonstrate that the Indabot platform, operating under ArUco-based guidance, offers a consistent, scalable, and low-cost autonomous phenotyping alternative for indoor agricultural environments. All performance metrics reported in this study were obtained exclusively within the Webots simulation environment; no physical validation of the platform has yet been conducted, and this constitutes a key limitation of the present work.

This is an open access article under the CC BY-SA 4.0 license.
(<https://creativecommons.org/licenses/by-sa/4.0/>)

1. INTRODUCTION

Global food security is under serious threat due to population growth, climate change, and the diminishing availability of arable land. Breeding efforts aimed at improving the productivity of wheat, which provides approximately 20% of the global caloric demand, rely heavily on accurate phenotypic measurements [1]. The high labor requirements and subjective nature of traditional manual methods have given rise to the concept of High-Throughput Phenotyping (HTP), which encompasses advanced sensing and robotic technologies [2]–[4]. Controlled Environment Agriculture (CEA) facilities are ideally suited for reproducible research thanks to the precise environmental control they provide [5]. The use of artificial intelligence-based models in such

environments enables detailed growth analyses [6]. However, the confined geometry of indoor spaces and the absence of GNSS signals restrict the deployment of existing field robots—such as BoniRob [7] and Thorvald II [8]—in these settings. More recent platforms such as PhenoRob-F [9] support both open-field and potted-plant phenotyping but rely on pre-defined navigation paths and lack mechanisms for individual plant-level identification via fiducial markers. While fixed indoor systems offer high precision, they lack flexibility and scalability. To overcome these limitations, ArUco fiducial markers offer a low-cost localization alternative [10]. The ArUco system enables 6-degree-of-freedom pose estimation through the Perspective-n-Point algorithm [11] and has been further enhanced through sensor fusion [12], 3D marker boards [13], and optimal placement strategies [14]. Although such

* Corresponding author. E-mail address: fideuysal@gmail.com
DOI: 10.58190/ijamec.2026.177

markers have been employed in agricultural applications such as robotic arm calibration [15], their use for sequential autonomous phenotyping navigation through individual plant identification has not yet been sufficiently investigated in the literature.

Since physical testing of autonomous robots is costly, simulation environments are indispensable [16]. The Webots platform [17] provides high accuracy in the kinematic simulation of agricultural robots operating at low speeds. Although Webots is widely used for testing agricultural algorithms, a notable gap remains in the literature regarding the simulation of marker-based indoor phenotyping strategies. This study presents a GPS-independent, low-cost autonomous phenotyping platform that, unlike existing phenotyping robots designed primarily for GPS-dependent open-field deployment, eliminates the cumulative drift problem inherent to SLAM by assigning a unique ID to each plant via ArUco markers in enclosed greenhouse environments. The algorithmic novelty of this study lies in the rotation-based random search navigation strategy formulated under the assumption that marker locations are not known a priori, together with the dual PID architecture that simultaneously performs forward distance regulation and pixel-based heading deviation correction. The modular and reproducible hardware design—achieved through a sigma-profile chassis, hoverboard hub motors, and consumer-grade components—provides a reference architecture for research groups that lack access to high-cost systems.

Finally, the statistical validation conducted across 13 independent Webots runs and supported by target-based dispersion analyses establishes a level of methodological rigor that is rarely reported in comparable systems.

2. MATERIAL AND METHODS

2.1. Indabot Hardware Architecture

The Indabot platform is constructed from 30×30 mm aluminum sigma profiles, which enable straightforward assembly without the need for welding. The robot has approximate dimensions of $300 \times 350 \times 250$ mm (L $\times$ W $\times$ H) and a total weight of approximately 15 kg. The structural layout and the assembled platform are illustrated in Figure 1. Locomotion is provided by two hoverboard-type BLDC hub motors with a 165 mm wheel diameter, configured in a differential drive arrangement. This design eliminates the need for external gears, belts, or chain transmissions. The motors are driven by an ODrive 3.6 BLDC controller implementing the Field-Oriented Control (FOC) algorithm and communicate bidirectionally with the main computing unit over UART at 115200 baud. The computing unit is an NVIDIA Jetson Orin NX single-board computer. The LiDAR is used to measure distances to surrounding objects, while the camera serves both for the recognition of ArUco markers and for image acquisition. The technical specifications of the hardware are summarized in Table 1.

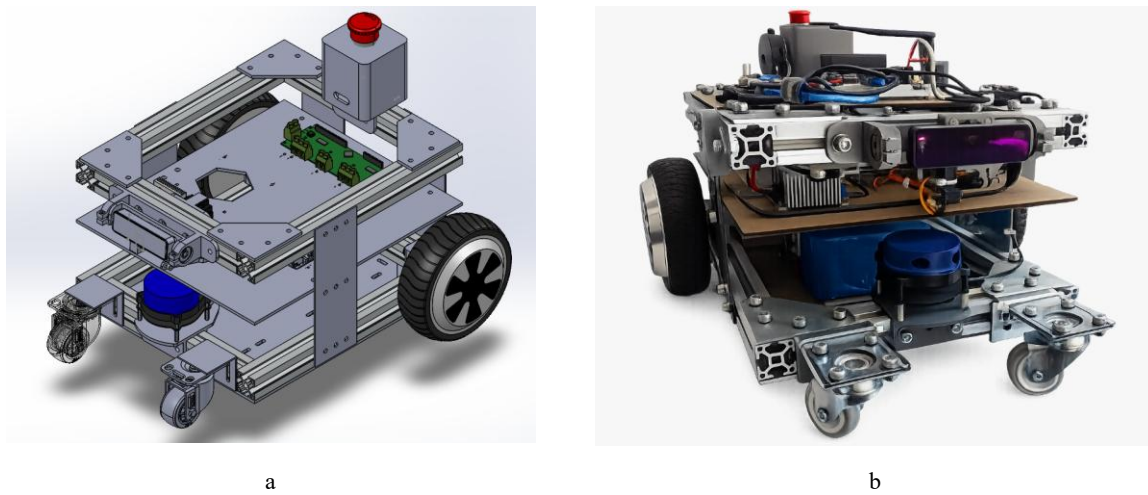


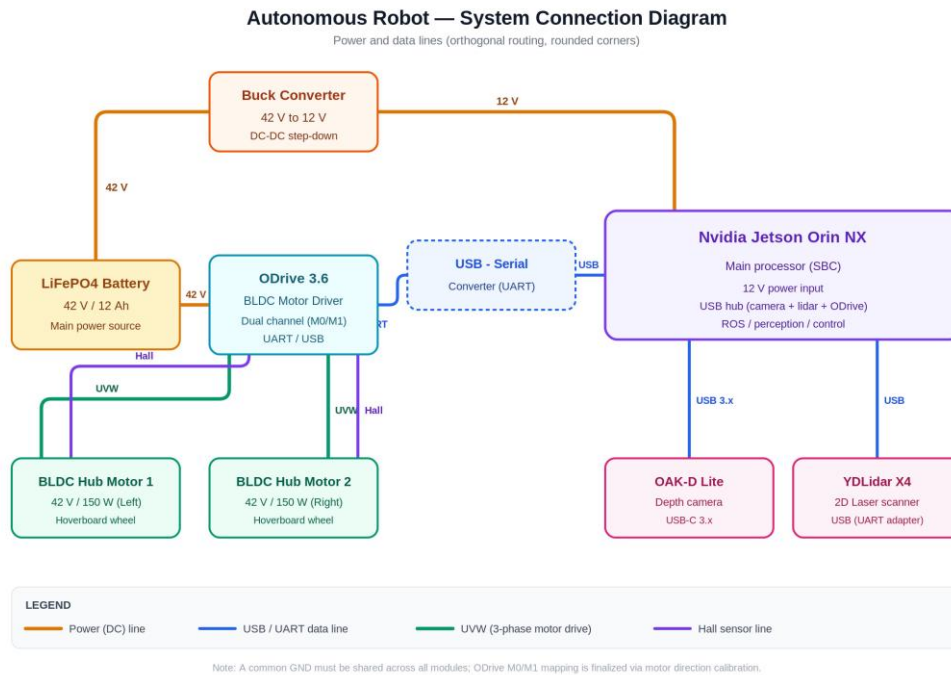
Figure 1. The Indabot autonomous phenotyping platform. **(a)** SolidWorks CAD model illustrating the structural layout and component placement. **(b)** Assembled physical platform showing the sigma-profile chassis, BLDC hub motors, OAK-D Lite stereo camera, and LiDAR sensor.

Table 1. Hardware specifications of the Indabot autonomous phenotyping platform.

Component	Technical Specification	Role in the System
Motor Driver	ODrive 3.6 (FOC)	BLDC speed/position control, battery voltage monitoring
Drive Motors	Hoverboard BLDC Hub (Ø165 mm)	Differential drive locomotion
Battery	42 V, 12 Ah LiFePO4 (~500 Wh)	Main power source
Computing Unit	NVIDIA Jetson Orin NX (8 GB)	Image processing, navigation logic
Depth Camera	Luxonis OAK-D Lite (IMX214)	ArUco detection, RGB phenotyping
LiDAR Sensor	YDLidar X4 (360°, 0.12–10 m)	Distance measurement, safety monitoring
Chassis Frame	30 × 30 mm sigma profile	Structural support, modular assembly
Power Regulation	DC-DC buck converter (12 V)	Peripheral hardware power supply

2.2. Power Management and Schematic

The electrical subsystem is organized around a 42 V, 12 Ah LiFePO4 battery (Figure 2). The battery directly powers the ODrive 3.6 and the two BLDC hub motors, while a DC-DC buck converter generates regulated 12 V output for the Jetson Orin NX and peripheral hardware. Optical isolation is implemented in the USB-to-serial converter. The separation of the motor and computing power lines minimizes electromagnetic interference arising from motor switching transients. The approximately 500 Wh capacity provides around 5 hours of continuous operation under typical phenotyping workloads.

**Figure 2.** Schematic diagram of the Indabot power management architecture.

2.3. Sensing and Computing Architecture

The sensing architecture combines a stereo depth camera for visual marker detection with a 2D laser range finder for distance feedback. The imaging task is handled by the Luxonis OAK-D Lite camera, which provides RGB and hardware-accelerated stereo depth streams, while range measurement is carried out by the YDLidar X4 operating over a 0.12–10 m range with 0.5° angular resolution at a scan rate of 6–12 Hz. During the approach maneuver, forward distance feedback is obtained from a narrow angular window centered on the robot's forward axis (LiDAR indices 175–185, $\pm 5^\circ$), and the minimum range value within this window is provided as input to the PID controller (Equation 1):

$$d_{\text{forward}} = \min(r_i), \quad i \in [175, 185] \quad (1)$$

Here, r_i denotes the range measurement at LiDAR index i . All sensor streams are processed on the NVIDIA Jetson Orin NX, which serves as the central computing node of the system.

2.4. ArUco Marker Detection and Plant Identification

Each wheat sample is tagged with an ArUco marker from the DICT_6X6_1000 dictionary, which provides 1,000 unique binary patterns encoded within a 6×6 bit grid. The detection pipeline, implemented in Python using the OpenCV ArUco module, comprises the following stages: (i) Image Acquisition — the Webots camera buffer is converted into RGBA and grayscale representations using NumPy and OpenCV; (ii) Candidate Extraction —

adaptive thresholding is applied to the grayscale image, and quadrilateral candidates are identified through contour detection and polygonal approximation; (iii) Marker Identification — each candidate quadrilateral is rectified via perspective transformation and binarized, then compared against the dictionary using Hamming distance matching; (iv) Center Computation — for each detected marker, the image-plane center is computed as the mean of opposite corners (Equation 2):

$$c_x = (x_{TL} + x_{BR}) / 2, \quad c_y = (y_{TL} + y_{BR}) / 2 \quad (2)$$

Here, c_x and c_y denote the horizontal and vertical pixel coordinates, respectively, of the ArUco marker's center point in the image plane. The values x_{TL} and y_{TL} represent the pixel coordinates of the marker's top-left corner, while x_{BR} and y_{BR} correspond to those of the bottom-right corner. These corner points are obtained from the four-corner vector returned in clockwise order by the `cv2.aruco.detectMarkers` function of the OpenCV library. Since the top-left and bottom-right corners constitute the diagonal corners of the marker, the arithmetic mean of these two points yields the geometric center of the marker.

The horizontal alignment error between the marker center and the camera's optical axis is defined as follows (Equation 3):

$$e_{\text{alignment}} = |c_x - W/2| \quad (3)$$

Here, $W = 640$ pixels denotes the image width. The marker is considered to be aligned when $e_{\text{alignment}} < 20$ pixels, which corresponds to an angular deviation of less than 2.25° given the camera's 73° horizontal field of view. Captured images are archived using a naming convention that incorporates the current timestamp and the ArUco marker ID associated with the corresponding plant.

2.5. PID-Based Motion Control

The robot's motion control system employs four independent discrete-time PID controllers. The standard discrete-time PID control law is formulated as follows (Equation 4):

$$u(k) = K_p \cdot e(k) + K_i \cdot \sum_{\tau=0}^k e(\tau) \Delta t + K_d \cdot [e(k) - e(k-1)] / \Delta t \quad (4)$$

Here, $u(k)$ is the controller output at time step k , $e(k)$ is the error signal, K_p , K_i , and K_d are the proportional, integral, and derivative gains, respectively, and $\Delta t = 0.05$ s is the sampling period derived from the Webots simulation time step. All controller outputs are clipped to the range $[-5, +5]$ rad/s to prevent actuator saturation. The controller parameters are listed in Table 2. The gain values listed in Table 2 were determined empirically through an iterative manual tuning procedure conducted within the Webots simulation environment, rather than through a formal auto-tuning method such as Ziegler–Nichols. For each controller, multiple candidate gain

combinations were tested, and their effect on the approach speed and the marker centering (alignment) error was systematically observed; the combination yielding the most stable and accurate performance for each controller was retained. This empirical approach was adopted because the simulated actuator dynamics differ from the idealized analytical plant models typically assumed by classical tuning methods, making manual adjustment a more practical choice for the present application.

Table 2. PID controller gains and functional assignments ($\Delta t = 0.05$ s; output range $[-5, +5]$ rad/s).

Controller	Kp	Ki	Kd	Setpoint	Function
pidx	3.0	0.025	0.08	1.0 m	Forward approach distance regulation
pidBackward	3.0	0.08	0.05	Variable	Reverse motion: return to pre-approach pose
pidYaw	3.0	0.2	0.07	0.0	Heading correction during in-place rotation
pidYawApproach	3.0	0.02	0.07	0.0	Steering during forward approach

The velocity controller (pidx) uses the forward distance measured by the LiDAR as its process variable and ensures that the target approach distance of 1.0 m is reached within a tolerance of ± 0.05 m. The heading controller (pidYawApproach) provides real-time steering correction during the drive-and-turn approach phase. In this mode, the pixel error is normalized to the range $[-1, +1]$ (Equation 5):

$$e_{\text{deviation}} = (W/2 - c_x) / (W/2) \quad (5)$$

Here, $e_{\text{deviation}}$ denotes the PID input signal, while $u_{\text{deviation}}$ denotes the PID output. The normalized error is fed into pidYawApproach, and its output $u_{\text{deviation}}$ modifies the left and right wheel velocities differentially: $v_{\text{left}} = v_{\text{forward}} + u_{\text{deviation}}$, $v_{\text{right}} = v_{\text{forward}} - u_{\text{deviation}}$. This dual PID architecture enables the robot to maintain visual centering while simultaneously reducing the distance to the target.

2.6. Webots Simulation Environment and Digital Twin

The navigation algorithm was developed and validated in the Webots R2023b environment prior to physical deployment. A three-dimensional robot model replicating the geometry, mass distribution, and joint configurations of the physical platform was created at 1:1 scale in SolidWorks, exported to URDF format using the SolidWorks-to-URDF plugin, and subsequently converted into the PROTO format used by Webots. The simulated robot incorporates virtual counterparts of all physical

sensors (Figure 3).

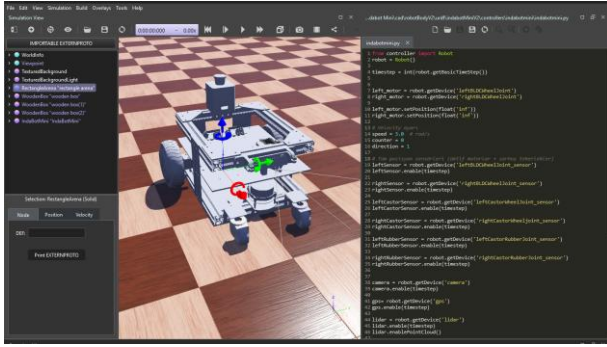


Figure 3. Webots simulation environment and robot control interface.

2.7. Navigation Scenario: Rotation-Based Random Search

A number of ArUco marker IDs equal to the number of plants is generated, starting from 0. The ArUco markers are placed on the plants without regard to ordering. The marker with ID 999 is reserved for terminating the operation; upon detecting marker 999, the robot completes the search task and initiates the parking procedure. The number and locations of the ArUco markers are not known in advance. Since the robot performs the search by rotating in place, the ArUco markers must be oriented to face the center of the simulation environment. The scenario consists of the following steps: (i) navigation to the center of the environment (`gotoMidPoint`), (ii) a 180° turn followed by a slow rotational scan, (iii) marker detection, alignment, and approach, (iv) image capture and timestamped archiving, and (v) return to the initial pose and continuation of the cycle. The robot measures its distance to the opposing wall using the angle and range data provided by the LiDAR within the 135°–225° sector (Equation 6):

$$d_{\text{wall}} = \max \{ |\cos(\theta_i)| \cdot r_i \}, \quad i \in [135, 225] \quad (6)$$

Once the center point is reached, the robot performs a 180° turn and initiates a slow rotational scan (left motor at 1.0 rad/s, right motor stationary). The robot maintains a list of the IDs of previously visited plants. When an unvisited marker is detected, the `alignToMarker` procedure rotates the robot until centering is achieved. Once the marker is precisely centered in the camera image, the `approachToMarker` procedure is executed to initiate the approach maneuver. During the approach phase, a dual PID strategy is employed: `pidX` regulates the forward velocity while `pidYawApproach` simultaneously provides heading correction. When the robot's distance to the plant reaches 1 m, the robot halts. An image is captured and archived together with the marker ID and timestamp. The ArUco marker ID corresponding to the plant is then added to the list of visited markers. The robot subsequently returns to its initial pose. The cycle is repeated until the termination marker with ID 999 is detected. Upon

detecting ID 999, the robot approaches the marker, performs a 180° turn, and reverses into the parking position. A snapshot of the scenario running in Webots is shown in Figure 4. The video recording of the simulation in the Webots environment is provided in [18].

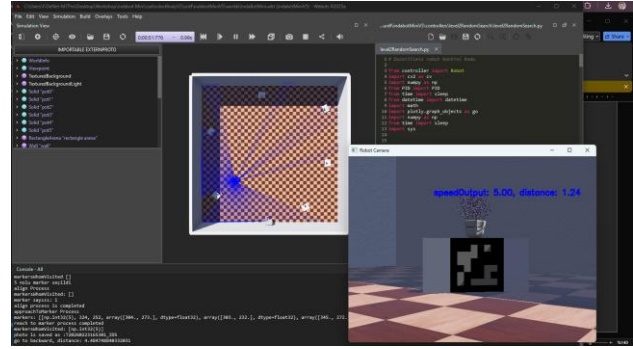


Figure 4. Webots simulation of the rotation-based random search scenario

3. RESULTS AND DISCUSSION

3.1. Marker Detection Performance

Under the uniform illumination of the simulation environment, the ArUco detection module achieved a 100% detection rate for markers within the camera's field of view at distances of up to approximately 4 m (Table 3). No false-positive detections or marker identity classification errors were recorded in any of the trials.

Table 3. Mission and detection performance metrics from 13 independent Webots runs (6 phenotyping + 1 parking target).

Performance Metric	Result
Marker detection rate	100%
Alignment accuracy (pixels)	≤ 20
Approach distance error (m)	± 0.05
Image capture success rate	100%
False-positive detections	0
Number of active PID controllers	4
Mission completion rate	100%

3.2. PID Performance and Statistical Results

The PID controllers exhibited stable convergence within the specified tolerance bands across all navigation phases, with no observable overshoot. The forward approach controller (`pidX`) reached the target distance of 1.0 m within a tolerance of ± 0.05 m in all 13 trials. Although $K_i = 0.025$ is modest in magnitude, it proved decisive in eliminating steady-state error under variable initial distances, while $K_d = 0.08$ provided sufficient damping to suppress oscillation during the final approach phase. The dual PID strategy ensured that the normalized pixel error remained below the 20-pixel threshold along more than 95% of the approach trajectory. The overall statistics obtained from 13 independent runs are presented in Table 4. The mean times across the 13 runs for the ordered ArUco marker IDs and the search-terminating marker (ID 999) are provided in Table 5.

Table 4. Overall statistics computed across 13 independent Webots runs.

Metric	Mean	SD	Min	Max
Total mission duration (s)	282.39	6.23	278.0	302.0
Mean detection time (s)	3.35	0.11	3.17	3.50
Mean alignment time (s)	3.44	0.15	3.17	3.67
Mean approach time (s)	19.44	0.43	18.50	20.00
Mean per-target visit time (s)	37.27	0.70	35.67	38.83
Mean alignment error (px)	3.06	0.51	2.50	3.83
Total duration of 6 regular visits (s)	223.62	4.17	214.0	233.0
Parking (999) sequence duration (s)	35.54	1.71	31.0	38.0

Table 5. Target-level mean timing and accuracy statistics (n = 13 runs per target).

