Universidade de Brasília
Faculdade de Tecnologia

DIGITAL TWIN–ENABLED QUALITY ASSURANCE ANALYSIS OF ROBOTIC CW-GTAW METAL ADDITIVE MANUFACTURING BASED ON DEEP LEARNING APPLIED TO 3D MESHES

João Vítor Arantes Cabral

DISSERTAÇÃO DE MESTRADO

PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS MECATRÔNICOS

Brasilía

2026

Universidade de Brasília
Faculdade de Tecnologia

ANÁLISE DE GARANTIA DA QUALIDADE NA MANUFATURA ADITIVA DE METAIS POR MÉTODO TIG-MADA ROBÓTICA BASEADA EM APRENDIZAGEM PROFUNDA APLICADA A MODELOS 3D HABILITADA POR GÊMEO DIGITAL

João Vítor Arantes Cabral

Dissertação de Mestrado submetida ao Departamento de Engenharia Mecânica da Universidade Brasília como parte dos requisitos necessários para a obtenção do grau de Mestre.

Advisor: Prof. Dr. Alberto José Álvares

Brasilía
2026

Cabral, João Vítor Arantes.
C117d DIGITAL TWIN–ENABLED QUALITY ASSURANCE ANALYSIS OF ROBOTIC CW-GTAW METAL ADDITIVE MANUFACTURING BASED ON DEEP LEARNING APPLIED TO 3D MESHES / João Vítor Arantes Cabral; orientador Alberto José Álvares. –– Brasilía, 2026.
156 p.

Dissertação de Mestrado em Programa de Pós-Graduação em Sistemas Mecatrônicos –– Universidade de Brasília, 2026.

1. Manufatura Aditiva. 2. Redes Neurais. 3. Malha baseada em nuvem de pontos. 4. Indústria 4.0. I. Álvares, Alberto José, orient. II. DIGITAL TWIN–ENABLED QUALITY ASSURANCE ANALYSIS OF ROBOTIC CW-GTAW METAL ADDITIVE MANUFACTURING BASED ON DEEP LEARNING APPLIED TO 3D MESHES

Universidade de Brasília
Faculdade de Tecnologia

DIGITAL TWIN–ENABLED QUALITY ASSURANCE ANALYSIS OF ROBOTIC CW-GTAW METAL ADDITIVE MANUFACTURING BASED ON DEEP LEARNING APPLIED TO 3D MESHES

João Vítor Arantes Cabral

Dissertação de Mestrado submetida ao Departamento de Engenharia Mecânica da Universidade Brasília como parte dos requisitos necessários para a obtenção do grau de Mestre.

Trabalho aprovado. Brasilía, 17 de março de 2026:

Prof. Dr. Alberto José Álvares,
UnB/FT/ENM
Orientador

Prof. Dr. Maksym Ziberov, UnB/FT/ENM
Examinador interno

Prof. Dr. Juan Sebastian Toquica Arenas,
UFC
Examinador externo

Brasilía
2026

This work is dedicated to my grandfather Lázaro Arantes and my grandmother, Alair Furquim. Your wisdom, dedication and perseverance helped me mold my character.

Acknowledgements

First and foremost, I want to thank God for giving me the strength and perseverance to complete this work.

I would like to thank my family for all the support, especially my mother Meire Arantes and my father Alyrio Cabral for immensely helping me throughout my journey and providing the conditions so I could pursue my dreams.

I would like to express my deepest thanks to my girlfriend Victoria Azucena, for all the support and motivation she gave me throughout the entire trajectory from my first scientific initiation until the conclusion of this work.

I am deeply grateful to Professor Álvares for his great influence on my journey as an aspiring scientist. I also thank him for the invaluable guidance throughout the development of this work.

Lastly, I would like to thank the support of the Fundação de Apoio à Pesquisa do Distrito Federal (FAPDF) and the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) for funding the development of this project.

For the purpose of grammar correction and text refinement only, Grammarly software version 1.2.235 and Gemini version 3.1 Pro were used throughout the development of this work.

Abstract

In the context of Industry 4.0, advanced paradigms for manufacturing and data-driven monitoring have emerged. Among these, Wire Arc Additive Manufacturing (WAAM) has revolutionized the production of metal components. To fully leverage this technology, Cyber-Physical Systems (CPS) and Digital Twins (DT) are increasingly employed to enable Artificial Intelligence (AI)-powered analysis of process data. However, few works have dealt with a comprehensive data analysis, based on Digital Twin systems, to study quality levels of manufactured parts using 3D models. With this background in mind, this current project uses a Digital Twin-enabled dataflow to constitute a basis for a proposed data analysis pipeline. The pipeline consists of analyzing metal AM-manufactured parts' surface roughness quality levels by the application of a Deep Neural Network (DNN) analytical model and enabling the assessment and tuning of deposition parameters by comparing AM-built models' 3D representation, obtained by photogrammetry scanning, with the positional data acquired during the deposition process and stored in a cloud database. Stored and analyzed data may be further used to refine the manufacturing of parts, calibration of sensors and refining of the DT model. Also, this work presents a comprehensive study on experiments carried out using the CW-GTAW (Cold Wire Gas Tungsten Arc Welding) process as the means of depositing metal, resulting in hollow parts whose geometries were evaluated by means of both 3D scanned data, obtained via photogrammetry, and positional/deposition process parameters obtained from the Digital Twin architecture pipeline, with a final step of validation on desktop laser-scanned based point clouds. Finally, an adapted PointNet DNN model was evaluated using two distinct training approaches: one utilizing simulated 3D models and another using real scanned data. While the simulated model achieved a validation accuracy of 90.62%, it exhibited limitations in generalizing one of the three quality classes; in contrast, the model trained on real deposited metal parts demonstrated robust generalization, obtaining an overall accuracy of 75.64% on the evaluation of surface roughness quality levels into 3 classes (good, fair, and poor).

Keywords: Additive Manufacturing. Neural Networks. Point cloud-based mesh. Industry 4.0.

Resumo

No contexto da Indústria 4.0, surgiram paradigmas avançados para a fabricação e monitoramento baseado em dados. Entre eles, a Manufatura Aditiva por Arco Elétrico (WAAM) revolucionou a produção de componentes metálicos. Para aproveitar ao máximo essa tecnologia, os Sistemas Ciberfísicos (CPS) e os Gêmeos Digitais (DT) são cada vez mais empregados para permitir a análise de dados de processo com base em Inteligência Artificial (IA). No entanto, poucos trabalhos abordaram uma análise de dados abrangente, baseada em sistemas de Gêmeos Digitais, para estudar os níveis de qualidade de peças manufaturadas utilizando modelos 3D. Com esse cenário em mente, o presente projeto utiliza um fluxo de dados habilitado por Gêmeo Digital para constituir a base de um pipeline de análise de dados proposto. O pipeline consiste em analisar os níveis de qualidade da rugosidade superficial de peças metálicas fabricadas por manufatura aditiva através da aplicação de um modelo analítico de Rede Neural Profunda (DNN), permitindo a avaliação e o ajuste dos parâmetros de deposição ao comparar a representação 3D dos modelos construídos, obtida por digitalização via fotogrametria, com os dados posicionais adquiridos durante o processo de deposição e armazenados em um banco de dados em nuvem. Os dados armazenados e analisados podem ser utilizados posteriormente para refinar a fabricação de peças, a calibração de sensores e o refinamento do modelo do DT. Além disso, este trabalho apresenta um estudo abrangente sobre experimentos realizados utilizando o processo CW-GTAW (Cold Wire Gas Tungsten Arc Welding) como meio de deposição de metal, resultando em peças ocas cujas geometrias foram avaliadas tanto por meio de dados digitalizados em 3D (fotogrametria) quanto por parâmetros de processo e posicionais obtidos a partir da arquitetura do Gêmeo Digital, com uma etapa final de validação baseada em nuvens de pontos adquiridas via scanner 3D de bancada a laser. Finalmente, um modelo DNN PointNet adaptado foi avaliado utilizando duas abordagens distintas de treinamento: uma utilizando modelos 3D simulados e outra utilizando dados digitalizados reais. Enquanto o modelo simulado alcançou uma precisão de validação de 90,62%, exibiu limitações na generalização de uma das três classes de qualidade; em contraste, o modelo treinado com peças metálicas reais depositadas demonstrou uma generalização robusta, obtendo uma precisão geral de 75,64% na avaliação dos níveis de qualidade de rugosidade superficial divididos em 3 classes (bom, regular e ruim).

Palavras-chave: Manufatura Aditiva. Redes Neurais. Malha baseada em nuvem de pontos. Indústria 4.0.

List of Figures

List of Tables

List of abbreviations and acronyms

ABB	ASEA Brown Boveri	43
AI	Artificial Intelligence	6
AM	Additive Manufacturing	19
CAD	Computer-aided Design	19
CAPP	Computer-Aided Process Planning	60
CPS	Cyber-Physical System	19
CW-GTAW	Cold-Wire Gas Tungsten Arc Welding	20
DBSCAN	Density-Based Spatial Clustering of Applications with Noise	35
DED	Direct Energy Deposition process	41
DNN	Deep Neural Network	88
DT	Digital Twin	19
DTh	Digital Thread	32
ETL	Extraction, transformation, load	47
FDM	Fusion Deposition Modeling	6
FFF	Fused Filament Fabrication	43
GTAW	Gas Tungsten Arc Welding	19
HDF5	Hierarchical Data Format 5	88
HMI	Human-Machine Interfaces	19
ICP	Iterative Closest Point	36
LiDAR	Light Detection and Ranging	34
LMD	Laser Metal Deposition	6
ML	Machine Learning	19
NKF	Neural Kernel Fields	34
NKSR	Neural Kernel Surface Reconstruction	35
QA	Quality Assurance	20
RA-CNN	Residual-attention CNN	44
RMSE	Root Mean Square Error	57

SfM	Structure-from-Motion	33
SGT	Spectral Graph Theory	43
SPSR	Screened Poisson Surface Reconstruction	54
SVM	Support Vector Machine	44
TCP	Tool Center Point	37
TS	Torch Speed	62
WAAM	Wire Arc Additive Manufacturing	19
WFS	Wire Feed Speed	62

Contents

1 Introduction

Driven by the advancements of Industry 4.0, a convergence of technologies such as the Internet of Things (IoT), big data, and Artificial Intelligence (AI) has been integrated into industrial environments. In parallel, innovative manufacturing paradigms and techniques have emerged within the realm of Additive Manufacturing (AM), including Wire Arc Additive Manufacturing (WAAM) and, specifically addressed in this study, metal 3D printing via Gas Tungsten Arc Welding (GTAW).

As highlighted by Pramanik et al. (DUTTA PRAMANIK et al., 2019), foundational elements of the Industry 4.0 transformation include Cyber-Physical Systems (CPS), Digital Twins (DT), and the application of big data to industrial workflows. These are complemented by pervasive and intelligent edge computing, which facilitates the deployment of AI and Machine Learning (ML) for manufacturing analytics. The implementation of these pillars has led to comprehensive quality and resource oversight, enhanced Human-Machine Interfaces (HMI), and more effective decision-making processes (SANTOS, 2018).

In the context of big data analysis for Additive Manufacturing processes, numerous applications can be developed using conventional ML techniques. These include predictive maintenance and the identification of anomalies during production in real or near-real time (BONG KIM; SHAO; JO, 2022). Another use case within DT frameworks involves identifying defects in fabricated components by comparing them against predefined tolerances using CAD models as benchmarks. This investigation builds upon that concept, which will be elaborated in the following chapter.

Although Digital Twin (DT) technologies are increasingly adopted for real-time monitoring in manufacturing, their application in in-depth data analysis of fabricated components—particularly through 3D model evaluation—remains limited. To bridge this gap, the present study introduces an innovative data analysis framework grounded in a DT-integrated dataflow. The central focus of this methodology is the evaluation of surface roughness in metal parts produced via Additive Manufacturing (AM), using a Deep Neural Network (DNN) to interpret and classify quality metrics. This approach involves correlating high-resolution 3D reconstructions of the final part—generated through photogrammetry—with positional and process data stored in the DT cloud infrastructure. By aligning these datasets, the system enables precise defect detection and spatial localization, while also facilitating the refinement of deposition parameters based on historical process insights. The proposed pipeline thus supports iterative optimization throughout the fabrication cycle.

The study presents a detailed case analysis using the Cold-Wire Gas Tungsten Arc Welding (CW-GTAW) technique to manufacture hollow metal components. A customized

PointNet-based DNN is implemented to segment and classify point cloud data into three surface quality categories: good, fair, and poor. This classification framework offers a semi-automated solution for quality assurance in AM workflows.

Despite the growing interest in WAAM and DT technologies, there is a noticeable scarcity of research focused on DT-driven analysis of CW-GTAW-based additive processes. This work addresses that gap by demonstrating how DT-enabled data analytics can enhance precision, traceability, and quality control in metal AM, especially for geometrically complex parts.

1.1 Problem

Traditional QA methods for metal AM-fabricated parts often rely on subjective visual assessments or disconnected, time-consuming inspection routines that lack scalability. Currently, there is a notable gap in the literature concerning the seamless integration of automated, objective quality inspection mechanisms with the physical manufacturing parameters of the shop floor. The lack of an efficient, data-driven QA pipeline integrated within a standardized Digital TWin architecture hinders the rapid and reliable detection of anomalies, surface roughness variations, and geometric non-conformities in the post-production phase. The framework presented in this study, then, seeks to bridge the gap between physical manufacturing parameters and automated quality inspection by replacing subjective visual assessments with objective, spectral graph-theoretical metrics applied to photogrammetry-derived 3D meshes.

1.2 Objectives

1.2.1 General objective

The primary purpose of this work is to propose and implement a framework solution for the automated quality assurance (QA) of manufactured parts related to metal Additive Manufacturing, bridging the gap between physical manufacturing parameters and automated quality inspection.

To achieve this, the solution uses a pre-implemented DT dataflow to collect large batches of data from a cloud database, which are related to parameters obtained from a metal AM robotic cell. The collected data is then used in parallel with 3D meshes, derived from point clouds of the manufactured parts via photogrammetry, to evaluate and compare surface quality levels and geometric conformity. Specifically, the study aims to adapt a Deep Learning-based approach (PointNet) utilizing 3D topological analysis applied to these meshes to replace subjective visual assessments with objective, spectral graph-theoretical metrics.

The main objective of this analytical comparison is to accurately classify levels of anomalies in the manufactured parts, identify errors in the calibration of equipment, and filter corrupted data. This evaluation operates based on the seamless streaming of data throughout the domains of a Digital Twin architecture, implemented based on the ISO 23247 standard (ISO 23247-1, 2021).

1.2.2 Specific objectives

To achieve the proposed general objective, the following specific objectives were defined and executed:

- **Digital Twin Integration:** Utilize an existing Digital Twin architecture based on the ISO 23247 standard as a data acquisition gateway, establishing a pipeline that correlates recorded deposition process parameters with the resulting physical geometry of the manufactured parts.
- **3D Scanning Pipeline:** Establish an easy-to-use, low-cost 3D mesh extraction pipeline based on common hardware (a smartphone) to acquire "as-built" 3D meshes, utilizing Iterative Closest Point (ICP) algorithms to validate their alignment against the "as-planned" Digital Twin trajectories and respective referential CAD models.
- **Topological Ground Truth Definition:** Define a mathematically objective ground truth for a proxy for perceived superficial roughness and defect categorization by calculating the Fiedler Number (algebraic connectivity) of the 3D mesh graphs, quantifying topological complexity rather than relying solely on visual interpretation.
- **Deep Learning Model Adaptation:** Adapt and implement the PointNet Deep Neural Network architecture to classify 3D mesh segments into quality categories (Good, Fair, Poor) based on their geometric features.
- **Comparative Training Analysis:** Evaluate and compare the performance of the PointNet model using two distinct training strategies: one based on mathematically generated (simulated) 3D segments and another based on real photogrammetry data, assessing the generalization capabilities and limitations of each approach.

1.3 Structure of the document

The work is divided into the following chapters:

- **Chapter 2 (Literature Review):** Reviews the state-of-the-art regarding Artificial Intelligence and Machine Learning methods applied to manufacturing, as well as modern computer vision strategies for point cloud generation and analysis. It covers

the physics of the CW-GTAW process, the formation mechanisms of geometric defects (such as waviness and stubbing), and the theoretical basis for Spectral Graph Theory metrics (Fiedler number) used as ground truth.

- **Chapter 3 (Methodology):** Outlines the proposed methodological approach, detailing the Digital Twin architecture based on the ISO 23247 standard. It defines the specific ETL (Extraction, Transformation, Load) pipeline developed to process data, the adaptation of the PointNet Deep Neural Network for surface roughness classification, and the definition of quantitative alignment metrics (RMSE and Hausdorff distance) used for geometric validation.
- **Chapter 4 (Case Study: Experimental Apparatus and Results):** Describes the experimental configuration of the robotic manufacturing cell and the deposition strategies tested (step-by-step versus helical). It presents the findings from applying the adapted Deep Neural Network model to both simulated and real-world datasets, including the metrological validation of the results against high-precision 3D laser scanning.
- **Chapter 5 (Discussion):** Examines the model's performance and limitations, also addressing the difference in performance and classification observed between simulated and real training data. It discusses the effectiveness of the unsupervised clustering approach for surface quality levels labeling and compares the accuracy of the proposed low-cost photogrammetry method against industrial metrology standards.
- **Chapter 6 (Conclusions):** Summarizes the study's main contributions to the field of Digital Twin-enabled Quality Assurance and suggests avenues for future research, including the expansion of the dataset to more geometries and the integration of real-time molten pool sensoring for the improvement of the Digital Twin framework.

2 Literature review

The state-of-the-art on which this dissertation is based comprises a review of recent advancements in Artificial Intelligence-enabled computer vision, clustering methods for point cloud technologies, and, finally, neural network-based methods for generating 3D object meshes from point cloud clusters. Besides the analytical part of the massive batches of data available through DT-enabled solutions, a small review of related articles on DT implementations that can be used as a basis for analytical model developments is presented in section 2.7. Concepts related to the process carried out by the robotic manufacturing cell will also be discussed.

In order to clarify this work's contributions to the state-of-the-art of DT-enabled geometrical analysis of AM fabricated parts compared to recent similar projects, firstly, in the next section, the bibliometric process used to support the research in question will be presented.

2.1 Bibliometrics

Using the Scopus database of articles as the basis of the bibliometric process, recent articles published in the last 9 years in the niche of this current work were separated using the following query:

(digital AND twin AND ((QUALITY AND assurance) OR (qa) OR (tolerance) OR analysis)) AND (manufactur*) AND ((additive) OR (subtractive))

This keyword entry resulted in 271 documents (from 2017 to February 2026, the maximum timespan for the specific combination of tokens of the selected query for the niche of DT-enabled QA in manufacturing), whose information was extracted from the Scopus database into a .csv file to be analyzed using the free VOSviewer software version 1.6.20.

In order to find the most cited documents and the main connections between the most relevant authors in relation to the research niche addressed, a network of connections was created using the VOSviewer software based on the number of citations (discarding documents with less than 2 citations).

This resulted in 144 eligible documents, whose clusters' links graph is in the map presented in Figure 1.

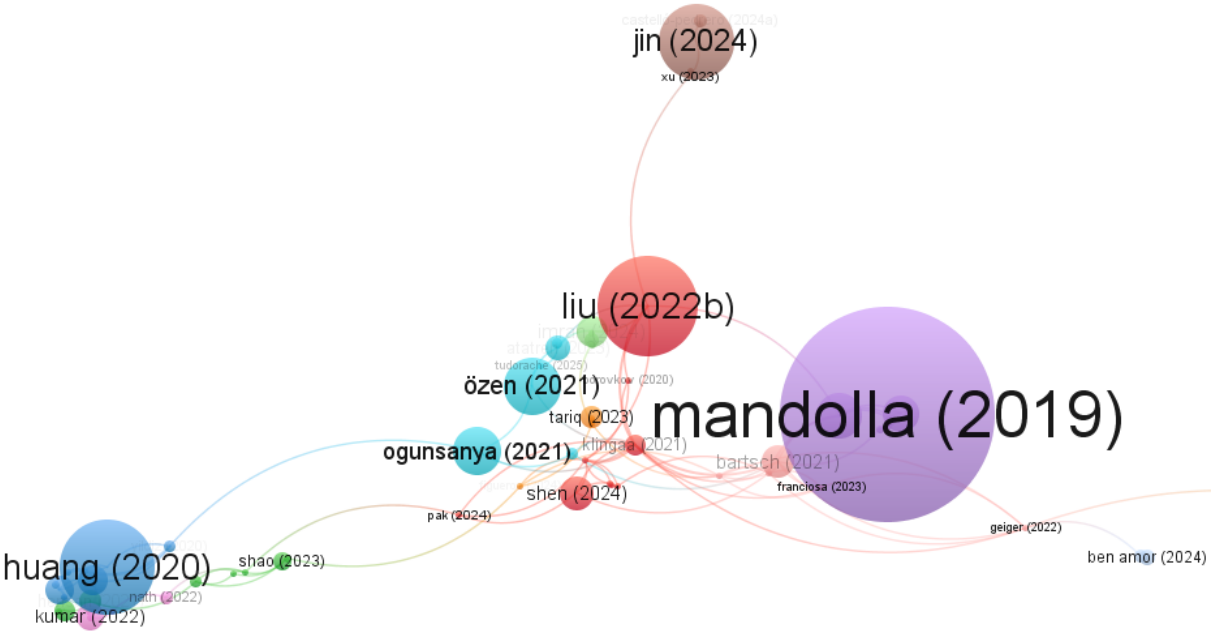


Figure 1 – Document clusters' links graph.

Analyzing each cluster and its documents, articles with enough adherence to the theme of this work were selected to be further reviewed. Other review papers were also used to support the research, resulting in more reviewed papers. After the revision process, 17 articles were thoroughly analyzed with their contributions being compared to this current work, resulting in the following Table 1.

Table 1 – Bibliometric comparison of DT-based works in manufacturing.