ID	n	Detection (s)	Alignment (s)	Approach (s)	Visit (s)	Error (px)	Distance (m)
0	13	2.69	3.38	21.85	40.54	3.00	5.24
1	13	3.77	3.23	15.85	30.00	3.23	2.90
2	13	2.38	3.62	23.08	43.62	2.23	5.63
3	13	2.23	3.38	17.92	33.92	3.46	3.69
4	13	3.38	3.62	20.08	39.62	3.23	4.76
5	13	5.62	3.38	17.85	35.92	3.23	3.62
999	13	3.62	3.31	15.77	35.54	2.69	2.90

Target-based analysis reveals that the per-target visit time exhibits a positive correlation with the initial distance to the target: the most distant markers (ID 0, mean 5.24 m; ID 2, mean 5.63 m) yielded the longest visit times (40.54 s and 43.62 s, respectively). In contrast, the closest marker (ID 1, 2.90 m) was the fastest to be fully visited (30.00 s). However, this correlation is not strictly monotonic, as variations in the robot's initial heading orientation at the start of each search cycle introduce minor deviations in detection and alignment times. The system maintained the alignment error within a narrow band (mean 3.06 ± 0.51 pixels) regardless of the target distance.

3.3. Comparative Analysis with State-of-the-Art Systems

A comparison of Indabot with prominent autonomous phenotyping platforms reported in the recent literature is presented in Table 6. While PhenoRob-F [9] offers deep learning-based feature extraction and BoniRob [7] employs multi-sensor fusion with high-resolution imaging, both platforms primarily target field deployments or structured potted-plant phenotyping with predefined navigation paths and entail substantially higher hardware costs. Indabot, in contrast, addresses a hitherto underserved niche by providing a low-cost, GPS-independent, indoor-specific phenotyping robot that leverages fiducial markers for plant-level identification. To enable a common basis for comparison, throughput is expressed as the number of targets processed per hour. Based on the mean mission completion time (282.4 ± 6.2 s for 6 phenotyping targets), Indabot achieves an operational throughput of approximately 76.5 targets per hour. For comparison, PhenoRob-F has been reported to process up to 1875 potted plants per hour under its potted-plant phenotyping mode [9]. This order-of-magnitude difference reflects the differing operational scopes of the two systems: PhenoRob-F performs rapid imaging of

plants at fixed, pre-mapped positions, whereas Indabot autonomously searches for, approaches, and visually centers on each target within an unstructured arena, a task that inherently requires additional navigation and alignment time.

Table 6. Comparative overview of the Indabot platform against state-of-the-art autonomous phenotyping robots.

Feature	Indabot (This Study)	PhenoRob-F	BoniRob	Thorvald II
Target environment	Indoor/greenhouse use	Field + potted plants	Open field	Mixed
Localization	ArUco + LiDAR	Predefined path + rail	GPS + IMU	GPS + Laser
Plant identification	Fiducial marker (unique ID)	Deep learning	Spectral	Row counting
Computing unit	Jetson Orin NX	Industrial PC	Embedded PC	Embedded PC
GPS dependency	None	Partial (field only)	Required	Required
Simulation	Webots R2023b	Not reported	Not reported	Gazebo
Estimated cost	Low–Medium	High	High	Medium–High
Modular design	Yes (Sigma profile)	Partial	Yes	Yes
Throughput	~76.5 targets/h (47.1 s/target, mission-based)	~1875 plants/h (potted-plant scenario) [9]	Not reported	Not reported

3.4. Limitations and Discussion

The experimentally tuned gain parameters yield response characteristics that are well suited to the low-speed, structured greenhouse environment in which dynamic obstacles are absent and marker locations are predictable. Compared with SLAM, which entails high computational cost and is susceptible to cumulative drift during long-term operation, the ArUco marker-based plant identification approach provides absolute and drift-free identification at a negligible computational expense. The limitations of this study are as follows: (i) the simulation environment provides idealized conditions characterized by uniform illumination, the absence of dynamic obstacles, and noise-free sensor models; factors such as natural illumination variability, reflections from greenhouse glazing, dust accumulation, and partial occlusion by foliage will require explicit mitigation strategies in physical deployment; (ii) the encoder-based dead reckoning used during turning maneuvers is susceptible to wheel slip and may lead to cumulative angular drift over long operational sequences; (iii) the system performs image capture but does not yet incorporate automated phenotypic trait extraction; and (iv) comprehensive validation on the physical platform will be reported in a subsequent publication.

To help benchmark the expected sim-to-real performance gap, comparable Gazebo-based greenhouse

robot studies have reported successful transfer of simulation-validated navigation algorithms to physical hardware with only minimal additional tuning [19], while a recent review of the sim-to-real "reality gap" in robotics indicates that performance degradation upon physical deployment is common but can be substantially mitigated through techniques such as domain randomization and closely matched physical parameters [20]; these findings provide a qualitative reference point for the physical validation of Indabot planned for future work.

4. CONCLUSIONS

This study has presented the Indabot platform for autonomous wheat phenotyping under ArUco marker guidance in controlled indoor environments. The system integrates an ODrive 3.6 BLDC driver, hoverboard hub motors, a 42 V LiFePO4 battery, an NVIDIA Jetson Orin NX, a Luxonis OAK-D Lite camera, and a YDLidar X4 LiDAR on a modular sigma-profile chassis, combining OpenCV-based ArUco detection, discrete-time PID control, and LiDAR distance feedback within a Python-based software framework.

Across 13 independent Webots runs, the system successfully visited all targets (6 phenotyping missions + 1 parking maneuver) with a 100% success rate, achieving a mission completion time of 282.4 ± 6.2 s (CV = 2.2%), a mean per-target approach time of 19.4 ± 0.4 s, and a marker centering error of 3.06 ± 0.51 pixels.

The target-based analysis reveals that the approach time exhibits a positive correlation with the initial distance to the target, while the system maintains a consistent pixel-level accuracy that is independent of range. The comparative analysis demonstrates that Indabot fills a distinct niche in low-cost, GPS-independent, fiducial marker-based indoor phenotyping. Future work on Indabot will proceed along the following directions: (i) deployment of the developed algorithms on the physical Indabot platform within an operational greenhouse facility and validation under real-world conditions; (ii) automated extraction of quantitative morphological traits—such as plant height, leaf area, and growth rate—from temporal image sequences; (iii) correction of the cumulative angular and positional drift incurred during long-term operation through visual odometry or ArUco-aided SLAM methods; (iv) development of multi-robot coordination strategies to enable the simultaneous phenotyping of large plant populations; and (v) quantitative evaluation, through controlled greenhouse experiments, of the labor savings, data throughput, and measurement consistency gains afforded by the autonomous system relative to manual methods.

Declaration of Ethical Standards

This study is based entirely on simulation-generated data and did not involve human participants, human tissue,

animals, or patient-specific records. Therefore, ethical approval and informed consent were not required for this work.

Credit Authorship Contribution Statement

The conceptualization, software development, simulation, and data curation were carried out by Fikret Uysal. The formal analysis and validation of results were performed by Kadir Sabanci. The original draft was written by Fikret Uysal; review, editing, and supervision were undertaken by Kadir Sabanci.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding / Acknowledgements

This work was supported by the Specialization Coordination Office for Precision Agriculture Applications and Innovative Processing Technologies (HASTUYİT) at Karamanoğlu Mehmetbey University, under project number 01-HASTU-DR-24.

Data Availability

The simulation scenario is publicly available at <https://doi.org/10.5281/zenodo.21372809>; additional resources and datasets are available from the corresponding author upon reasonable request.

References

- [1] B. Shiferaw et al., "Crops that feed the world 10. Past successes and future challenges to the role played by wheat in global food security," *Food Security*, vol. 5, pp. 291–317, 2013, doi: 10.1007/s12571-013-0263-y.
- [2] J. L. Araus and J. E. Cairns, "Field high-throughput phenotyping: The new crop breeding frontier," *Trends in Plant Science*, vol. 19, no. 1, pp. 52–61, 2014, doi: 10.1016/j.tplants.2013.09.008.
- [3] A. Atefi et al., "Robotic technologies for high-throughput plant phenotyping: Contemporary reviews and future perspectives," *Frontiers in Plant Science*, vol. 12, p. 611940, 2021, doi: 10.3389/fpls.2021.611940.
- [4] R. Xu and C. Li, "A Review of High-Throughput Field Phenotyping Systems: Focusing on Ground Robots," *Plant Phenomics*, vol. 2022, p. 9760269, 2022. doi: 10.34133/2022/9760269
- [5] R. R. Shamshiri et al., "Advances in greenhouse automation and controlled environment agriculture: A transition to plant factories and urban agriculture," *International Journal of Agricultural and Biological Engineering*, vol. 11, no. 1, pp. 1–22, 2018, doi: 10.25165/j.ijabe.20181101.3210.
- [6] C. Zhao et al., "Crop phenomics: Current status and perspectives," *Frontiers in Plant Science*, vol. 10, p. 714, 2019, doi: 10.3389/fpls.2019.00714.
- [7] A. Ruckelshausen, P. Biber, M. Doma, H. Gremmes, R. Klose, A. Linz, F. Rahe, R. Resch, M. Thiel, D. Trautz, and U. Weiss, "BoniRob – An autonomous field robot platform for individual plant phenotyping," in *Precision Agriculture '09: Papers Presented at the 7th European Conference on Precision Agriculture*, Wageningen Academic Publishers, 2009, pp. 841–847, doi: 10.1163/9789086866649_101
- [8] L. Grimstad and P. J. From, "Thorvald II – A modular and re-configurable agricultural robot," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4588–4593, 2017, doi: 10.1016/j.ifacol.2017.08.1005.

- [9] M. Yang, Z. Li, J. Cui et al., "PhenoRob-F: An autonomous ground-based robot for high-throughput phenotyping of field crops," *Plant Phenomics*, vol. 7, p. 100085, 2025, doi: 10.1016/j.plaphe.2025.100085
- [10] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014, doi: 10.1016/j.patcog.2014.01.005.
- [11] M. Kalaitzakis et al., "Fiducial markers for pose estimation: Overview, applications and experimental comparison of the ARTag, AprilTag, ArUco and STag markers," *Journal of Intelligent & Robotic Systems*, vol. 101, p. 71, 2021, doi: 10.1007/s10846-020-01307-9.
- [12] R. Adámek, M. Brable, P. Vávra, B. Dobossy, M. Formánek, and F. Radil, "Analytical Models for Pose Estimate Variance of Planar Fiducial Markers for Mobile Robot Localisation," *Sensors*, vol. 23, no. 12, p. 5746, 2023. doi: 10.3390/s23125746
- [13] P. Oščádal, D. Heczko, A. Vysocký, J. Mlotek, P. Novák, I. Virgala, M. Sukop, and Z. Bobovský, "Improved Pose Estimation of ArUco Tags Using a Novel 3D Placement Strategy," *Sensors*, vol. 20, no. 17, p. 4825, 2020, doi: 10.3390/s20174825
- [14] S. Hinderer, M. Scheffler, and B. Yang, "Investigation of ArUco Marker Placement for Planar Indoor Localization," in *Proc. IEEE 22nd International Conference on Advanced Robotics (ICAR)*, 2025, doi: 10.1109/ICAR65334.2025.11338671.
- [15] L. He, Y. Sun, L. Chen, Q. Feng, Y. Li, J. Lin, Y. Qiao, and C. Zhao, "Advance on Agricultural Robot Hand-Eye Coordination for Agronomic Task: A Review," *Engineering*, vol. 51, no. 8, pp. 263–279, 2025. doi: 10.1016/j.eng.2025.01.022
- [16] A. Farley, J. Wang, and J. A. Marshall, "How to pick a mobile robot simulator: A quantitative comparison of CoppeliaSim, Gazebo, MORSE and Webots with a focus on accuracy of motion," *Simulation Modelling Practice and Theory*, vol. 120, p. 102629, 2022, doi: 10.1016/j.simpat.2022.102629.
- [17] O. Michel, "Cyberbotics Ltd. Webots™: Professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004, doi: 10.5772/5618.
- [18] F. Uysal, "Navigation Scenario: Rotation-Based Random Search," *Zenodo*, 2026. [Online]. Available: <https://doi.org/10.5281/zenodo.21372809>.
- [19] S. Narvekar, S. Atar, V. Gupta, L. Penubaku, and K. Arya, "agriFrame: Agricultural framework to remotely control a rover inside a greenhouse environment," *arXiv preprint arXiv:2504.09079*, 2025, doi: 10.48550/arXiv.2504.09079.
- [20] E. Aljalbout, J. Xing, A. Romero, I. Akinola, C. R. Garrett, E. Heiden, A. Gupta, T. Hermans, Y. Narang, D. Fox, D. Scaramuzza, and F. Ramos, "The Reality Gap in Robotics: Challenges, Solutions, and Best Practices," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 9, pp. 403–432, 2026, doi: 10.1146/annurev-control-031924-100130.

*Research Article***C-PPDSS: A Comprehensive Patient Personal Data Sovereignty System for Real-Time PII Redaction in Healthcare LLM Workflows**

**Edoghogho Olaye^a , Williams O. Aigbe^b , David Onwuka^{a,*} , Daniel Obuh^a ,
Amarachi A. Madukwe^a**

^a University of Benin, Edo State, Nigeria

^b Chemnitz University of Technology, Germany and Fraunhofer-Institute for Electronic Nano Systems (ENAS) Chemnitz, Germany

ARTICLE INFO

Article history:

Received 23 May 2026

Accepted 28 August 2026

Keywords:

C-PPDSS,
Healthcare AI,
Large Language Models,
Patient Data Privacy,
PII Redaction,
PIIRANHA

ABSTRACT

Health surveillance in modern healthcare increasingly relies on AI, raising concerns about the exposure of patients' identifiable information (PII) in automated workflows. This paper presents the Comprehensive Patient Personal Data Sovereignty System (CPPDSS), a privacy-preserving framework designed to enforce real-time PII redaction during outbound interactions with large language models. C-PPDSS employs a fine-tuned Custom PIIRANHA model, based on Microsoft's DeBERTa-v2 architecture, to intercept and sanitize HTTP POST requests containing sensitive healthcare data before they reach external LLM services. Evaluated on a held-out test set of 20,927 annotated examples, Custom PIIRANHA achieved 99.51% token-level accuracy and a micro-averaged F1 score of 95.45%, outperforming a fine-tuned Distilled BERT baseline across all major metrics. Quantitative benchmarking demonstrated a mean inference latency of 229.66 ms per request, a model memory footprint of 305.7 MB, and a sustained throughput of 4.35 requests per second on CPU-only hardware, confirming suitability for real-time deployment on modest intranet infrastructure. By securing patient information at the network level, C-PPDSS bridges a gap left by previous static-document-focused systems, enabling safe AI integration in healthcare workflows. Its architecture ensures reproducibility and feasibility for deployment in hospital intranets or standard commodity hardware setups.

This is an open access article under the CC BY-SA 4.0 license.
(<https://creativecommons.org/licenses/by-sa/4.0/>)

1. INTRODUCTION

From 24 May 2018, the General Data Protection Regulation (GDPR) of the European Union aims to protect individuals' privacy and data security which have a link to the Union's territory or market. This was demonstrated by the European Union's implementation of the comprehensive Data Protection Regulation. It mandates all Data Protection Authorities (DPAs) with the task to ensure companies clearly outline user agreements and forbids them from misleading or pressuring users into surrendering their privacy rights, and in the case of failure to adhere, they impose sanctions [1]. Individuals' data privacy and security become a concern that must be addressed, especially in this era of artificial intelligence (AI) [2], [3]. The adoption of AI in the healthcare sector

has rapidly improved disease prediction and diagnosis, treatment optimization, and overall health management. AI-powered health surveillance systems promise to enhance healthcare delivery efficiency, save lives, and reduce costs [4], [5].

Despite these opportunities and unprecedented capabilities, AI adoption in healthcare has also raised profound challenges around data privacy, equity, and accountability. As the healthcare sector increasingly relies on AI to monitor, predict, and diagnose patients' health, the absence of robust frameworks for digital sovereignty risks exposing sensitive health data to cross-border flows that may override national regulations and compromise privacy [6]. By "digital sovereignty," we follow [7] in treating the concept not merely as a descriptive label for

* Corresponding author. E-mail address: david.onwuka@eng.uniben.edu
DOI: 10.58190/ijamec.2026.178

actors exerting control over digital technologies, but as a normative framework centered on authority. This emphasizes that sovereignty in the digital domain requires legitimate control that is assessed through participation, accountability, consent and the ability to serve public needs and protect rights. Such an approach is particularly relevant to healthcare adopting AI-powered systems, where control must also be evaluated against standards of fairness and trust.

Framing the issue in this way raises critical questions: Who ultimately controls these systems? How are data collected, extracted, loaded, and transformed before, during, and after analysis?

Researchers have explored the challenges of data protection and privacy that result from the integration of AI in healthcare, highlighting risks of data breaches, algorithmic bias, and erosion of fundamental patient rights [8], [9]. To keep data localized, federated learning approaches have been proposed, yet these often lack end-to-end governance models that ensure true data sovereignty [10], [11].

In light of these challenges, this work seeks to address the following research questions:

How can sensitive health data be intercepted and protected before transmission to external AI or LLM systems?

Can privacy-preserving data redaction be performed in real time within healthcare workflows without disrupting clinical efficiency?

What governance and architectural frameworks can ensure that data sovereignty is preserved at both the institutional and national levels in AI-powered health surveillance systems?

By investigating these questions, this paper argues that the design and governance of AI-powered health technologies must prioritize innovation in ethical frameworks and controls that reconcile competing priorities of privacy, sovereignty, and equity to avoid eroding fundamental rights.

2. LITERATURE REVIEW

The integration of AI in healthcare raises urgent concerns around data privacy and sovereignty. While AI enhances diagnostic accuracy and treatment planning, it poses significant risks to patient confidentiality and data security [12]. Differential privacy and federated learning emerge as potential solutions to preserve patient privacy while utilizing healthcare data effectively [9]. The need for robust governance and risk management strategies is emphasized, with proposals like the 'Integrated Security and Ethics Model' addressing the dynamic nature of AI in healthcare [13]. Recent research has explored various approaches to ensure data security and sovereignty in healthcare surveillance. Patient-centric platforms using

proxy re-encryption allow patients to control access to their health records, enhancing data sovereignty [14]. Edge of Things (EoT) computing frameworks employing fully homomorphic encryption offer secure, real-time processing of IoT-generated patient data while preserving privacy [15]. Existing security measures include encryption, access controls, and secure authentication mechanisms, with a focus on compliance with healthcare acts like HIPAA [16]. Emerging technologies such as blockchain and artificial intelligence are being explored to fortify data security in healthcare [17], [18], [19]. [20] proposed a model combining Fuzzy C-means Clustering and Support Vector Machines for improved cloud security in surveillance, achieving 94.4% accuracy. [21] reviewed AI-driven cybersecurity in e-health systems, emphasizing AI's role in continuous monitoring, threat detection, and response. They also discussed future research directions, including quantum-resistant encryption and privacy-preserving AI techniques. [22] examined various AI and ML methods for healthcare system security, such as anomaly detection and risk prediction. The study evaluated different algorithms, including neural networks and decision trees, using metrics like accuracy and precision to assess their effectiveness in protecting patient data and IT networks from cyber threats. [6] developed a system that utilized an optical character recognition powered by AI to obfuscate personal identifiable patient's data capture before sending to the machine learning models for training. These papers collectively underscore the potential of AI and ML in bolstering security measures for sensitive data in healthcare and surveillance contexts. However, the protection of patient data before integration into large language models (LLMs) has received little scholarly attention. Our work involves the development of a Comprehensive Patient Personal Data Sovereignty System (C-PPDSS) in order to ensure on-premise personally identifiable information interception. Specifically, when healthcare practitioners use large language models (LLMs), our system ensures that the input received by the LLMs contains no personally identifiable patient information. Unlike approaches that focus primarily on data protection during the training stage, our work addresses the usage stage, ensuring that sensitive patient data are intercepted and obfuscated before being transmitted to the model for inference.

3. METHODOLOGY

The Comprehensive Patient Personal Data Sovereignty System (C-PPDSS) was designed to ensure that no personally identifiable information (PII) leaves the local environment when healthcare professionals interact with external large language models (LLMs). The system detects and filters personally identifiable information locally before data leaves the hospital, ensuring

compliance with regulations and preserving digital sovereignty. Relying on external personally identifiable information processing services was rejected due to risks of data misuse. Instead of using large, general-purpose LLMs like Mistral, the team developed a lightweight, fine-tuned model tailored to efficiently identify and replace healthcare-related personally identifiable information with placeholders.

3.1. System Design

The system was architected as a modular pipeline consisting of three primary components:

Request interception

Personally identifiable information filtration

Request transformation

The overall architecture of the Comprehensive Patient Personal Data Sovereignty System (C-PPDSS) is shown in Figure 1, which provides a high-level view of how these components interact. The architecture illustrates the flow of data from outbound requests, through interception and PII redaction, to transformation and forwarding to external large language models (LLMs). This diagram serves as a reference map for the subsections that follow, where each module of the pipeline is described in detail.