Article	No. Citations	Proposal or Implementation	Objective of Analysis	Monitored process	Uses DT in Analysis	Standard or Architecture	Protocols Used	Local or Cloud Database	Analysis Model (heuristic, AI, fuzzy, etc.)
(WANG, J. et al., 2019)	303	DT model for rotating machinery in manufacturing	Simulation for fault diagnosis of rotor systems	Subtractive manufacturing	Yes	Own	N/A	Simulated data	Heuristic
(CAVALCANTE et al., 2019)	325	Simulation of supplier selection in digital manufacturing	Risk mitigation design in supply chain management	General	No	N/A	N/A	Simulated	KNN, logistic regression, supervised machine learning
(SAMPEDRO; PUTRA; ABISADO, 2023)	18	Ensemble machine learning approach (3D-AmplifAI) for DT fault monitoring	Fault monitoring and detection in 3D printers	Additive manufacturing (3D printing)	Yes	Digital twin-based system for smart factories	N/A	N/A	Ensemble Machine Learning (3D-AmplifAI)

Table 1 – Bibliometric comparison of DT-based works in manufacturing. (continued)

Article	No. Citations	Proposal or Implementation	Objective of Analysis	Monitored process	Uses DT in Analysis	Standard or Architecture	Protocols Used	Local or Cloud Database	Analysis Model (heuristic, AI, fuzzy, etc.)
(LIU, Q. et al., 2018)	295	DT for production line design	Optimize production efficiency	Production line	Yes	N/A	OPC	Local	Heuristic
(BOTKINA et al., 2018)	155	Digital Twin for cutting machine	DT for subtractive manufacturing based on LISA architecture	Subtractive manufacturing	Yes	LISA, ISO 13399	Enterprise Service Bus (ESB)	Local	Proposes machine learning
(RAO; KONG; DUTY; SMITH, et al., 2015)	9	Use of point cloud for analysis of manufactured parts	SGT applied to dimensional integrity analysis of manufactured parts	Additive manufacturing	No	N/A	N/A	N/A	Heuristic, SGT, fiedler
(KRÜCKEMEIER; ANDERL, 2022)	17	Concept for virtual part inspection using Digital Twin Shadow	Replace physical inspection with virtual quality prediction	Additive Manufacturing (FDM/FLM)	Yes	Digital Twin Shadow	N/A	N/A	Heuristic, process data mapping to toolpath

Table 1 – Bibliometric comparison of DT-based works in manufacturing. (continued)

Article	No. Citations	Proposal or Implementation	Objective of Analysis	Monitored process	Uses DT in Analysis	Standard or Architecture	Protocols Used	Local or Cloud Database	Analysis Model (heuristic, AI, fuzzy, etc.)
(DECKER; WANG, Y.; HUANG, Q., 2020)	24	Scan of parts manufactured via 3D printing for surface analysis	Align manufactured parts with 3D models identifying manufacturing defects	Additive manufacturing	No	N/A	N/A	N/A	Heuristic
(MANDOLLA et al., 2019)	312	DT for AM based on Blockchain	Application of Blockchain-based DT in the aerospace industry	Additive manufacturing	Yes	N/A	Blockchain protocol (DIGEST messages)	Distributed Blockchain	N/A
(PANTELIDAKIS; MYKONIATIS, 2024)	16	Real-time Digital Twin Ecosystem (DTE) for CNC using Unity	Model machine behavior and detect collisions	Subtractive manufacturing (CNC)	Yes	DTE architecture (APDC & VRC)	LinuxCNC Python interface	Local real-time data	Heuristic (collision detection)

Table 1 – Bibliometric comparison of DT-based works in manufacturing. (continued)

Article	No. Citations	Proposal or Implementation	Objective of Analysis	Monitored process	Uses DT in Analysis	Standard or Architecture	Protocols Used	Local or Cloud Database	Analysis Model (heuristic, AI, fuzzy, etc.)
(ZHANG, H. et al., 2017)	326	DT and statistical analysis for improving the glass manufacturing process	Improvement in the efficiency and speed of the monitored process	Glass production line	Yes	SCADA	OPC, Ethernet	Cloud	Heuristic
(XU, Y. et al., 2019)	291	Deep transfer learning for fault analysis	Fault analysis	Assembly line	Yes	Own architecture for DFDD	N/A	Local PVS	Deep learning, transfer learning, DFDD
(TASHI et al., 2018)	20	3D printing of mesh generated via scan	Create prototypes of cultural artifacts	Additive manufacturing	No	N/A	N/A	N/A	Heuristic, PCAM parameterization

Table 1 – Bibliometric comparison of DT-based works in manufacturing. (continued)

Article	No. Citations	Proposal or Implementation	Objective of Analysis	Monitored process	Uses DT in Analysis	Standard or Architecture	Protocols Used	Local or Cloud Database	Analysis Model (heuristic, AI, fuzzy, etc.)
(TARIQ et al., 2023)	33	Framework for digital factory integrating DT with hybrid manufacturing	Streamline manufacturing operations and discuss future outlook	Hybrid manufacturing (LMD and subtractive)	Yes	N/A	N/A	N/A	AI, IoT (conceptual framework)
(TABAR et al., 2020)	33	Optimization of spot welding sequence using Digital Twin data	Minimize geometrical deviation in sheet metal assemblies	Spot welding (Assembly)	Yes	N/A	N/A	Inspection database	Genetic Algorithm (GA), Variation Simulation
(FERNANDES; BALDO; DONATELLI, 2021)	6	Digital Twin framework for geometrical variation management	Analyze deviations and simulate virtual assembly to predict fit-up issues	Pipe spool manufacturing (oil & gas)	Yes	Own framework	N/A	N/A	Heuristic, Geometric analysis (virtual assembly)

Table 1 – Bibliometric comparison of DT-based works in manufacturing. (continued)

Article	No. Citations	Proposal or Implementation	Objective of Analysis	Monitored process	Uses DT in Analysis	Standard or Architecture	Protocols Used	Local or Cloud Database	Analysis Model (heuristic, AI, fuzzy, etc.)
(FU; LIU, S.; LI, P., 2025)	23	DT-driven process management method for smelting monitoring	Visualize and control smelting process (lining thickness & temperature)	Converter Steelmaking	Yes	5-Dimension DT Model	N/A	N/A	Point Cloud Processing, Inverse Heat Conduction (IHCP)
Current work	N/A	Deep learning for mesh creation and tolerance analysis of manufactured parts and system failures	Improve process efficiency and quality of manufactured parts	Additive manufacturing (CW-GTAW WAAM)	Yes	ISO 23247	MQTT, HTTP, and Ethernet	Firebase Firestore cloud	Deep learning and Fuzzy

Regarding the bibliometric analysis summarized in Table 1, distinct trends and prevailing gaps in the implementation of Digital Twins for manufacturing are revealed. A significant observation is the lack of standardization in system architectures. The majority of the reviewed works, such as (WANG, J. et al., 2019), (LIU, Q. et al., 2018), (ZHANG, H. et al., 2017), and (FERNANDES; BALDO; DONATELLI, 2021), rely on proprietary or undefined architectures, with (BOTKINA et al., 2018) being a rare exception that cites specific standards (LISA, ISO 13399). This fragmentation hinders interoperability and scalability in industrial environments. With that in mind, the current work addresses this gap by strictly adhering to the ISO 23247 standard, proposing a reference framework that ensures the Digital Twin is not merely an isolated experiment but a scalable, standard-compliant solution.

In the context of data management and connectivity, a noticeable dichotomy exists between legacy simulation-based approaches and modern IoT-driven implementations. A considerable number of high-impact studies, including (CAVALCANTE et al., 2019) and (WANG, J. et al., 2019), validate their models using simulated data or local databases without defining specific communication protocols. While recent works like (PANTELIDAKIS; MYKONIATIS, 2024) and (TARIQ et al., 2023) begin to integrate real-time data handling, they often rely on direct hardware interfaces rather than open IoT standards. In this sense, the framework proposed herein leverages a cloud-native approach using MQTT and HTTP protocols with Firebase Firestore, reflecting the industry's shift towards distributed, accessible, and low-latency data infrastructures.

Furthermore, concerning analytical intelligence, a transition from deterministic heuristics to stochastic and Artificial Intelligence (AI) models is evident, albeit incomplete. As shown in the table, many applications—spanning from glass manufacturing (ZHANG, H. et al., 2017) to subtractive CNC processes (PANTELIDAKIS; MYKONIATIS, 2024) and additive manufacturing inspection (KRÜCKEMEIER; ANDERL, 2022; DECKER; WANG, Y.; HUANG, Q., 2020)—still rely on heuristic algorithms or simple geometric mapping. While advanced methods such as Genetic Algorithms (TABAR et al., 2020) and Ensemble Machine Learning (SAMPEDRO; PUTRA; ABISADO, 2023) are emerging, the application of Deep Learning to 3D geometric quality assurance remains underexplored. This work fills this specific gap by integrating a Deep Neural Network (adapted from PointNet) to process unstructured 3D meshes directly, moving beyond simple scalar thresholding.

In this sense, although a singular document encompassing the exact integrated framework structure proposed in this work—combining ISO 23247 standardization, cloud-native architecture, and Deep Learning-based geometric QA—was not identified within the boundaries of this review, the increasing relevance of this research avenue is evident. As illustrated in Figure 2, a marked upward trajectory in the number of publications related to Digital Twins applied to manufacturing contexts is observed over the last decade, particularly intensifying after 2019. This trend validates the growing academic and industrial effort

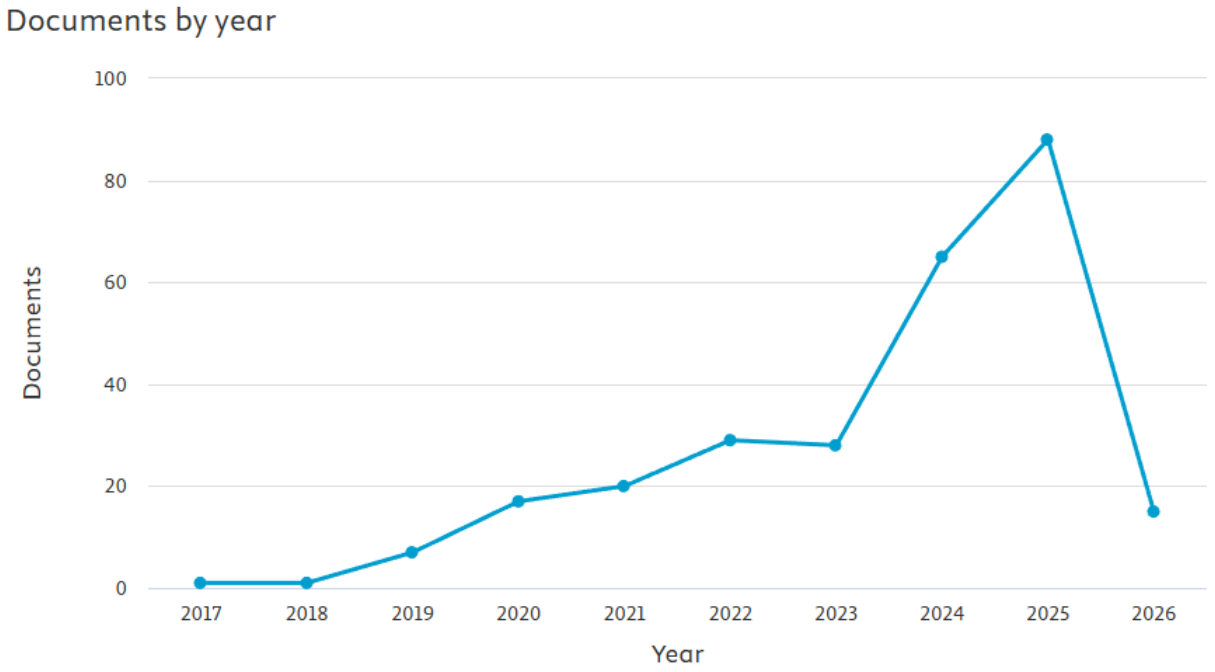


Figure 2 – Published documents' quantity trend from 2017 to 2026.

towards developing more sophisticated QA models integrated into DT-enabled environments, even if current solutions remain fragmented across different technical approaches.

Also, the domain of Additive Manufacturing (AM) presents specific challenges that are often treated in isolation. Existing literature tends to focus either on the monitoring of process parameters, such as temperature fields in (FU; LIU, S.; LI, P., 2025), or on post-process geometric scanning (RAO; KONG; DUTY; SMITH, et al., 2015), rarely integrating both into a unified Digital Twin loop for complex processes like Wire Arc Additive Manufacturing (WAAM). Works such as (MANDOLLA et al., 2019) introduce novel concepts like Blockchain for AM but do not address quality assurance of deposited parts. The present study bridges this divide by correlating process physics (CW-GTAW) with geometric defects and perceived surface quality through a unified Digital Thread.

Finally, while the concept of the Digital Twin is well-established, its practical implementation often lacks the convergence of standardization, cloud connectivity, and advanced AI. The existing body of knowledge provides strong foundations in individual areas—heuristics for collision detection, machine learning for fault diagnosis, and simulation for process planning. However, there is a clear need for a holistic framework that combines these elements. In this sense, the current work contributes to this niche by presenting an ISO 23247-compliant, cloud-enabled, and Deep Learning-driven Digital Twin specifically tailored for the quality assurance of deposited walls in robotic AM.

Beyond those cited papers, works related to computer vision, positional data extraction methods, AI models, additive manufacturing processes, etc., were also considered to base the methodology of this current work. Those will be further discussed in the following

sections.

2.2 Methods of positional data (point clouds) extraction

Point clouds are discrete point representations of real assets; usually, those points are collected via infrared cameras or LiDARs. In the case of this project, point clouds can be generated using both positional data collected from the history of variables in the cloud database, and by scanning manufactured 3D parts using photogrammetry and LiDAR, followed by a third method of 3D scanning with higher precision for metrology validation, Multi-Stripe Laser Triangulation.

2.2.1 Photogrammetry-based 3D models generation

Another option for 3D mesh generation is the use of photogrammetry. Photogrammetry is based on triangulation, where images taken from two different perspectives help determine the positions of points on an object's surface. This technique plays a crucial role in depth perception and various spatial analyses. A key challenge in photogrammetry is determining the camera's internal and external orientations, along with the spatial coordinates of image points. Internal orientation relates to camera-specific parameters like focal point and lens distortions, while external orientation defines the camera's position and direction using six parameters: three projection center coordinates (X0, Y0, Z0) and three rotation angles (omega, phi, and kappa) (JAIN, 2020).

This approach of generating 3D point clouds and 3D models from photogrammetry (Structure-from-Motion, SfM) (OZYESIL et al., 2017) operates by analyzing the dataset of overlapping photographs to identify distinct feature points—such as high-contrast corners, edges, and texture gradients—across multiple viewing angles. By triangulating the relative displacement of these points, the engine mathematically computes their precise depth and spatial coordinates, reconstructing the physical geometry of the hollow parts with metric fidelity. Furthermore, the platform functions as a hybrid framework by integrating Machine Learning (ML) algorithms to augment this geometric process. Specifically, neural networks are utilized for semantic analysis and automatic subject masking (essentially segmentation), effectively distinguishing the metallic deposition from environmental background noise. This allows the workflow to leverage AI to optimize signal-to-noise ratio and texture refinement while strictly relying on photogrammetric triangulation to ensure the dimensional accuracy of the resulting mesh.

2.2.2 LiDAR and Laser triangulation extraction methods

The Light Detection and Ranging (LiDAR) technology integrated into modern mobile devices operates on the Direct Time-of-Flight (dToF) principle. Unlike the continuous

scanning line of laser triangulation systems, these sensors emit a pulsed matrix of infrared light using a Vertical Cavity Surface Emitting Laser (VCSEL) array. The distance to a specific point on the target surface is calculated by measuring the round-trip time required for a photon to travel from the emitter to the object and return to the Single-Photon Avalanche Diode (SPAD) detector. This relationship is governed by the constant speed of light c in a vacuum, as described by (HANSARD et al., 2012).

Another common method, related this time to laser scanning, is the Multistripe Laser Triangulation (MLT) technology, which is an active non-contact scanning technique that advances traditional single-line methods by projecting an array of multiple laser stripes simultaneously onto the target surface. Implemented in systems such as the NextEngine scanner (ANIWAA, 2026), this technology utilizes a "twin array" architecture consisting of multiple solid-state lasers and dual complementary optical sensors offset by a specific baseline angle (PLATT; HASIOTIS; HIRMAS, 2010).

This configuration allows the acquisition pipeline to capture dense geometric data in a single pass while significantly mitigating occlusion effects, as surface points hidden from one sensor's perspective are likely captured by the secondary sensor. Studies have validated MLT as a robust method for quantitative 3D analysis, demonstrating that its multi-beam approach provides superior noise resilience and edge preservation compared to standard structured light or time-of-flight sensors when digitizing small-to-medium-sized artifacts with intricate surface details (PLATT; HASIOTIS; HIRMAS, 2010) (GUIDI et al., 2010).

2.3 Neural network-based 3D mesh generation

Parting from the clusters of points obtained by applying clustering algorithms to original point clouds, it is possible to analyze those clusters, generating 3D meshes that correspond to the real manufactured parts.

Since 2021, new technologies have been developed by NVIDIA researchers in partnership with the University of Toronto, Vector Institute, and other institutions in the field of 3D reconstruction of surfaces and objects from point clouds.

The first authorial technology from NVIDIA reviewed in this current work is the Neural Kernel Fields (NKF) (WILLIAMS et al., 2021), which uses a neural network-based solution to learn kernel parameters from input data. The model developed was trained using a dataset of shapes, which was then tested on point cloud inputs of 3D objects to generate output 3D meshes with less noise and smoother surfaces when compared to other previously developed methods mentioned in the study.

Using PyTorch accelerated with CUDA cores, the Neural Kernel Surface Reconstruction (NKSR) (HUANG, J. et al., 2023) algorithm represents an advancement over its

predecessor, NKF. The model was trained using data obtained by LiDAR point cloud frames. Every single frame is analyzed with the intent of characterizing each point as belonging or not to a certain surface and filtering noisy points. Using tens of thousands of iterations to train the developed model, the algorithm achieves unmatched results in 3D mesh generation compared to previous developments such as NKF and POCO (BOULCH; MARLET, 2022).

Alternatively, another method of generating 3D meshes may be based on captured images of produced parts. In (XU, J. et al., 2024), a method of creating 3D meshes from grids of images of objects captured from different angles is presented. The cited methodology is called InstantMesh and is established upon a previous method, Instant3D (LI, J. et al., 2023). Both methods are based on sparse-view reconstruction, which utilizes input images to train a deep learning neural network using transformer decoding. The main difference between them is that the first uses images as inputs for the generation of 3D models, whilst the latter makes use of tokenized natural language as input for the already trained model.

2.4 Clustering algorithms

Since the ‘90s, many advancements have been made in the development of new or improved old ones’ clustering algorithms. This kind of algorithm is mainly used to understand boundaries between surfaces, objects, and groups of points in general.

One of the main contributions in the field of clustering focused on computer vision is presented in (ESTER et al., 1996) in the form of the algorithm DBSCAN (Density-Based Spatial Clustering of Applications with Noise). The DBSCAN algorithm is based on the “density-reachability” and “density-connectivity” approaches to cluster points while ignoring noise. To achieve such a discretization, points within a cluster are also distinguished as central or border points. The implementation of the algorithm in general results in a fast and reliable method of clustering when compared to previously developed ones (k-means, clarans).

(RODRIGUEZ, M. Z. et al., 2016) presents a comprehensive review of many methods of clustering, such as k-means, EM, hcmodel, and the already cited DBSCAN. The results of the comparative models applied to the algorithms suggest that some of them, such as EM (BRADLEY; FAYYAD; REINA, 2000) and hcmodel, present better accuracy results in some benchmark scenarios created through randomly generated features (groups of points) in artificial datasets.

2.5 ICP referential correction of 3D models

One of the main problems when comparing point clouds or meshes is the reference based on which each point cloud or mesh was obtained. To correct this, the Iterative Closest

Point (ICP) might be applied to the set of points of the point clouds generated. The objective of applying this algorithm to the generated point clouds is to minimize the error in the distances between corresponding points when superimposing two point cloud frames, thereby enabling the analysis of errors and tolerances among the manufactured parts.

One work related to this problem is presented in (ZHANG, J.; YAO; DENG, 2020), which introduces a fast and robust point-to-plane ICP algorithm. The implemented algorithm builds on a traditional point-to-point ICP solution to achieve faster convergence; the proposed formulation is then extended to a point-to-plane ICP solution.

An application of the ICP concept in generating point cloud-based objects and surfaces is presented in (NEWCOMBE et al., 2011). In the cited work, a reconstruction of surfaces using Kinect is presented; one problem faced during the capture of point cloud data is the constant change of angle and position of the sensor, making it difficult to scan the entire surface of an object. To overcome this, the ICP algorithm is applied to each captured frame, estimating the pose of the sensor and making it possible to join the frames captured to form a surface.

(DECKER; WANG, Y.; HUANG, Q., 2020) proposes a method of alignment correction of 3D scanned AM-produced parts with their original designs, this way enabling the assessment of geometric differences. This method can facilitate the process of quality assurance of additive-manufactured parts by reducing errors commonly found when trying to superpose meshes or point clouds. Some examples of possible errors are non-uniform sampling or scanning, or even misalignment resulting from a local minimal distance (not the optimal) found when applying traditional ICP algorithms. Their proposed method aims to reduce those errors by adding initial steps before applying the ICP algorithm. Those steps include (but are not only): manual alignment of the 3D model of the design with the scanned real part point cloud one, segmentation of the scanned model so that noisy or background points are separated from the intended shape and resampling on the point cloud data to achieve uniform point density. After the execution of an ICP algorithm on the pre-aligned 3D models, a deviation calculation is performed on the shapes, ensuring that the entire method was correctly performed.

2.6 GTAW process

The GTAW process is characterized by the utilization of a non-consumable electrode for the induction of the electric arc. In order to protect both the electrode and the molten pool of metal, an inert gas is used, which comes out of the GTAW torch while the parts are manufactured (SINGH, R., 2020).

The shielding gases typically employed in this process are argon and helium. Generally, the operation utilizes Direct Current Electrode Negative (DCEN) on the tungsten

electrode; however, Alternating Current (AC) may also be applied to achieve distinct welding effects.

Regarding this process, the main limitation is the orientation of the TCP (Tool Center Point, at the tip of the torch) while the electric arc is active, because the TCP orientation has to be perpendicular to the direction of material deposition, and the metal wire feeding must happen in the opposite direction of the deposition (ANTONINI, 2014). While this configuration imposes kinematic constraints on the robotic manipulator—specifically regarding rotation and travel speed—it results in a superior surface finish on the manufactured component. This characteristic is highlighted in Figure 3.

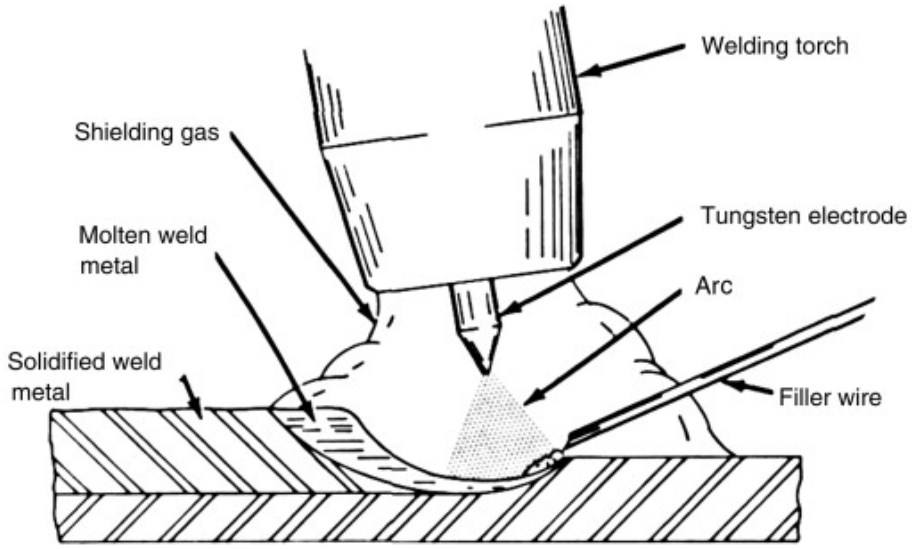


Figure 3 – GTAW welding diagram (Source: (ANTONINI, 2014)).

At the GRACO laboratory, the metal additive Cold-Wire GTAW process is implemented using a Fronius MW5000 power source connected to a torch mounted on the robot's end-effector, enabling the fabrication of 3D metal parts within the additive manufacturing robotic cell.

The Cold-Wire GTAW process was chosen as the process for depositing metal AM hollow parts in this work, for being a direct energy deposition process that usually results in better superficial finishing and quality when compared to other processes, such as the more commonly used GMAW derivatives (TOMAR; SHIVA, 2023). A brief comparative analysis of the advantages and disadvantages of GTAW and GMAW-based processes is presented in Table 2.

Despite usually having higher costs and a lower productivity, the higher quality finishing, allied with the tuning of parameters throughout time, should enable a wide range of perceived quality levels along experiments, so a qualitative study based on the application of AI models is supported. Also, the lower number of studies regarding the CW-GTAW process, especially in DT-enabled metal AM applications, should further support the

Table 2 – Comparative analysis of GTAW (TIG) vs GMAW (MIG) in WAAM processes

Criterion	**TIG (GTAW)**	**MIG (GMAW)**
Geometric precision	Superior (SINGH, J.; IL'YENKO; OLIVEIRA, 2025)	Inferior (SINGH, J.; IL'YENKO; OLIVEIRA, 2025)
Thermal control	Fine and stable (GHOSH, 2024)	High input (GHOSH, 2024)
Surface finish	Excellent (GHOSH, 2024)	Rough (GHOSH, 2024)
Productivity	Low (SINGH, J.; IL'YENKO; OLIVEIRA, 2025)	High (SINGH, J.; IL'YENKO; OLIVEIRA, 2025)
Operational cost	Higher (time-consuming) (GHOSH, 2024)	Lower (GHOSH, 2024)
Popularity in WAAM literature	Lower (GHOSH, 2024)	Dominant (GHOSH, 2024)

motivation of having a DT-enabled framework for surface roughness classification developed and analyzed.

2.6.1 Heat input and thermal dynamics

The stability of the deposition process in Wire Arc Additive Manufacturing is fundamentally governed by the heat input, which dictates the thermal cycles, cooling rates, and the fluidity of the molten pool (EAGAR, 1993). The heat input (Q) represents the amount of electrical energy transferred to the substrate per unit length of the weld bead. According to classic welding theory, it is calculated using the arc voltage (V), welding current (I), and travel speed (v), adjusted by the thermal efficiency factor (η) of the process, as shown in Equation 2.1.

$$Q = \eta \frac{V \cdot I}{v} \tag{2.1}$$

For the Gas Tungsten Arc Welding (GTAW) process utilized in this framework, the thermal efficiency η is typically estimated between 0.60 and 0.70. In the manufacturing of thin-walled structures, the heat input plays a decisive role in the formation of geometric defects. Excessive heat input leads to heat accumulation in the upper layers, reducing the viscosity of the molten material and causing the bead to flow outward (bulging), which compromises the dimensional accuracy. Conversely, insufficient heat input prevents proper wetting of the substrate, potentially leading to a lack of fusion. Consequently, variations in the robot's travel speed or current modulation directly alter the factor Q, manifesting as physical artifacts that the Digital Twin is tasked with detecting.

2.6.2 Surface waviness and effective width

The geometric quality of the deposited walls is quantitatively assessed through the concept of Surface Waviness (SW), which measures the lateral surface irregularity of the additive part. In the context of near-net-shape manufacturing, waviness is a critical metric because it determines the volume of material that must be machined away to achieve the final design tolerances. To quantify this, the literature (MARTINA et al., 2012) distinguishes between two primary metrics: the Total Width (W_{total}) and the Effective Width (W_{eff}). The Total Width is measured as the maximum distance between the outer peaks of the deposited wall surfaces. In contrast, the effective width is defined as the width of the largest rectilinear solid that can be inscribed within the wall boundaries, representing the actual usable material. Surface waviness is derived from the discrepancy between these two values, as described in Equation 2.2 and visualized in Figure 4.

$$SW = \frac{W_{total} - W_{eff}}{2} \tag{2.2}$$

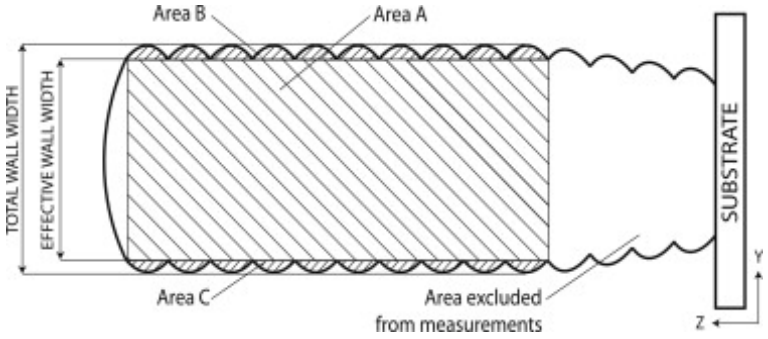


Figure 4 – Surface waviness visualization through the difference between the effective and total wall width (Source: (MARTINA et al., 2012)).

A higher SW value indicates significant surface roughness or instability in the deposition path. In the context of this study, the "waviness" classification performed by the neural network effectively identifies regions where process instabilities—induced by the heat input fluctuations described previously—caused the W_{total} to oscillate significantly while reducing the W_{eff}. This relationship establishes the physical ground truth for the classification labels used in the Digital Twin's dataset. Higher deviations in surface waviness across layers result in lower surface connectivity and, consequently, a lower Fiedler number, as contextualized in the following subsection of the current work.

2.6.3 Topological connectivity and spectral analysis

To complement the local geometric analysis of waviness, the framework utilizes Spectral Graph Theory to assess the global topological consistency of the deposited layers. By transforming the point cloud into a k-nearest neighbors graph, the structural integrity of the mesh can be quantified using the Fiedler number. This metric is defined by the eigenvalues

of the graph's Laplacian matrix, as expressed in Equation 2.3, with the contextualization of positional data-based point clouds from deposited parts for the assessment of the Fiedler number.

$$\lambda_2 = \text{eig}_2(\mathcal{L}), \qquad \mathcal{L} = I - D^{-1/2} A D^{-1/2} \tag{2.3}$$

Where:

- λ_2 — Fiedler number, the second-smallest eigenvalue of the normalized Laplacian $\mathcal{L}$
- $\text{eig}_2(\cdot)$ — function that returns the second-smallest eigenvalue of a matrix
- $\mathcal{L}$ — normalized Laplacian matrix of the graph
- I — identity matrix of size $n \times n$, where n is the number of nodes in the graph
- A — adjacency matrix of the k-nearest neighbors graph constructed from the point cloud
- D — degree matrix, a diagonal matrix where $D_{ii} = \sum_j A_{ij}$

The Fiedler number (λ_2), also known as the algebraic connectivity, serves as a global descriptor that correlates directly with the physical and geometric parameters discussed in previous sections. While heat input (Q) governs the fluid dynamics of the melt pool and surface waviness (SW) measures the resulting local profile irregularities, λ_2 captures the overall structural cohesion of the digital representation. A process with stable heat input produces a uniform bead with low waviness, resulting in a graph with high connectivity and a larger λ_2 value. Conversely, significant fluctuations in heat input lead to severe waviness (large deviations between total and effective widths), which creates topological "bottlenecks" or constrictions in the mesh graph. These geometric defects significantly reduce the algebraic connectivity, allowing λ_2 to function as a global quality index that validates the local defect classification.

2.6.4 Common defects found in the Direct Energy Deposition process

As the GTAW WAAM process is characterized by also being a Direct Energy Deposition (DED) process (COSTELLO et al., 2023), there are 3D parts' building and layering deposition defects that must be taken into consideration while analyzing manufactured parts and refining deposition parameters. Liu et al. (LIU, M. et al., 2021) systematically categorize and examine defects in DED, focusing specifically on part quality issues such as geometrical, morphological, and microstructural anomalies. By linking these defects to their root causes and potential solutions, the study aims to equip manufacturers with strategies to mitigate

challenges and improve the reliability of DED processes. The main defects categorized in the review article are summarized as follows (ALVARES, A. J. et al., 2025).

Balling Defect: This phenomenon arises when the material liquefies before making contact with the building support plate, leading to the formation of a conspicuously bright and oversized molten metal sphere suspended in midair.

Dripping Defect: This occurs when the wire momentarily loses connection with the build plate, prompting the process control to react swiftly. While minor dripping may not always pose a problem, excessive occurrences make it impossible to print over the defect without exacerbating the issue.

Necking Defect: This defect precedes dripping, characterized by the melt pool remaining attached to the wire while stretching and thinning significantly before reaching the substrate. This thinning indicates an inadequate material supply, ultimately causing surface tension inconsistencies. These defects manifest internally, where the new material adheres only to adjacent lines rather than the underlying layer. Additionally, the external surface may develop an uneven, wavy pattern, with distorted layer lines.

Overbuilding Defect: Contrary to necking, this defect results from an energy imbalance shifting towards the substrate throughout the printing process. As each successive layer becomes slightly taller than the specified height in the slicing software, the gap between the part and the print head may diminish. Signs of overbuilding include the melt pool embedding itself into the substrate and the material ceasing to emit a glow upon contact. In arc processes, this can cause the electrode to dip into the melt pool (tungsten inclusion) or the wire to stub against the solidifying front.

Stubbing Defect: This issue arises when the energy balance disproportionately favors the substrate, preventing adequate energy transfer to the material. The result is incomplete wire fusion at best, or severe bending and detachment of the wire at worst. Excessive force applied to the print head can trigger a complete failure in the process, potentially damaging the machine.

To systematically address these defects within the specific context of the experimental CW-GTAW setup, a Cause-and-Effect (Ishikawa) diagram was developed, as presented in Figure 5. This diagram maps the observed anomalies—specifically surface waviness, stubbing, and geometric deviations—to their root causes. This structured characterization serves as the basis for the proposed mitigation strategies discussed in the results chapter subsection 4.2.2, such as the adoption of the helical deposition path to stabilize the thermal gradient and minimize the defects described above.

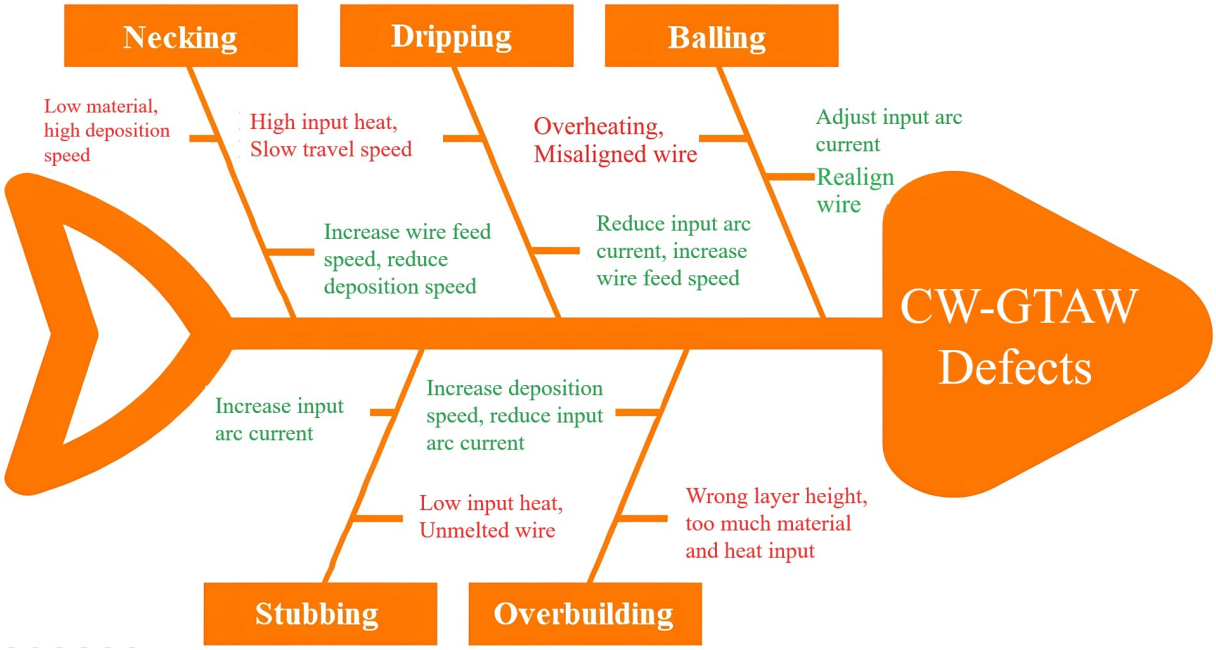


Figure 5 – Ishikawa diagram for root causes of superficial defects and solutions for the AM CW-GTAW process (Adapted from: (ALVARES, A. J. et al., 2025)).

2.7 Related works

Recently, many applications of DT have been developed with the focus of obtaining huge chunks of data from monitored assets and industrial processes. In this sense, many projects have already applied AI models to analyze data streamed through DT architectures. Regarding the analysis of products and processes lifecycle, a comprehensive implementation of a Digital Twin for additive manufacturing has been developed by (LIU, C. et al., 2022). In their work, many domains of DT applications were considered in the developed framework, such as quality measurement, product design, and process planning, besides the traditional manufacturing and monitoring applications usually found in DT implementations. The extensive clusters' variables and parameters used in their pipeline enable a wide range of data analysis; in this regard, a deep learning-enabled defect analysis of deposited layers is also present in the framework.

(DOUNGTAP et al., 2023) shows the development of a pipeline used for the creation of DT applications based on analyzing 2D images and assets from a database. The data collected is then processed by a pre-trained deep learning model to create a 3D point cloud representation, which will then be post-processed to create a 3D mesh of the original 2D objects. Those generated 3D meshes are then used in a virtual Digital Twin 3D simulation for visualization and apparel customization.

In the specific application context of point cloud data-driven AM, (TASHI et al., 2018) presents a method for generating CAD files for manufacturing via traditional FDM layer deposition. The real objects were scanned using a 3D scanner or image processing, but a

solution for creating point cloud models from 2D shapes through a parametric approach to points generation is also present.

When considering projects that use 3D point cloud modeling to analyze additively manufactured parts, Rao et al. (RAO; KONG; DUTY; SMITH, 2016) apply a dimensional quality assurance method based on Spectral Graph Theory (SGT) to analyze parts fabricated through polymer extrusion Fused Filament Fabrication (FFF). The dimensional integrity of the tested parts was evaluated through the SGT-based topological invariant called Fiedler Number ($\lambda 2$), which was proved as an efficient dimensional quality estimator throughout the analysis developed.

Finally, the work presented in this current work is based on a previous project in which a Digital Twin was developed for a metal AM robotic cell (CABRAL, J. V. A., 2023), which was also based on a previous project on the development of a DT framework for subtractive manufacturing on a Haas Mini Mill Machining Center (CABRAL, J.; RODRIGUEZ, E.; ALVARES, A., 2023). The cell is composed of an ABB IRB 2600 robotic arm, an ABB IRBP A250 positioning table, and a Fronius MW5000 welding machine with a GTAW torch attached to the manipulator end of the robotic arm. In Figure 6, a detailed illustration - adapted from the previous work - of the whole DT framework architecture based on the ISO 23247 is presented.

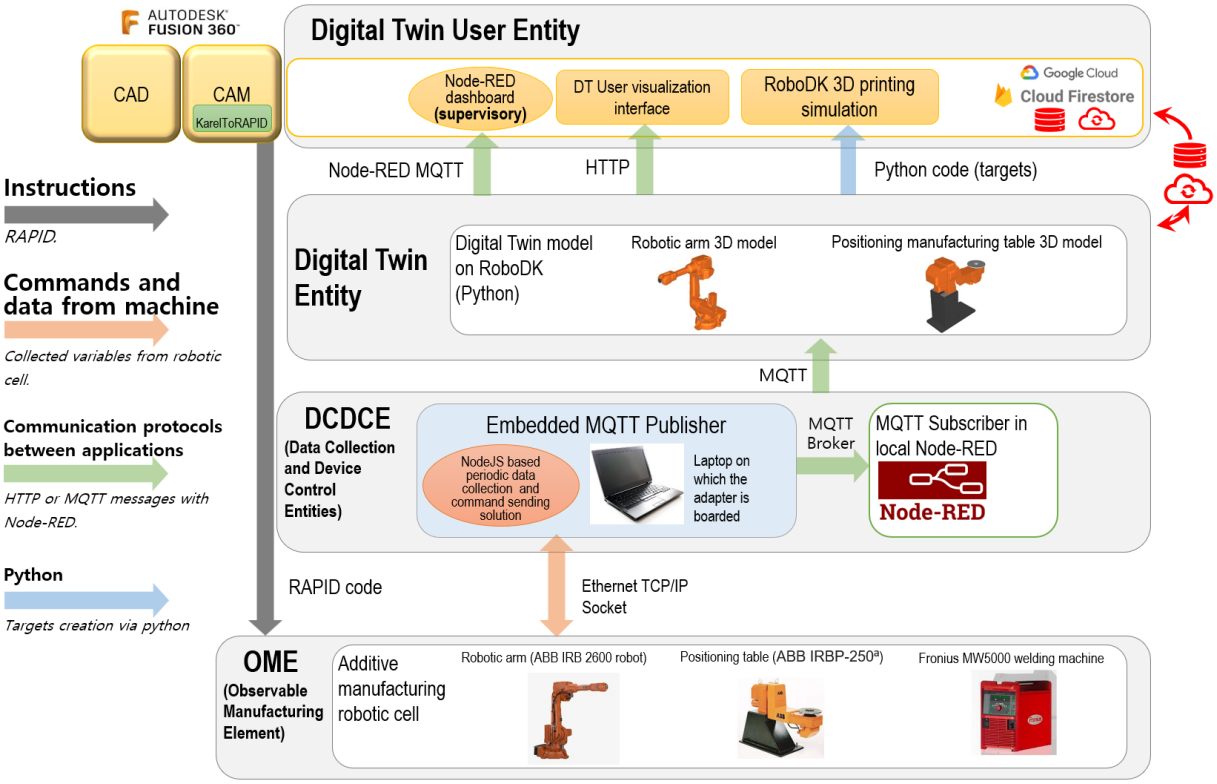


Figure 6 – Digital Twin framework architecture based on ISO 23247 applied to a real metal Additive Manufacturing robotic cell (CABRAL, J. V. A.; ÁLVARES; CARVALHO, 2024).

The collection of data occurs based on signals monitored by the ABB ICR5 Ro-

botWare 5.13.0 controller. Those signals are then sent to a local TCP/IP server via Ethernet. This server is embedded into an MQTT adapter on a laptop; all the data streamed from the controller is then structured and transmitted through an MQTT dataflow so that a dashboard and 3D simulation of the mirrored movement of the real assets can be established.

The most important feature related to this current work is the dataflow that ends in data published in a cloud Firebase Firestore, so the history of data can be accessed and analyzed. In this sense, the data stored in the cloud will be used in this work to generate 3D point clouds of the path traversed by the Tool Center Point (TCP) during the deposition of metal. The main objective is to compare real manufactured parts with the data obtained through the DT monitoring. This new application will be detailed in Chapters 3 and 4.

2.8 Quality assurance analysis on metal additively manufactured parts

In order to further support the analysis framework discussed through the methodology in chapter 3, a revision on the quality assessment of metal AM-based deposited parts is needed.

According to a recently published review by (JIN et al., 2024), machine learning has a pivotal role in improving additive manufacturing (AM) overall performance and deposition results. Process parameter optimization enables a better topology on fabricated parts and a more uniform deposited metal cord, potentially without defects. Some of the common defects encountered in metal-deposited parts are metal cracks, porosity and residual internal stresses, which might compromise the mechanical strength of the resulting part (BELLINI et al., 2021).

In the context of wire-arc additive manufacturing (WAAM), (WU et al., 2021) proposes a heuristic-based analysis (Box–Behnken Design response surface) to find optimized deposition parameters based on three key parameters: auxiliary gas nozzle angle, gas flow rate, and nozzle-to-substrate distance. Those optimizations led to better morphology, refined grains, and improved properties due to the stirring and cooling effects of the auxiliary gas.

(HOU et al., n.d.) uses a convolutional neural network (CNN) approach to detect and classify superficial defects on metal AM deposited parts. Using a residual-attention CNN (RA-CNN) algorithm combined with a support vector machine (SVM), it is possible to extract features from images of manufactured parts and classify their defects into many categories, such as pitted surface, scratches, patches, etc., and also in binary classification (defect or non-defect).

Fountas et al. (FOUNTAS; KECHAGIAS; VAXEVANIDIS, 2023) presents a machine learning model based on a virus-evolutionary genetic algorithm (VEGA) for parameter

optimization over the selective laser sintering—selective laser melting (SLS-SLM) process for metal deposition. The study explores three experimental cases, each with different objectives: maximizing the density of Ti6Al4V specimens, maximizing both hardness and tensile strength of Ti6Al4V samples, and minimizing surface roughness while maximizing density and hardness of L316 stainless steel powder. It is shown in the results that the VEGA algorithm outperforms other ones, such as Greywolf, Multi-verse, Antlion, and dragonfly algorithms, regarding the optimization of the SLS/SLM processes.

On dimensional accuracy of fused deposition modeling (FDM), Mohamed et al. (MOHAMED; MASOOD; BHOWMIK, 2021) propose a deep learning neural network framework for optimizing the deposition process. The study suggests a new experimental design technique combining definitive screening design (DSD) and deep learning feedforward artificial neural networks (ANNs) to evaluate and predict the effects of six key operating variables on the dimensional accuracy of FDM-fabricated parts. The approach aims to optimize the FDM process by accurately modeling and analyzing the impact of these variables, leading to improved dimensional accuracy and stability of the manufactured parts.

(LEE et al., 2021) provides a comprehensive overview of quality monitoring techniques in metal additive manufacturing (AM). The paper highlights the similarities between AM and conventional metal manufacturing processes like welding, emphasizing shared challenges such as layer misalignment, dimensional errors, and residual stress. In metal AM, image data helps detect melt pool characteristics, spatter, and keyhole formation while identifying dimensional errors. Plasma spectrum data enables elemental composition analysis and cladding track measurement, improving spectrum acquisition. Acoustic signal data assesses thermal properties and predicts porosity, contributing to overall process optimization.

In Chua et al. (CHUA; AHN; MOON, 2017), a multi-step analysis approach to numerical quality assessment is presented. The proposed inspection process progresses through three stages: single-layer, multi-layer, and final inspection, using technologies such as infrared cameras, pyrometers, and computed tomography scans. Single-layer inspection captures real-time data to establish correlations between process parameters and melt pool characteristics. Multi-layer inspection detects void locations based on temperature inconsistencies, while final inspection ensures the overall quality of the printed part through detailed verification. The ultimate goal is to refine monitoring systems for AM through continuous feedback, improving quality control and defect detection.

(LI, X. et al., 2020) presents a deep learning-based approach for quality monitoring in metal additive manufacturing, leveraging semi-supervised learning to handle low-quality images, taken on the inspection process of metal AM fabricated parts. In the domain of visual defect detection for Laser Metal Deposition (LMD-wire), (ALVARES, A. J. et al., 2025) proposed a comprehensive deep learning framework for automated quality control. The study established a specific taxonomy for common deposition anomalies, categorizing

defects into five distinct classes: Balling, Dripping, Necking, Overbuilding, and Stubbing. By conducting a comparative analysis between YOLOv5s and Faster R-CNN architectures, the authors demonstrated the superior efficiency of the YOLOv5s model, which achieved a mean Average Precision (mAP) of 94.6%. This work validates the capability of Convolutional Neural Networks (CNNs) to generalize defect detection across different quality levels using 2D image datasets, providing a foundational baseline for AI-driven inspection in metal additive manufacturing.

When compared to other QA inspection methodologies found in the state-of-the-art of metal AM manufacturing, it is clear that not many of them approach this problem with 3D modelling analysis and AI (deep neural networks, machine learning) applied to such a type of data extraction and structuring. Thus, the research novelty of this thesis lies in providing a numerical qualitative approach to quality assessment of GTAW metal manufactured parts, with the proposed approach based on comparisons among 3D models and analytical methods applied to those.

Using data acquisition methods supported by a Digital Twin architecture and CAD/-CAM deposition planning, it is also possible to support decision-making on the refining of the deposition process itself and to compare expected results with those obtained in the real deposited parts. In the next chapters 3 and 4, the general methodology of the proposed QA approach to the GTAW process in the context of a DT-enabled robotic cell and the results of the analysis framework implementation will be presented and thoroughly discussed.

3 Methodology

To develop a quality assurance model for the manufactured parts from the robotic cell, it is necessary to establish a pipeline architecture for processing the data collected from the history of variables available in the cloud Firebase Firestore database.

The proposed pipeline of data analysis will be based on the principles of the ETL cycle: extraction, transformation, and load. With that in mind, data extracted from the implemented Firestore database will be organized in batches to be analyzed and transformed according to programs executed by the DT-enabled AM robotic cell. After the application of AI-driven analysis on the data extracted, the processed data may then be uploaded back to a separate collection on the database to be further used at a later moment.

Starting from the previous project of the original Digital Twin framework presented in Figure 6, a new Digital Twin architecture is conceived based on the previous framework, including the proposed ETL pipeline in the Application and Service Sub-system Entity, as presented in Figure 7.