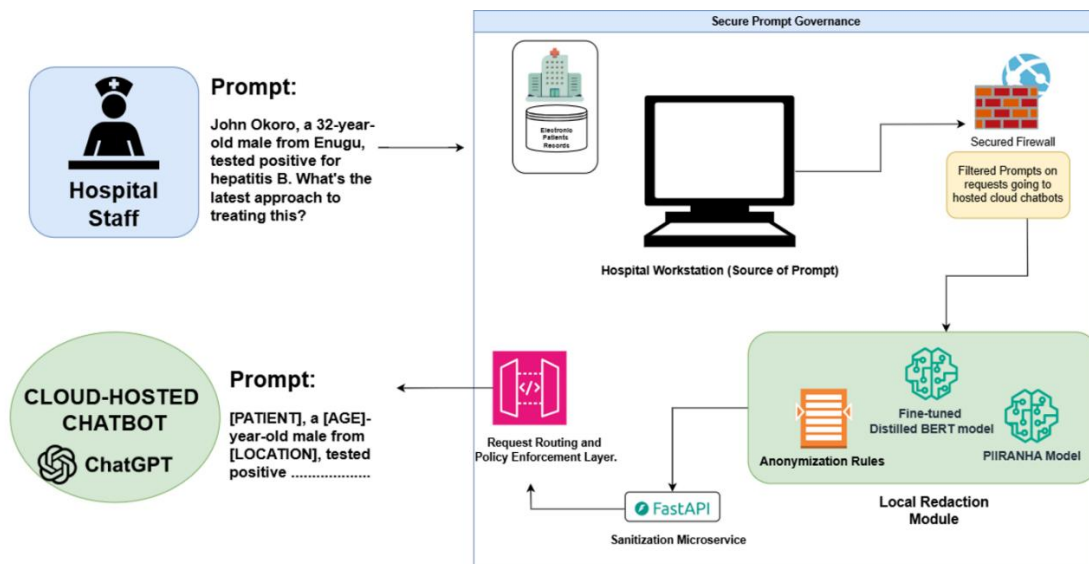


Figure 1. Overview of the C-PPDSS Architecture and Data Flow Pipeline

3.1.1. Request Interception

A system-level monitoring mechanism was implemented to detect outbound HTTP POST requests directed at OpenAI's ChatGPT service. Using browser developer tools, we first inspected the structure of these requests, confirming that user prompts were transmitted as JSON payloads within HTTPS POST calls to `api.openai.com/v1/chat/completions` as shown in Figures 2 and 3. To automate this process, we deployed a local HTTPS proxy to capture traffic at the network boundary. This enabled us to log, parse, and analyze the outbound request body in real time, providing visibility into exactly what information—potentially including PII—was leaving the hospital environment. The captured request structure formed the basis for the subsequent redaction module, which intercepts and replaces sensitive fields before forwarding the sanitized query to the LLM.

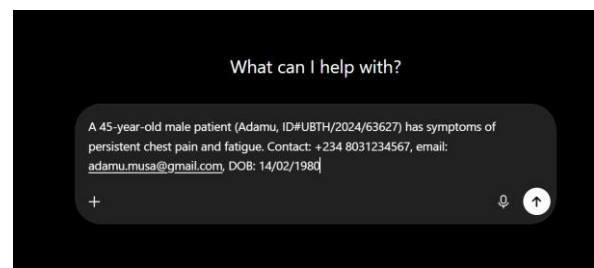


Figure 2. Sample ChatGPT web interface input containing potential patient PII

```

{
  "timestamp": "2026-08-11T13:52:02.715",
  "request": {
    "url": "https://chatgpt.com/backend-api/conversation",
    "headers": {
      ...
    },
    "content_type": "application/json",
    "body": {
      "action": "next",
      "messages": [
        {
          "id": "c3134bff-bf6b-4e02-bac1-84beae1e07e",
          "author": {
            "role": "user"
          },
          "create_time": 1746194852.685,
          "content": {
            "content_type": "text",
            "parts": [
              "A 45-year-old male patient (Adamu, ID#BIRTH/2024/63627) has symptoms of persistent chest pain. contacts: +234 8031234567, email: adamu.musa@gmail.com, DOB: 14/02/1980."
            ]
          },
          "metadata": {
            "selected_github_repos": [],
            "selected_all_github_repos": false,
            "serialization_metadata": {
              "custom_symbol_offsets": []
            },
            "dictation": false
          }
        ]
      }
    }
  }
}

```

Figure 3. System Log Entry showing a Captured HTTP POST Request containing a User Prompt

3.1.2. PII Filtration

Both candidate models were fine-tuned on the AI4Privacy PII-Masking-200k dataset (AI4Privacy, 2022), a publicly available Hugging Face dataset containing over 200,000 synthetically generated sentences annotated with personally identifiable information, including names, addresses, phone numbers, and identification codes. All data used across both fine-tuning procedures was either publicly sourced or synthetically generated, with no real patient data collected or processed at any stage.

The dataset was partitioned using a fixed random seed (seed = 42) into three non-overlapping splits, as shown in Table 1.

Table 1. Dataset Split Configuration

Split	Examples	Proportion
Training	167,416	80%
Validation	20,927	10%
Test	20,927	10%
Total	209,270	100%

The test set contains 56 distinct PII label types. Of these, 14 categories, GIVENNAME, SURNAME, EMAIL, TELEPHONENUM, ACCOUNTNUM, SOCIALNUM, PASSWORD, USERNAME, STREET, CITY, ZIPCODE, BUILDINGNUM, CREDITCARDNUMBER, and DATEOFBIRTH, fall within PIIRANHA's detection taxonomy and were used for evaluation scoring in Section 4. The remaining 42 label types (e.g., BITCOINADDRESS, USERAGENT, JOBTYP) lie outside PIIRANHA's classification scope and were excluded from scoring accordingly.

Both models were fine-tuned on a single personal computer to emphasize reproducibility on accessible, non-specialized hardware. The specifications of the shared fine-tuning environment are provided in Table 2.

Table 2. Specifications for Model Fine-tuning

Component	Specification	Purpose
CPU	Intel Core i5 (4 cores)	Parallel processing
GPU	—	Accelerated fine-tuning (not available)
RAM	8GB DDR4	Handling large datasets
Storage	256GB	Dataset and model checkpoint storage
Framework	PyTorch 2.13, Hugging Face Transformers	Model architecture and training

3.1.3. Fine-tuned Distilled BERT-Based Model

A distilled BERT model was fine-tuned on the dataset and split described above. To adapt the dataset to healthcare-specific contexts, additional annotations were curated by our team to capture domain-specific identifiers, including prescription codes and medical record numbers. Training required 78 hours of compute time under the environment specified in Table 2, following the configuration in Table 3.

Table 3. Distilled BERT-Based Model Fine-Tuning Configuration

Parameter	Value
Base model	distilbert-base-uncased (66M parameters)
Tokenizer	BERT WordPiece tokenizer
Maximum sequence length	512 tokens
Number of epochs	4
Batch size	16
Gradient accumulation steps	1 (effective batch size = 16)
Learning rate	3×10^{-5}
Learning rate scheduler	Linear decay (no warm-up)
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$)
Random seed	42
Training framework	PyTorch 2, Hugging Face Transformers
Training duration	Approximately 78 hours (CPU-only)

The model demonstrated high precision in detecting standard PII categories, such as email addresses, phone numbers, and medical record numbers. It exhibited signs of overfitting, however, suggesting that further optimization or larger compute resources would be required for production-grade deployment.

3.1.4. PIIRANHA Model with Region-Specific Enhancements

To address the limitations of the fine-tuned distilled BERT model, we adopted PIIRANHA, a model built on Microsoft's DeBERTa (Decoding-enhanced BERT with disentangled attention) [23]. PIIRANHA was further fine-tuned on a supplementary, region-specific synthetic dataset constructed to reflect Nigerian healthcare contexts, described below.

3.1.5. Region-Specific Dataset Construction

A supplementary dataset of 900 examples was generated programmatically using a template-based slot-

filling approach; no real patient data was collected or used. The templates represented common clinical scenarios, including hospital admissions, referral letters, and discharge summaries. These templates were populated with randomly sampled values from a curated list of 312 common Nigerian given names and 287 surnames, Nigerian telephone number formats (+234 XXXXXXXXXX), hospital medical record number formats following regional conventions (e.g., UBTH/YYYY/NNNN), and Nigerian street addresses and city names. Entity annotations were generated automatically from each template's slot positions, requiring no manual labelling. Generated entities were mapped to PIIRANHA's label taxonomy: patient names to GIVENNAME and SURNAME, hospital record numbers to ACCOUNTNUM, telephone numbers to TELEPHONENUM, and email addresses to EMAIL. The dataset was split into 720 training, 90 validation, and 90 test examples.

3.1.6. Fine-Tuning Configuration

PIIRANHA was fine-tuned on a combination of a sampled subset of the AI4Privacy training split and the region-specific dataset described above, under the configuration in Table 4.

Table 4. Custom PIIRANHA Fine-Tuning Configuration

Parameter	Value
Base model	iiiorg/piiranha-v1-detect-personal-information (DeBERTa-v2, ~0.3B parameters)
Tokenizer	DeBERTa-v2 SentencePiece tokenizer
Maximum sequence length	512 tokens
Number of epochs	3
Per-device batch size	1
Gradient accumulation steps	8 (effective batch size = 8)
Learning rate	2×10^{-5}
Learning rate scheduler	Linear decay with warm-up
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$)
Random seed	42
Training framework	PyTorch 2.13, Hugging Face Transformers
Training duration	Approximately 8 hours (CPU-only)

Both models were evaluated using standard classification metrics including precision, recall, and F1-score, on the withheld test set described in Table 1. These metrics were chosen due to their relevance in assessing both the accuracy and completeness of PII detection.

3.1.7. Request Transformation

After PII detection, the system applies a transformation stage that replaces sensitive entities with structured placeholders (e.g., replacing a patient's name with "[PATIENT]" or a medical record number with "[RECORD_ID]"). This substitution is performed at the token level to preserve the syntactic and semantic integrity of the request while ensuring that no identifiable information leaves the local environment.

To maintain consistency, identical entities within a

request are replaced with the same placeholder token. The transformed request is then reconstructed into its original JSON schema and forwarded to the target LLM endpoint. This design ensures that the LLM receives a valid, well-structured request without knowledge of the underlying sensitive attributes.

The response from the LLM is routed back through the system and delivered to the user without modification, creating the illusion of direct interaction while ensuring that all outbound communications remain free of personally identifiable information.

3.2. Model Evaluation and Deployment

Both models were evaluated on a held-out test set to determine their effectiveness in redacting personally identifiable information (PII). The evaluation process employed four key performance metrics commonly used in natural language processing tasks: Accuracy, Precision, Recall, and F1 Score. These metrics provided a balanced assessment of each model's ability to correctly identify and redact PII while minimizing false positives and false negatives.

To ensure fairness and reproducibility, the same test set was applied to both models. This enabled a direct comparison of their strengths and weaknesses in the healthcare-specific domain. Following evaluation, the deployment decision was guided by the superior performance of the Custom PIIRANHA model across accuracy, precision, recall, and F1 score. The deployment pipeline consisted of:

Containerization of the model into a lightweight runtime environment.

Serving the model as a RESTful API using FastAPI, selected for its ability to support low-latency, high-performance machine learning inference.

Integration of the API as an internal microservice within the request interception pipeline, enabling real-time PII redaction before prompts reach external large language models (LLMs).

4. RESULTS

4.1. Model Evaluation and Comparison

The Comprehensive Patient Personal Data Sovereignty System (C-PPDSS) was evaluated using two machine learning models: a fine-tuned Distilled BERT and a custom PIIRANHA model (based on Microsoft's DeBERTa). Both models were tested on the same held-out dataset to ensure fairness and reproducibility.

Table 5. Model Evaluation of the Fine-Tuned Distilled BERT and the Custom PIIRANHA

Metric	Fine-Tuned Distilled BERT	Custom PIIRANHA
Accuracy	94.9%	99.51%
Precision	90.85%	94.93%
Recall	91.36%	95.98%
F1 Score	92.60%	95.45%

The Custom PIIRANHA model outperformed the fine-tuned Distilled BERT across all major evaluation metrics, with a 4.61% increase in accuracy and a 2.85-point improvement in F1 score. These gains reflect a better balance between precision and recall, and motivated the choice of PIIRANHA as the model carried forward into deployment.

To understand where this improvement came from, we examined PIIRANHA's performance at the level of individual PII categories.

Table 6. Per-Category Token-Level Performance of the Custom PIIRANHA Model

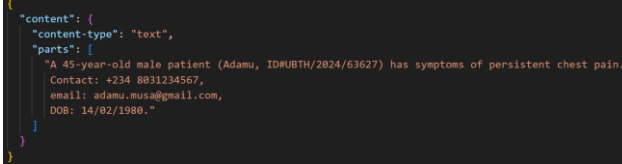
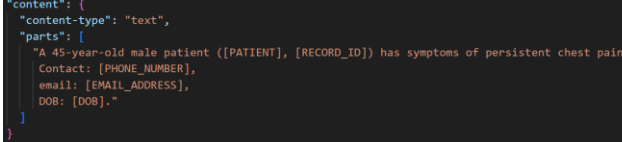
PII Category	Precision	Recall	F1-Score	Support (Tokens)
EMAIL	0.9960	1.0000	0.9980	1,245
SOCIALNUM	0.9920	1.0000	0.9960	374
TELEPHONENUM	1.0000	0.9898	0.9949	488
USERNAME	0.9912	0.9978	0.9945	449
PASSWORD	0.9885	1.0000	0.9942	1,118
STREET	0.9774	0.9774	0.9774	265
ACCOUNTNUM (MRN)	0.9468	0.9944	0.9700	179
ZIPCODE	0.9356	0.9561	0.9458	228
CITY	0.9447	0.9447	0.9447	217
GIVENNAME	0.9156	0.9632	0.9388	653
SURNAME	0.9050	0.9241	0.9144	237
CREDITCARDNUMBER	0.8714	0.9590	0.9131	537
BUILDINGNUM	0.8824	0.9000	0.8911	100
DATEOFBIRTH	0.7104	0.6068	0.6545	384
Micro-Average	0.9493	0.9598	0.9545	5,644
Macro-Average	0.9262	0.9442	0.9377	5,644
Accuracy (All Tokens)	—	—	0.9951	112,063

Structured alphanumeric PII categories including EMAIL, TELEPHONENUM, SOCIALNUM, PASSWORD, and USERNAME, achieved F1-scores of 0.994 or higher, reflecting the model's accuracy on entities with well-defined lexical patterns. ACCOUNTNUM, used

as a proxy for hospital medical record numbers, achieved an F1-score of 0.9700 with a recall of 0.9944, confirming reliable detection of healthcare-specific patient identifiers — the category most directly relevant to clinical deployment.

4.2. Real-Time Redaction Verification

Following this evaluation, the PIIRANHA model was deployed in the final system. It was containerized and served via a RESTful API using FastAPI, providing low-latency inference suitable for real-time request interception. Integrated as an internal microservice, the model successfully redacted sensitive information from outgoing HTTP POST requests before they reached external LLMs, confirming that the system could operate in real-time healthcare-specific use cases without introducing noticeable delays.


Figure 4. System log entry showing unredacted PII in an outbound request before C-PPDSS processing

Figure 5. System log entry showing redacted placeholders in an outbound request after C-PPDSS processing

4.3. System Performance Benchmarks

With redaction accuracy and functional correctness established, the deployed C-PPDSS microservice was evaluated for real-time operational suitability using 200 HTTP POST requests containing synthetic clinical prompts (mean prompt length: 154 characters; mean PII entities per prompt: 3.8), benchmarked on the hardware specification described in Table 2.

Table 7. C-PPDSS System Performance Benchmark Results

Metric	Measured Value
Model load time	2.50 s
Model RAM footprint	305.7 MB
Total process memory (runtime)	1,116.7 MB (~1.1 GB)
Mean inference latency	229.66 ms
Median latency (p50)	217.00 ms
95th percentile latency (p95)	325.69 ms
99th percentile latency (p99)	344.23 ms
Min / Max latency	153.63 ms / 368.76 ms
Throughput	4.35 requests/s
CPU utilization (mean)	196.6% (~2 of 4 cores)

The mean inference latency was 229.66 ms per request, with the 95th and 99th percentile latencies remaining below 350 ms. Given that typical LLM responses take between 2 and 10 seconds, the redaction process adds only a small fraction of the overall response time and is unlikely to be perceptible to end users.

Resource utilization stayed within the capabilities of the deployment hardware. The PIIRANHA model occupied 305.7 MB of memory after loading, while total runtime process memory averaged approximately 1.1 GB — well below the available 8 GB of system memory. Average CPU utilization was 196.6%, corresponding to roughly two fully utilized cores under psutil per-core accounting, indicating efficient multi-threaded execution without saturating the host system. Together, these measurements support low-latency operation and low resource consumption for the deployed C-PPDSS.

5. DISCUSSION

5.1. Interpretation of Model Performance

The evaluation of the two models — fine-tuned Distilled BERT and the custom PIIRANHA — points to clear differences in their suitability for healthcare-specific PII redaction. While Distilled BERT achieved a strong accuracy of 94.9% and an F1 score of 92.60%, PIIRANHA outperformed it, reaching an accuracy of 99.51% and an F1 score of 95.45%. The improved recall (95.98%) of the Custom PIIRANHA carries particular weight in this setting: healthcare applications demand a conservative approach, where missing a sensitive identifier can pose serious risks to patient privacy.

5.2. Answering the Research Questions

These findings map directly onto the three research questions posed at the outset of this study. First, the local HTTPS interception mechanism demonstrated that sensitive healthcare information can be identified and replaced before transmission to external LLM services. Second, the measured mean inference latency of 229.66 ms confirms that privacy-preserving redaction can be

performed in real time without introducing noticeable delays during clinical AI interactions. Third, by ensuring that all PII detection and substitution occur within the local deployment environment before requests are forwarded, the proposed architecture supports institutional data sovereignty while still allowing healthcare practitioners to benefit from external AI services.

5.3. Positioning Relative to Existing Work

Compared to existing systems like the original Patient Personal Data Sovereignty System (PPDSS) [6], which focused on static document sanitization, C-PPDSS introduces a paradigm shift by enforcing privacy at the network level. By intercepting outbound HTTP POST requests in real time, the system closes a gap that prior work left largely unaddressed: conversational prompts sent to large language model (LLM) APIs. This matters directly for interactive clinical workflows such as drafting medical notes, referral letters, or clinical summaries, where the sensitive content is generated on the fly rather than stored in a document

Table 8. Systematic Comparison of Privacy-Preservation Approaches for Healthcare AI

Approach	Example Work	Scope	Real-Time	Network-Level	LLM Usage Stage
Static document masking	Olaye et al. [6]	Documents	No	No	Pre-training only
NER-based redaction	C-PPDSS	Conversational	Yes	Yes	Usage/Inference
Federated learning	Anunita et al. [10]	Model training	No	No	Training only
Homomorphic encryption	Alabdulatif et al. [15]	IoT data	Partial	No	Pre-processing
Proxy re-encryption	Rodrigues et al. [14]	Health records	No	No	Storage/Access
Blockchain-based	Bhaskar et al. [17]	Data logging	No	No	Audit trail only

As Table 8 shows, existing privacy-preserving approaches concentrate on document sanitization, secure storage, encrypted computation, or model training. C-PPDSS instead operates at the inference stage, intercepting conversational requests before they leave the local environment. This lets sensitive information be removed at the network boundary without requiring any changes to the external LLM service itself.

5.4. Deployment Feasibility

The deployment architecture reinforces the practicality of this approach. The containerized FastAPI microservice combined low latency with minimal resource consumption, making the system viable on modest hardware or within hospital intranet environments.

Training and deployment were both shown to be feasible on a personal computer setup, without reliance on high-performance clusters or cloud-based infrastructure, which is a meaningful factor for reproducibility and for adoption in resource-constrained healthcare settings.

5.5. Error Analysis

Per-category evaluation (Table 6) shows the model performing consistently across most structured PII categories. EMAIL, TELEPHONENUM, PASSWORD, USERNAME, and SOCIALNUM all achieved F1-scores above 0.99, confirming reliable detection of identifiers with well-defined lexical patterns. ACCOUNTNUM, used as a proxy for hospital medical record numbers, likewise maintained a high recall of 0.9944 following region-specific fine-tuning.

The lowest performance was observed for DATEOFBIRTH ($F1 = 0.6545$). Most false negatives traced back to differences in date formatting: numeric, textual, and ISO representations, were all present in the evaluation data, and because these formats tokenize differently, detection accuracy varied across them. Incorporating date normalization before inference is expected to close much of this gap in future versions of the system.

GIVENNAME and SURNAME also scored lower than the structured identifiers, since name detection depends more on contextual understanding than on fixed lexical patterns. During deployment, this limitation was mitigated through word-level boundary expansion: when any subword token within a name is flagged as PII, the complete word is replaced, preventing partial disclosure of patient names.

5.6. Effect of Region-Specific Fine-Tuning

Including the supplementary Nigerian healthcare dataset improved the model's ability to recognize identifiers common to local clinical practice including hospital medical record number formats, Nigerian telephone prefixes, and common Nigerian personal names among them. The high recall obtained for ACCOUNTNUM suggests that adapting the model with region-specific examples makes it better suited to healthcare deployments where local identifier formats diverge from those represented in general-purpose datasets.

5.7. Multilingual Applicability

The base PIIRANHA model (iiiorg/piiranha-v1-detect-personal-information) supports six languages: English, Spanish, French, German, Italian, and Dutch. The current C-PPDSS evaluation is limited to English-language clinical prompts. Extension to multilingual healthcare settings such as Francophone African health systems, would require language-specific synthetic training data and evaluation sets, which is identified as future work.

5.8. Limitations

Several limitations remain. First, the current implementation is limited to text-based HTTP POST payloads and does not extend to multimodal data such as scanned medical documents, voice recordings, or medical images. Second, although both the AI4Privacy dataset and the supplementary Nigerian dataset are synthetically generated, real-world deployments may surface identifiers, abbreviations, or code-switching patterns that were not represented during training. Third, the system currently targets OpenAI-compatible HTTPS POST requests and does not yet support communication channels such as WebSocket or gRPC. Finally, the underlying DeBERTa architecture processes a maximum of 512 tokens per request, requiring long clinical narratives to be segmented before inference, a constraint that may increase processing time for lengthy documents.

Even with these limitations, the results suggest that lightweight, domain-specific models like PIIRANHA can play a meaningful role in enabling secure AI adoption in healthcare. By balancing accuracy with efficiency, the system offers a practical blueprint for integrating privacy-preserving safeguards into everyday clinical AI workflows without sacrificing usability.