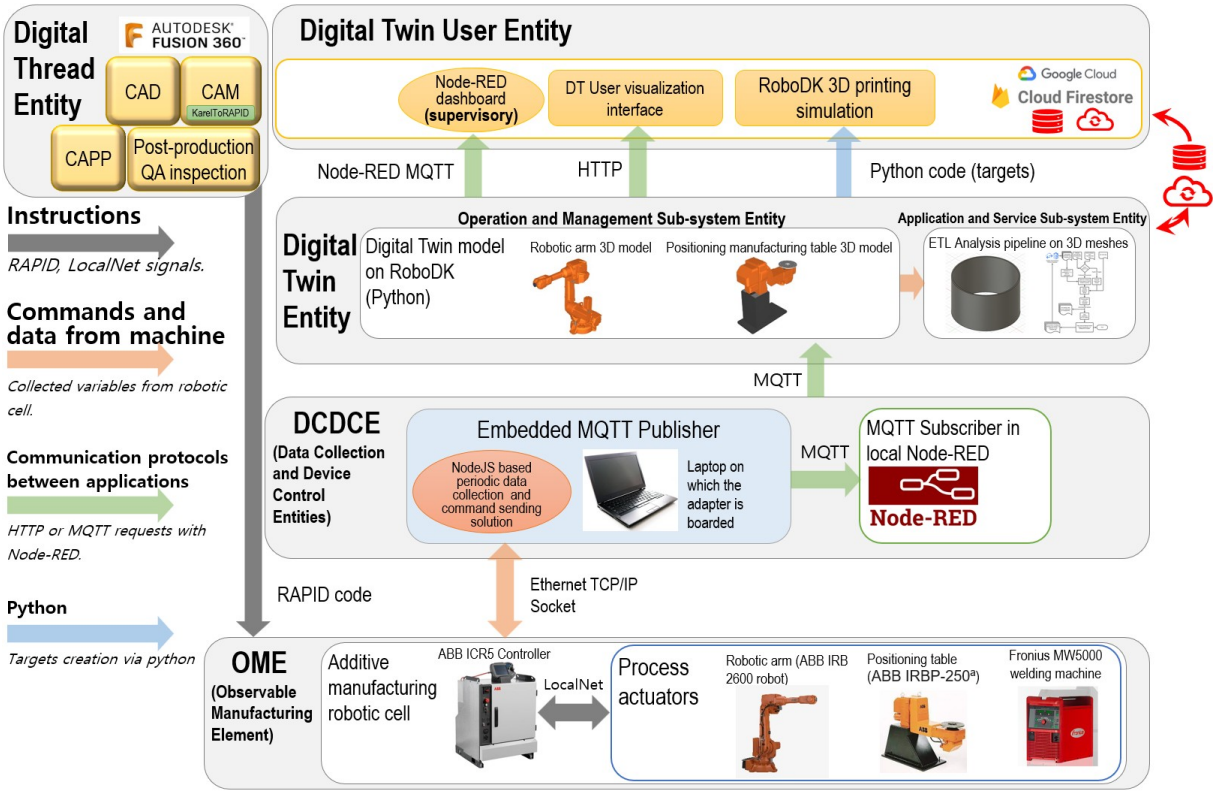


Figure 7 – New Digital Twin framework architecture based on ISO 23247 applied to a real metal Additive Manufacturing, an ETL pipeline was included as the Application and Service Sub-system, with its corresponding offline QA inspection in the newly conceived Digital Thread Entity domain.

To enable comparisons between expected results and obtained ones along the production cycle, it is necessary to collect positional data for the generation of 3D point cloud models to be compared with production parameters collected through the Digital Twin. CAD models and meshes of planned 3D parts are also used in this kind of quality assurance analysis.

In order to achieve the new proposed DT architecture based on the ISO 23247 and the previously conceived DT framework for the metal AM robotic manufacturing cell, a series of methodological steps explored throughout this work is required, which are evidenced in the process flowchart in Figure 8.

In the following sections, the proposed extraction and transformation of data will be detailed, having in mind the following questions: How to extract point cloud data of manufactured parts? How to align 3D meshes of different phases of the production cycle (point cloud extracted from data collected from the DT and point cloud generated through the finished part)? What methods of point cloud correspondence analysis and differences measurement between meshes are to be used in the data analysis? Finally, and most importantly, is it possible to establish a correlation between problems and differences observed between planned parts' models, collected positional data, production parameters, and 3D point clouds generated from manufactured parts with production faults, sensor calibration, and problems with the DT pipeline?

3.1 Digital Twin data and dashboard

The Digital Twin data acquisition pipeline presented in Figure 7 operates in 100 ms data-collection cycles, capturing data on the current program execution, welding source parameters, and positional parameters from both the robotic arm and the positioning table. Data is fetched through cycles of interruptions in the ABB RAPID code program that is being executed by the ABB IRC5 controller during deposition, sending messages through LocalNet Ethernet TCP/IP to a TCP/ IP socket server embedded in a laptop connected via Ethernet cabling connected to the ICR5 controller LocalNet module, as detailed in the previous work (CABRAL, J. V. A., 2023).

Fetched data is then organized in MQTT topics for local data distribution throughout nodes, with subscribers embedded in a local Node-RED separate process, concatenating the information into topics to be published via HTTP to the Google Cloud Firebase Firestore database. The description of variables, along with their corresponding topics, is available in Table 3.

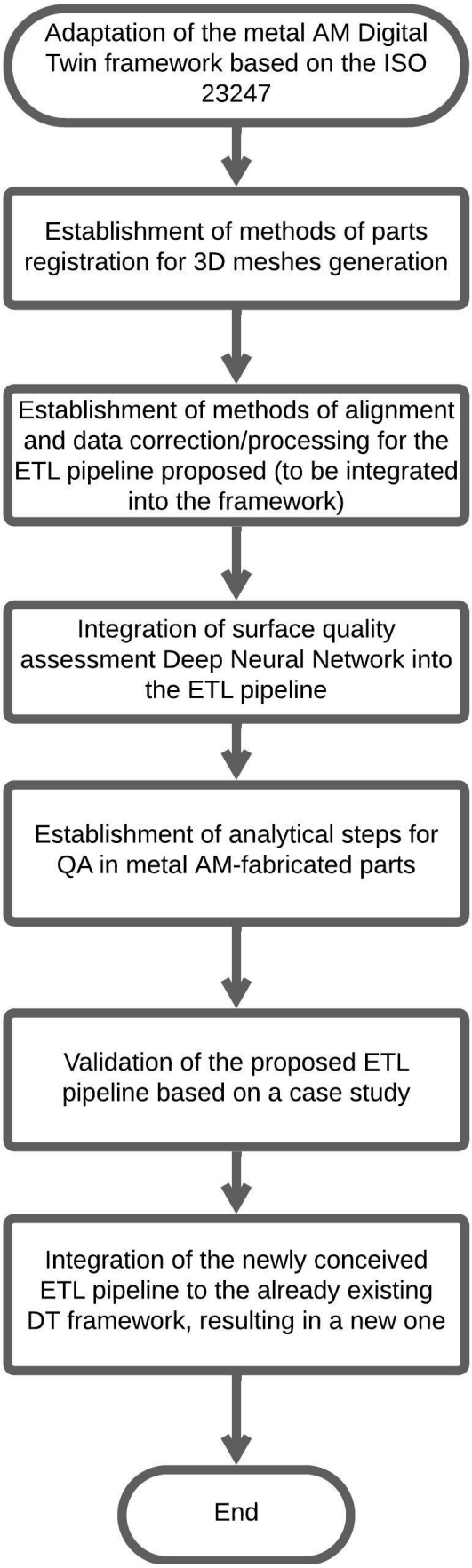


Figure 8 – Methodological steps flowchart for the adaptation of the already existing metal AM Digital Twin architecture framework into a new one, incorporating QA paradigms as an ETL pipeline incorporated into the Application and Service Sub-system and post-production QA steps integrated into the Digital Twin Thread Entity.

Table 3 – Variables collected in the ABB RAPID code and corresponding MQTT topics. Adapted from: (CABRAL, J. V. A., 2023).

Variable	**Collection method**	**Variable description**	**Corresponding topics**
Status	When the *socket* module is started in a RAPID program, it sends the message "Status;ONLINE" to the adapter server via TCP/IP *socket*; when the *socket* module is closed, the message "Status;OFFLINE" is sent.	Describes if the adapter is online and connected to the IRC5 controller.	"abb/irb2600/status"
Mode	The controller always sends a message via TCP/IP *socket* with the content "auto", since the *socket* module only works in the automatic execution mode of the controller.	Current execution mode of the IRC5 controller (manual or automatic).	"abb/irb2600/mode"
Execution	Program name defined in the RAPID code.	Name of the program being executed.	"abb/irb2600/execution"
Tool	RAPID function CTool() which returns the current tool variables.	Describes the current work tool: offset between the coupler and the TCP, orientation, and inertia if necessary.	"abb/irb2600/tool"
Cwobj	RAPID function CWobj() which returns the current work object variables.	"Current work object", refers to the current work coordinates upon which the robot calculates the global movement coordinates.	"abb/irb2600/cwobj"
Speed	A system output signal is defined in the IRC5 controller where the system variable "Tcp Speed" can be obtained.	Current TCP speed.	"abb/irb2600/speed"
Mspeed	RAPID function MaxRobSpeed().	Maximum linear robot speed in mm/s.	"abb/irb2600/mspeed"
RSpeed	Obtained via the RAPID function CSpeedOverride().	Rotation speed of the positioner table in percentage.	"abb/irb2600/rspeed"

Welding	Obtained via the I/O signal monitoring the Fronius source by the IRC5 controller "doFr1ArcOn".	Defines the values "TRUE" or "FALSE" for the current state of the deposition process, in terms of arc presence.	"abb/irb2600/welding"
WVoltage	Obtained via the I/O signal monitoring the Fronius source by the IRC5 controller "aiFr1WeldingVoltage".	Current welding voltage in volts.	"abb/irb2600/wvoltage"
WCurrent	Obtained via the I/O signal monitoring the Fronius source by the IRC5 controller "aoFr1Power".	Current welding current in amperes.	"abb/irb2600/wcurrent"
WSpeed	Obtained via the I/O signal monitoring the Fronius source by the IRC5 controller "aoFr1WireSpeedWfi".	Current wire feed speed in mm/s.	"abb/irb2600/wspeed"
PosRotation	Rotation of the external robot axes obtained from the "extax" attribute of the output of the RAPID function CRobT() "current robot target".	Rotation of the external robot axes (positioner table rotation in degrees).	"abb/irb2600/posrotation"
CRobT	Translational ("trans") and rotational ("rot") attributes of the robot's current "target" obtained via the RAPID function CRobT().	Current robot position variables ("target"), Cartesian coordinates, and orientation in quaternion form.	"abb/irb2600/crobt" "abb/irb2600/x_pos" "abb/irb2600/y_pos" "abb/irb2600/z_pos" "abb/irb2600/rorient"
Timestamp	Obtained via the RAPID functions CDate() and CTime()	Current date and time for reference of the end of the variable collection cycle by the controller.	"abb/irb2600/timestamp"

During the deposition process, it is possible to monitor acquired parameters and assess the integrity of the acquired data throughout the pipeline via an embedded Node-RED dashboard in a separate application thread, which can be observed in Figure 9.

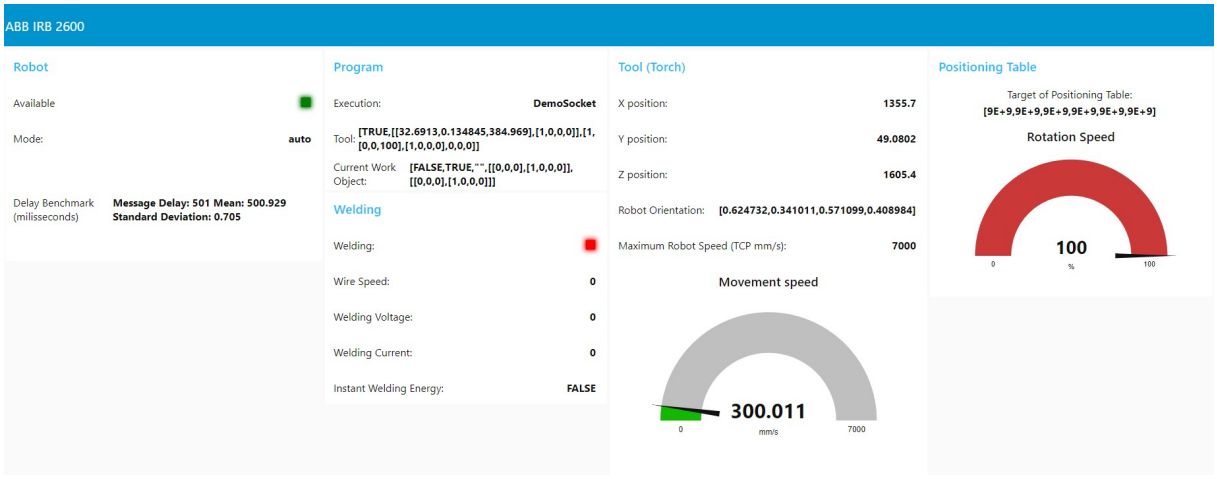


Figure 9 – Local MQTT-based Node-RED dashboard of acquired process variables visualized during acquisition (CABRAL, J. V. A., 2023).

During data acquisition test cycles, it was determined that packages that were sent to the Google Cloud Firebase Firestore database had a bimodal stable sampling distribution centered at 100 ms (10 Hz) with a secondary mode at 190 ms (retransmissions, as seen in the histogram in Figure 10), with an estimated accuracy on first transmission of 95%, and no significant data loss.

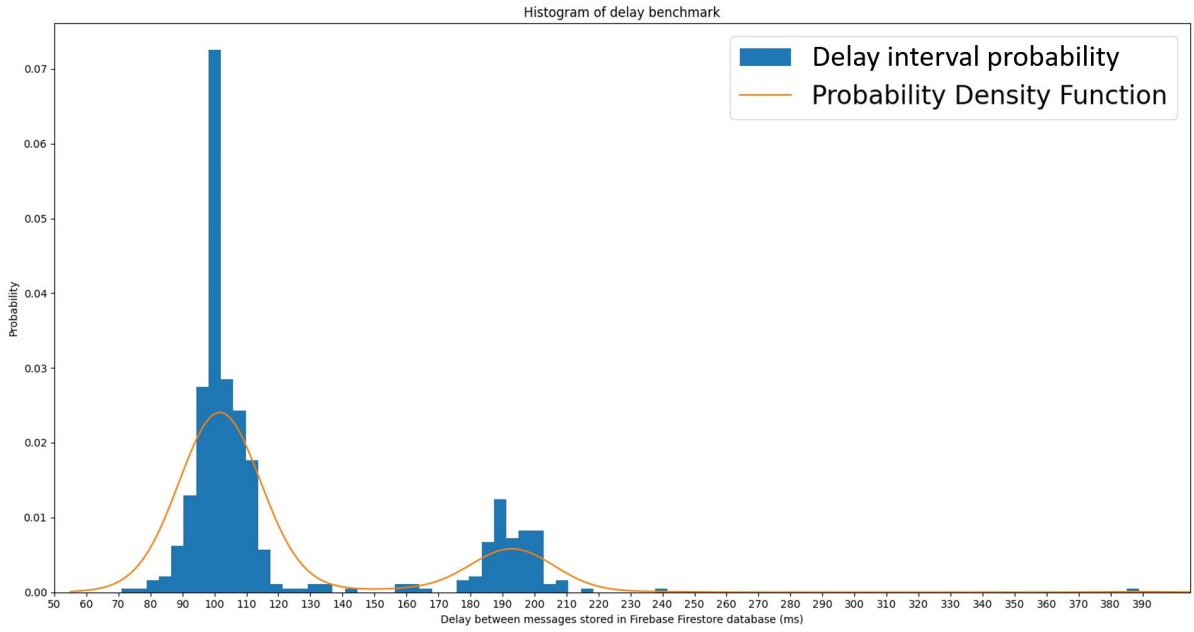


Figure 10 – Histogram of end-to-end message latency from the ABB IRC5 controller to the Firebase Firestore database. The distribution shows a consistent sampling frequency centered at 100 ms (10 Hz), confirming the stability of the extraction pipeline. Secondary peaks (e.g., $\approx$ 190 ms) correspond to successful TCP/IP retransmissions, validating the data integrity protocol.

The Digital Twin is also capable of simulating the movement of the robotic arm, positioning table, and toolpath during deposition, through acquired positional, rotational, and welding activity data fetched through the local MQTT broker. This simulation is done in RoboDK version 5.9 software. It is also possible to simulate the toolpath for parts generation before starting the deposition process, simulating through the virtual workstation configured in RoboDK. An example of a simulated toolpath for the helical single-pass per layer deposition strategy for a hollow cylinder is available in Figure 11.

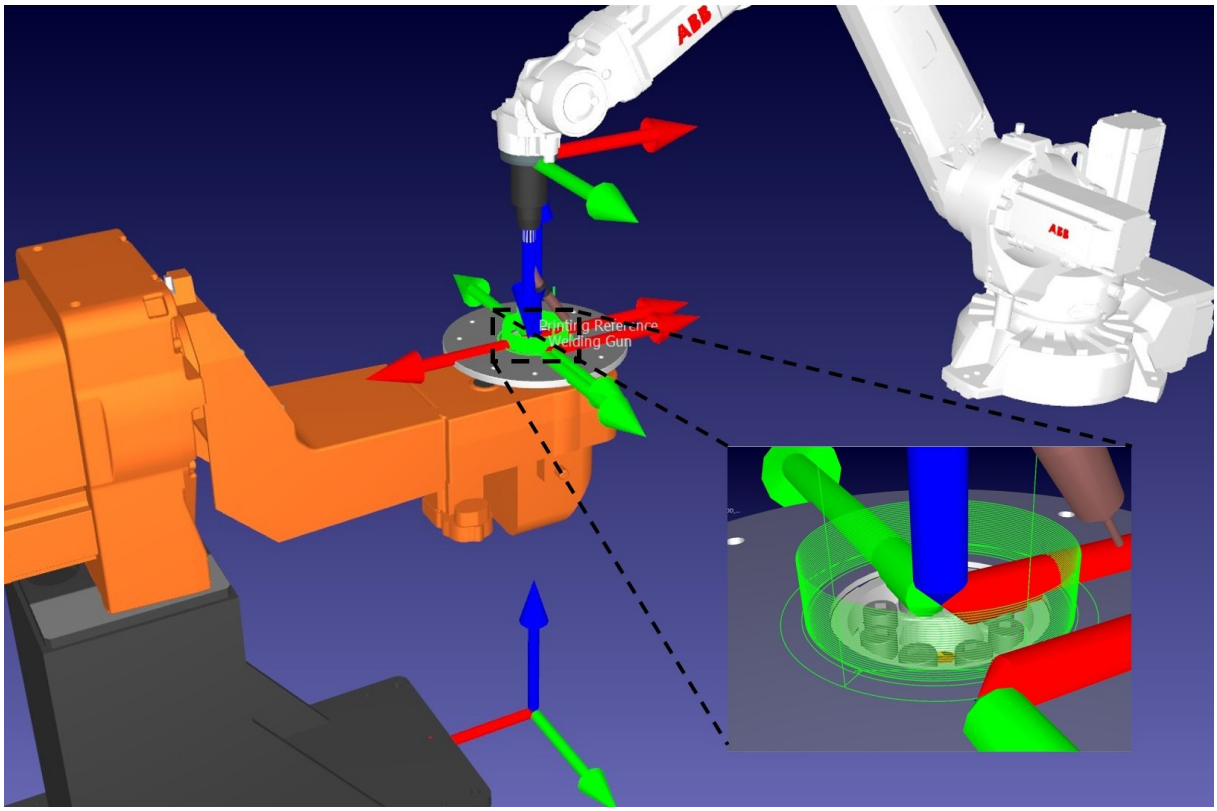


Figure 11 – Simulated virtual RoboDK workstation for the digital representation of the real robotic metal AM cell. Helical single-pass per layer continuous TCP ascending Toolpath for a 120 mm diameter, 30 mm height, 5 mm wall width hollow cylinder highlighted in the bottom right corner.

Demonstrations of the functioning of the Digital Twin, along with its 3D simulation of movement and real-time dashboard, are available in https://www.youtube.com/watch?v=UGdrRNocmDg and https://www.youtube.com/watch?v=V7_o5d0dj0Y.

The collected welding parameters are further used to analyze the deposited parts and deposition strategies in section 4.2. Furthermore, collected positional data from the Firebase Firestore database is used to comment on trajectory definition and to generate 3D point clouds, which are used for the analysis of deposited parts in subsection 4.2.2.

3.2 Extraction of data and methods of point cloud generation

This section is divided into a few subtopics related to problem-solving in the extraction and pre-processing of data for further analysis. Although positional data is already easily obtained through the cloud database, the raw data must be converted into point cloud data and have its density of points corrected to generate 3D meshes afterwards. Regarding the 3D CAD models, STL files, which are generally used in 3D printing, can be generated in commonly used programs such as Autodesk Fusion, Rhino, Blender, etc. In this work, the CAD files for reference are generated via Autodesk Fusion version 2605.1.39 software.

The main challenge involved in data extraction is the uniform collection of points for the 3D point cloud frames, more specifically, the density of the points distribution depending on the sampling method. Moreover, noisy data might also be filtered, generating clusters of scanned objects.

Clustering methods such as DBSCAN clustering (ESTER et al., 1996) may be applied to the generated 3D point cloud in order to filter noisy points generated through program execution and data collection of the Digital Twin pipeline.

After proper clustering of data points, a likely problem with the positional data collected via the Digital Twin is oversampling during data-collection cycles. Reducing the density of the point cloud frame is then necessary to lower the computational power required to generate the 3D meshes; moreover, it is important to maintain uniform density across frames to facilitate the calculation of differences.

To correct this problem, a method named Screened Poisson Surface Reconstruction (SPSR) by Kazhdan et al. (KAZHDAN; HOPPE, 2013) is used. In this method, the gradient of points along boundaries of surfaces on point cloud and meshes data is adjusted so a smoother result of the objects' surfaces is achieved, granting a better distribution of points and better noise filtering.

The second dataset needed for the QA analysis is related to the real manufactured parts. To filter noisy points and generate better 3D point clouds from manufactured parts, DBSCAN might also be applied after point extraction if the data are obtained via LiDAR, photogrammetry, or laser triangulation 3D scanning of the real parts. In this case, the density correction SPSR might be applied because of scanning problems such as insufficient sampling rate, angles of the part not properly covered during the scanning process, and possible duplicate points.

Regarding photogrammetry-based positional data extraction, an iPhone 16 Pro Max is used in macro mode (48MP Ultra Wide camera) using PolyCam software for point acquisition. LiDAR-based positional data extraction is obtained using the same hardware and the same

software.

For laser-based extraction of datapoints used in the validation of the scans processed through the ETL pipeline, Multi-stripe Laser Triangulation is used via the NextEngine 2020i desktop 3D laser scanner along with the ScanStudio PRO software. Macro mode focus was used with the SD mode of 4.4k points per inch (PPI), utilizing a 16-segment rotation strategy to ensure full surface coverage. This configuration establishes a sampling density of $\rho_s = 4{,}400$ PPI, resulting in a theoretical spatial resolution of $R_{sp} = \frac{25.4\,\text{mm}}{4400} \approx 5.8\,\mu\text{m}$ (0.0058 mm). The nominal accuracy of the device is ± 0.127 mm ($\pm 0.005''$) (inches), ensuring that the meso-scale geometric features of the WAAM layers are preserved for the validation comparison. The 3D laser scanner is illustrated in Figure 12.

Figure 12 – NextEngine 2020i 3D laser scanner (ANIWAA, 2026).

3.3 Generation of 3D meshes

Following the discussion of the generation of 3D models of manufactured parts presented in sections 2.5 and 3.2, the next step after correcting the possible errors in the point cloud generation involves generating the 3D meshes based on those positional data clusters.

Since the CAD models used as a base for the manufacturing are already considered a mesh, this process will be skipped on that data. Regarding the positional (x, y, and z axes) data collected via the DT and possibly also point clouds collected from the manufactured data via LiDAR or 3D scanning, both datasets are suitable for processing through mesh generation algorithms. The chosen way of generating 3D point cloud models through vectorial data is using Open3D's (OPEN3D TEAM, 2025) Python library for creating triangle mesh STL files.