6. CONCLUSION

This work presented the Comprehensive Patient Personal Data Sovereignty System (C-PPDSS), a privacy-preserving framework designed to safeguard sensitive information in healthcare interactions with large language models. By focusing on real-time interception of outbound HTTP POST requests and integrating lightweight yet accurate machine learning models, the system demonstrated a practical approach to enforcing data sovereignty at the point of data generation.

The evaluation of two candidate models highlighted the superiority of the custom PIIRANHA model, which achieved a 99.51% accuracy and an F1 score of 95.45%, outperforming the fine-tuned Distilled BERT baseline. Its favourable balance of recall and resource efficiency makes it particularly well-suited for healthcare contexts, where both privacy and responsiveness are critical.

Beyond its technical contributions, the system addresses a significant gap left by prior privacy-preservation efforts that concentrated on static documents while overlooking conversational AI use cases. By embedding privacy safeguards within the interaction pipeline, C-PPDSS demonstrates how healthcare institutions can confidently integrate AI tools into clinical workflows without compromising patient trust or regulatory compliance.

Future work will explore extending the system to multimodal data, adapting to encrypted communication protocols, and expanding the training corpus to include more diverse healthcare identifiers. These steps will

further strengthen its robustness and scalability.

In conclusion, C-PPDSS establishes a foundation for secure, real-time PII redaction in healthcare AI applications. Its combination of accuracy, efficiency, and architectural simplicity positions it as a viable solution for protecting sensitive data in an era of rapidly expanding AI adoption.

CRedit Authorship Contribution Statement

Edoghogho Olaye: Conceptualization, Methodology, Software. Williams O. Aigbe: Validation, Investigation, Writing. David Onwuka: Software, Formal Analysis, Writing. Daniel Obuh: Software, Data Curation. Amarachi A. Madukwe: Review & Editing.

Declaration of Competing Interest

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Funding

This work was supported by the Tertiary Education Trust Fund National Research Fund grant (TETFund Ref: NRF2023/CC/STI/00031/VOL.1).

Ethical Standards

All data used in this study was either publicly available (AI4Privacy dataset, licensed under Creative Commons) or synthetically generated. No real patient data was collected or used. The study complies with all applicable ethical standards.

Data Availability

The AI4Privacy PII-Masking-200k dataset used in this study is publicly available through Hugging Face. The synthetic region-specific dataset and supporting materials generated for this study are available from the corresponding author upon reasonable request.

Acknowledgments

The authors thank TETFund and the Internet Integrity Initiative for the open-source release of the PIIRANHA model (iiiorg/piiranha-v1-detect-personal-information).

References

- [1] J. P. Albrecht, "How the GDPR Will Change the World," *Eur. Data Prot. Law Rev.*, vol. 2, no. 3, pp. 287–289, 2016, doi: 10.21552/EDPL/2016/3/4.
- [2] V. Rathod, S. Nabavirazavi, S. Zad, and S. S. Iyengar, "Privacy and Security Challenges in Large Language Models," in *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA: IEEE, Jan. 2025, pp. 00746–00752. doi: 10.1109/CCWC62904.2025.10903912.
- [3] G. Sperli, "Exploring Privacy and Security Risks in LLMs: Data Leakage, Prompt Injection, and Membership Inference," in *2025 International Symposium on Multimedia (ISM)*, Naples, Italy: IEEE, Dec. 2025, pp. 5–12. doi: 10.1109/ISM66958.2025.00011.
- [4] H. Alami, P. Lehoux, C. Papoutsis, S. E. Shaw, R. Fleet, and J.-P. Fortin, "Understanding the integration of artificial intelligence in healthcare organisations and systems through the NASSS framework: a qualitative study in a leading Canadian academic centre," *BMC Health Serv. Res.*, vol. 24, no. 1, p. 701, Jun. 2024, doi: 10.1186/s12913-024-11112-x.
- [5] P. Goktas and A. Grzybowski, "Shaping the Future of Healthcare: Ethical Clinical Challenges and Pathways to Trustworthy AI," *J. Clin. Med.*, vol. 14, no. 5, p. 1605, Feb. 2025, doi: 10.3390/jcm14051605.
- [6] E. Olaye, I. Omafovbe, W. O. Aigbe, and D. Obuh, "Personalized Governance Strategy for Patient Data in a Digital One Health Surveillance System," *Procedia Comput. Sci.*, vol. 254, pp. 20–29, 2025, doi: 10.1016/j.procs.2025.02.060.
- [7] H. Roberts, "Digital sovereignty and artificial intelligence: a normative approach," *Ethics Inf. Technol.*, vol. 26, no. 4, p. 70, Dec. 2024, doi: 10.1007/s10676-024-09810-5.
- [8] B. Murdoch, "Privacy and artificial intelligence: challenges for protecting health information in a new era," *BMC Med. Ethics*, vol. 22, no. 1, p. 122, Dec. 2021, doi: 10.1186/s12910-021-00687-3.
- [9] S. M. Williamson and V. Prybutok, "Balancing Privacy and Progress: A Review of Privacy Challenges, Systemic Oversight, and Patient Perceptions in AI-Driven Healthcare," *Appl. Sci.*, vol. 14, no. 2, p. 675, Jan. 2024, doi: 10.3390/app14020675.
- [10] K. P. Anunita, S. R. Devisri, and C. Arumugam, "Federated Learning: Countering Label Flipping Attacks in Retinal OCT," *Procedia Comput. Sci.*, vol. 254, pp. 58–67, 2025, doi: 10.1016/j.procs.2025.02.064.
- [11] C. Mennella, U. Maniscalco, G. De Pietro, and M. Esposito, "Ethical and regulatory challenges of AI technologies in healthcare: A narrative review," *Heliyon*, vol. 10, no. 4, p. e26297, Feb. 2024, doi: 10.1016/j.heliyon.2024.e26297.
- [12] P. L. Chong et al., "Integrating artificial intelligence in healthcare: applications, challenges, and future directions," *Future Sci. OA*, vol. 11, no. 1, p. 2527505, Dec. 2025, doi: 10.1080/20565623.2025.2527505.
- [13] S. Gawankar, S. Nair, V. Pawar, A. Vhatkar, and P. Chavan, "Patient Privacy and Data Security in the Era of AI-Driven Healthcare," in *2024 8th International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, Pune, India: IEEE, Aug. 2024, pp. 1–6. doi: 10.1109/ICCUBEA61740.2024.10775004.
- [14] B. Rodrigues, I. Amorim, I. Costa, and A. Mendes, "Patient-centric health data sovereignty: an approach using Proxy re-encryption," 2023, *arXiv*. doi: 10.48550/ARXIV.2307.01175.
- [15] A. Alabdulatif, I. Khalil, X. Yi, and M. Guizani, "Secure Edge of Things for Smart Healthcare Surveillance Framework," *IEEE Access*, vol. 7, pp. 31010–31021, 2019, doi: 10.1109/ACCESS.2019.2899323.
- [16] S. Chandra, S. Ray, and R. T. Goswami, "Big Data Security in Healthcare: Survey on Frameworks and Algorithms," in *2017 IEEE 7th International Advance Computing Conference (IACC)*, Hyderabad, India: IEEE, Jan. 2017, pp. 89–94. doi: 10.1109/IACC.2017.0033.
- [17] T. Bhaskar, M.N. Narsaiah, and M. Ravikanth, "Central Medical Centre Healthcare Data Security with Lightweight Blockchain Model in IoT Sensor Environment," *JSIHS*, vol. 1, no. 1, pp. 15–26, Dec. 2023, doi: 10.69996/jsihs.2023002.
- [18] G. P. Tobia et al., "Privacy in Italian Clinical Reports: A NLP-Based Anonymization Approach," in *2025 IEEE 13th International Conference on Healthcare Informatics (ICHI)*, Rende, Italy: IEEE, Jun. 2025, pp. 630–635. doi: 10.1109/ICHI64645.2025.00077.
- [19] L. Mainetti and A. Elia, "Detecting Personally Identifiable Information Through Natural Language Processing: A Step Forward," *Appl. Syst. Innov.*, vol. 8, no. 2, p. 55, Apr. 2025, doi: 10.3390/asi8020055.
- [20] L. R. Kumar, C. Ashokkumar, P. S. Pandey, S. K. Kanniah, B. J., and M. I. T. Hussan, "Security Enhancement in Surveillance Cloud Using Machine Learning Techniques," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 11, no. 3s, pp. 46–55, Mar. 2023, doi: 10.17762/ijrtcc.v11i3s.6154.
- [21] M. Nankya, A. Mugisa, Y. Usman, A. Upadhyay, and R. Chataut, "Security and Privacy in E-Health Systems: A Review of AI and Machine Learning Techniques," *IEEE Access*, vol. 12, pp. 148796–148816, 2024, doi: 10.1109/ACCESS.2024.148796.

- 10.1109/ACCESS.2024.3469215.
- [22] N. V. D. P. Raju, "Design and Implementation of Artificial Intelligence and Machine Learning Techniques for Security of Healthcare Systems," in *2024 Asian Conference on Intelligent Technologies (ACOIT)*, KOLAR, India: IEEE, Sep. 2024, pp. 1–7. doi: 10.1109/ACOIT62457.2024.10941611.
- [23] J. Ganesh and A. Bansal, "Transformer-based Automatic Mapping of Clinical Notes to Specific Clinical Concepts," in *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, Torino, Italy: IEEE, Jun. 2023, pp. 558–563. doi: 10.1109/COMPSAC57700.2023.00080.



Research Article

Sampling Strategies and Metric Sensitivity in Imbalanced Predictive Maintenance: A Comparative Study with Ensemble Methods

Berk Mercan ^{a,*} , Eren Ağar ^a , Mert Çimentepe ^a , Meltem Apaydın Üstün ^a ^a Tekirdağ Namık Kemal University, Çorlu Faculty of Engineering, Electrical and Electronics Engineering Department, Çorlu, Tekirdağ, Türkiye

ARTICLE INFO

Article history:

Received 29 June 2026

Accepted 14 September 2026

Keywords:

Class imbalance,
Feature engineering,
Predictive maintenance,
Random Forest,
SHAP,
XGBoost

ABSTRACT

This study presents a systematic comparative analysis of class balancing strategies for predictive maintenance of industrial equipment using the AI4I 2020 Predictive Maintenance Dataset, which comprises 10,000 samples with a severe class imbalance of 3.4% failure instances. Two tree-based ensemble classifiers, Random Forest (RF) and XGBoost, are evaluated under nine balancing configurations: Baseline, class-weight balancing, SMOTE, ADASYN, Random Undersampling (RUS), SMOTE-Tomek, SMOTEENN, SMOTE+RUS, and SVMSMOTE. Four domain-informed engineered features, Delta_T, Power_Proxy, Temp_Ratio, and Torque_per_Wear, are derived from physical process knowledge and integrated into the modeling pipeline. Models are evaluated using Precision, Recall, F1-score, and PR-AUC, with particular emphasis on PR-AUC due to the severe class imbalance. The results show that RF is robust to class imbalance without complex resampling. Tuned Baseline RF achieves the best RF performance with an F1-score of 86.4% and a PR-AUC of 0.886, while Tuned Balanced RF provides a comparable F1-score of 85.7% and a PR-AUC of 0.875. In contrast, XGBoost benefits more from sampling strategy and hyperparameter optimization: Tuned SVMSMOTE + XGB achieves the strongest tuned XGBoost performance with an F1-score of 83.0% and the highest PR-AUC of 0.914. These findings demonstrate that sampling methods do not universally improve classifier performance; rather, their effectiveness depends on the learning algorithm. In addition, SHAP-based interpretability analysis shows that Rotational Speed, Torque, Tool Wear, and engineered load-related features contribute substantially to failure prediction. Overall, PR-AUC is recommended as a primary evaluation metric for imbalanced industrial predictive maintenance datasets.

This is an open access article under the CC BY-SA 4.0 license.
(<https://creativecommons.org/licenses/by-sa/4.0/>)

1. INTRODUCTION

Unplanned equipment failures in modern manufacturing environments impose significant economic burdens through production downtime, emergency repair costs, and safety risks. Traditional maintenance strategies, namely reactive (fix-on-failure) and preventive (schedule-based), have proven inadequate: the former results in costly unplanned stoppages, while the latter leads to unnecessary maintenance actions on healthy equipment. Predictive Maintenance (PdM), enabled by the convergence of Industry 4.0 technologies—Internet of Things (IoT) sensor networks, big data platforms, and machine learning—represents a paradigm shift toward condition-based, data-

driven maintenance decision-making that addresses both failure detection latency and unnecessary preventive interventions [1].

Machine learning-based PdM systems learn failure patterns from multivariate sensor streams (temperature, vibration, torque, rotational speed, tool wear) and generate probabilistic failure forecasts, enabling maintenance teams to intervene precisely when needed. However, a fundamental methodological challenge undermines the reliability of these systems: class imbalance. In real industrial environments, failure events are inherently rare, and predictive maintenance datasets are therefore typically characterized by severe class imbalance [1,2]. Standard

* Corresponding author. E-mail address: 1230657013@nku.edu.tr
DOI: 10.58190/ijamec.2026.179

classification algorithms trained on imbalanced data tend to be biased toward the majority (healthy) class, achieving deceptively high accuracy while failing to detect faults—the very events the system is intended to identify. Addressing this imbalance is therefore not merely a preprocessing step but a core design consideration that directly affects the operational utility of a PdM system.

The literature offers a wide range of techniques to mitigate class imbalance, broadly categorized as data-level methods (oversampling, undersampling, and hybrid methods), algorithm-level methods (cost-sensitive learning, class weighting), and ensemble-level methods. Despite the availability of these approaches, a critical gap persists: the interaction between the choice of balancing strategy and the choice of learning algorithm remains insufficiently characterized in the PdM domain. Most existing comparative studies either evaluate a limited set of strategies, focus on a single classifier, or lack interpretability analysis to explain model decisions to domain practitioners.

This paper addresses these gaps through a structured experimental framework designed to evaluate the interaction between class balancing strategies and learning algorithms in imbalanced predictive maintenance. Random Forest and XGBoost, two widely used tree-based ensemble classifiers in industrial machine learning applications, are benchmarked against nine class balancing configurations on the publicly available AI4I 2020 Predictive Maintenance Dataset [3]. These configurations include the baseline model without balancing, class-weight adjustment, synthetic oversampling methods, undersampling, and hybrid sampling strategies. In addition to model development, this study incorporates domain-informed feature engineering to improve both predictive performance and practical relevance by embedding basic physical relationships among temperature, torque, rotational speed, and tool wear variables. Furthermore, SHAP-based interpretability analysis is included to examine the contribution of individual features to model predictions and to provide more transparent explanations for domain practitioners [4].

The main contributions of this study are as follows: (i) a full factorial comparison of algorithm–sampling strategy interactions across 18 model configurations using Random Forest and XGBoost; (ii) the incorporation of physical domain knowledge through engineered features; (iii) a metric-oriented evaluation framework emphasizing F1-score, Recall, and PR-AUC for imbalanced industrial classification tasks; (iv) a comparative analysis of the relative effects of sampling strategies and hyperparameter optimization for RF and XGBoost; and (v) a SHAP-based interpretability analysis to support the explanation of model decisions.

The scope of this study is deliberately limited to tree-based ensemble methods. This choice is motivated by the

structure of the AI4I 2020 dataset, which consists of tabular sensor-derived features rather than raw sequential time-series or unstructured signal data. Although deep learning architectures have shown strong performance in time-series and signal-based predictive maintenance applications, they generally require larger datasets, higher computational resources, and often provide limited interpretability. In contrast, tree-based ensemble methods offer a practical balance between predictive performance, computational efficiency, and interpretability for structured industrial datasets. Therefore, deep learning architectures are not included in the experimental comparison and are left as a direction for future work.

1.1. Related Work

Research on PdM can be broadly categorized into three groups based on the signal modalities and modeling paradigms employed: classical threshold-based and signal processing approaches, classical machine learning approaches, and deep learning approaches.

Classical and Signal Processing-Based Approaches.

Early PdM systems relied on physics-based thresholds and hand-crafted signal features derived from current, vibration, and temperature sensors. These methods exploit known failure physics: for instance, stator current monitoring in induction motors can identify rotor bar breakage through sideband frequency analysis in the current spectrum [5]. While interpretable and computationally inexpensive, these approaches require detailed prior knowledge of failure mechanisms and do not generalize well across equipment types or operating conditions. Their sensitivity is also limited in detecting incipient faults that manifest as subtle statistical deviations below deterministic alarm thresholds [6].

Classical Machine Learning Approaches.

The availability of labeled sensor datasets has enabled the application of supervised classifiers to PdM. Several studies have compared traditional classifiers—Support Vector Machines (SVM), k-Nearest Neighbors (kNN), Decision Trees, and ensemble methods—on fault detection tasks using static feature vectors extracted from sensor signals. Kalafatelis et al. evaluated multiple classifiers combined with the Synthetic Minority Oversampling Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), and Random Undersampling (RUS) on an offset printing machine dataset, demonstrating that the interaction between sampling method and classifier architecture is a primary determinant of minority class recall rather than the sampling method alone [7]. Similarly, Hakami demonstrated that GAN-generated synthetic data, when used to augment training sets prior to RF and XGBoost training, can alleviate the scarcity of failure samples, achieving ANN accuracy of 88.98% and RF accuracy of 74.15% on a benchmark dataset; the author also noted that temporal dependencies

in sensor data are not captured by static feature extraction [8]. Patel and Shanbhag evaluated multiple ML models on the AI4I 2020 dataset and observed that Random Forest achieved an F1-score of 85% under the standard SMOTE oversampling method but did not investigate whether alternative balancing strategies could further improve or degrade this performance [9]. A broader comparative study by Zaidi et al. found that multi-sensor data fusion—combining vibration, temperature, acoustic emission, current, and speed signals—raised fault classification performance from 84% to over 99% when using kNN, SVM, RF, and Decision Trees, underscoring the importance of feature richness alongside algorithmic choice [10].

Deep Learning Approaches. Temporal sensor data has motivated the application of recurrent and convolutional architectures for PdM. Borré et al. applied a hybrid CNN-LSTM model to vibration signals from electrical motors, using three-axis accelerometer data to detect anomalies via empirical cumulative distribution-based outlier detection; the temporal attention mechanism enabled the model to focus on degradation patterns occurring in specific time windows prior to failure [11]. A comparative evaluation of deep learning architectures by Li and Li demonstrated that hybrid CNN-LSTM models achieve 96.1% accuracy and 95.2% F1-score on industrial sensor datasets, outperforming standalone architectures, while also noting that these gains come at substantially higher computational cost and reduced interpretability [12]. For short-horizon failure forecasting in industrial pumps, Alghtus et al. combined sliding-window time series features with RF and XGBoost classifiers, achieving 65.6% recall at 15- and 30-minute prediction horizons, and found that optimal history window length depends on the target prediction horizon and the specific degradation dynamics of the monitored system [13]. The systematic review by Carvalho et al. further highlighted that while LSTMs and CNN variants achieve state-of-the-art results in PdM, the black-box nature of deep models creates barriers to adoption in safety-critical manufacturing contexts where maintenance decisions must be explainable to operators [2].

Although deep learning architectures such as CNN and LSTM have shown strong performance in sequential fault detection tasks, they are often regarded as black-box models with limited interpretability and higher computational cost. In industrial PdM settings where model transparency is critical for maintenance decision-making, this study focuses on ensemble tree-based methods that offer both competitive performance and interpretability for structured tabular datasets [14,15,16]. SHAP analysis is further employed to support the explanation of feature contributions in the selected models [4].

This paper seeks to address a specific research gap identified in the literature. While ensemble classifiers have

been shown to perform strongly on structured industrial sensor data, and while various class balancing strategies are available, the interaction between learning algorithm, sampling strategy, hyperparameter optimization, metric sensitivity, and interpretability remains insufficiently examined for the AI4I 2020 dataset. Specifically, this study evaluates Random Forest and XGBoost across nine distinct class balancing configurations, integrates domain-informed feature engineering, applies hyperparameter optimization to selected high-performing configurations, and incorporates SHAP-based interpretability analysis. In this way, the study provides practical guidance for designing transparent and metric-aware ML-based PdM pipelines under severe class imbalance.

2. MATERIALS AND METHODS

2.1. Dataset

In this study, we used the AI4I 2020 Predictive Maintenance Dataset [3], accessible through the UCI Machine Learning Repository. The dataset simulates sensor readings from a CNC machine tool in a manufacturing environment, comprising 10,000 observations across six input variables and a binary failure target. It is designed to reflect realistic industrial conditions, including the severe class imbalance characteristic of real PdM scenarios: 9,661 samples (96.6%) correspond to healthy operation, while 339 (3.4%) represent failure events. No missing values are present in the dataset. The resulting class distribution is illustrated in Figure 1. Table 1 summarizes the input variables and the target variable.

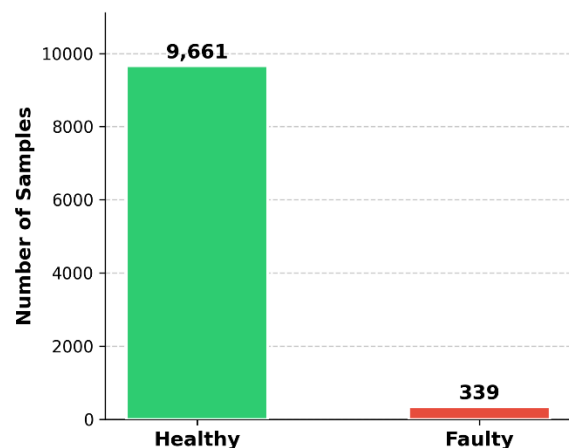


Figure 1. Class distribution of the AI4I 2020 dataset

Table 1. Feature descriptions of the AIAI 2020 dataset

Feature	Type	Description	Range / Values
Air Temperature [K]	Continuous	Ambient air temperature	~295–305 K
Process Temperature [K]	Continuous	Machining process temperature	~305–315 K
Rotational Speed [rpm]	Continuous	Spindle rotational speed	~1000–2800 rpm
Torque [Nm]	Continuous	Applied torque on spindle	~0–77 Nm
Tool Wear [min]	Continuous	Cumulative tool usage duration	0–253 min
Type	Categorical	Product quality class	L / M / H
Target	Binary	Failure label (output)	0: Non-Faulty, 1: Faulty

2.1.1. Feature Engineering

We constructed four derivative features from the raw sensor variables, guided by physical process knowledge of Computer Numerical Control (CNC) machining dynamics [17]. These features are designed to capture non-linear interaction effects that are not directly observable from individual sensor readings. The engineered features and their physical rationale are given in Table 2.