Through combining PolyCam software and Open3D's algorithms for registration, alignment, and mesh generation, high-fidelity digital representations of the physical parts are reconstructed via photogrammetry. The core mechanism employed by the PolyCam solution is fundamentally rooted in the Structure from Motion (SfM) (OZYESIL et al., 2017) technique discussed in section 2.2.1.

For metrological validation of the 3D scans analyzed through the ETL pipeline, 3D meshes and 3D point clouds are also generated via the ScanStudio PRO software for the NextEngine 2020i desktop 3D laser scanner. Post-extraction of points, align and fuse functions (UNIVERSITY OF FLORIDA, n.d.) are used for referential correction of the extracted scans of point clouds and for generating watertight surfaces for the exported 3D meshes from the program, respectively.

The implementation of 3D meshes generation algorithms will be further commented on in chapter 4.

3.4 Alignment of 3D point clouds to enable comparison of differences

As cited in section 2.5, a collection of ICP methods of aligning two 3D point cloud frames can be used to guarantee the correct comparison of generated point cloud data.

To achieve the desired fine alignment of frames, the Python Open3D point-to-point ICP algorithm will be applied to the point cloud data. This approach was selected as it constitutes the standard, robust method for minimizing the Euclidean distance between point sets without relying on surface normal estimations, which can be computationally expensive or noisy on rough surfaces (BESL; MCKAY, 1992).

The point-to-point algorithm present in the Open3D Python library is based on the seminal article by Besl and McKay (BESL; MCKAY, 1992), which presents a general-purpose solution for the accurate registration of 3D shapes. The approach revolves around iteratively searching for the closest point in the reference model for each point in the source model, and estimating the rigid-body transformation (rotation and translation) that minimizes the mean squared error (MSE) of these correspondences.

It is important to note that complex geometries with many edges close to each other

might generate non-optimal alignment across iterations of the ICP algorithm, as a local best alignment might result, not achieving the best possible alignment (DECKER; WANG, Y.; HUANG, Q., 2020).

3.5 Comparison between point cloud frames and evaluation metrics

As a heuristic threshold for comparing 3D models for further AI-based QA analysis, developed models can be compared after all meshes are generated to gather data on their differences. To achieve this, each point of the mesh frames must be compared to the other frames, and the distance between the closest points of the two frames is registered.

To validate the alignment quality among the Digital Twin-generated point cloud, the photogrammetry scans, CAD models, and ground truth ones, quantitative error metrics are computed following ICP convergence. This framework utilizes the Root Mean Square Error (RMSE) (DECKER; WANG, Y.; HUANG, Q., 2020), presented in Equation 3.1 to confirm the robustness of the global registration.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|p_i - q_i\|^2} \tag{3.1}$$

Where:

- **RMSE** — global alignment error metric quantifying the average deviation between the point clouds
- N — total number of point correspondences used
- p_i — the i-th point in the source point cloud P
- q_i — the corresponding point in the reference point cloud Q for p_i
- $\|\cdot\|$ — Euclidean norm (L2 norm) between vectors (use $\|p_i - q_i\|$ for point-to-point distance)

Furthermore, the Hausdorff distance (presented in Equation 3.2) (CABRAL, J. V. A.; ALVARES, A. J., et al., 2026) is calculated to measure the maximum local deviation between the planned trajectory and the actual deposited surface. This metric is critical as it quantitatively captures the amplitude of significant physical defects—such as localized balling and layer overbuilding—allowing these geometric anomalies to be distinguished from the global surface alignment.

$$d_H(P,Q) = \max\left\{ \max_{p\in P} \min_{q\in Q} \|p-q\|, \max_{q\in Q} \min_{p\in P} \|q-p\| \right\} \tag{3.2}$$

Where:

- $d_H(P,Q)$ — symmetric Hausdorff distance representing the maximum geometric discrepancy between the two datasets
- P — source point cloud set
- Q — reference point cloud set
- $\max_{p\in P} \min_{q\in Q} \|p-q\|$ — forward distance: for each $p \in P$ find the nearest $q \in Q$, then take the largest such nearest distance
- $\max_{q\in Q} \min_{p\in P} \|q-p\|$ — backward distance: for each $q \in Q$ find the nearest $p \in P$, then take the largest such nearest distance

3.6 Deep neural network model for quality classification of manufactured parts

The quality assurance framework within the Digital Twin-enabled robotic cell employs a specific AI model for defect classification. A PointNet (QI et al., 2016) deep neural network is utilized to evaluate the superficial roughness and conformity of metal parts manufactured via the GTAW process.

PointNet, a deep neural network designed for processing point cloud data, has shown promising applications in classifying and segmenting 3D models and meshes. Unlike traditional image-based approaches, PointNet directly analyzes 3D point clouds, capturing fine-grained surface details that are crucial for assessing roughness variations. This capability is particularly valuable in GTAW-based AM, where thermal fluctuations and deposition inconsistencies can lead to surface defects.

In a Digital Twin framework, the adapted PointNet model can be integrated to provide feedback on surface quality, enabling the fine-tuning of welding parameters. By analyzing point cloud data previously acquired from laser scanners or photogrammetry, the model can detect deviations in roughness all along the surface of the scanned parts. Thus enabling data-backed decision making.

3.7 Fuzzy-based method of discriminating manufacturing parameters problems, calibration, and model inaccuracies

Regarding inaccuracies on deposition and discrepancies among extracted 3D models, firstly, fuzzy logic is proposed to analyze the processed data fetched through comparisons of the positional data of planning, positional data of DT data, and the finished manufactured parts through 3D scanned photogrammetry. This logic is based on the ICP alignment metrics of RMSE and Hausdorff, based on point-to-point comparisons of the 3D point clouds, establishing a threshold of tolerance of RMSE and Hausdorff distance values previously discussed in section 3.5.

Using the original CAD models of the parts as a basis for the threshold and comparing their differences from the other models, a fuzzy strategy for decision-making-oriented analysis is proposed based on the logic presented in the table in Figure13.

O = Sufficiently equal. **X** = Unequal.			Decision-making analysis based on inequalities among the 3D meshes models
3D point clouds from CAD files	3D point clouds from DT positional data	3D point clouds from SPSR applied to scanned real manufactured parts	
O	O	O	The part was correctly manufactured, no action must be taken.
O	O	X	There is probably an error related to the welding parameters (wire feed speed or TCP speed) or idle time between layers deposited (required for the molten metal pool to solidify).
O	X	O	Since the part was correctly manufactured but the 3D mesh produced from the DT data is wrong, there is probably an error with the DT pipeline (data transmission, rounding errors, etc.).
X	O	O	Probably there is a problem with the sensors' calibration of the robotic cell, the welding parameters are correct since the DT mesh model and the manufactured part one are sufficiently equal.
X	X	X	The RAPID program of material deposition was incorrectly programmed, the TCP path required to produce the part is wrong and the welding parameters might also have errors.

Figure 13 – Decision-making based on inequalities among generated 3D meshes.

Using Open3D's (OPEN3D TEAM, 2025) Iterative Closest Point (ICP) point-to-point

registration, the geometric fidelity of the Digital Twin is evaluated by superimposing generated positional point clouds against photogrammetry scans of the physical parts. To quantify the correspondence, specific validation thresholds were established based on the physical constraints of the CW-GTAW process. An acceptable Root Mean Square Error (RMSE) baseline was defined at approximately 1.5 mm to account for the intrinsic roughness of as-built WAAM surfaces. Furthermore, a maximum Hausdorff Distance tolerance of 4.4 mm was established—derived from the fixed 4 mm arc length plus a 10% tolerance margin. Deviations exceeding this upper bound indicate a violation of the process stability window (e.g., electrode collision or arc extinction) rather than simple surface roughness. Consequently, adherence to these geometric tolerances constitutes a sufficient correspondence, enabling the subsequent application of the fuzzy-based logic decision-making framework presented in Figure 13.

Highlighting the contribution of the fuzzy strategy within the proposed methodological steps, it is important to note that this logic acts as a vital link among the Digital Twin visualizations, the photogrammetry scans, and the original CAD models. By applying specific geometric tolerances, this strategy enables the validation of the entire framework, justifying the application of the Digital Twin as a precursor for enhancing the Computer-Aided Process Planning (CAPP). Furthermore, this fuzzy-based approach serves as a foundational precursor for automated decision-making, establishing a baseline that can be further explored and expanded in future works.

3.8 General methodology and pipeline

The methodology presented will be applied to the many generated models of manufactured parts to support decision-making and the analysis of the quality of manufactured parts. The pipeline of data makes use of the ETL cycle principles to form a framework for the desired analysis. In this sense, the general planned pipeline workflow can be seen in Figure14.

It is important to highlight that, besides the general decision-making analysis applied through the presented pipeline, the generated 3D mesh files might also be used to compare the generated models visually, adding another factor to the machine operators' decisions regarding the manufacturing process.

In the next chapter, all the modules and data involved in the pipeline framework illustrated in Figure14 will be explained in detail with their respective algorithms and results of implementation discussion.

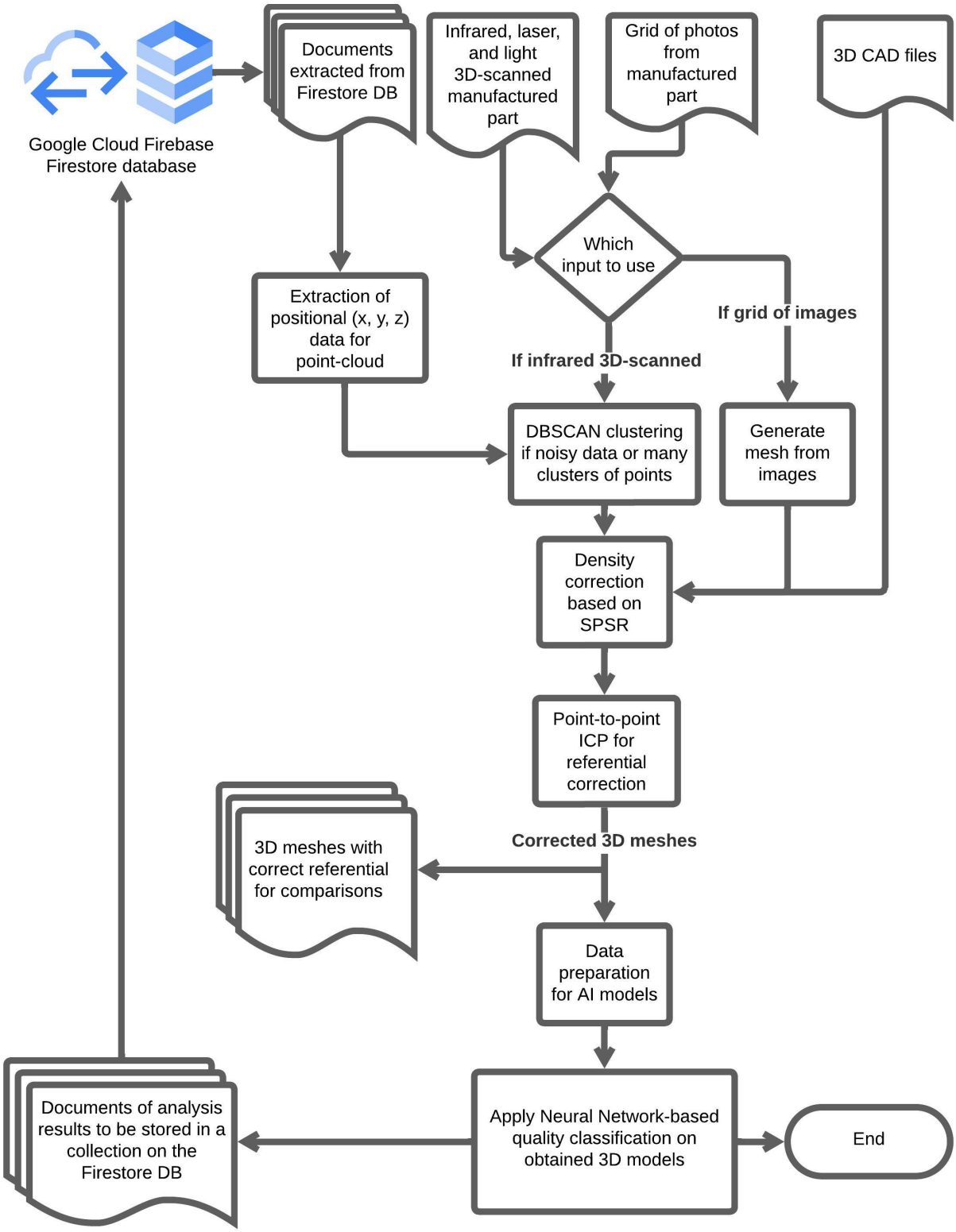


Figure 14 – Planned pipeline for data transformation and surface quality classification based on generated 3D meshes.

4 Case study: experimental apparatus and results

Throughout this chapter, the specific parts that compose the entire ETL data pipeline framework proposed in chapter 3 will be discussed in their respective implementations and results.

An overview of the parameters utilized on metal deposition at the robotic manufacturing cell will also be provided. This study is aimed at finding the relation between the fine-tuning of the GTAW process and the overall quality of fabricated parts based on the general quality of parts classification models.

Such quality assurance models are built upon the structure that the framework provides and will be scrutinized in section 4.4; that section also provides training details and the results of testing with simulated 3D parts with different levels of expected quality. Results of the application of said AI and heuristic models in the context of metal-manufactured parts QA analysis are discussed in section 4.5.

4.1 Deposition system setup

The experimental apparatus for WAAM, illustrated in Figure 15, was comprised of a 6-axis ABB IRB 2600ID-15/1.85 robotic arm, an IRBP A-250 two-axis positioner, and a Fronius MagicWave 5000 power source. All components were integrated and operated by an IRC5 controller via the LocalNet network. For the deposition, ER70S-6 mild steel wire (1.2 mm diameter) was used as the additive material, deposited onto AISI 1020 steel plates ($150 \times 150 \times 4.75$ mm). The process was shielded by pure argon at a 14 l/min flow rate, and a constant arc length of 4 mm was maintained.

For the validation of the framework comprised in this current work, a thin-walled hollow cylindrical geometry was selected as the case study. This geometric profile is highly representative of industrial applications, particularly within the Oil & Gas sector for piping and pressure vessel components. Furthermore, the choice of a tubular shape significantly simplifies the generation of the Tool Center Point (TCP) trajectory, allowing for a continuous deposition path without the need for complex slicing algorithms required for free-form solids. This geometric simplicity ensures that the analysis focuses on the surface variations and process instabilities rather than path planning artifacts, providing a clear baseline for the Digital Twin's validation.

The programming of the continuous unidirectional deposition path required an

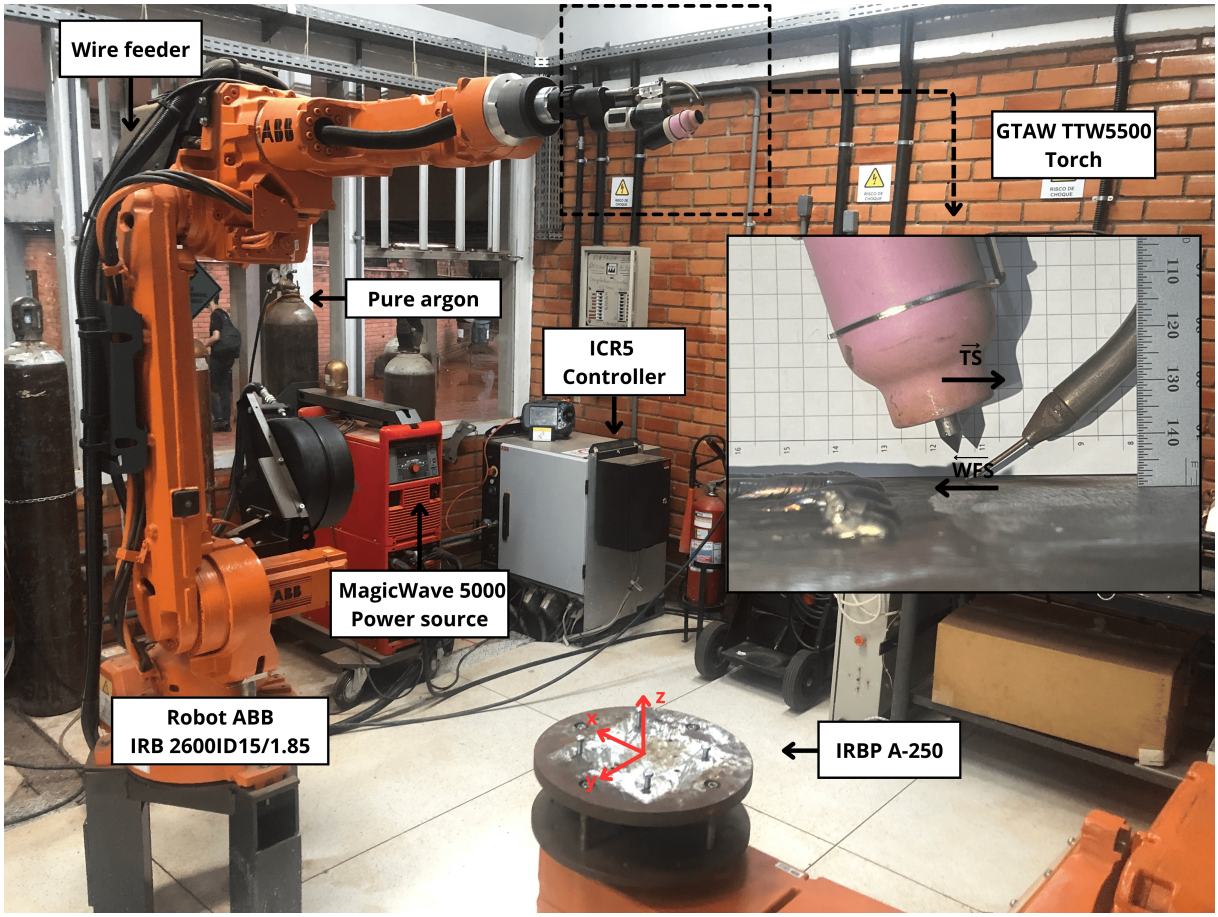


Figure 15 – Cold Wire GTAW deposition system (CABRAL, J. V. A.; ALVARES, A. J., et al., 2026). Amplified visualization of the deposition setup is highlighted in the right box, TS: Torch Speed; WFS: Wire Feed Speed.

estimation of the expected average layer height (1 mm). This value was determined using data from preliminary experiments, which involved depositing discontinuous and unidirectional arc planar walls. During these initial tests, TCP positioning data was collected at the electrode contact points, allowing for the calculation of the mean layer height used in the main depositions.

As discussed in the previously developed work on the original Digital Twin framework (CABRAL, J. V. A., 2023), cylindrical coordinates (r, θ, Z) are used to describe the trajectory of the TCP during deposition. In this strategy, variation of the TCP position along the Y axis is used to increase or reduce the radius of the cylinder - which isn't used in this current work as the diameter of the deposited tubular parts is unchanged throughout their layers - whereas Z position is used to change from layer to layer, the angle θ is then controlled during the deposition by the rotation of the positioning table, completing the deposition perimeter of each layer. This also means that the deposition speed is directly related to the turntable spin's speed and controlled by input parameters in the RAPID code to the external axis of the positioning table. This behavior is exemplified in the diagram in Figure 16.

As will be further detailed in subsection 4.2, four tubular specimens were produced to

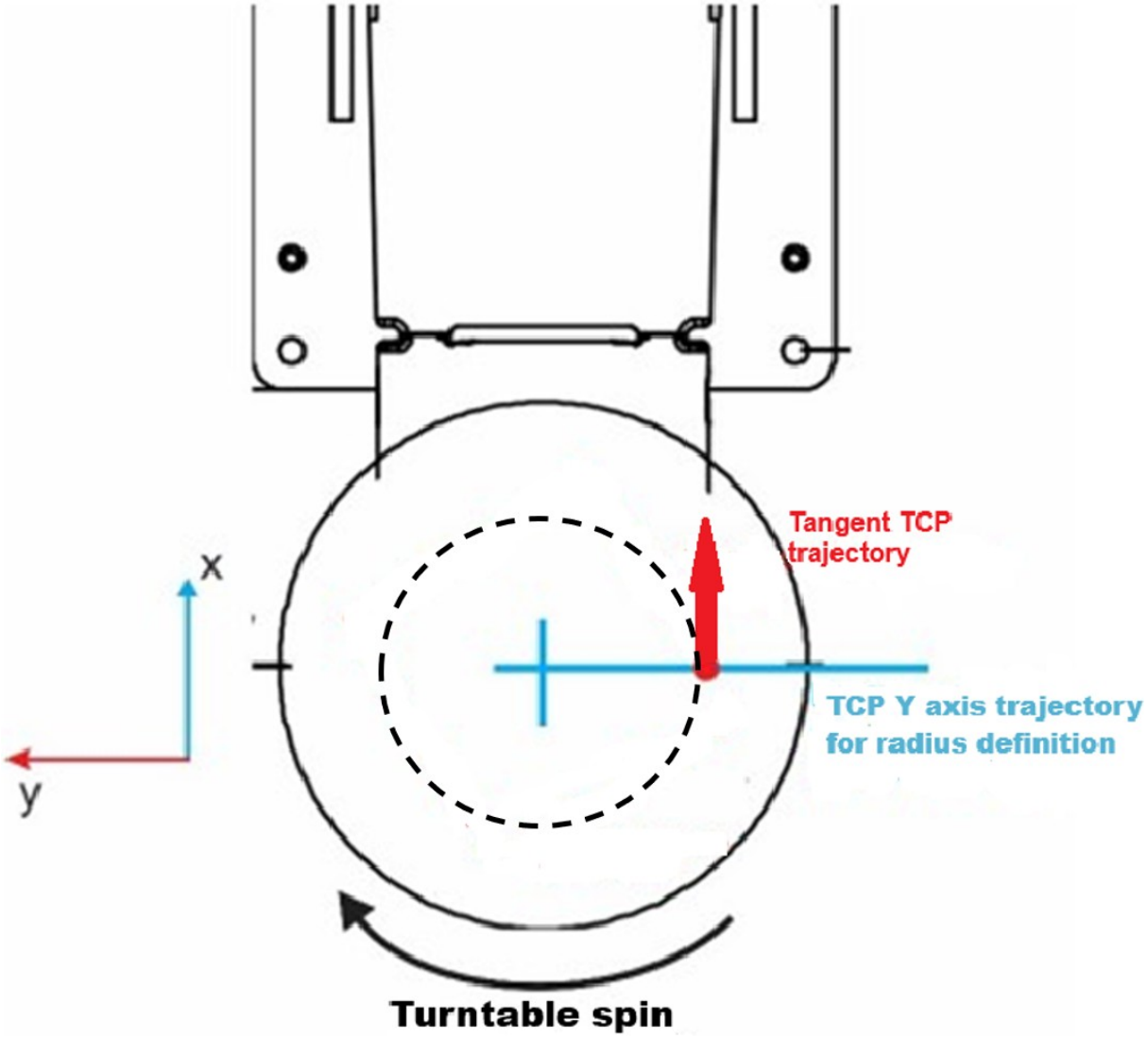


Figure 16 – Description of the TCP's movement based on the rotation of the positional table for the cylindrical coordinates approach. Adapted from: (SALES DE MATOS, 2022).

evaluate two different manufacturing strategies. The first three specimens were all fabricated using a "step-by-step" deposition trajectory, but with distinct process parameters for each. The fourth specimen, in contrast, was produced using a "helical" trajectory, which also had a unique parameter set. This experimental design allows for a clear assessment of how variations in both the deposition path and process parameters influence the final geometry and material distribution of the fabricated cylinders.

4.2 Experimental analysis and parameters acquired during deposition

In this section, the experimental parameters of deposition will be discussed. Besides the welding parameters themselves (welding current, wire feed speed), there are also parameters related to the robotic arm and positioning table, such as linear speed, table rotation

speed, and continuous or discontinuous Z-axis ascending rate.

Another aspect of the deposition process to be discussed is the trajectory definition of each experiment and deposition strategies of each manufactured part, which will be covered in subsection 4.2.2.