Table 2. Engineered features and their physical rationale

Feature	Formula	Physical Rationale
Delta_T	$T_{\text{process}} - T_{\text{air}}$	Thermal gradient between process and ambient; elevated values may signal abnormal heat generation
Power_Proxy	$\text{Torque} \times \text{RPM}$	Proxy for instantaneous mechanical power; captures combined load effect
Temp_Ratio	$T_{\text{process}} / T_{\text{air}}$	Normalized thermal loading; compensates for ambient temperature drift
Torque_per_Wear	$\text{Torque} / (\text{Wear} + 1)$	Mechanical load normalized by tool age; captures load-wear interaction

2.2. Preprocessing Methods

The categorical product type variable (Type: L, M, H) was transformed into binary indicator variables via One-Hot Encoding. One category was dropped to avoid multicollinearity, yielding two binary columns (Type_L and Type_M). The resulting feature matrix comprises 11 variables after encoding and feature engineering.

The dataset was partitioned into training (80%) and test (20%) sets using stratified random sampling to preserve the 96.6/3.4 class ratio in both subsets, resulting in 8,000 training and 2,000 test samples. All continuous features were scaled using RobustScaler, which scales data by the interquartile range rather than the standard deviation. This choice is motivated by the presence of 418 outliers in rotational speed and 69 in torque, identified via the IQR

method. Because these outliers reflect genuine extreme operating conditions rather than measurement error, they were retained, and the robust scaling strategy mitigates their influence on model training. To prevent data leakage, the scaler was fitted exclusively on the training set and subsequently applied to the test set.

2.3. Machine Learning Methods

Random Forest (RF) [14]. RF is a bagging-based ensemble method that constructs a collection of decision trees on bootstrap samples of the training data, with random feature subsets evaluated at each split. Predictions are aggregated by majority vote. The bootstrap mechanism provides implicit class overrepresentation variation across trees, which confers a degree of natural robustness to class imbalance. RF is computationally efficient, non-parametric, and produces feature importance estimates through mean decrease in Gini impurity.

XGBoost [15]. XGBoost implements gradient boosted trees using a regularized objective function. Unlike bagging, boosting builds trees sequentially, with each tree trained to correct the residual errors of the ensemble. The sequential focus on misclassified samples makes XGBoost sensitive to the initial class distribution; in imbalanced settings, the `scale_pos_weight` parameter provides a direct mechanism for up-weighting minority class errors. XGBoost is consistently competitive on tabular datasets and offers fast inference.

Nine class balancing configurations were applied to each classifier: Baseline (no balancing), class-weight adjustment (balanced), SMOTE [18], ADASYN [19], Random Undersampling (RUS), SMOTE-Tomek, SMOTEENN, SMOTE+RUS (hybrid), and SVMSMOTE [20]. All sampling transformations are encapsulated in imblearn Pipeline objects [21], ensuring that synthetic sample generation occurs exclusively within each training fold of the outer 5-fold stratified cross-validation loop, and that validation and test data are never used in generating synthetic samples. For the selected high-performing RF and XGBoost configurations, hyperparameter optimization was conducted via RandomizedSearchCV (`n_iter=15`, F1-score as the optimization criterion) within the training partition using stratified cross-validation. The RF search space included `n_estimators`, `max_depth`, `min_samples_split`, `min_samples_leaf`, `max_features`, and `class_weight`, whereas the XGBoost search space included `n_estimators`, `max_depth`, `learning_rate`, `subsample`, `colsample_bytree`, `min_child_weight`, and `gamma`. Final test set evaluation was performed only once on the held-out 20% partition that was not used during model selection or hyperparameter search. All experiments used a fixed random seed (`random_state=42`) for reproducibility, with scikit-learn 1.3 and imbalanced-learn 0.11.

2.4. Performance Metrics

The reported performance indicators include Precision, Recall, F1-score, and PR-AUC. Precision measures the proportion of predicted failure cases that are actually failures, while Recall measures the proportion of actual failures that are correctly detected. They are defined in Eq. (1) and Eq. (2), respectively:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2)$$

where TP , FP , and FN denote true positives, false positives, and false negatives, respectively. The F1-score is the harmonic mean of Precision and Recall and is calculated as given in Eq. (3):

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

F1-score provides a balanced measure of false positives and false negatives, which is especially useful for imbalanced classification problems. ROC-AUC, which quantifies the classifier's ability to discriminate between classes across different decision thresholds, is not reported here because it can be overly optimistic under severe class imbalance.

2.5. SHAP-Based Interpretability Analysis

To improve the interpretability of the selected final model, SHAP (SHapley Additive exPlanations) analysis was applied [4]. SHAP assigns an importance value to each feature by quantifying its contribution to the model prediction. In this study, SHAP analysis was performed on the final tuned XGBoost model with the strongest overall minority-class performance, namely Tuned SVMSMOTE + XGB. Mean absolute SHAP values were used to rank the features according to their overall contribution to the model's predictions on the held-out test set.

3. RESULTS AND DISCUSSION

3.1. Random Forest Results

Table 3 presents the performance of all nine Random Forest configurations obtained through stratified 5-fold cross-validation on the training partition and final evaluation on the held-out test set. In line with the imbalanced nature of the dataset, the comparison focuses on recall, F1-score, and PR-AUC rather than accuracy or ROC-AUC, since accuracy can be dominated by the majority class and ROC-AUC may provide overly optimistic interpretations under severe class imbalance. The cross-validation columns summarize the mean fold-wise performance, whereas the test columns reflect the final generalization performance on unseen data.

A methodological caution is necessary when interpreting the test results: the held-out test partition

contains only 68 failure instances, corresponding to 3.4% of the 2,000 test samples. Therefore, a small number of misclassified minority samples can noticeably affect the F1-score, and minor performance differences should not be overinterpreted. With this consideration, Baseline RF achieves the best overall test performance, reaching the highest F1-score of 86.4% and the highest PR-AUC of 0.886. Balanced RF follows as the second-best configuration, with a test F1-score of 82.6% and a PR-AUC of 0.871. These results indicate that Random Forest performs strongly on the original data distribution and that additional sampling strategies do not necessarily improve minority-class performance for this dataset.

Table 3. Cross-validation and test set performance of Random Forest configurations

Model	CV F1 (%)	CV Recall (%)	Test Recall (%)	Test F1 (%)	Test PR-AUC
Baseline RF	83.8	73.5	79.4	86.4	0.886
Balanced RF	79.0	66.4	73.5	82.6	0.871
SMOTE + RF	71.1	81.9	80.9	67.1	0.834
ADASYN + RF	68.6	80.8	82.4	69.6	0.832
RUS + RF	42.1	94.5	95.6	39.4	0.704
SMOTE-Tomek + RF	71.1	81.5	79.4	65.9	0.827
SMOTEENN + RF	59.3	83.0	83.8	58.5	0.781
SMOTE+RUS + RF	75.7	81.5	82.4	73.7	0.853
SVMSMOTE + RF	75.3	76.8	79.4	76.6	0.854

Synthetic oversampling and hybrid methods, including SMOTE, ADASYN, SMOTE-Tomek, and SMOTEENN, generally increased recall but reduced precision, which prevented a net improvement in F1-score. For instance, SMOTE + RF achieved a higher CV recall of 81.9% than Baseline RF, but its CV F1-score fell to 71.1%, which remained considerably lower than the Baseline RF. This indicates that synthetic samples generated by interpolation may introduce additional noise into the feature space and disturb the decision boundaries learned by the Random Forest model.

RUS + RF represents the most extreme precision-recall trade-off. Although it achieved the highest CV recall of 94.5% and the highest test recall of 95.6%, its CV F1-score fell to 42.1% and its test F1-score remained only 39.4%. This means that the model detected many failure cases but also generated a large number of false alarms. Therefore, recall alone is not sufficient for evaluating model performance in imbalanced predictive maintenance tasks. Among the sampling-based configurations, SVMSMOTE + RF and SMOTE+RUS + RF achieved relatively better test F1-scores of 76.6% and 73.7%, respectively; however, they still did not outperform the Baseline RF. Overall, these findings suggest that, for the AI4I 2020 dataset, complex sampling strategies do not improve Random Forest performance compared with the baseline configuration.

3.2. Hyperparameter Optimization Results (RF)

The two highest-performing configurations, Baseline RF and Balanced RF, were selected for subsequent hyperparameter optimization based strictly on their cross-validation F1-scores evaluated within the training partition. The held-out test set remained completely unseen during this selection process. Hyperparameter optimization was conducted using RandomizedSearchCV within the training partition, with inner-fold cross-validation and F1-score as the optimization criterion. Table 4 presents the optimal hyperparameter combinations, while Table 5 compares the post-optimization test set performance of the selected RF models.

This tuning step was performed to examine whether hyperparameter optimization could further improve the already strong performance of Random Forest on the original and class-weighted data distributions. Since Baseline RF and Balanced RF achieved the two best results in the initial RF comparison, focusing on these configurations provides a consistent and data-driven basis for evaluating the effect of tuning.

Table 4. Optimal hyperparameters identified by RandomizedSearchCV for Random Forest

Parameter	Tuned Baseline RF	Tuned Balanced RF
n_estimators	200	400
max_depth	None (unlimited)	20
min_samples_split	2	2
min_samples_leaf	1	2
max_features	log2	sqrt
class_weight	None	balanced

Table 5. Test set performance of the tuned Random Forest models

Model	Test Precision (%)	Test Recall (%)	Test F1 (%)	Test PR-AUC
Tuned Baseline RF	94.7	79.4	86.4	0.886
Tuned Balanced RF	93.1	79.4	85.7	0.875

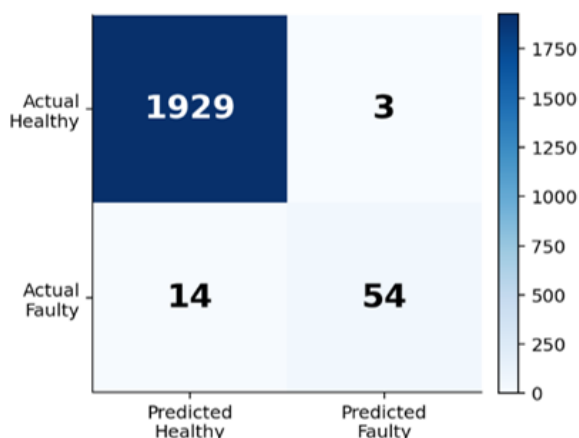


Figure 2. Confusion matrix of the Tuned Baseline RF model

As shown in Table 5, hyperparameter optimization did not alter the performance ranking observed in the initial RF comparison; Tuned Baseline RF retained its leading position. Notably, however, tuning produced a meaningful improvement for Balanced RF, raising its F1-score from 82.6% (pre-tuning) to 85.7%, while Baseline RF remained unchanged at 86.4%. Tuned Baseline RF remained the best-performing RF configuration, achieving a test F1-score of 86.4% and a PR-AUC of 0.886. Tuned Balanced RF achieved a very close F1-score of 85.7% and a PR-AUC of 0.875, with both tuned RF models reaching the same recall value of 79.4%. However, Tuned Baseline RF maintained slightly higher precision, indicating a better precision-recall balance. Figure 2 presents the confusion matrix of the Tuned Baseline RF model on the test set. Overall, these results suggest that Random Forest performs strongly on the original data distribution, while class-weight adjustment provides a competitive but not superior alternative for this dataset. The corresponding precision-recall curve is shown in Figure 3.

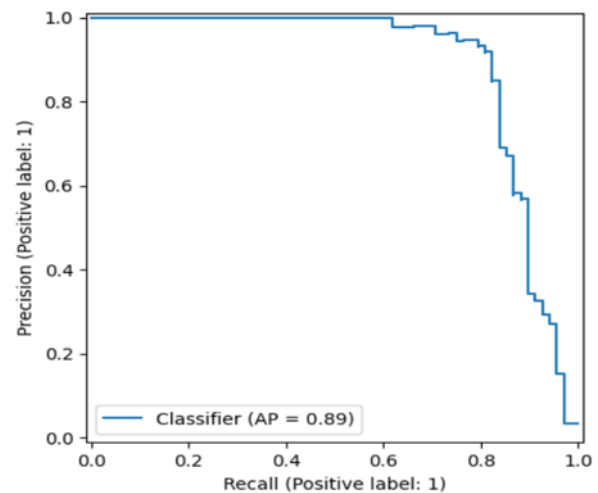


Figure 3. Precision-recall curve of the Tuned Baseline RF model

3.3. XGBoost Results

Table 6 presents the cross-validation and test set performance of all nine XGBoost configurations. To maintain consistency with the Random Forest results, the table reports cross-validation metrics first, followed by test set metrics. Since the dataset is severely imbalanced, the comparison focuses on recall, F1-score, and PR-AUC rather than ROC-AUC. This allows the evaluation to emphasize minority-class detection and the precision-recall trade-off more directly.

Table 6. Cross-validation and test set performance of XGBoost configurations

Model	CV F1 (%)	CV Recall (%)	Test Recall (%)	Test F1 (%)	Test PR-AUC
Baseline XGB	69.8	61.2	70.0	75.4	0.833
Balanced XGB	70.0	69.1	77.1	77.7	0.811
SMOTE + XGB	63.9	74.1	84.3	71.9	0.843
ADASYN + XGB	63.5	76.3	82.9	67.8	0.811
RUS + XGB	35.9	89.9	95.7	43.4	0.648
SMOTE-Tomek + XGB	64.6	75.9	84.3	73.3	0.859
SMOTEENN + XGB	52.9	79.9	91.4	64.0	0.823
SMOTE+RUS + XGB	65.9	74.5	78.6	75.9	0.861
SVMSMOTE + XGB	72.1	75.2	81.4	79.2	0.871

In contrast to the Random Forest results, XGBoost benefited more clearly from selected sampling strategies. SVMSMOTE + XGB achieved the best overall performance before hyperparameter tuning, with the highest test F1-score of 79.2% and the highest test PR-AUC of 0.871. This suggests that, unlike Random Forest, XGBoost can benefit from synthetic augmentation of samples located near decision boundaries through more sophisticated techniques such as SVMSMOTE.

The results also show why recall alone should not be used as the primary performance criterion. RUS + XGB achieved the highest test recall of 95.7%, indicating that it detected most failure cases. However, its test F1-score remained only 43.4%, and its PR-AUC dropped to 0.648, showing that the model produced too many false alarms. Similarly, SMOTEENN + XGB reached a high recall of 91.4%, but its F1-score remained limited at 64.0%. Therefore, a high recall value does not necessarily indicate a well-balanced model in imbalanced predictive maintenance problems.

Among the more balanced configurations, SVMSMOTE + XGB provided the strongest precision-recall trade-off, while SMOTE+RUS + XGB achieved the second highest PR-AUC value of 0.861. Overall, these findings indicate that sampling strategies affect XGBoost differently from Random Forest. While complex sampling methods degraded or failed to improve Random Forest performance, boundary-aware and hybrid sampling approaches provided more competitive results for XGBoost before hyperparameter optimization.

3.4. Hyperparameter Optimization Results (XGBoost)

Based on their initial cross-validation F1-scores evaluated strictly on the training partition, the two top-performing XGBoost configurations (Balanced and SVMSMOTE) were selected for subsequent hyperparameter optimization using RandomizedSearchCV. Table 7 presents the winning parameter combinations and Table 8 compares their post-optimization test set performance. The search space spanned `n_estimators`, `max_depth`, `learning_rate`,

`subsample`, `colsample_bytree`, `min_child_weight`, and `gamma`—parameters governing XGBoost’s tree construction depth, stochastic column and row subsampling, and leaf impurity regularization. The optimal hyperparameter combinations differed between the two tuned XGBoost configurations, indicating that the upstream balancing strategy affected the preferred model complexity. Tuned Balanced XGB selected a shallower and faster-learning configuration, whereas Tuned SVMSMOTE + XGB favored a larger ensemble with deeper trees and a lower learning rate. This suggests that boundary-aware synthetic sampling benefited from a more expressive boosted-tree structure combined with shrinkage regularization.

Table 7. Optimal hyperparameters identified by RandomizedSearchCV for XGBoost

Parameter	Tuned Balanced XGB	Tuned SVMSMOTE + XGB
<code>n_estimators</code>	200	500
<code>max_depth</code>	5	8
<code>learning_rate</code>	0.1	0.05
<code>subsample</code>	0.7	0.9
<code>colsample_bytree</code>	0.8	0.7
<code>min_child_weight</code>	1	1
<code>gamma</code>	0.1	0.1

Table 8. Test set performance of the tuned XGBoost models

Model	Test Precision (%)	Test Recall (%)	Test F1 (%)	Test PR-AUC
Tuned Balanced XGB	81.2	82.4	81.8	0.856
Tuned SVMSMOTE + XGB	79.2	87.1	83.0	0.914

Hyperparameter tuning produced clear improvements for both selected XGBoost configurations. Tuned Balanced XGB achieved a test F1-score of 81.8% and a PR-AUC of 0.856, improving over its pre-tuning performance. However, Tuned SVMSMOTE + XGB achieved the strongest overall performance among the tuned XGBoost models, with a test F1-score of 83.0% and the highest PR-AUC of 0.914. It also provided the highest recall value of 87.1%, indicating stronger failure detection capability.

The comparison between the two tuned models reveals a clear precision-recall trade-off. Tuned Balanced XGB achieved slightly higher precision, whereas Tuned SVMSMOTE + XGB achieved higher recall, F1-score, and PR-AUC. This indicates that boundary-aware synthetic sampling, when combined with hyperparameter optimization, provided a more effective balance between detecting failure cases and maintaining overall minority-class performance. Therefore, unlike Random Forest, for which the baseline configuration remained dominant, XGBoost benefited more from the combination of class-balancing strategy and hyperparameter tuning.

3.5. SHAP-Based Interpretability Results

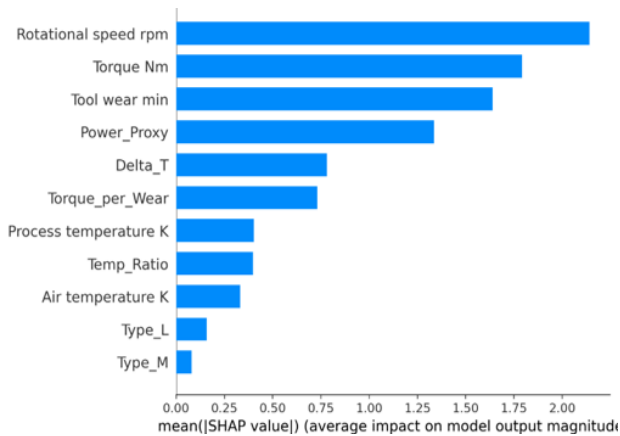


Figure 4. Mean absolute SHAP values of the Tuned SVMSMOTE + XGB model

Figure 4 presents the mean absolute SHAP values for the Tuned SVMSMOTE + XGB model. The results show that Rotational Speed, Torque, and Tool Wear are the most influential original sensor variables in the model's decision process. Among the engineered features, Power_Proxy, Delta_T, and Torque_per_Wear also contribute substantially, indicating that domain-informed feature engineering added useful predictive information beyond the raw sensor measurements. Overall, the SHAP analysis supports the practical relevance of both the original operating variables and the physically motivated engineered features in predicting machine failure.

3.6. Discussion

The most consequential finding of this study is that the effectiveness of class balancing strategies is strongly dependent on the learning algorithm. For Random Forest, the baseline and class-weighted configurations remained dominant. Tuned Baseline RF achieved the best RF performance with an F1-score of 86.4% and a PR-AUC of 0.886, while Tuned Balanced RF achieved a very close F1-score of 85.7% and a PR-AUC of 0.875. These results suggest that the bagging-based structure of Random Forest provides a degree of robustness under class imbalance, and that complex synthetic sampling strategies are not necessarily beneficial for this model.

For XGBoost, the pattern was different. Before hyperparameter optimization, SVMSMOTE + XGB achieved the strongest XGBoost performance, with a test F1-score of 79.2% and a PR-AUC of 0.871. After tuning, Tuned SVMSMOTE + XGB remained the strongest tuned XGBoost configuration, achieving an F1-score of 83.0% and the highest PR-AUC of 0.914. Tuned Balanced XGB also improved after optimization, reaching an F1-score of 81.8% and a PR-AUC of 0.856. These findings indicate that XGBoost benefits more from the combination of boundary-aware synthetic sampling and hyperparameter optimization than Random Forest.

The metric-level results further show that recall alone is not sufficient for model selection in imbalanced predictive maintenance. For example, RUS + XGB achieved the highest test recall of 95.7%, but its F1-score remained only 43.4%, and its PR-AUC dropped to 0.648. This indicates that the model detected many failure cases but generated too many false alarms. Similarly, RUS + RF achieved the highest CV recall among RF configurations, but its precision and F1-score were substantially lower than those of Baseline RF. Therefore, F1-score and PR-AUC provide a more reliable basis for evaluating minority-class performance than recall alone.

The SHAP analysis provides an additional interpretability perspective for the final tuned XGBoost model. Rotational Speed, Torque, and Tool Wear emerged as the most influential original sensor variables, while Power_Proxy, Delta_T, and Torque_per_Wear were the most relevant engineered features. This supports the practical relevance of the feature engineering step and shows that the model relies on physically meaningful operating variables rather than arbitrary patterns. For industrial PdM applications, this interpretability is important because maintenance decisions should be understandable to engineers and operators.

From a practical perspective, the results suggest that class balancing strategies should not be selected independently of the learning algorithm. For Random Forest, the baseline configuration should be considered a strong starting point, and class-weight adjustment may be used as a competitive alternative. For XGBoost, boundary-aware sampling strategies such as SVMSMOTE can provide additional benefit, especially when combined with hyperparameter optimization. Overall, the findings support a metric-aware and interpretable PdM pipeline in which sampling strategy, learning algorithm, hyperparameter tuning, and explainability are evaluated together rather than in isolation.

4. CONCLUSIONS

This study conducted a systematic comparison of nine class balancing strategies applied to Random Forest and XGBoost on the AI4I 2020 Predictive Maintenance Dataset. The results demonstrate that the effectiveness of class balancing strategies is algorithm dependent. For Random Forest, the baseline and class-weighted configurations produced the strongest results. Tuned Baseline RF achieved the best RF performance with an F1-score of 86.4% and a PR-AUC of 0.886, while Tuned Balanced RF achieved a comparable F1-score of 85.7% and a PR-AUC of 0.875. These findings indicate that Random Forest can remain robust under severe class imbalance without requiring complex synthetic resampling strategies.

For XGBoost, the results showed a different pattern.

Boundary-aware sampling combined with hyperparameter optimization provided the most effective tuned XGBoost performance. Tuned SVMSMOTE + XGB achieved an F1-score of 83.0% and the highest PR-AUC of 0.914, outperforming Tuned Balanced XGB in terms of recall, F1-score, and PR-AUC. This confirms that XGBoost benefits more from the combination of sampling strategy and hyperparameter optimization than Random Forest.

The study also confirms that PR-AUC and F1-score are more informative than accuracy or ROC-AUC under severe class imbalance. High recall alone was shown to be insufficient, since some configurations achieved high recall by producing excessive false alarms. Therefore, model selection in imbalanced predictive maintenance should consider the full precision-recall trade-off rather than relying on a single metric.

In addition, SHAP-based interpretability analysis showed that Rotational Speed, Torque, and Tool Wear were the most influential original sensor variables, while Power_Proxy, Delta_T, and Torque_per_Wear were the most relevant engineered features. This supports the practical value of domain-informed feature engineering and improves the transparency of the final tuned XGBoost model.

Overall, the findings suggest that class balancing strategies should not be selected independently of the learning algorithm. For industrial predictive maintenance pipelines, Random Forest can serve as a strong and robust baseline, while XGBoost may benefit from boundary-aware sampling and careful hyperparameter optimization. Future work should extend this framework to additional industrial datasets, investigate cost-sensitive threshold optimization, and evaluate interpretable deep learning models for sequential sensor data.

Declaration of Ethical Standards

The authors declare that this research complied with all applicable ethical guidelines. No experiments involving human subjects or animals were conducted.

Credit Authorship Contribution Statement

Berk Mercan: Conceptualization, Methodology, Software, Formal Analysis, Investigation, Data Curation, Visualization, Literature Review, Writing – Original Draft, Writing – Review and Editing.

Eren Açar: Methodology, Software, Formal Analysis, Validation, Data Curation, Investigation, Writing – Review and Editing.

Mert Çimentepe: Literature Review, Validation, Writing – Review and Editing.

Meltem Apaydın Üstün: Supervision, Conceptualization, Methodology, Project Administration, Writing – Review and Editing.

Declaration of Competing Interest

The authors declare no competing financial or personal interests.

Funding / Acknowledgements

This study was supported by the Scientific and Technological Research Council of Türkiye (TÜBİTAK) under the 2209-A University Students Research Projects Support Program, 2024/1 call, project number 1919B012470616. The authors gratefully acknowledge TÜBİTAK for its support. The responsibility for the content of this publication belongs solely to the authors. Outputs produced within the scope of the program will be reported to TÜBİTAK through the BİDEB Application and Monitoring System.

Data Availability

The AI4I 2020 Predictive Maintenance Dataset used in this study is publicly available through the UCI Machine Learning Repository at <https://doi.org/10.24432/C5HS5C>.

References

- [1] I. Hector and R. Panjanathan, "Predictive maintenance in Industry 4.0: A survey of planning models and machine learning techniques," *PeerJ Computer Science*, vol. 10, Art. no. e2016, 2024. doi: 10.7717/peerj-cs.2016
- [2] T. P. Carvalho et al., "A systematic literature review of machine learning methods applied to predictive maintenance," *Computers & Industrial Engineering*, vol. 137, Art. no. 106024, 2019. doi: 10.1016/j.cie.2019.106024
- [3] S. Matzka, "AI4I 2020 predictive maintenance dataset," UCI Machine Learning Repository, 2020. doi: 10.24432/C5HS5C
- [4] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. 31st Int. Conf. Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, 2017, pp. 4765–4774. arXiv:1705.07874.
- [5] O. E. Hassan et al., "Induction motor broken rotor bar fault detection techniques based on fault signature analysis – a review," *IET Electric Power Applications*, vol. 12, no. 7, pp. 895–907, 2018. doi: 10.1049/iet-epa.2018.0054
- [6] D. Neupane et al., "Data-driven machinery fault diagnosis: A comprehensive review," *Neurocomputing*, vol. 627, Art. no. 129588, 2025. doi: 10.1016/j.neucom.2025.129588
- [7] A. S. Kalafatelis et al., "An effective methodology for imbalanced data handling in predictive maintenance for offset printing," in *Proc. 11th Int. Conf. Mechatronics and Control Engineering (ICMCE 2023)*, Lecture Notes in Mechanical Engineering, Springer, Singapore, 2024, pp. 89–98. doi: 10.1007/978-981-99-6523-6_7
- [8] A. Hakami, "Strategies for overcoming data scarcity, imbalance, and feature selection challenges in machine learning models for predictive maintenance," *Scientific Reports*, vol. 14, Art. no. 9645, 2024. doi: 10.1038/s41598-024-59958-9
- [9] K. Patel and A. Shanbhag, "Exploring ML for predictive maintenance using imbalance correction techniques and SHAP," in *Proc. 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, Prague, Czech Republic, 2022, pp. 1–10. doi: 10.1109/ICECET55527.2022.9873073
- [10] S. H. H. Zaidi et al., "A systematic review of anomaly and fault detection using machine learning for industrial machinery," *Algorithms*, vol. 19, no. 2, Art. no. 108, 2026. doi: 10.3390/a19020108
- [11] A. Borré et al., "Machine fault detection using a hybrid CNN-LSTM attention-based model," *Sensors*, vol. 23, no. 9, Art. no. 4512, 2023. doi: 10.3390/s23094512
- [12] W. Li and T. Li, "Comparison of deep learning models for predictive maintenance in industrial manufacturing systems using sensor data," *Scientific Reports*, vol. 15, no. 1, Art. no.

- 23545, 2025. doi: 10.1038/s41598-025-08515-z
- [13] K. M. A. Alghtus et al., "Short-horizon predictive maintenance of industrial pumps using time-series features and machine learning," arXiv preprint arXiv:2508.19974, 2025. doi: 10.48550/arXiv.2508.19974
- [14] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. doi: 10.1023/A:1010933404324
- [15] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794. doi: 10.1145/2939672.2939785
- [16] F. Velasco-Loera, M. Alcaraz-Mejia, and J. L. Chavez-Hurtado, "An interpretable hybrid fault prediction framework using XGBoost and a probabilistic graphical model for predictive maintenance: A case study in textile manufacturing," *Applied Sciences*, vol. 15, no. 18, Art. no. 10164, 2025. doi: 10.3390/app151810164
- [17] Y. Altintas, *Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2012. doi: 10.1017/CBO9780511843723
- [18] N. V. Chawla et al., "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002. doi: 10.1613/jair.953.
- [19] H. He et al., "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN)*, Hong Kong, 2008, pp. 1322–1328. doi: 10.1109/IJCNN.2008.4633969.
- [20] H. M. Nguyen et al., "Borderline over-sampling for imbalanced data classification," *International Journal of Knowledge Engineering and Soft Data Paradigms*, vol. 3, no. 1, pp. 4–21, 2011. doi: 10.1504/IJKESDP.2011.039875
- [21] G. Lemaitre et al., "Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: <http://jmlr.org/papers/v18/16-365.html>.

*Research Article*

YOLOv10 and RT-DETR for Weld Defect Detection: Experimental Evaluation of Dataset Splits and Explainability

Adem Dilbaz^{a,*}

^a Sahin Tanker Limited Company, Asagipinarbasi Organized Industrial Zone, 521 Street 7, 42250, Konya, Türkiye

ARTICLE INFO

Article history:

Received 30 July 2026

Accepted 14 September 2026

Keywords:

CAM,
Deep learning,
Object detection,
RT-DETR,
Weld defect detection,
YOLOv10

ABSTRACT

Automated weld defect inspection is essential for improving product quality, reducing inspection costs, and minimizing human error in industrial manufacturing. Although deep learning-based object detectors have demonstrated promising performance, comprehensive comparisons under identical experimental conditions remain limited. This study presents a systematic evaluation of YOLOv10 and RT-DETR architectures for weld defect detection using the publicly available Weld Defects (WD) dataset. Five object detection models, namely YOLOv10-N, YOLOv10-L, YOLOv10-X, RT-DETR-L, and RT-DETR-X, were trained and evaluated under identical hyperparameter settings. In addition, the influence of different training-validation split ratios on YOLOv10 performance was investigated. Detection performance was assessed using Precision, Recall, F1-score, mAP@50, and mAP@50–95, while qualitative analyses were conducted through representative detection results and class activation mapping (CAM) visualizations. Experimental results showed that RT-DETR-X achieved the highest localization performance with a mAP@50–95 of 0.458 and the best F1-score, whereas YOLOv10-X obtained the highest precision (0.768) while maintaining competitive overall performance. Explainability analysis further demonstrated that RT-DETR-X generated more spatially coherent attention maps, whereas YOLOv10-X produced more localized but occasionally fragmented activations. The findings provide a comprehensive comparison of CNN- and Transformer-based detectors and offer practical guidance for selecting suitable deep learning models for reliable automated weld defect inspection in industrial applications.

This is an open access article under the CC BY-SA 4.0 license.
(<https://creativecommons.org/licenses/by-sa/4.0/>)

1. INTRODUCTION

In industrial manufacturing, welding is widely adopted as a primary joining process owing to its high joint strength, economic efficiency, and capability to produce durable structures [1]. It plays a vital role in various industries, including automotive, aerospace, shipbuilding, construction, and pipeline engineering, where the quality of welded joints directly affects the structural integrity, reliability, and safety of the final products [2]. However, welding processes are susceptible to various defects that may compromise joint performance and lead to costly failures if not detected at an early stage. Conventional weld inspection methods rely heavily on manual visual examination and expert interpretation, making the inspection process time-consuming, labor-intensive, and

prone to human error [3]. Recent advances in deep learning have enabled the development of automated inspection systems that provide reliable and efficient weld defect detection from visual data. Consequently, numerous studies have investigated convolutional neural network (CNN) and transformer-based architectures to improve the accuracy and robustness of automated weld inspection. Some of the studies related to this research are presented below.

Zhao et al. used RT-DETR, an end-to-end object detector that eliminates the need for Non-Maximum Suppression (NMS). On the MS COCO dataset, RT-DETR-R50 achieved 53.1 AP with an inference latency of approximately 9.2 ms, outperforming several YOLO-based detectors in the accuracy-speed trade-off. The

* Corresponding author. E-mail address: ademdilbaz38@gmail.com
DOI: 10.58190/ijamec.2026.180

Transformer-based architecture provides strong global feature modeling capabilities compared with conventional CNN-based detectors. However, YOLO remains more suitable for resource-constrained and embedded applications due to its lower computational complexity [4].

In one study, Sivakumar et al. developed a weld defect classification framework based on Swin Transformer and DeiT using the SWRD dataset. Image denoising and enhancement techniques were applied to improve input quality before training. The proposed method achieved a 99.38% F1-score in binary classification and 98.30% F1-score in multi-class classification, demonstrating excellent performance in weld defect identification. Overall, the study showed that Transformer-based models can provide highly accurate and reliable weld defect classification [5].

Xu et al. used DBP-YOLO, a lightweight weld seam defect detection model based on YOLOv8n that enhances feature representation and multi-scale feature fusion through the proposed C2f-BiFusion and DFFN modules. The model also incorporates the Focal PIoU loss function to improve bounding box regression accuracy. Experimental results showed that DBP-YOLO achieved 83.5% mAP (mean Average Precision) at an Intersection over Union (IoU) threshold of 0.50, representing a 3.7% improvement over the baseline YOLOv8n while reducing the number of parameters by 1.21 million. These results demonstrate that the proposed model provides accurate and computationally efficient weld seam defect detection [6].

Parmar et al. evaluated YOLOv5, YOLOv8, YOLO11, and RT-DETR for weld defect detection using thermographic images acquired during TIG welding. The models were compared in terms of precision, recall, mAP, and inference latency. RT-DETR-X demonstrated superior feature extraction capability but incurred the highest inference latency, whereas YOLOv8 achieved the best overall balance between detection accuracy and computational efficiency. The results confirmed the effectiveness of thermography combined with deep learning for automated weld quality inspection [7].

Zhang et al. developed LMS-RTDETR, a lightweight Transformer-based model for X-ray weld defect detection in resource-constrained environments. The proposed framework integrates multi-scale feature aggregation, efficient attention, and an optimized NWD-Inner GIoU loss to improve detection performance. Experimental results showed a 3.10% improvement in mAP@50-95, while reducing the number of parameters by 38.34% and GFLOPs by 29.30%. These findings demonstrate that LMS-RTDETR achieves high detection accuracy with significantly lower computational complexity [8].

Peng et al. used YOLO-PDR, a lightweight defect detection network designed to improve the detection of small and low-contrast defects in complex textured environments. The proposed model integrates parallel

texture fusion, dilated block reparameterization, and a lightweight detail enhancement head to enhance feature representation while reducing computational complexity. Experimental results demonstrated a 97.4% mAP@50, with only 2.11 M parameters and 5.9 GFLOPs, while reducing the number of parameters and computational cost by 18.6% and 6.35%, respectively, compared with the baseline YOLOv11 model. These findings indicate that YOLO-PDR achieves an excellent balance between detection accuracy and computational efficiency [9].

In another study, Zhang et al. developed YOLO-Air, a lightweight YOLOv11-based model for weld surface defect detection. The proposed model enhances feature extraction while reducing computational complexity through optimized backbone, neck, and detection head designs. Experimental results showed a 1.3% improvement in mAP@50, while reducing the number of parameters by 17.3% and computational complexity by 31.7% compared with the baseline YOLOv11 model. These results demonstrate that YOLO-Air provides an efficient and accurate solution for weld surface defect detection [10].

Mafuwe et al. experimentally compared YOLO and RT-DETR for pest detection on yellow sticky traps by investigating the effects of learning rate, optimizer, and batch size. Their results showed that RT-DETR achieved its best performance using the SGD optimizer with a batch size of 32 and a learning rate of 0.01, reaching a mAP@50 of 0.963. In contrast, YOLO demonstrated greater robustness to higher learning rates and achieved perfect classification performance on the test set, with no observed misclassifications. Overall, both models achieved comparable detection performance after hyperparameter optimization [11].

Guo et al. proposed MCTNet, a CNN-Transformer hybrid network for aluminum alloy die-casting defect detection. The proposed model integrates multilevel feature interaction, super-resolution feature fusion, and an adaptive feature weighting detection head to improve the detection of multiscale and irregular defects. Experimental results showed 4.8% and 5.7% improvements in mAP@50 and mAP@50:90, respectively, compared with the baseline model. These findings demonstrate that MCTNet achieves superior detection accuracy over both CNN-based and Transformer-based approaches [12].

Zhang et al. proposed Weld-YOLO, a YOLO11-based model for weld X-ray defect detection. The model incorporates a dynamic gating pinwheel convolution (DGPCnv) and a dual attention block enhancement (DABE) module to improve feature extraction and multi-scale representation. These enhancements increase robustness against defects with varying sizes and orientations. Experimental results showed that Weld-YOLO achieved 97.5% mAP@0.5 and 93.4% recall. The proposed approach improved detection accuracy while

maintaining a lightweight architecture with only a slight increase in computational cost [13].

In a study, Yang et al. used a multi-image feature sequence fusion method based on YOLOv8 for radiographic casting defect detection. The approach employs gamma transformation to generate dynamic image sequences and a multi-image feature fusion network to enhance spatiotemporal feature representation, followed by confidence similarity-based localization refinement. The proposed method was experimentally compared with state-of-the-art detectors, including RT-DETR, YOLOv10, and YOLOv11, demonstrating superior detection performance. It achieved a 96.71% recall and 77.09% mAP, significantly reducing missed detections and improving localization accuracy for low-semantic defects [14].

Zhang et al. proposed FC-YOLO, an enhanced YOLOv7-based model for aircraft skin defect detection. The model integrates the Coordinate & Channel Attention (ELAN-C) module, an Adaptive-Path Aggregation Network (APAN) with ASFF, and the SCYLLA-IoU loss function to improve multi-scale feature fusion and localization accuracy. Experimental results on the ASD-DET and NEU-DET datasets showed that FC-YOLO outperformed both YOLOv8 and RT-DETR, achieving higher detection accuracy while improving the robustness of aircraft skin defect inspection [15].

Huan et al. proposed EF-YOLO, an enhanced lightweight defect detection network for quality inspection of *Paeoniae Radix Alba* slices. The proposed method integrates a self-developed detection head (LSD-Head) with a Feature Fusion Attention (FCA) mechanism to improve defect feature representation. Experimental comparisons were conducted with several state-of-the-art detectors, including YOLOv5, YOLOv8, YOLOv10, YOLOv8 variants, and RT-DETR. The proposed EF-YOLO achieved 74.1% mAP@0.5 with only 5.4 GFLOPs and 2.62 M parameters, providing competitive accuracy with significantly lower computational complexity [16].

Dilbaz and Ozkan previously investigated the YOLOv3-S and YOLOv5-M models for automated spot-weld defect detection within a simulation-based inspection framework [17]. In contrast, the present study provides a broader comparative evaluation of the CNN-based YOLOv10 and Transformer-based RT-DETR architectures for weld defect detection using a different and larger dataset of real-world images, while also considering different dataset split configurations and incorporating explainability analyses.

Dilbaz previously conducted a comparative evaluation of YOLO12 and YOLO13 architectures for steel surface defect detection using a fixed test sample and different validation ratios on a dataset of 1,800 images [18]. In contrast, the present study uses a different dataset consisting of approximately 5,200 images, incorporates

XAI analyses, and introduces a comparative assessment of CNN-based YOLOv10 and Transformer-based RT-DETR models. Moreover, the inclusion of X-variants enables a more comprehensive investigation of detection accuracy and computational efficiency under industrial inspection scenarios.

Among recent YOLO-based detectors, YOLOv10, originally developed by researchers from Tsinghua University and later adopted by the Ultralytics framework, introduces an end-to-end object detection architecture with consistent dual label assignment and NMS-free inference [19]. These improvements enhance detection efficiency and accuracy, making YOLOv10 suitable for industrial visual inspection applications.

Despite the considerable progress achieved by recent YOLO variants and Transformer-based detectors such as RT-DETR, most existing studies focus on improving individual architectures or evaluating limited model versions. Comparative analyses between advanced YOLO-based detectors and Transformer-based approaches under comparable experimental conditions remain limited, particularly for weld surface defect detection. To address this research gap, this study presents a comprehensive evaluation of the Ultralytics YOLOv10 and RT-DETR models using the publicly available Weld Defects (WD) dataset under identical experimental settings and hyperparameters. Furthermore, the effect of different training-validation split ratios on YOLOv10 performance is systematically investigated. The evaluated models are compared using standard object detection metrics, including precision, recall, F1-score, mAP@50, and mAP@50–95, to analyze their accuracy, robustness, and suitability for industrial weld inspection applications.

The main contributions of this study can be summarized as follows:

- A comprehensive comparative evaluation of YOLOv10 and RT-DETR architectures is conducted for weld defect detection using the same dataset and identical experimental conditions.
- The effect of different training-validation split ratios on YOLOv10 performance is systematically investigated to analyze their effect on evaluation stability and detection performance.
- The evaluated models are compared using widely adopted object detection metrics, including precision, recall, F1-score, mAP@50, and mAP@50–95, providing a comprehensive analysis of detection capability.
- The trade-off between detection accuracy and computational efficiency is investigated by considering model complexity and inference performance for practical industrial inspection scenarios.
- The study provides practical insights into the selection of suitable deep learning-based detectors and data partition strategies for reliable weld inspection systems. Furthermore, the trained RT-DETR model weights are

made available to support reproducibility and future research.

- XAI analysis is performed using Grad-CAM visualizations to examine the spatial attention patterns of the evaluated models and provide complementary insights into their decision-making behavior in challenging weld defect scenarios.

The remainder of this paper is organized as follows. Section 2 describes the materials and methods, including the dataset, model architectures, experimental settings, and evaluation metrics. Section 3 presents the experimental results, followed by a detailed discussion of the findings in Section 4. Finally, the conclusions and future research directions are presented in Section 5.

2. Materials and Methods

2.1. Weld Defects Dataset

The WD dataset, a publicly available benchmark dataset for weld defect detection, was used in this study. The dataset contains a total of 5,198 annotated images obtained from the Roboflow Universe platform, including 4,797 training images, 279 validation images, and 122 test images in its original configuration [20]. In this study, for all evaluated train–validation split ratios (46:3:1, 46:6:1, and 46:9:1), the same independent test set of 122 images was kept unchanged. It consists of three classes: Good Weld, Bad Weld, and Defect, representing different weld conditions and defect categories.