4.2.1 Welding source parameters and deposition experiments

The welding parameters among the experimental depositions of hollow metal parts varied as the process was being refined throughout the iterations of deposition. In this subsection, the selected parameters for experimentation and their results on surface roughness and overall finishing/superficial defects will be evaluated. All of the experiments with different welding parameters testing were conducted at the same AM robotic cell, composed of the robotic arm ABB IRB 2600, the positioning table ABB IRBP A250, the ICR5 controller, and the welding source Fronius MW5000. Also, all the experiments utilized the same material for the wire feeding input at the Fronius torch, which was a 1.2 mm diameter low-carbon ER70S6 steel wire.

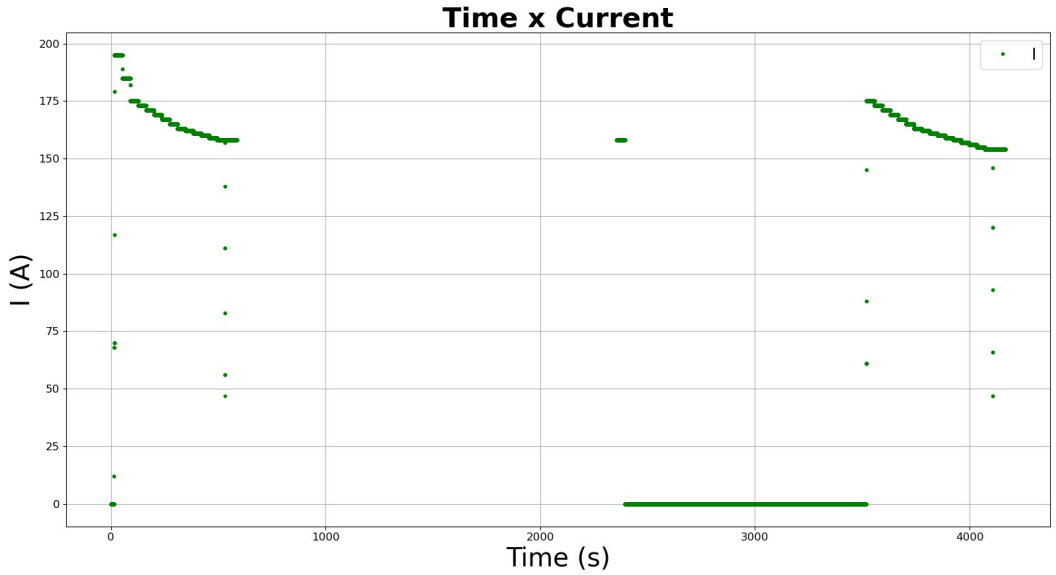
The first four successful cylinder-hull deposition experiments were considered sufficient to provide a comprehensive dataset of different quality levels and identified defects for applying the prototype of the proposed framework with a DNN-based QA pipeline. Various parameters and trajectory strategies were adopted to yield different overall quality results and dimensional properties, which will be discussed throughout this subsection.

The first welding source parameters strategy wasn't successful in providing a continuous cord during deposition, and several balling, stubbing, and overbuilding visible superficial structural defects (LIU, M. et al., 2021) were observed on the finished part. Since it was the first time experimenting with hollow parts manufacturing at the manufacturing cell, some parameters, such as maximum and minimum value and step variations on current, and input wire feed speed, were off. Those parameters were then reviewed and improved throughout the conducted experiments. Albeit not having presented good results on finishing and general structure, this first deposition was further used in the validation of the AI models regarding quality assurance and superficial roughness classification, further commented on in sections 4.4 and 4.5.

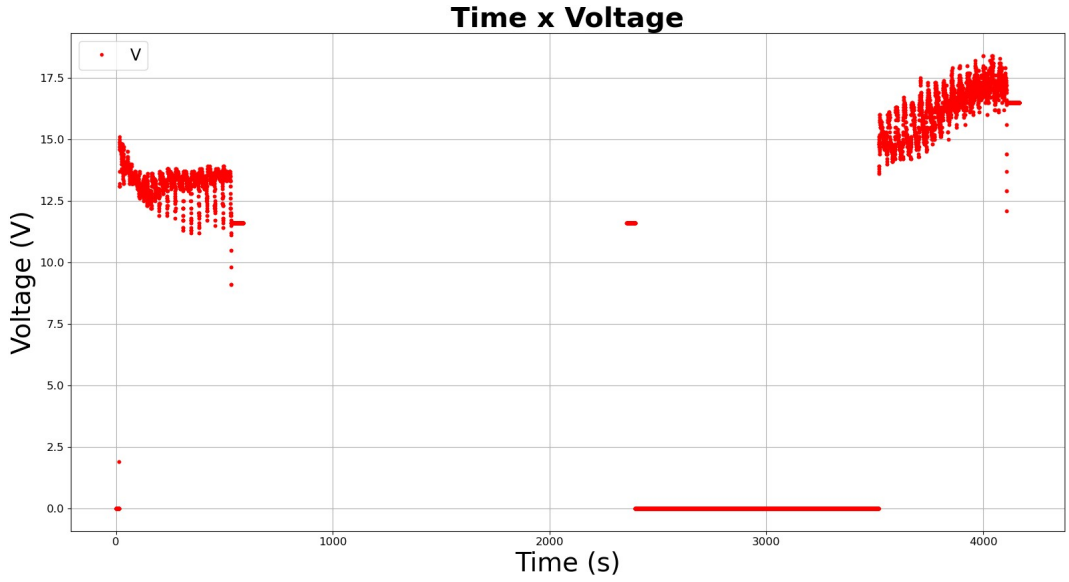
Welding parameters used in the first test of part deposition are related in Table 4.

Those parameters were varied during the deposition and captured by the Digital Twin data pipeline to be stored in Google Cloud. The current was varied with a strong, steep decrease, which might have contributed to the superficial defects observed. Graphs obtained from the history of data available in the Cloud Firestore database are shown in the grid of images in Figure 17, presenting current, voltage, and wire feed speed variation through time.

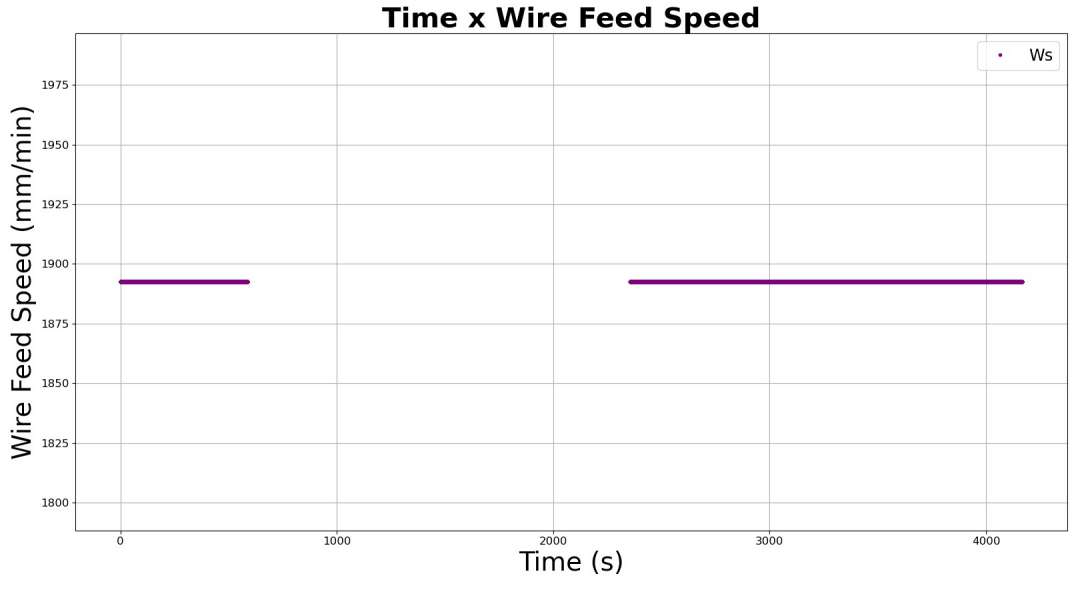
The cooling interval is clearly observable in the graphs, which indicate a pause of



(a)



(b)



(c)

Figure 17 – (a) Current graph of deposited part using the first step-by-step strategy, (b) Voltage graph, and (c) Wire feed speed graph (constant throughout the deposition).

Table 4 – GTAW Parameters used in the first experiment using the first strategy mentioned.

GTAW Deposition Process Setup Overview (First experiment, step-by-step strategy)	
Arc Current (A) first half	195 → 158 (firstly 2 steps of -10 A then steps of -2 A and steps of -1 A at the second half of the first session)
Arc Current (A) second half	175 → 154 (steps of -2 A and steps of -1 A at the second half of the second session)
Arc Voltage (V) first half	Approx. 11.00 – 15.19
Arc Voltage (V) second half	Approx. 13.50 – 18.50
Wire Feed Speed (mm/min)	1893 (constant)
Travel Speed (mm/s)	5 (constant)
Trajectory strategy	Step-by-step approach with cooling time

approximately 25 minutes between the two deposition sessions. A video recording specifically documenting the second session of this first experiment is available at https://youtu.be/IwUhHcJp_5w. This footage was captured using a smartphone camera equipped with a welding filter, mounted adjacent to the positioning table to visualize the deposition process.

The noted superficial defects (major stubbing, overbuilding, balling and necking, besides the general overbuilding in a “ladder” effect) can be seen in the picture taken from the deposited part in Figure 18. The dimensional parameters obtained from the finished part can be observed in Table 5.

Figure 18 – Pictures taken from the deposited part of the first step-by-step successful experiment.

The first experimental deposited part presented poor finishing with a great variation of thickness among layers and a bad quality overall, needing several improvements for a smoother part with better general quality through the refining of deposition parameters and trajectory strategies. Acquired dimensional parameters from the deposited part after the experiment are shown in Table 5.

Throughout the experiments, in particular from the first experiment to the second and third experiments, variables obtained through the Digital Twin data pipeline were

Table 5 – Deposition dimensional parameters acquired from the first deposited part.

Dimensional Parameters	
Mean height (mm)	28.5
Number of layers	30
Mean layer height (mm)	0.95
Mean thickness (mm)	6.7
Programmed deposition diameter (mm)	80
Passes per layer	1

utilized to tune deposition parameters, as sudden variations (drops) in voltage as observed in Figure 17 (b) usually indicate the presence of major defects, such as stubbing in this case, as the input wire wasn't fully melting before contacting the molten pool, causing parts of the wire to stick and merge with the wall, resulting in the effect observed in the pictures in Figure 18.

For the second successful experiment, a variation of the first strategy of cooling time between halves of deposition was applied, this time, welding parameters were better tune in order to achieve better finishing and to mitigate major defects, the main parameter being the increase of the travel speed from 5 to 7 mm/s, as seen along another minor changes such as a lower initial welding current in the deposition input parameters Table 6.

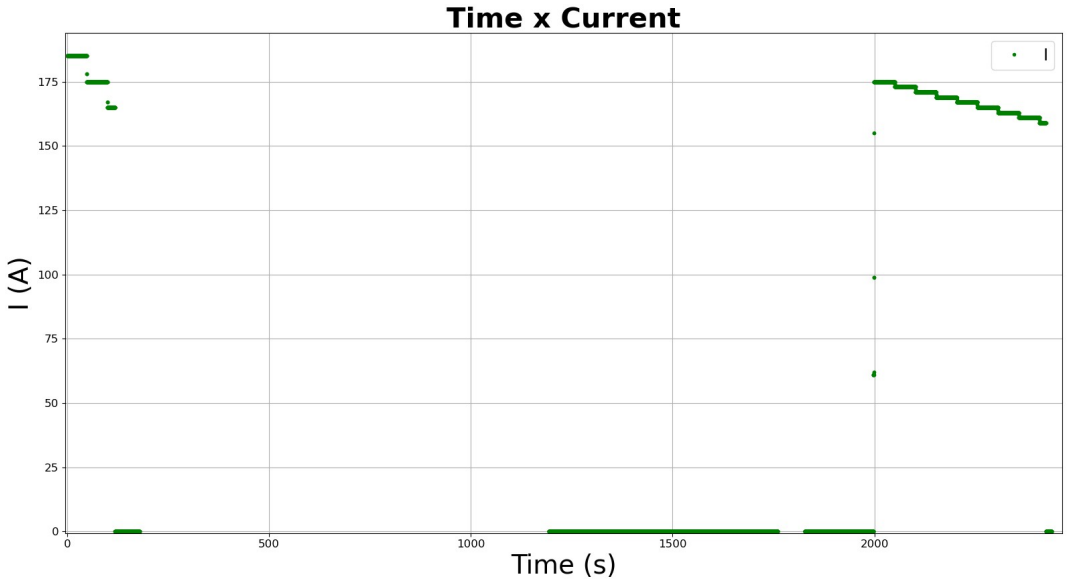
Table 6 – GTAW Parameters used in the second experiment using the first strategy mentioned.

GTAW Deposition Process Setup Overview (Second experiment, step-by-step strategy)	
Arc Current (A) first half	185 → 165 (2 steps of -10)
Arc Current (A) second half	175 → 159 (steps of -2 A)
Arc Voltage (V) first half	Approx. 14.88 – 11.86
Arc Voltage (V) second half	Approx. 15.80 – 11.78
Wire Feed Speed (mm/min)	1893 (constant)
Travel Speed (mm/s)	7 (constant)
Trajectory strategy	Step-by-step approach with cooling time

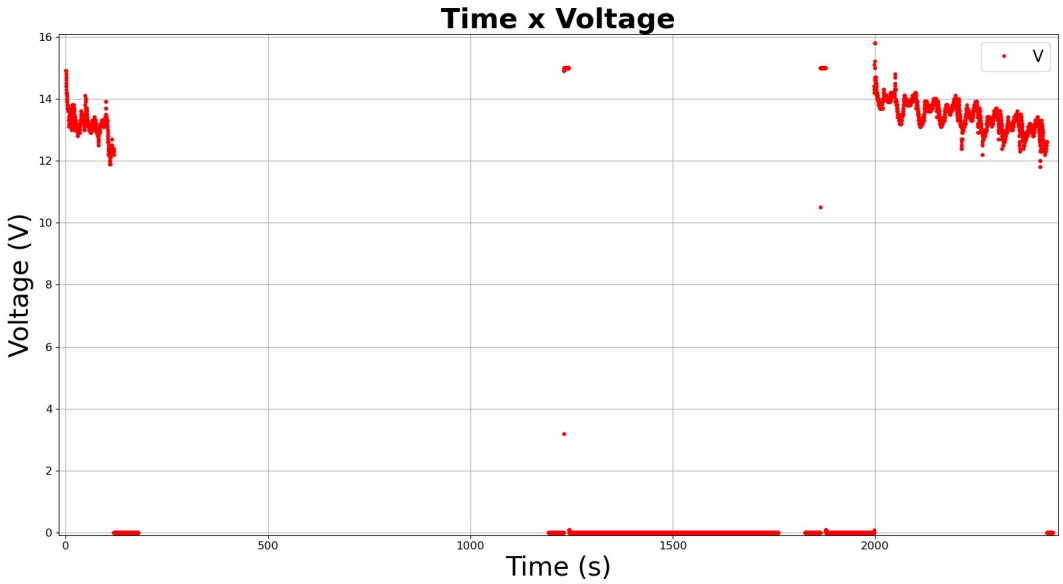
Another aspect to be observed in the deposition graphs in Figure 19 is the reduction of the number of sudden drops in voltage when compared to the first experiment, indicating the mitigation of major defects such as the stubbing observed in the previous deposited part.

The better overall finishing with fewer superficial defects (minor necking and the expected ladder effect of lateral overbuilding) when compared to the first experiment can be inferred from the pictures of the real deposited part in Figure 20.

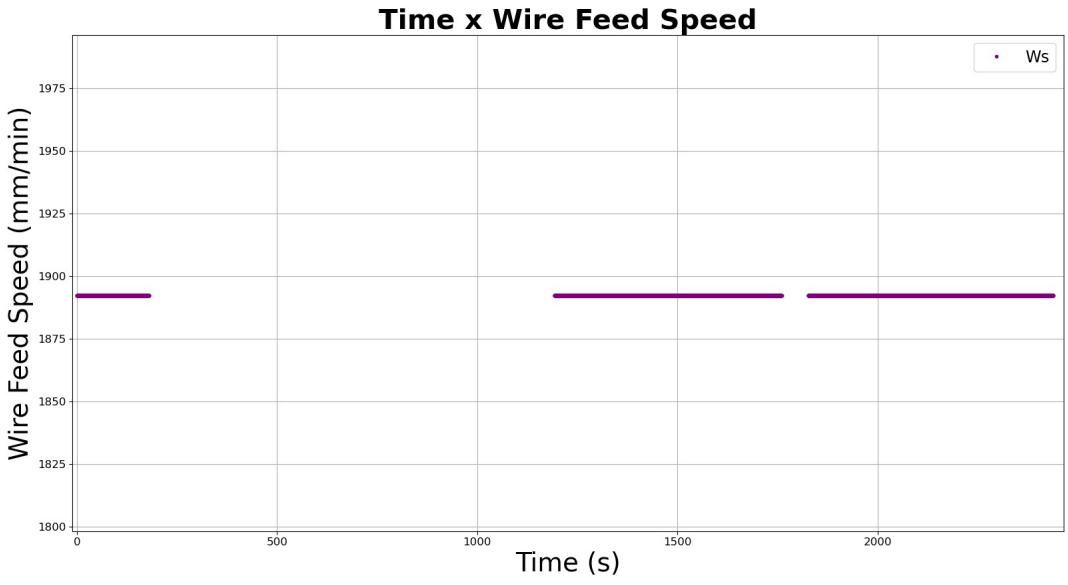
The second successful experiment resulted in a part of lower height (12.4 mm) with a lower number of layers and consequently passes when compared to the first part, as presented in Table 7.



(a)



(b)



(c)

Figure 19 – (a) Current graph of the second deposited part using the first step-by-step strategy, (b) Voltage graph, and (c) Wire feed speed graph (constant throughout the deposition).

Figure 20 – Pictures taken from the deposited part of the second successful experiment.

Table 7 – Deposition dimensional parameters acquired from the second deposited part.

Dimensional Parameters	
Mean height (mm)	12.4
Number of layers	12
Mean layer height (mm)	1.033
Mean thickness (mm)	6.5
Programmed deposition diameter (mm)	80
Passes per layer	1

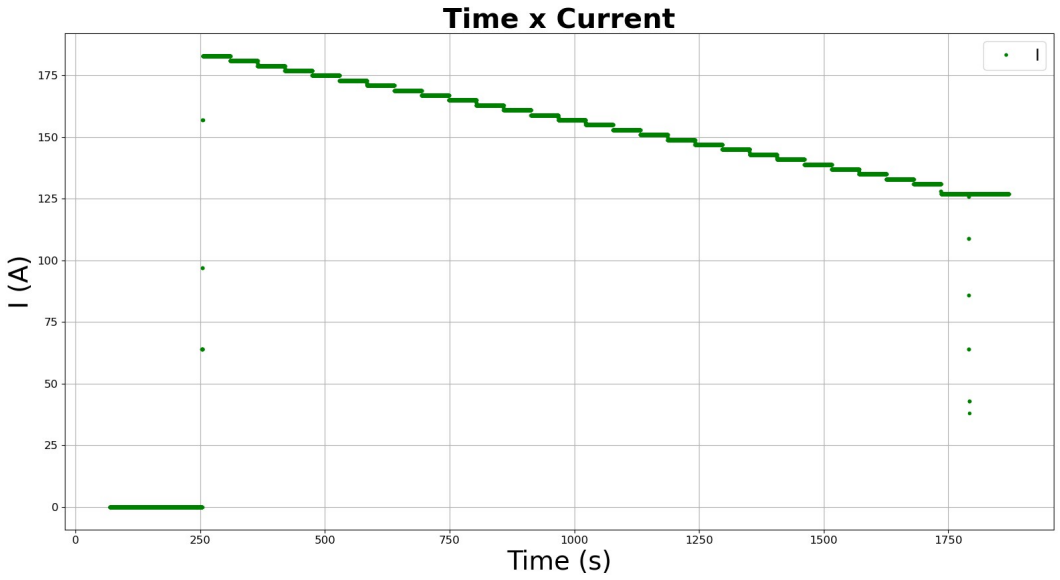
For the third experiment, a new strategy was applied, with a lower cooling time applied during the transitions from layer to layer, without a long period of time between halves deposited, with only interruptions being the time of transition from layer to layer, raising the TCP for a new Z position for a new layer deposition.

The parameters used for the third successful experiment are shown in Table 8, while their variation over the deposition time is presented in the graphs in Figure 21.

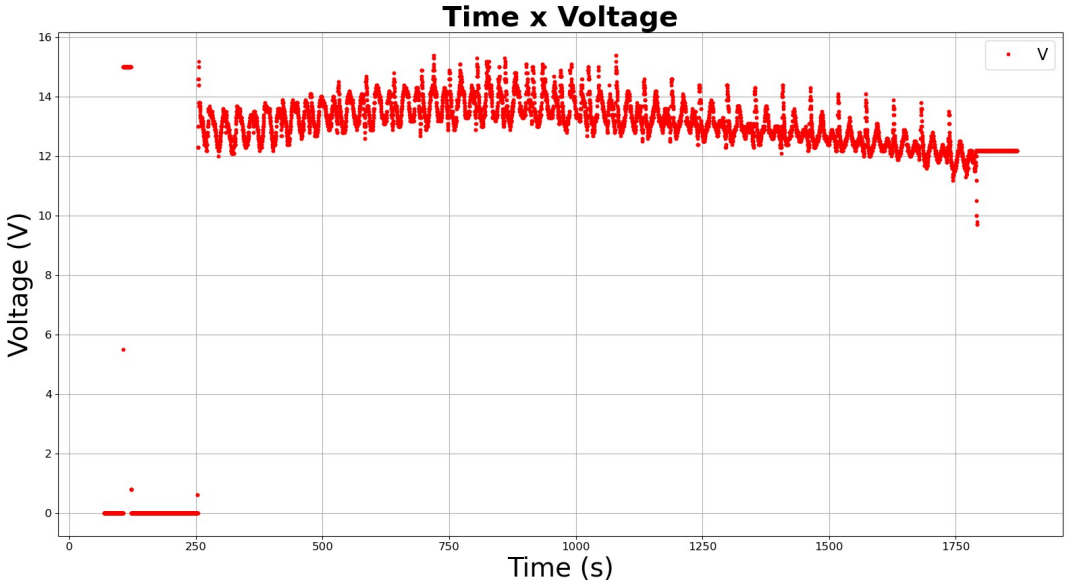
Table 8 – GTAW parameters used in the third experiment using the second step-by-step strategy.

GTAW Deposition Process Setup Overview (Third experiment, second step-by-step strategy)	
Arc Current (A)	183 → 127 (steps of -2 A)
Arc Voltage (V)	Approx. 11.14 – 15.42
Wire Feed Speed (mm/min)	1893 (constant)
Travel Speed (mm/s)	7 (constant)
Trajectory strategy	Step-by-step approach without cooling time

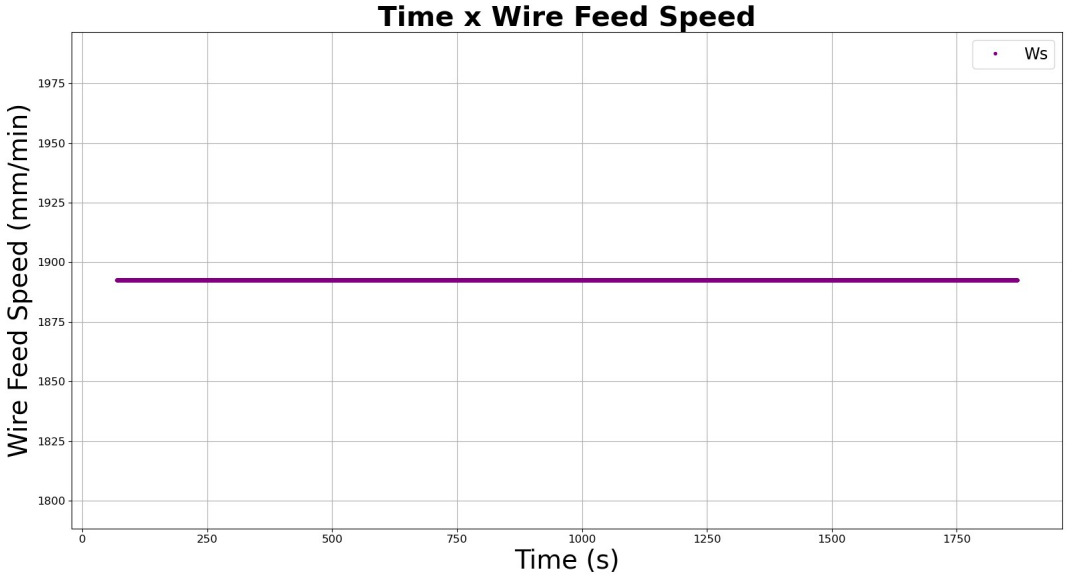
The improvement in surface quality stems from parameter selection refinement (by increasing the travel speed as in the previous experiment), which prevents previous wire sticking defects while allowing a higher cooling rate, effectively mitigating the detrimental effects of excessive heat accumulation.



(a)



(b)



(c)

Figure 21 – (a) Current graph of deposited part using the second step-by-step strategy, (b) Voltage graph, and (c) Wire feed speed graph (constant throughout the deposition).

Throughout the third deposition, the pre-programmed *Z*-axis torch increment, calibrated from previous experiments, introduced deviations in the expected arc gap, leading the metal transfer mode to fluctuate depending on the wire's entry position into the arc. Initially, a continuous bridge transfer mode was observed, whereas in the subsequent layers the transfer became intermittent, a shift associated with reduced layer height and higher molten pool temperature. As the current decreased, both the distance between the deposited layer and the tungsten electrode and the metal transfer mode were restored. This behavior is clearly reflected in the arc voltage versus deposition time shown in Figure 21b.

In addition, the metal transfer mode in some spots of the first three layers was observed to be partially in the form of droplets, a significant feature that impacted the surface roughness of the final part. Those factors highlight the interaction between current, workpiece geometry and metal transfer mode as determining elements for the final deposition quality.

The modifications to the strategy adopted presented better results in overall superficial finishing and cord defects. Also, a crucial factor influencing the result was the difference in diameter (increased from 8 cm to 12 cm, as presented in Table 9), which allowed a longer cooling time for the workpiece before the welding torch passed through again.

Table 9 – Deposition dimensional measurements for the third experiment (second step-by-step strategy).

Dimensional Measurements	
Mean effective height (mm)	30.9
Number of layers	29
Mean layer height (mm)	1.065
Mean total thickness (mm)	5.4
Programmed deposition diameter (mm)	120
Passes per layer	1

The fair quality superficial roughness with noted defects (mainly necking and overbuilding caused by the accumulating material when ascending from one layer *Z* position to another) can be inferred from the pictures taken of the finished part in Figure 22.

Compared to the first deposition, the second and third successful experiments achieved better results in terms of general surface finish and bead defect reduction, as illustrated in Figure 22.

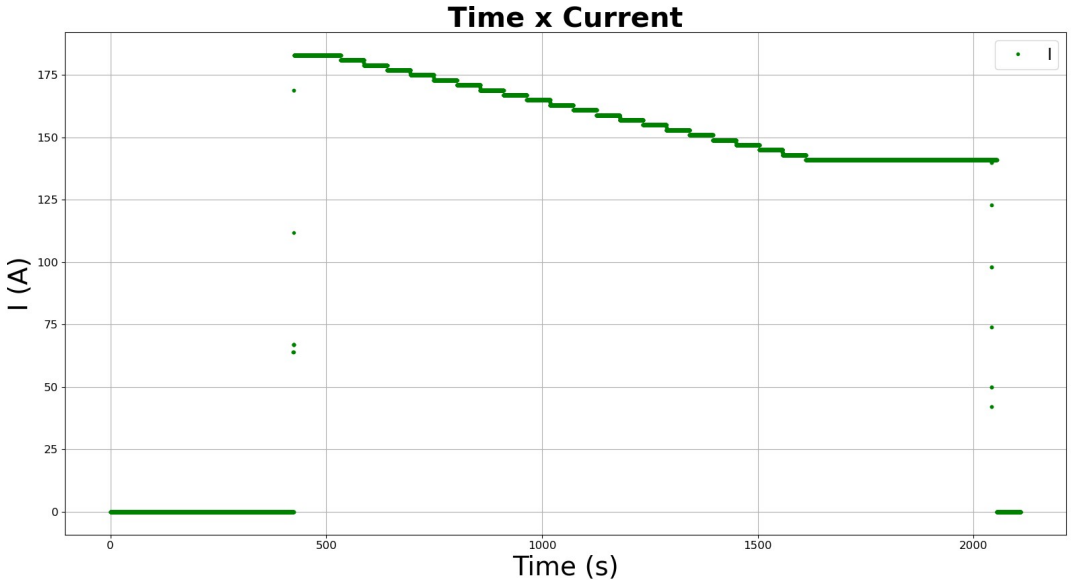
Figure 22 – Pictures taken from the deposited part resulting from the third successful experiment (the outer metal part, the inner one is the part from the second experiment).

A continuous helical deposition strategy was subsequently evaluated. The specific parameters selected for this experimental setup are summarized in Table 10.

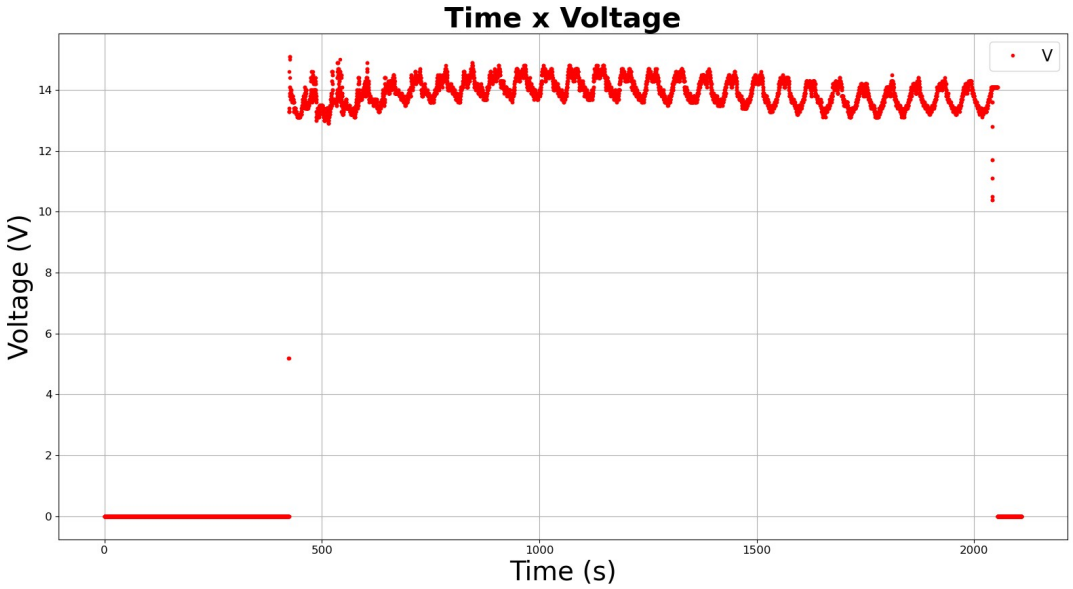
Table 10 – GTAW parameters used in the fourth experiment using the helical Z position ascension strategy.

GTAW Deposition Process Setup Overview (Fourth experiment, helical strategy)	
Arc Current (A)	183 → 141 (steps of -2 A)
Arc Voltage (V)	Approx. 12.84 – 15.14
Wire Feed Speed (mm/min)	1893 (constant)
Travel Speed (mm/s)	7 (constant)
Trajectory strategy	Helical approach

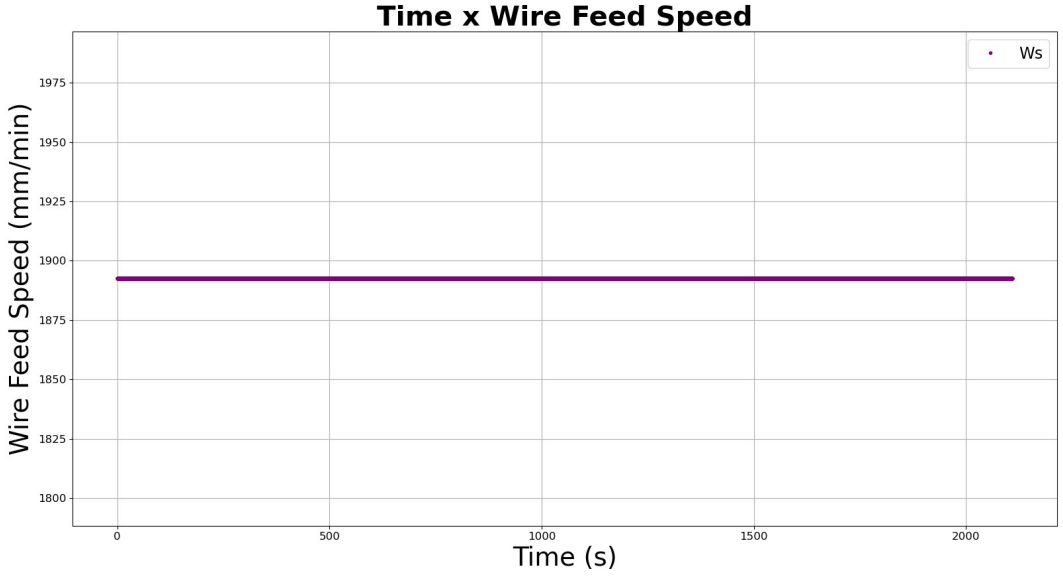
Having also a lower steep downward variation of the current when compared to the beginning of the first experiment, the variations in the parameters throughout the deposition can be seen in the graphs of the grid in Figure 23.



(a)



(b)



(c)

Figure 23 – (a) Current graph of deposited part using the helical strategy, (b) Voltage graph, and (c) Wire feed speed graph (constant throughout the deposition).

Overall, the deposition remained stable, characterized by a bridge transfer mode, which can be inferred from the periodicity observed in Figure 23 (b).

Finally, the overall good quality of superficial roughness with little noted defects (some overbuilding at the top layer and minor necking) can be inferred from the pictures taken of the finished part in Figure 24.

Figure 24 – Pictures taken from the deposited part of the helical strategy experiment.

The dimensional parameters obtained from the helical experiment's finished part can be observed in Table 11.

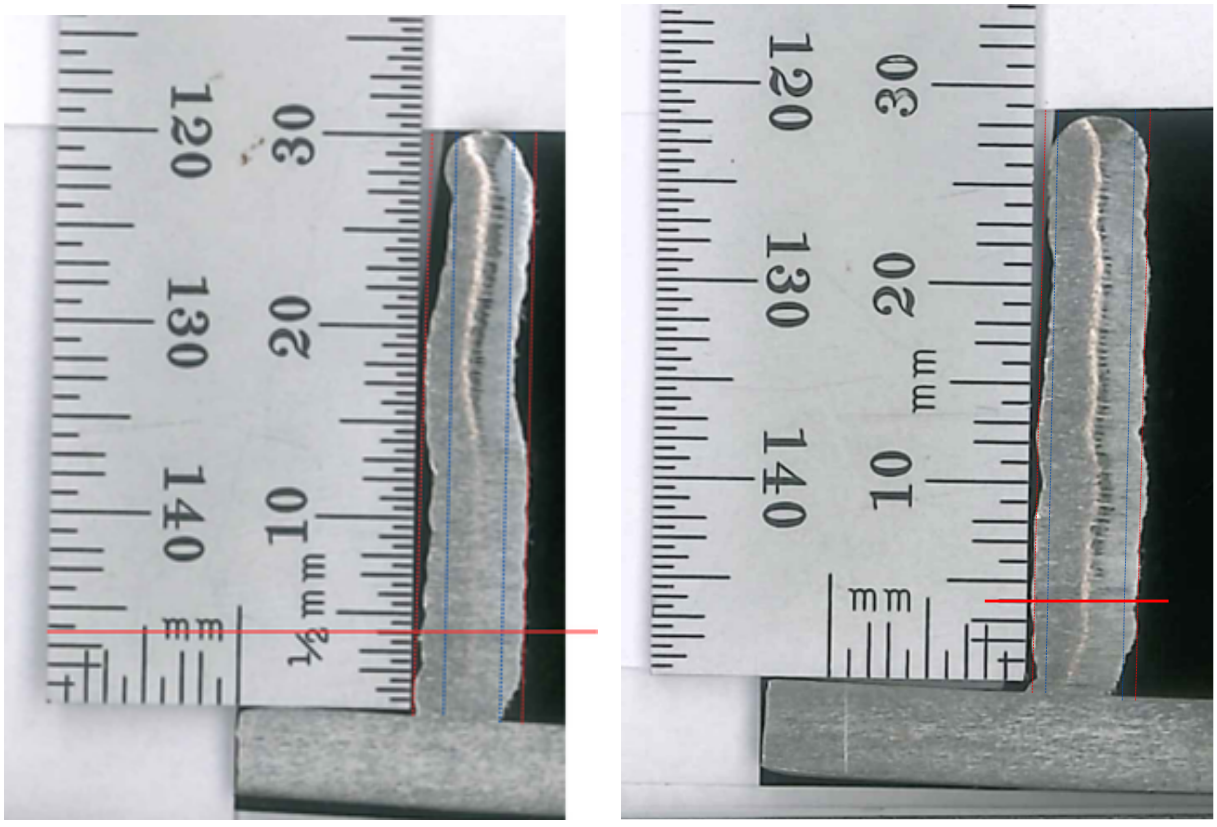
Table 11 – Deposition dimensional measurements for the fourth experiment (helical strategy).

Dimensional Parameters	
Mean height (mm)	29.4
Number of layers	29
Mean layer height (mm)	1.014
Mean total thickness (mm)	5.2
Programmed deposition diameter (mm)	120
Passes per layer	1

4.2.2 Trajectory definition and metal deposition strategies

During extensive deposition tests conducted at the robotic manufacturing cell within the GRACO laboratory, the helical deposition method emerged as the superior strategy along the Z-axis. This finding aligns with recent literature, which often highlights the advantages of continuous deposition paths, such as helical or spiral trajectories, over unidirectional (e.g., zigzag) methods in WAAM (VENTURINI et al., 2018; SARMA; KAPIL; JOSHI, 2025). The helical approach was further optimized by dynamically adjusting the arc current, progressively reducing its value as the wall height increased during the deposition process.

Figure 25 presents the cross-sections of the third and fourth parts after the cutting process. For clarity, the red dashed lines demarcate the total wall width, while the blue dashed lines indicate the effective wall width. The solid red line marks the boundary of the fourth deposited layer. Measurements commenced at the fourth layer and proceeded at 1 mm intervals along the entire part height. The results for the third specimen showed an effective wall width of 2.9 mm and a total wall width of 5.4 mm. Alternatively, the fourth specimen measured 3.9 mm and 5.3 mm, respectively. The mean wall thickness was calculated as 5.40 $\pm$ 0.07 mm for the third specimen and 5.20 $\pm$ 0.05 mm for the fourth. Furthermore, data from the deposition tests revealed a significant difference in surface quality. The fourth specimen, fabricated using a helical deposition strategy, exhibited approximately 43% lower surface waviness, 1.2 mm vs. 0.68 mm (which is given by the amplitude of total and effective width), compared to the third part, despite both being produced with identical parameters up to the 22nd layer.



(a) (b)

Figure 25 – (**a**) Cross-sectional third part, produced by stepwise strategy, and (**b**) cross-sectional fourth part, produced by helical strategy. Blue and red lines in the pictures highlight the layer's effective and total wall width, respectively, used for measurements and comparisons between the experiments.

According to Wang et al. (WANG, H.; KOVACEVIC, 2001), heat continuously accumulates in each deposited layer until a balance is reached between the heat input and dissipation to the previous layers and the environment. Since this heat buildup influences the deposited morphologies and structures, it becomes necessary to adjust the process pa-

rameters according to the height. Therefore, a gradual reduction in the current was applied as a strategy to keep the melt pool width within a narrow range, increasing process stability. In this study, the current was the only parameter varied and in the GTAW process, it acts independently of the wire feed speed, thus being the main factor controlling the deposited layer width. The width increase caused by the rise in interpass temperature was counterbalanced by the reduction resulting from the decrease in current. Since the cross-sectional area of the layer remained constant (the cross-sectional area of the layer is determined by the relationship between the wire feed speed and the travel speed, since these parameters were not varied, it can be concluded that the area remained unchanged), any change in width implied a corresponding adjustment in the layer height, which thus remained within a stable and uniform range.

However, since the proposed QA analysis methods need to be tested and validated in various scenarios, other trajectory strategies are also contemplated through the validation process, such as layer deposition in sessions with pauses for cooling down after a certain number of layers deposited, and non-continuous step-by-step ascent from one layer to another, but without pauses. The first method allows the general thermal gradient of the system (metal support build plate + deposited part) to settle before another session of layer deposition, whilst the second one briefly allows the molten pool to begin solidifying before starting a new layer.

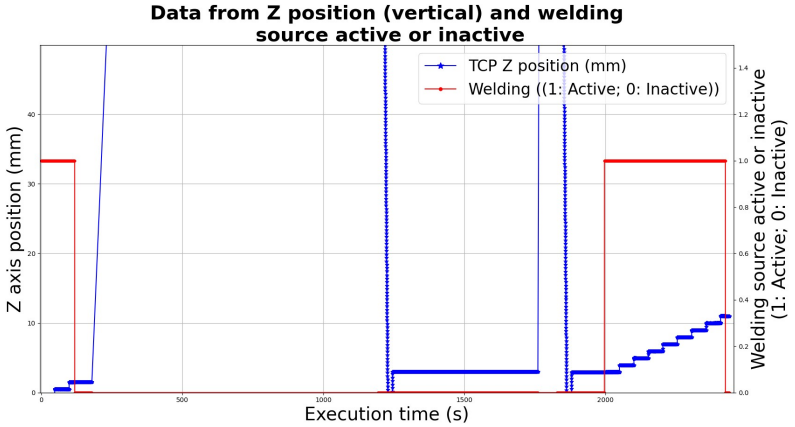
A comparative analysis of the deposition strategies revealed that the stepwise ascending method, both with and without interpass cooling, resulted in inconsistent bead geometry and surface irregularities in the as-built condition, unlike the more uniform outcomes of the continuous ascending helical approach. Also, the step-by-step trajectory definition strategy inherently results in a "ladder" aspect of layering thickness at the point of TCP ascending, which is due to material being fed into the molten pool while the positioning table is not turning, accumulating material into a single point of the part's wall.

The variation of the *Z* position throughout the deposition process of the first, second, and third (helical) strategies and experiments can be observed in a grid of graphs obtained via the Cloud Firestore database provided in Figure 26.

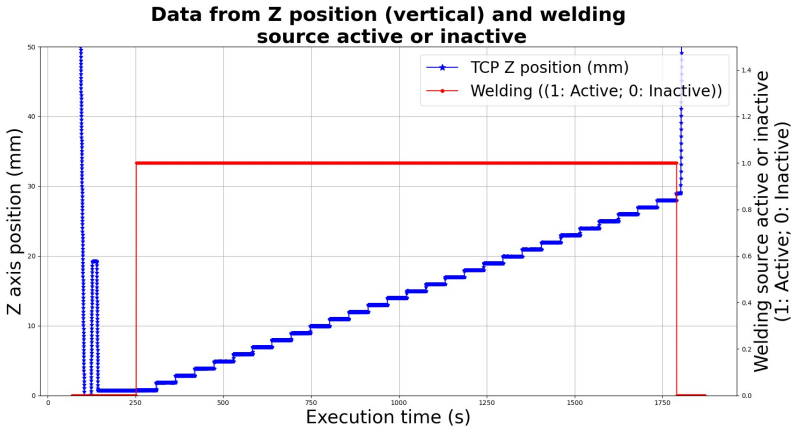
The trajectory definition for material deposition has a direct impact on the superficial layering quality in the deposited part's walls. This is due to the material accumulation on the step-based ascension in *Z* position in the first and second strategies, characterizing an overbuilding defect in the wall. The difference is clearly visible in the provided graphs of *Z* position throughout the deposition time of the defined strategies, which also reinforces the importance of the Digital Twin in supporting decision-making in production cycles and refining parameters for the deposition of metal parts in the manufacturing cell.

Another thing to consider is the mentioned cooling time between two sessions of layer deposition, which is easily noticeable in the *Z* position graph (a) in Figure 26. The rough

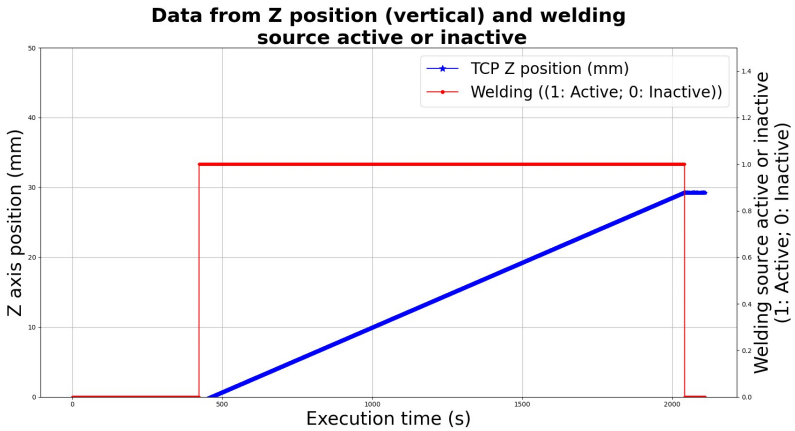
(a)



(b)



(c)



(d)

Figure 26 – (a) Z position graph of deposited part using the two sessions cooling strategy, (b) Z position graph of the second experiment, (c) Z position graph of the second strategy part, and (d) Z position graph of the helical strategy deposited part.

finishing and the easily visible discontinuous cord from the superior half of the deposited part are possibly due to the great thermal inertia and abrupt thermal gradient caused by the difference of temperature of the different halves. Better continuous cooling strategies (WANG, H.; CAO, et al., 2021; REISGEN et al., 2020) might be considered in the future to address those problems and prevent the observed defects from happening.

Both the helical method and the step-by-step methods presented a beginning of balling and necking effects in the first two layers. However, as the energy input/output and thermal gradient were stabilized as layers were deposited, those defects faded, and the cord became continuous for most of the length of the deposited parts' walls.

The comparison between the overall quality obtained on parts deposited with those deposition methods can be inferred by the grid of images in Figure 27. Those images were generated from the 3D models scanned via PolyCam software, using the photogrammetry method of 3D surface generation.