In the original WD dataset configuration, approximately 92.3%, 5.4%, and 2.3% of the images were allocated to the training, validation, and testing sets, respectively, corresponding to a 46:3:1 split ratio. However, to systematically investigate the effect of training–validation partitioning on detection performance, the dataset was reorganized using different training–validation split ratios while maintaining the same experimental conditions. These alternative train–validation configurations were used to analyze their influence on model evaluation reliability, generalization capability, and detection performance.

The dataset includes weld images containing both defective and non-defective regions, enabling the evaluation of deep learning-based object detection models for automated weld inspection. Representative examples from the dataset are presented in Figure 1.



Figure 1. Sample Images from the WD Dataset: Defect, Bad Weld, and Good Weld, Respectively

The images in the WD dataset are provided in .jpg format and represented in the RGB color space. Each image contains annotated weld regions belonging to one or more of three classes: Class 0 (Bad Weld), Class 1 (Defect), and Class 2 (Good Weld). A total of 3,424 images contain Class 0, 2,456 images contain Class 1, and 3,208 images contain Class 2. Since a single image may contain instances from multiple classes, these image counts are not mutually exclusive. The dataset exhibits substantial variations in weld appearance, shape, illumination, background conditions, and defect characteristics, providing a challenging benchmark for the training and evaluation of deep learning-based weld quality inspection models.

2.2. YOLOv10 Architecture

YOLOv10 is a state-of-the-art one-stage object detection model designed to achieve high detection accuracy while minimizing computational cost and inference latency [21]. The architecture introduces an end-to-end detection framework that eliminates redundant post-processing operations, enabling efficient object detection. By combining effective feature extraction, multi-scale feature fusion, and an optimized prediction mechanism, YOLOv10 provides robust performance across a wide range of object detection tasks, including industrial visual inspection and automated weld defect detection. The network consists of three main components: a backbone that extracts hierarchical feature representations from the input image, a neck that fuses multi-scale features to improve the detection of objects with different sizes, and a detection head that directly predicts object classes, confidence scores, and bounding box coordinates in an end-to-end manner [22].

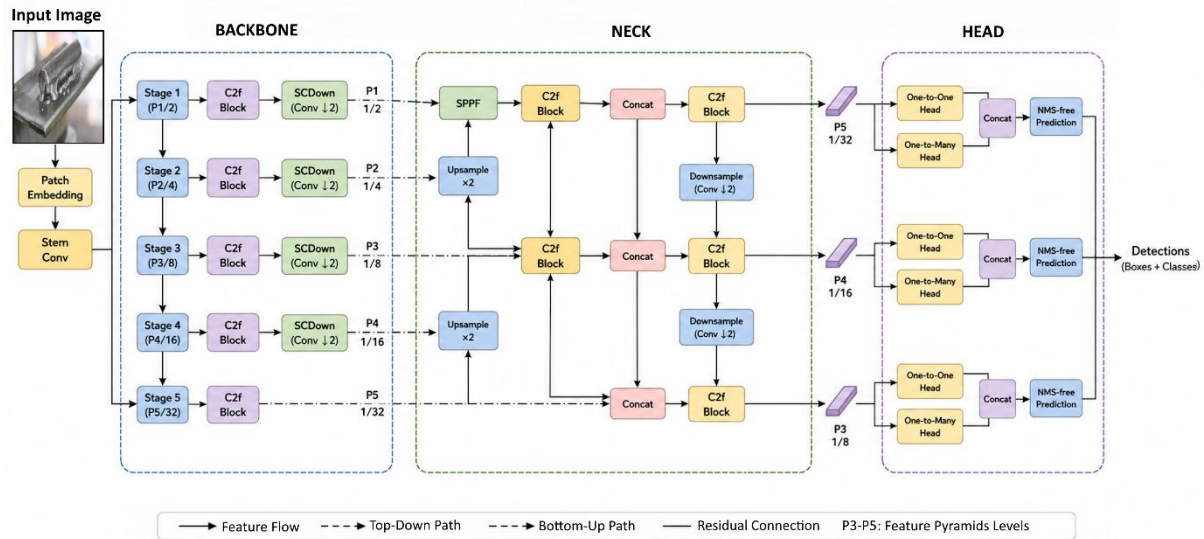


Figure 2. Layers and Implementation Stages of the YOLOv10 Architecture

Figure 2 illustrates the overall architecture of the YOLOv10 model. The backbone extracts hierarchical feature representations using convolutional layers, C2f blocks, and SCDown modules for efficient spatial downsampling [23]. The extracted features are then fused in the neck through an FPN-PAN structure, where the SPPF module enlarges the receptive field and enhances multi-scale feature representation with low computational cost [24]. Finally, the end-to-end NMS-free detection head directly predicts object classes, confidence scores, and bounding box coordinates without redundant post-processing. The decoupled prediction branches improve classification and localization performance, while the SiLU activation function facilitates efficient feature learning and CIoU loss contributes to accurate bounding box regression during training [25].

2.3. RT-DETR with HGNetv2 (High-Performance GPU Network v2) Architecture

RT-DETR is a Transformer-based object detector that combines convolutional feature extraction with Transformer-based object decoding in an end-to-end detection framework. Unlike conventional CNN-based detectors, RT-DETR eliminates heuristic post-processing steps and directly predicts object classes and bounding boxes through a Transformer decoder, achieving an effective balance between detection accuracy and inference speed [26].

The backbone of RT-DETR is based on HGNetv2, a lightweight convolutional architecture specifically designed for efficient GPU inference. Compared with the widely used ResNet backbone, HGNetv2 adopts hierarchical aggregation blocks that improve feature reuse while reducing computational redundancy. Instead of relying on residual bottleneck structures with numerous skip connections, HGNetv2 employs efficient convolutional operations and optimized feature

aggregation, enabling stronger multi-scale feature extraction with lower computational complexity. As a result, HGNetv2 achieves a better trade-off between detection accuracy and inference time, making it particularly suitable for industrial vision applications.

Following the backbone, RT-DETR utilizes a Hybrid Encoder to process multi-scale feature maps. The Hybrid Encoder integrates Attention-based Intra-scale Feature Interaction (AIFI) and Cross-scale Cross Feature Fusion Module (CCFM) to effectively capture both local details and global contextual information while maintaining computational efficiency. Finally, a Transformer decoder performs end-to-end object prediction by directly estimating object classes and bounding box coordinates without requiring NMS, thereby reducing inference latency and improving deployment efficiency [27].

Table 1. Comparison of the Architectural Characteristics of YOLOv10 and RT-DETR

Feature	YOLOv10	RT-DETR with HGNetv2
Core Design	CNN-based one-stage detector with end-to-end NMS-free detection	Transformer-based end-to-end object detector
Backbone	Conv, C2f, and SCDown modules	High-Performance GPU Network v2
Neck	FPN-PAN with SPPF for multi-scale feature fusion	Hybrid Encoder with AIFI and CCFM modules
Detection Head	Decoupled NMS-free detection head	Transformer Decoder with object queries
Detection Strategy	Dense prediction with dual assignment	Query-based object prediction
Feature Representation	Multi-scale convolutional feature extraction and fusion	Hybrid CNN–Transformer representation with global contextual modeling

Table 1 summarizes the key architectural differences between YOLOv10 and RT-DETR with the HGNetv2 backbone. YOLOv10 employs a CNN-based architecture

with an end-to-end NMS-free detection head, whereas RT-DETR combines the HGNetv2 backbone, a Hybrid Encoder, and a Transformer decoder for end-to-end object detection. To quantitatively evaluate the performance of these architectures, several standard object detection metrics were employed. In addition, confusion matrices were analyzed to provide a detailed assessment of class-wise prediction performance.

2.4. Normalized Confusion Matrix

In multi-class object detection, evaluating the prediction performance of individual classes is essential for identifying classification errors and assessing the overall reliability of the detection model. To provide a detailed class-wise analysis, the confusion matrix is commonly employed as a complementary evaluation tool. By examining the relationship between actual and predicted class labels, it helps reveal both correctly classified samples and misclassification patterns among different classes.

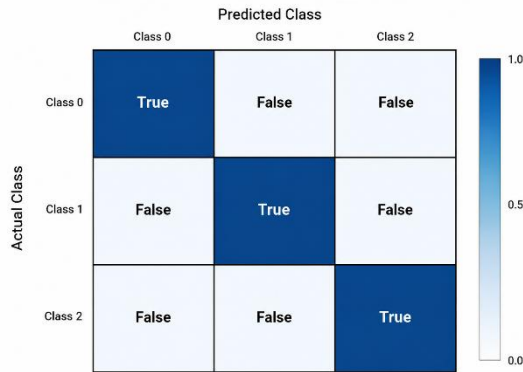


Figure 3. Normalized Confusion Matrix of Classification Results

The confusion matrix summarizes the prediction results for each class by comparing the predicted labels with the corresponding ground-truth labels. It can be presented either as raw counts or in a normalized form. As shown in Figure 3, the normalized confusion matrix illustrates the relationship between ground-truth and predicted classes. The values are expressed as proportions ranging from 0 to 1, allowing a more intuitive comparison of classification performance across classes regardless of differences in sample size. Additionally, off-diagonal entries represent instances in which samples are incorrectly assigned to other classes, providing insight into class-specific confusion patterns.

2.5. Performance Metrics

To quantitatively assess the detection performance of the proposed models, standard evaluation metrics derived from the confusion matrix were employed. The confusion matrix compares the predicted labels with the corresponding ground-truth labels and summarizes the numbers of true positives (TP), false positives (FP), and

false negatives (FN). Based on these values, four widely adopted object detection metrics, namely Precision, Recall, F1-score, and mAP, were used to comprehensively evaluate the classification and localization performance of the YOLOv10 and RT-DETR models. The mathematical formulations and corresponding descriptions of these evaluation metrics are summarized in Table 2.

Table 2. Performance Metrics, Formulas, and Explanation

Performance Metrics	Mathematical Formulas	Explanation
Precision (P)	$\frac{TP}{TP + FP}$	Measures the accuracy of positive predictions by indicating how many detected objects are correctly classified
Recall (R)	$\frac{TP}{TP + FN}$	Measures the ability of the model to identify all relevant objects in the dataset
F1-score	$\frac{2 \times P \times R}{P + R}$	Represents the harmonic mean of precision and recall, balancing both metrics in a single score
mAP	$\frac{1}{N_c} \times \sum_{i=1}^{N_c} AP_i$	Represents the mean detection performance by averaging the average precision across all object classes.

Average Precision (AP) measures the detection performance for a single object class by calculating the area under the precision–recall (P–R) curve [28]. The mAP is obtained by averaging the AP values across all object classes. Depending on the evaluation protocol, mAP can be calculated at a single IoU threshold, such as mAP@50 (IoU = 0.50), or averaged over multiple IoU thresholds ranging from 0.50 to 0.95 with a step size of 0.05 (mAP@50–95), providing a more comprehensive assessment of detection performance.

2.6. YOLO and RT-DETR Variants

YOLOv10 is provided in multiple model variants, including nano (N), small (S), medium (M), large (L), and extra-large (X). Although these variants share the same network architecture, they differ in terms of network depth, width, number of parameters, and computational complexity. Smaller variants are computationally efficient and well suited for resource-constrained environments, whereas larger variants offer greater feature representation capacity and generally achieve higher detection accuracy on more challenging datasets. The RT-DETR family includes large and extra large variants, offering different trade-offs between detection accuracy and computational efficiency. Unlike the YOLO family, RT-DETR does not offer smaller model scales in its official model family, as its architecture primarily targets high-performance object detection.

3. Experimental Results

All experiments were conducted in the Google Colab environment. The experimental configurations adopted in this study are presented in Table 3. All models were optimized using the AdamW optimizer with an initial

learning rate of 0.001 and a momentum value of 0.937. Weight decay was applied through parameter groups, with zero decay assigned to bias and normalization-related parameters and a decay factor of 0.0001 applied to weight parameters. The batch size was fixed at 16 throughout all experiments. The models were trained for 50 and 100 epochs using the same optimization strategy to ensure a fair and unbiased comparison among the evaluated models.

Table 3. Experimental setup configuration for all models

Component	Specification
Operating System	Google Colab (Linux)
GPU	NVIDIA Tesla T4 (15 GB VRAM)
Python	3.12.13
PyTorch	2.11.0 + CUDA 12.8 (torch-2.11.0+cu128)
CUDA	CUDA 12.8
Ultralytics	V8.4.105

To ensure a fair and consistent comparison, all baseline models were initialized using the official pretrained weights provided by the Ultralytics framework (v8.4.105). Each model retained its original network architecture without any architectural modifications. All models were trained and evaluated under identical experimental conditions.

The comparative analysis of the evaluated models in terms of the number of parameters and FLOPs is presented in Table 4. The evaluated models include the YOLOv10-X, YOLOv10-L, YOLOv10-N, RT-DETR-L, and RT-DETR-X architectures, representing different levels of model capacity and computational complexity. All models were initialized using official pretrained weights and trained on the same weld defect dataset under identical experimental conditions and hyperparameter settings. The architectural characteristics provide insights into the trade-offs between model complexity and computational efficiency. The comparative evaluation results based on standard object detection metrics are presented in the following section.

For clarity and to facilitate reproducibility, the exact numbers of images and corresponding approximate percentage allocations for the training, validation, and test sets under each split configuration are summarized in Table 5, providing a detailed overview of the dataset

distributions used across all evaluations.

Table 4. Architectural Characteristics of the YOLOv10 and RT-DETR

Model	Pre-trained	Layers	Params (M)	Flops (G)
YOLOv10-N	yolov10n	272	2.70	8.4
YOLOv10-L	yolov10l	365	25.77	127.2
YOLOv10-X	yolov10x	401	31.66	171.0
RT-DETR-L	rt detr-l	470	32.81	108.0
RT-DETR-X	rt detr-x	592	67.3	232.3

Table 5. WD Dataset Configurations and Split Ratios Used Across All Evaluations

	Configuration 1	Configuration 2	Configuration 3
Train–Validation–Test Ratio	46:3:1	46:6:1	46:9:1
Training Images	4,797	4,490	4,246
Validation Images	279	586	830
Test Images	122	122	122
Training (%)	92.3	86.4	81.7
Validation (%)	5.4	11.3	16.0
Test (%)	2.3	2.3	2.3

Following training under identical experimental conditions, the performance of all models was evaluated using standard object detection metrics. The comparative quantitative results, detailed performance analysis, and qualitative evaluations are presented in the following section.

The YOLOv10-N model was trained and evaluated under identical experimental settings using an input image size of 640×640 pixels. The dataset was divided into training, validation and test sets using three different split ratios: 46:3:1, 46:6:1, and 46:9:1. Model performance was quantitatively assessed using standard object detection metrics, while qualitative comparisons were conducted using representative validation images. The confusion matrices obtained from all YOLOv10-N experiments are presented in Figure 4.

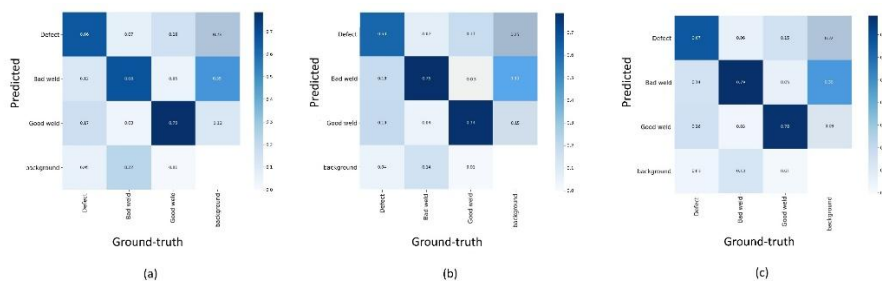


Figure 4. Normalized Confusion Matrices of the YOLOv10-N under different dataset split ratios: (a) 46:3:1, (b) 46:6:1, and (c) 46:9:1

Among the evaluated dataset split ratios, the 46:9:1 configuration achieved the best overall classification

performance. It produced the highest diagonal values for all three defect classes, indicating improved class discrimination and reduced inter-class confusion. In contrast, the 46:6:1 split showed a noticeable decrease in the Defect class accuracy, while the 46:3:1 split yielded comparatively lower overall performance. These results demonstrate that the 46:9:1 split provides the most balanced and reliable classification performance for YOLOv10-N. Following this confusion matrix analysis, a comprehensive quantitative evaluation was performed using the Precision, Recall, and mAP@50–95 metrics. The test results of the five evaluated object detection models, namely YOLOv10-N, YOLOv10-L, YOLOv10-X, RT-DETR-L, and RT-DETR-X, under identical experimental conditions are summarized below.

Table 6. Comparative Precision, Recall, and mAP@50–95 Performance of YOLO- and RT-DETR-Based Models

Model	Epoch	Precision	Recall	mAP@50-95
YOLOv10-N	50	0.691	0.574	0.369
YOLOv10-N	100	0.706	0.644	0.401
YOLOv10-L	50	0.722	0.614	0.416
YOLOv10-L	100	0.759	0.699	0.448
YOLOv10-X	50	0.719	0.601	0.410
YOLOv10-X	100	0.768	0.702	0.453
RT-DETR-L	50	0.681	0.695	0.405
RT-DETR-L	100	0.739	0.715	0.450
RT-DETR-X	50	0.690	0.679	0.432
RT-DETR-X	100	0.751	0.709	0.458

Table 6 summarizes the quantitative performance of the five evaluated models under both 50- and 100-epoch training settings. Among these models, YOLOv10-N consistently exhibited the lowest overall detection performance across the evaluated metrics. In contrast, YOLOv10-X and RT-DETR-X achieved the most competitive overall results. YOLOv10-X attained the highest precision (0.768), whereas RT-DETR-X achieved

the highest mAP@50–95 (0.458). Notably, RT-DETR-L produced the highest recall (0.715), indicating its superior ability to identify defect instances. Although YOLOv10-L and RT-DETR-L also demonstrated strong detection performance, their overall results were slightly inferior to those of YOLOv10-X and RT-DETR-X. Therefore, the subsequent analyses primarily focus on the comparison among YOLOv10-N, YOLOv10-X, and RT-DETR-X, including the F1-score–confidence curves, Precision–Recall curves, qualitative detection results, and explainability evaluations. YOLOv10-N was included as the baseline model, whereas YOLOv10-X and RT-DETR-X were selected because they represent the highest-performing YOLO-based and Transformer-based object detection architectures, respectively.

Figure 5 presents the F1-score–confidence curves for the selected models under identical training and hyperparameter settings after 50 epochs. Among the three models, RT-DETR-X achieved the highest overall F1-score, reaching 0.68 at a confidence threshold of 0.572, indicating superior balance between precision and recall. YOLOv10-X ranked second with a maximum F1-score of 0.65 at a confidence threshold of 0.374. YOLOv10-N achieved peak F1-scores of 0.63 at confidence thresholds of 0.266. Class-wise analysis further shows that the Good Weld category consistently attained the highest F1-score across all models, whereas the Defect class remained the most challenging, exhibiting substantially lower F1 values throughout the confidence range. In addition, RT-DETR-X maintained comparatively higher and more stable F1 performance over a broader confidence interval, suggesting greater robustness to confidence threshold selection than the YOLOv10 variants.

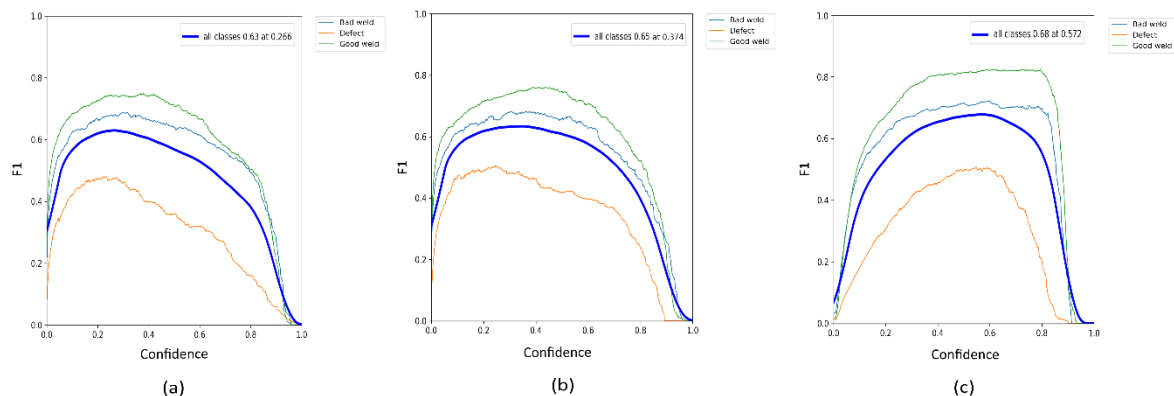


Figure 5. F1-score–confidence curves for the selected models under identical hyperparameter settings: (a) YOLOv10-N, (b) YOLOv10-X, and (c) RT-DETR-X

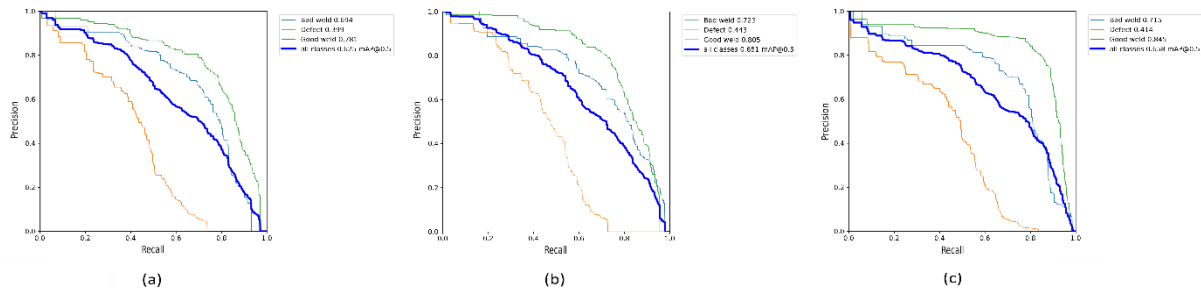


Figure 6. Precision-Recall curves for the selected models under identical hyperparameter settings: (a) YOLOv10-N, (b) YOLOv10-X, and (c) RT-DETR-X

Figure 6 presents the Precision–Recall (PR) curves for the selected object detection models under identical training and hyperparameter settings after 50 epochs. Overall, RT-DETR-X achieved the highest detection performance, obtaining an overall mAP@0.50 of 0.658, followed by YOLOv10-X (0.651) and YOLOv10-N (0.625). Class-wise analysis indicates that the Good Weld category consistently produced the highest average precision across all models, increasing from 0.781 for YOLOv10-N to 0.805 for YOLOv10-X and reaching 0.845 with RT-DETR-X. Similarly, the Bad Weld class demonstrated gradual improvement, with AP values of 0.694, 0.723, and 0.715, respectively. In contrast, the Defect category remained the most challenging class, although its AP increased from 0.399 (YOLOv10-N) to 0.443 (YOLOv10-X) and 0.414 (RT-DETR-X). Furthermore, the PR curve of RT-DETR-X remained closer to the upper-right corner over a wider recall range, indicating a more favorable balance between precision and recall. These results demonstrate that the Transformer-based RT-DETR-X provides superior overall localization and classification performance on the weld defect dataset while maintaining more robust detection characteristics across varying recall levels.

Figure 7 presents a qualitative comparison of the detection results produced by YOLOv10-X and RT-DETR-X on representative weld defect samples. Overall, both models successfully identify the primary defect regions; however, several notable differences are observed. In sample (a), YOLOv10-X produces duplicate detections for the same defect, whereas RT-DETR-X generates a single, well-localized bounding box. In sample

(c), RT-DETR-X introduces an additional false-positive detection by identifying a non-existent defect in the lower image region, while YOLOv10-X correctly localizes only the actual defect. In samples (b) and (d), both models accurately detect the weld defects with comparable localization performance. In sample (e), both models identify the defective weld, although YOLOv10-X predicts an additional overlapping bounding box, resulting in redundant detections. These observations indicate that YOLOv10-X is more susceptible to duplicate detections, whereas RT-DETR-X exhibits a tendency to generate occasional false-positive predictions in challenging background regions. Despite these limitations, both models demonstrate reliable qualitative detection performance across representative welding scenarios.

As shown in Figure 8, YOLOv10-X generally produces more localized activation maps, whereas RT-DETR-X captures defect regions more comprehensively. In particular, samples (d) and (e) demonstrate that RT-DETR-X focuses more consistently on the weld defect, while YOLOv10-X exhibits either scattered background attention (d) or limited attention confined to a small portion of the weld joint (e). These results suggest that RT-DETR-X provides more coherent explanations in challenging welding scenarios. However, Grad-CAM visualizations may not always accurately reflect the actual decision-making process of the model, and some activation patterns may not correspond precisely to the true defect regions. Therefore, these visualizations should be interpreted as complementary evidence rather than definitive representations of model reasoning.

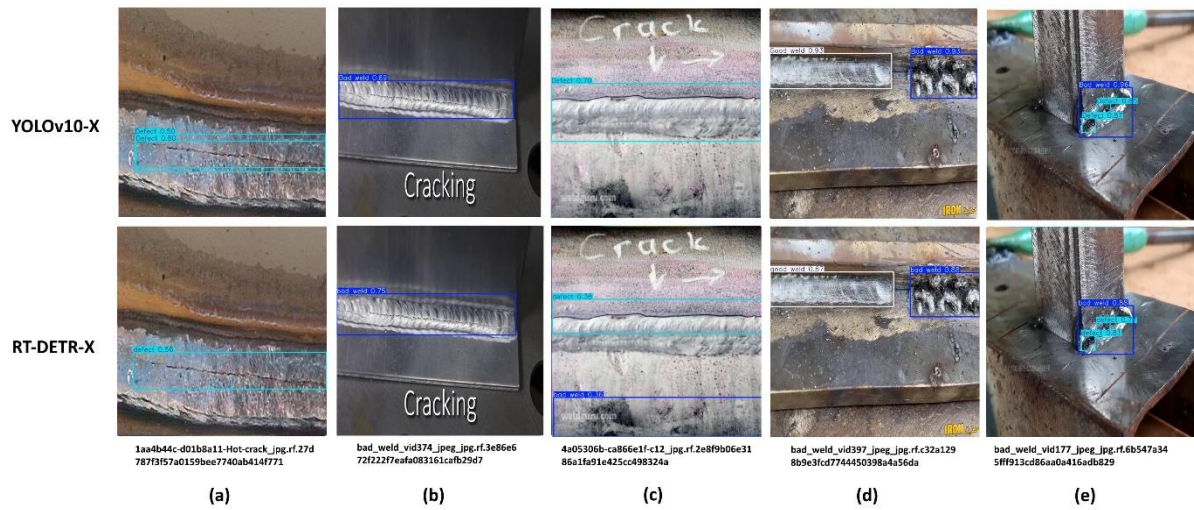


Figure 7. Qualitative comparison of object detection results obtained by YOLOv10-X and RT-DETR-X on representative weld defect images.

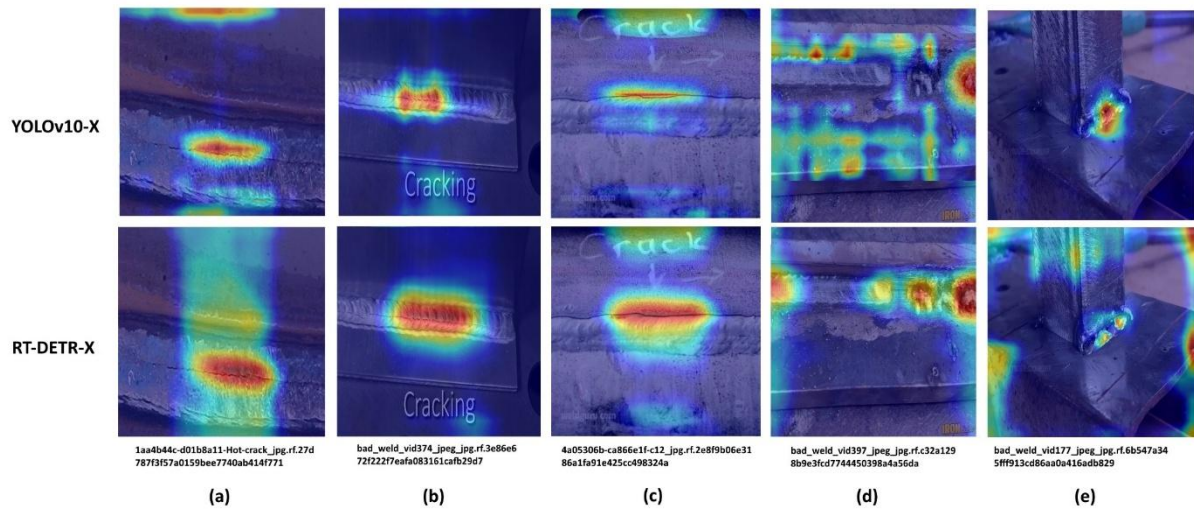


Figure 8. Comparative Grad-CAM visualizations of YOLOv10-X and RT-DETR-X on representative weld defect images.

4. Results and Discussion

The present study conducted a comprehensive comparative evaluation of five object detection models, namely YOLOv10-N, YOLOv10-L, YOLOv10-X, RT-DETR-L, and RT-DETR-X, under identical experimental conditions using the WD dataset. Among the evaluated models, RT-DETR-X achieved the strongest overall detection performance, outperforming YOLOv10-X by approximately 1.1% in relative terms $mAP@50-95$ while also attaining the highest F1-score. As illustrated by the Precision–Recall curves (Figure 6), scaling the YOLOv10 architecture from the N to the L variant resulted in a 4.2% improvement in $mAP@50$, followed by an additional 1.1% improvement with the X variant. Similarly, the F1–confidence analysis (Figure 5) showed that the optimal F1-score increased by 3.2% from YOLOv10-N to YOLOv10-L and by a further 4.6% from YOLOv10-L to YOLOv10-X, yielding an overall improvement of approximately 7.9% across the YOLOv10 family. These findings demonstrate that increasing the model capacity

consistently improved the balance between precision and recall while enhancing detection robustness.

The analysis of different dataset partitioning strategies demonstrated that the 46:9:1 training–validation split provided the most reliable classification performance for YOLOv10-N. Specifically, the classification accuracy of the Bad Weld class improved by 10.3% when the training–validation split was changed from 46:3:1 to 46:6:1, followed by a further 5.3% improvement under the 46:9:1 configuration. Overall, the classification accuracy of the Bad Weld class increased by 16.2% compared with the initial configuration. Among the evaluated split configurations, the 46:9:1 split achieved the highest observed classification accuracy while maintaining comparable overall detection performance.

The comparative evaluation further revealed different characteristics of the YOLOv10 and RT-DETR architectures. RT-DETR-X consistently achieved superior localization performance, as reflected by the highest $mAP@50-95$ and the most stable F1-score across different confidence thresholds. In contrast, YOLOv10-X

demonstrated the highest precision, indicating fewer false-positive predictions overall. These findings suggest that the Transformer-based architecture benefits from stronger global contextual modeling, whereas the CNN-based YOLO architecture provides highly reliable object classification with efficient feature extraction.

The quantitative results reveal distinct characteristics of the YOLOv10 and RT-DETR architectures. RT-DETR-X consistently achieved superior localization performance, as reflected by the highest mAP@50–95 and the most stable F1-score across different confidence thresholds. In contrast, YOLOv10-X achieved the highest precision, indicating fewer false-positive predictions while maintaining competitive overall accuracy. These findings suggest that the Transformer-based architecture benefits from stronger global feature modeling, enabling more accurate localization of complex and elongated weld defects, whereas the YOLO-based architecture provides reliable classification through efficient feature extraction.

The Grad-CAM visualizations (Figure 8) provide additional insight into the decision-making behavior of the evaluated models. YOLOv10-X generally produced localized activation maps concentrated around the predicted defect regions. In contrast, RT-DETR-X generated more spatially coherent attention maps covering a larger portion of the weld defect, particularly in challenging samples with irregular defect boundaries. For example, RT-DETR-X consistently focused on the complete defective weld region in representative samples, whereas YOLOv10-X occasionally produced fragmented activations or concentrated on only a limited portion of the defect. These observations indicate that the Transformer-based architecture captures global contextual information more effectively, resulting in more comprehensive and spatially consistent visual explanations. Nevertheless, a more comprehensive assessment of model interpretability could be achieved by comparing Grad-CAM with additional CAM methods, such as Grad-CAM++, Eigen-CAM, or related XAI approaches. Such a multi-method comparison would provide a more robust evaluation of the consistency and reliability of the highlighted defect regions and strengthen the interpretability analysis of the evaluated models.

Overall, the experimental results demonstrate that both YOLOv10 and RT-DETR provide reliable solutions for automated weld defect detection. RT-DETR-X achieved the highest overall detection and localization performance, whereas YOLOv10-X offered competitive accuracy together with the highest precision. The complementary quantitative, qualitative, and explainability analyses consistently support these findings, providing practical guidance for selecting appropriate object detection architectures according to the accuracy requirements and deployment constraints of industrial weld inspection systems. Future work will focus on validating the proposed

framework on additional weld inspection datasets and improving its suitability for practical industrial deployment through inference latency analysis, model compression, and hardware-specific optimization techniques.

5. Conclusions

This study presented a comprehensive experimental evaluation of YOLOv10 and RT-DETR architectures for automated weld defect detection using the publicly available Weld Defects (WD) dataset. Five models, namely YOLOv10-N, YOLOv10-L, YOLOv10-X, RT-DETR-L, and RT-DETR-X, were evaluated under identical training and evaluation conditions. In addition, the influence of different training–validation split ratios on YOLOv10 performance was investigated to assess their effect on model reliability.

The experimental results demonstrated that RT-DETR-X achieved the strongest overall detection performance, obtaining the highest mAP@50–95 (0.458) and F1-score (0.68), while YOLOv10-X achieved the highest precision (0.768). These findings indicate that RT-DETR-X provides stronger localization and a more favorable precision–recall balance, whereas YOLOv10-X offers an effective alternative when precision and false-positive reduction are prioritized.

The investigation of dataset partitioning showed that the 46:9:1 training–validation–test configuration provided the most reliable classification performance for YOLOv10-N. Compared with the original 46:3:1 configuration, the overall classification accuracy increased by 16.2%. These results suggest that the choice of training–validation split can influence the observed classification performance of the model.

The qualitative and explainability analyses further supported the quantitative results. RT-DETR-X generally produced more spatially coherent Grad-CAM activation maps and captured larger portions of defective weld regions, whereas YOLOv10-X generated more localized activations. Qualitative results also indicated that YOLOv10-X was more prone to duplicate detections, while RT-DETR-X occasionally produced false-positive predictions in challenging background regions.

Overall, both YOLOv10 and RT-DETR demonstrated their suitability for automated weld defect inspection, while exhibiting different strengths. RT-DETR-X provided the best overall localization and F1 performance, whereas YOLOv10-X achieved the highest precision. Therefore, model selection should be based on the specific requirements of the target industrial inspection system.

As future work, the evaluated models will be tested on additional weld-defect datasets and real-world industrial inspection scenarios to further assess their generalization and robustness. Future studies will also investigate

inference efficiency, model compression, and hardware-specific optimization to improve the suitability of the models for real-time industrial deployment. In addition, advanced explainable artificial intelligence techniques will be explored to provide deeper insights into model decision-making and to quantitatively assess the correspondence between model attention and actual weld-defect regions.

Declaration of Ethical Standards

The article does not contain any studies with human or animal subjects.

Credit Authorship Contribution Statement

Adem Dilbaz: Conceptualization, Methodology, Software, Investigation, Data Curation, Formal Analysis, Visualization, Writing – Original Draft, Writing – Review & Editing.

Declaration of Competing Interest

The author declares that there are no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding / Acknowledgements

This research received no external funding.

Data Availability

<https://universe.roboflow.com/yolov8-daldb/weld-defects-mlopr>

References

- [1] X. Chen, Y. Duan, Z. Chen, and X. Chen, "Multimodal defect detection in CMT welding based on melt pool images and acoustic signals," in 2025 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2025, pp. 330–335.
- [2] G. Chudasama, S. Kashyap, J. Patel, V. Kalyankar, and J. Shewale, "AISI 2205 DSS for body-in-white automotive structure: Effect of welding current and time on the performance of spot welded joint," *Welding International*, vol. 36, no. 12, pp. 769–778, 2022.
- [3] V. Gaidhane and S. Kolhe, "Enhanced weld defect detection through advanced image preprocessing techniques," in 2025 International Conference on Electronics, Computing, Communication and Control Technology (ICECCC), 2025, pp. 1–6.
- [4] F. Zuo, J. Liu, M. Fu, L. Wang, and Z. Zhao, "An efficient anchor-free defect detector with dynamic receptive field and task alignment," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 6, pp. 8536–8547, 2024.
- [5] B. Xiao, Y. Yang, Y. He, and G. Ma, "Multi-type weld defect detection in galvanized sheet MIG welding using an improved YOLOv10 model," *Materials*, vol. 19, no. 6, Art no. 1178, 2026.
- [6] C. Zhang, C. Xu, W. Shan, and Z. Han, "DBP-YOLO: A dynamic bifurcated-path fusion network for weld seam defect detection," *IEEE Access*, vol. 13, pp. 163749–163768, 2025, doi: 10.1109/ACCESS.2025.3609890
- [7] K. Parmar, A. Biber, M. Olesch, R. Sharma, and U. Reisgen, "Leveraging thermal images for automatic surface defect detection in TIG welding process of thin metal sheets," *Welding in the World*, vol. 1, pp. 1–17, 2026.
- [8] X. Zhang, H. Chen, Z. Liu, and S. Anli, "Lightweight DETR algorithm for X-ray weld defect detection," *Scientific Reports*, vol. 15, Art no. 41873, 2025, doi: 10.1038/s41598-025-25916-2.
- [9] S. Peng, H. Bao, J. Li, and J. Gao, "YOLO-PDR: A lightweight defect detection network for hot-pressed light guide plates based on parallel texture fusion and multi-scale feature expression," *Measurement Science and Technology*, vol. 37, no. 12, p. 125902, 2026.
- [10] R. Zhang, C. Pan, Z. Chen, J. Fan, Z. Chen, and B. Yuan, "An improved lightweight YOLOv11 algorithm for weld surface defect detection," *Scientific Reports*, vol. 16, no. 1, Art no. 11440, 2026.
- [11] K. Mafuwe, R. V. S. Dulam, C. Kambhamettu, and M. S. Crossley, "Comparative hyperparameter optimization of object detection models for precision monitoring of cucumber beetles and similar insects on yellow sticky cards," *Scientific Reports*, vol. 16, Art no. 22099, 2026, doi: 10.1038/s41598-026-51483-1.
- [12] C. Guo, Y. Du, Y. Yang, S. Wang, D. Liu, and M. Cong, "MCTNet: Defect detection in aluminum alloy die-casting using a multilevel CNN-transformer interaction network," *IEEE Transactions on Instrumentation and Measurement*, vol. 74, pp. 1–11, 2025, doi: 10.1109/TIM.2025.3589704.
- [13] Y. Zhang, H. Zhang, T. Ma, Z. Xu, J. Jiang, and F. Xue, "Weld-YOLO: A YOLOv11-based model for high accuracy defect detection in weld X-ray images," in *Proc. SPIE*, vol. 14247, 2026, Art. no. 1424714.
- [14] D. Yang, H. Jiang, H. Yang, Y. Wang, Z. Liu, X. Zhang, and H. Cheng, "Casting defect-detection method based on multi-image feature of fusion artificial intelligence model," *Applied Intelligence*, vol. 55, no. 11, p. 774, 2025.
- [15] W. Zhang, J. Liu, Z. Yan, M. Zhao, X. Fu, and H. Zhu, "FC-YOLO: An aircraft skin defect detection algorithm based on multi-scale collaborative feature fusion," *Measurement Science and Technology*, vol. 35, no. 11, p. 115405, 2024.
- [16] M. Huan, T. Wang, Q. Xu, L. Gao, Z. Lu, X. Shi, M. Qian, L. Jia, and C. Yao, "An efficient lightweight object detection algorithm for defect identification in Paeoniae Radix Alba with rapid area and diameter estimation using OpenCV," *Food Science & Nutrition*, vol. 13, no. 9, Art. no. e70947, 2025.
- [17] A. Dilbaz and İ. A. Ozkan, "Simulation-based spot welding inspection on automotive chassis using YOLO-powered image processing," *International Journal of Automotive Engineering and Technologies*, vol. 14, no. 3, pp. 170–180, 2025, doi: 10.18245/ijaet.1729908.
- [18] A. Dilbaz, "YOLOv12 and YOLOv13 for steel surface defect detection: Experimental evaluation," *Intelligent Methods in Engineering Sciences*, vol. 5, no. 2, pp. 55–68, 2026, doi: 10.58190/imiens.2026.175.
- [19] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "YOLOv10: Real-time end-to-end object detection," *Advances in Neural Information Processing Systems*, vol. 37, pp. 107984–108011, 2024, doi: 10.52202/079017-3429.
- [20] Roboflow Universe, "Weld defects dataset," 2024. [Online]. Available: <https://universe.roboflow.com/yolov8-daldb/weld-defects-mlopr> [Accessed: Aug. 1, 2026]
- [21] L. Jia, Y. Zhu, and B. Li, "DPCNet: A dual path cross perception network for small object detection in UAV imagery," *PLOS ONE*, vol. 21, no. 3, Art no. e0344091, 2026.
- [22] B. İnan, İ. A. Özkan, and N. Doğan, "Deep learning-based skin disease detection: Comparative performance analysis of YOLOv8 and YOLOv11 models," *International Journal of Applied Methods in Electronics and Computers*, vol. 13, no. 4, pp. 121–130, 2025.
- [23] S. Haitao, Z. Li, X. Xiaoxue, S. Wang, and Y. Zhang, "Rapid identification of mangrove leaves based on improved YOLOv10 model," *Forests*, vol. 16, no. 7, Art no. 1068, 2025.
- [24] Y. Shen and C. Zhang, "Research on the application of YOLOv10n in helmet detection," *IAENG International Journal of Computer Science*, vol. 53, no. 2, pp. 721–729, 2026.
- [25] M. Hussain, "YOLOv5, YOLOv8 and YOLOv10: The go-to detectors for real-time vision," *arXiv preprint arXiv:2407.02988*, 2024.
- [26] L. Liu, R. Yuan, Y. Lv, W. Bo, X. Hu, and E. D. Gedikli, "An efficient high-resolution remote sensing object detection approach using a multilevel optimized RT-DETR model," *IEEE Sensors Journal*, vol. 25, no. 22, pp. 42306–42315, 2025.

- doi: 10.1109/JSEN.2025.3617485.
- [27] Q. Chen, X. Su, X. Zhang, J. Wang, J. Chen, Y. Shen, C. Han, Z. Chen, W. Xu, F. Li, and S. Zhang, "LW-DETR: A transformer replacement to YOLO for real-time detection," arXiv preprint arXiv:2406.03459, 2024.
- [28] Q. Wang, Y. Li, B. Luo, Z. Tu, and C. Li, "Dual-correlation hypergraph network for unaligned RGBT video object detection and a large-scale benchmark," arXiv preprint arXiv:2607.08191, 2026.
- [29] F. Liu, F. Wang, M. Liu, and D. Zhang, "RT-DETR-Crack: A tunnel lining crack detection method via multiscale feature enhancement and spatial attenuation mechanism," *IAENG International Journal of Applied Mathematics*, vol. 56, no. 4, pp. 1593–1606, 2026.