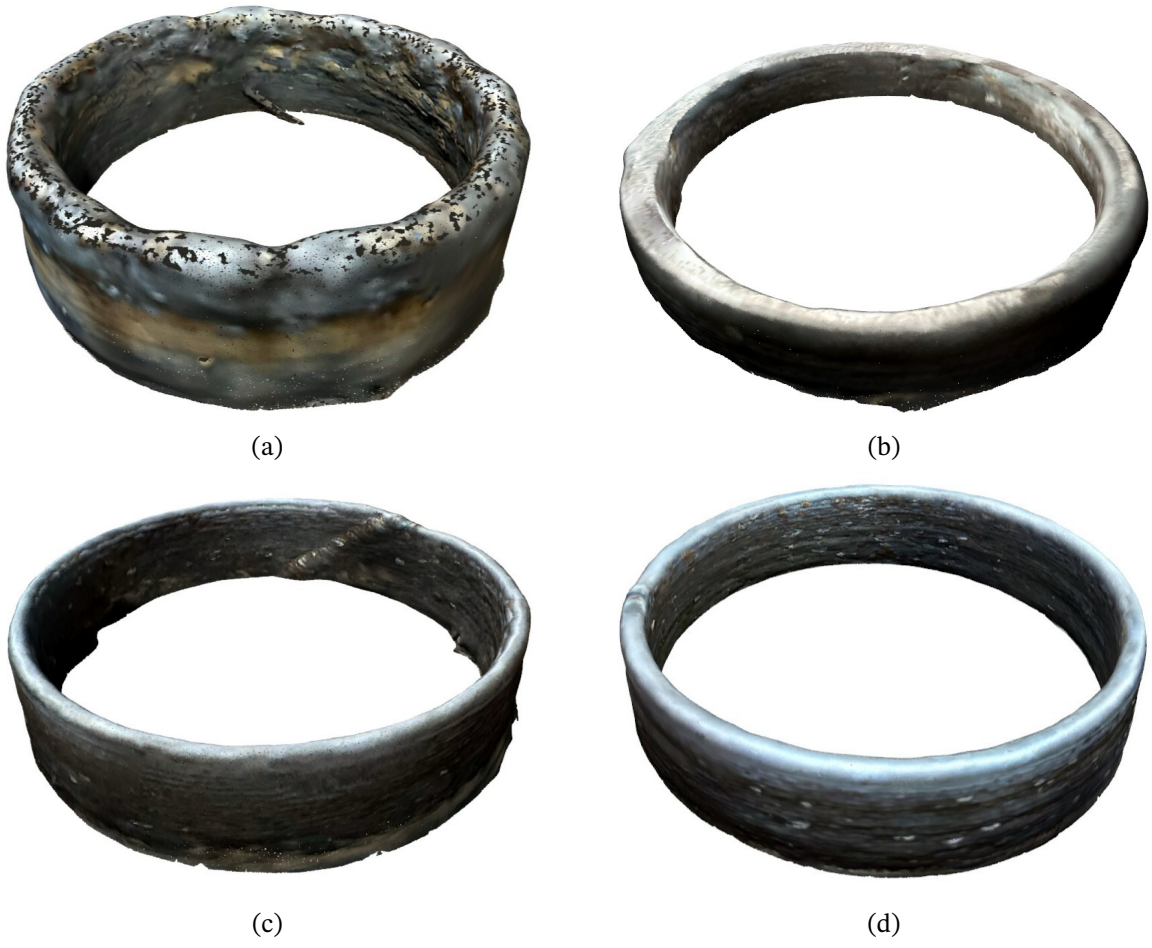


Figure 27 – (a) Scanned 3D model of the deposited part from the first experiment, (b) Scanned 3D model of the second experiment's part, (c) Scanned 3D model of the third experiment's part, and (d) Scanned 3D model of the helical strategy experiment.

As perceptible in the grid of images, the step-by-step deposition presents a rougher finish when compared to the helical one. Also, the discontinuous height ascent produces

excesses of material in specific parts when the tip of the tool is going up by melting more steel in that little time period, when the linear horizontal deposition speed is slowed down for the layer switching process.

The differences in the layering processes of the 3 conducted experiments are also perceivable in the form of the captured point clouds from the Digital Twin pipeline. The X and Y positions are corrected via the stored θ angle of the positioning table to reconstruct the observed point cloud during deposition, using also the Z position as the measurement of the height of the deposited layer. The grid of images of those point clouds is present in Figure 28.

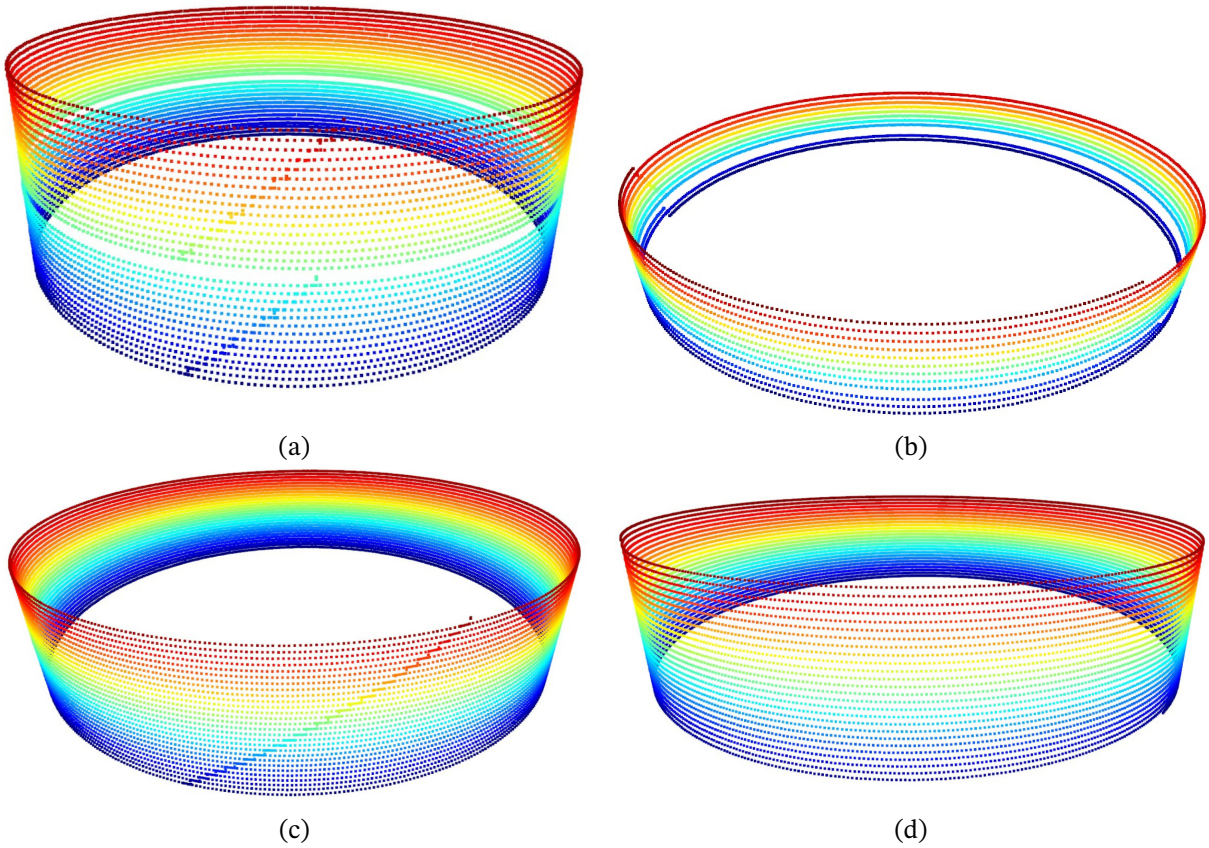


Figure 28 – (**a**) 3D point cloud of the deposited part from the first experiment, (**b**)3D point cloud of the second experiment's part, and (**c**) 3D point cloud of the third experiment's part (**d**) 3D point cloud of the helical strategy experiment. The gradient of colors is used for better contrast and visualization, as well as to highlight the change in height throughout the Z-axis layering.

In the first experiment in Figure 28, there are two separate point clouds for each deposition section, with the step-by-step layering being clearly visible in each one of them. In the second experiment, there is only a single deposition section, with a step-by-step layering clearly visible through the wall of the captured point cloud, whereas in the fourth experiment, corresponding to the helical strategy, there is a smooth, continuous ascent of the TCP for the layering, resulting in a continuous deposition without accumulated material and captured

points in the form of a “ladder” that was observed in the previous experiments.

4.3 Data ETL pipeline and modules

In this section, all modules that compose the expected pipeline framework already presented in Figure 14 are explained in their own following subsections.

The ETL pipeline is a sub-entity of the Digital Twin Entity in the proposed Digital Twin architecture framework presented in the methodology and illustrated in Figure 6.

4.3.1 Data extraction

The first part of the pipeline comprises the extraction of data to be transformed and loaded into the local repositories and the cloud infrastructure for the history of variables and analysis results’ documents. The data extraction is a crucial part for the correct functioning of the entire data pipeline; if any errors occur during parts scanning or mesh generation, those errors will be propagated through the entirety of the implemented data framework.

Throughout the following sub-subsections, the utilized methods for data extraction and 3D meshes generation methods for comparisons and CAD/CAM deposition trajectory definition will be summarized.

4.3.1.1 3D CAD files for TCP trajectory and deposition parameters definition

The 3D models used as reference for the deposition of the parts were created on Autodesk Fusion version 2605.1.39. Beyond being a reference for trajectory definition for hollow parts’ deposition, it serves as a basis for comparison with generated 3D models and point clouds from both Digital Twin-acquired positional data and from the 3D scanning via photogrammetry.

Defined measurements for the 3D CAD files as a reference for the perfect case scenario planned for deposition are presented in Table 12.

Table 12 – Reference dimensions for the ideal deposition 3D CAD models used in the experiments.

Reference 3D CAD Models		
Experiment	**Diameter (mm)** *(center-to-center)*	**Height (mm)**
1	80	30
2	80	12
3	120	30
4	120	30

Acquired positional data from the generated .STL files of the 3D CAD meshes are further used to measure the differences when aligned with the DT-based positional data and 3D photogrammetry scanned one, in subsection 4.5.5.

4.3.1.2 LiDAR scanning 3D point cloud acquisition

The first method for generating 3D point clouds for dimensional analysis considered was to use a LiDAR sensor. However, due to limitations on the scan range and minimal measurement resolution of the LiDAR present in the iPhone 16 Pro Max, generated 3D models of real manufactured parts presented insufficient geometric fidelity to be analyzed through the proposed ETL pipeline. A LiDAR would work in the context of macro geometrical features evaluation on larger parts with a higher estimated tolerance in manufacturing, since most LiDARs have an estimated accuracy of around 5 mm in measurement.

In the future, the acquisition of a better handheld infrared 3D scanner might enable the generation of 3D meshes of manufactured parts using the Infrared Structured Light detection technique (GENG, 2011), such as the Creality Otter (SHENZHEN CREALITY 3D TECHNOLOGY CO., LTD., 2024).

4.3.1.3 Photogrammetry scanning 3D point cloud acquisition

The chosen method of 3D scanning, utilizing the iPhone 16 Pro Max's camera, is the photogrammetry-based scan of the manufactured parts. Having a theoretical spatial resolution of approximately 0.3 mm in photo-based scans, the PolyCam (POLYCAM, 2025) software app version 4.0.11 photogrammetry solution was the chosen image fetch and processing mechanism. While preliminary manual verification estimated a global dimensional scale drift (absolute error in the cylinder's height/diameter) of 0–3 mm due to the markerless scanning approach, this global deviation does not affect the downstream analysis, as the proposed PointNet classification relies on unit-sphere-normalized segments.

For the GTAW-based deposited samples of hollow parts, a sampling of $160 - 180$ pictures per deposited part was taken (RAW mode for maximum resolution, macro mode, $20 - 30\ cm$ of distance from the deposited part) for generating the 3D meshes in the PolyCam iPhone app. This dataset size, which is approximately 3 to 4 times the standard recommendation for this geometry (POLYCAM, 2025), ensures high overlap ($> 80\%$) to mitigate occlusion and filter transient specular reflections during 3D mesh generation. The complete dataset, including source images and photogrammetry parameters (distance from referential, camera orientation, blur score), is available in (CABRAL, J. V. A., 2025b).

In Figure 29, it is possible to compare the picture taken from a deposited part with its respective 3D model as seen by the photogrammetry solution from PolyCam.

Acquired 3D meshes from the photogrammetry method applied to hollow metal

(a) (b)

Figure 29 – (a) Real picture from the third experiment's part (the outer tubular wall), (b) Scanned 3D model of the third experiment's part.

parts are used for validation of the analytical AI-based models described in the section 4.4.

4.3.1.4 Digital Twin-based positional data 3D point cloud acquisition

Since the original Digital Twin data pipeline culminates in storing near real-time acquired data, if the robotic cell's sensors are correctly calibrated and the trajectory is well-defined based on the reference CAD models for deposition, the Digital Twin-acquired positional data is then a good source of comparison for geometrical features of the deposited parts.

Among the variables acquired from the robotic AM cell, positional data, welding source parameters, linear TCP speed, positioning table rotation angle, and TCP orientation can be used to precisely reconstruct the deposition trajectory. All this collected information can then be used to support both AI-based and heuristic analytical models applied to the data related to each deposition experiment.

Specifically for the reconstruction of point clouds using positional data, the positioning table's rotating angle was also used to correct the X and Y positions by applying a geometric transformation to the raw coordinates. It was corrected by iterating through each data point to calculate its radial distance from the origin, thus deriving the corrected X and Y values using cosine and sine based on the converted radian angle, ultimately returning the adjusted trajectory X and Y positions.

All the 3D models of deposited hollow parts described in the data extraction subsection must be corrected and prepared before being analyzed by the proposed AI models. The transformation processes, corrections and overall data structuring of the ETL pipeline are detailed in the following subsection.

4.3.2 Data transformation

Comprising the main part of the ETL pipeline framework, the data transformation part is composed of many modules of the flowchart and is divided into many programs that must be executed in a specific sequence in order to properly handle and prepare the data for the analytical algorithms and AI models.

Throughout the following subsections, the modules related to the transformation part of the ETL pipeline in Figure 14 are explained in detail.

4.3.2.1 SPSR density correction and superficial normals generation

Following the referential correction and noise filtering of the 3D meshes, a final normalization step is implemented to guarantee the homogeneity of the 3D point clouds for subsequent analysis. The Screened Poisson Surface Reconstruction (SPSR) (KAZHDAN; BOLITHO; HOPPE, 2006) is applied to rectify inhomogeneities in point distribution, effectively balancing regions of varying density caused by the uneven scanning of the manufactured parts.

This method is applied to correct uneven scanning of manufactured 3D parts, reducing the occurrence of high densities in some point graph neighborhoods while leaving others with almost no scanned points.

As a convention to maintain consistency across future analyses, the SPSR application was standardized to 98304 output points per point cloud. This specific number was chosen because it is a multiple of 2048 points (see 4.4.1) and because it maintains a good level of detail across the entire surface, as confirmed by experimental validation on 3D scans. The SPSR is applied based on Open3D's points sampling SPSR method (KAZHDAN; BOLITHO; HOPPE, 2006).

SPSR was selected as the primary reconstruction method due to its robustness and efficiency. While alternative techniques such as the Ball Pivoting Algorithm (BPA) were evaluated, they were ultimately rejected due to their high computational cost, which proved prohibitive for the rapid processing required in the proposed pipeline. Conversely, SPSR offers a computationally efficient solution that acts as a low-pass filter, smoothing high-frequency photogrammetry noise while preserving the meso-scale geometric deviations necessary for quality classification. The target output density was standardized to 49,152 points per model. This specific resolution aligns with the neural network's architecture, facilitating the precise extraction of 24 segments of 2,048 points. Empirical testing indicated that doubling this resolution to 98,304 points drastically increased the computational load for mesh processing and training without yielding a statistically significant improvement in classification accuracy.

After the density correction of the point clouds, the immediate vertex and mesh

normals are computed using the corresponding methods on the open3D library. Then, new triangle meshes are generated based on the corrected points and calculated normals. The storage of such 3D meshes is further commented on at sub-subsection 4.3.3.1.

In the following sub-subsections, the last steps of data transformation for load and further analysis are commented on, including the segmentation of the transformed point clouds and vectorization/compression for being compatible with AI models' input parameterization.

4.3.2.2 Point cloud segmentation

A crucial part of the data pre-processing done before inputting it to the classification AI models is the segmentation of acquired data. This part of the pipeline is directly responsible for how the analytical models "see" the manufactured parts, thus being vital for the computer vision part of the dataflow. In fact, any modifications in this process, as small as they might be, have profound impacts on the weights of feature inputs and activation of neural nodes, contributing even further to the unpredictability of the classification models' learning process, training results, and validation ones.

The first approach that was experimented with, having in mind the simulated 3D models as input for the DNN model (which will be further explained in subsection 4.4.2), is the algorithm that executes a vertical decomposition of the point cloud (available in A.1.1), linearly dividing the object's geometry along its vertical axis. By calculating the total height of the scanned part, the system slices the continuous mesh into 24 distinct, equidistant horizontal layers. This layering strategy effectively mimics the physical layer-by-layer deposition nature of the WAAM process, isolating the geometry of specific deposition levels into discrete blocks.

The primary implication of this segmentation is the standardization of variable metrology data into the fixed-size input format required by the PointNet architecture. Regardless of the original density of the scan, each of the 24 resulting segments is resampled to contain exactly 2048 points, ensuring a consistent tensor shape for the deep learning model. By treating each slice as an independent unit, the approach effectively removes global positional bias, forcing the classifier to evaluate the topological roughness and surface irregularities of that specific segment in isolation.

After some iterations of experimental training and testing of AI models, it was decided to use a radial split of the 3D models along the Z axis for the real parts' segments approach for training and evaluation (steps which will be further explained in subsection 4.4.3). This was made to uniformize the segments of the point clouds, making it easier to normalize input ranges and reducing distortions in the edges of the segments. Another advantage of this method of segmentation is the possibility of evaluating continuity among deposited layers. By capturing the full height of the part within each 15-degree angular segment, the

model can analyze the vertical alignment and bead stacking consistency, which enables the detection of defects that span across multiple layers, a feature that would be lost with a simple horizontal layer-wise segmentation.

The discussed method of segmentation is illustrated in Figure 30.

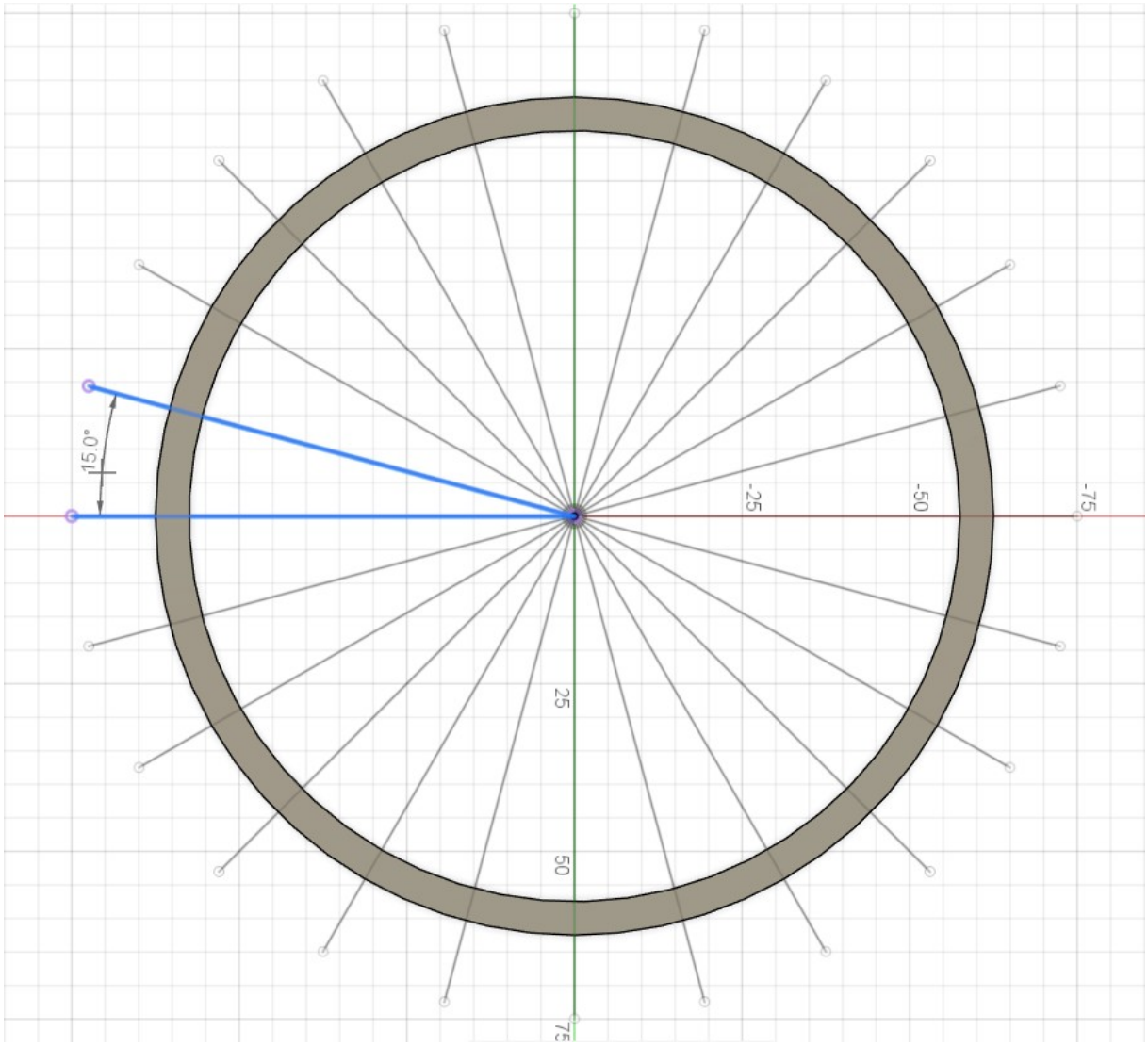


Figure 30 – Azimuthal slicing of a CAD reference file for exemplifying the segmentation of a real deposited part as part of the transformation process of the ETL pipeline.

Since the input segments are made of 2048 points in the point cloud each, and the SPSR sampling of the 3D point clouds is set at 49152 points for each STL input, the radial angle of slicing throughout the Z axis is set at 15º. Since the initial orientation is constrained by the radial segmentation logic, the alignment is robust, ensuring that the neural network processes purely topological features rather than positional artifacts. Other segmentation values such as 7.5º and 30º were tested, but the model wasn't able to properly generalize across all the 3 classes on those approaches, especially the 7.5º azimuthal slicing approach, which more than doubled training length because of the large dataset, increasing from 4 hours of the 15º approach to more than 10 hours of training routine.

4.3.2.3 ICP correction

The `apply_icp` function in the segmentation module A.1.2 applies the Iterative Closest Point (ICP) algorithm to align two segments of a point cloud. After initializing the point clouds, the function sets an ICP threshold of 0.02, which specifies the maximum correspondence distance. A 4×4 identity matrix, `trans_init`, is used as the starting guess for the transformation. The ICP registration step is performed using `registration_icp`, which aligns the source cloud with the target cloud via point-to-point estimation. The result includes a transformation matrix that minimizes the distance between corresponding points across the two clouds.

Finally, the calculated transformation is applied to the source cloud through `.transform(transformation)`, and the transformed coordinates are copied back into the original NumPy array. The function returns the updated `source` array, now spatially aligned to `target`. Each input CSV file contains 24 stacked horizontal segments along the positive Z axis, so this function can be applied iteratively to align all layers with respect to a base segment, reconstructing or analyzing the 3D structure progressively.

4.3.2.4 Data vectorization and compression

After the segmentation process, the data is not yet ready to be fed into the DNN model's input. Being the last step before the application of the PointNet model, the data vectorization and compression convert the input segments into structured tensors containing positional coordinates, surface normals, and class labels.

To prepare the raw segments for the PointNet architecture, the pipeline first standardizes the input density to ensure a consistent tensor shape. The algorithm reads the segmented CSV files and enforces a fixed point count ($N = 2048$) to match the neural network's input requirements. If a segment exceeds this limit, iterative voxel downsampling is applied to reduce density while preserving geometric structure; conversely, if the segment is too sparse, random point duplication is performed to pad the dataset. This step ensures that all input vectors have identical dimensions regardless of the original scan resolution.

Following density standardization, the data undergoes rigorous geometric normalization to ensure training stability. The point cloud is centered at the origin by subtracting the centroid coordinates from every point. Subsequently, the entire segment is scaled to fit within a unit sphere by dividing the coordinates by the maximum distance from the origin. This transforms the data from absolute spatial coordinates to relative geometric features, making the analysis invariant to the global scale of the part. Simultaneously, surface normal vectors are normalized to unit length to ensure consistent orientation data.

However, identifying the exact effects of this unit sphere normalization on the distortion of superficial defects is a complex task, especially when considering feature identification

and its effects on neural network training metrics. With this in mind, the unit normalization sphere might deform superficial defects according to the input segment format, potentially also impacting feature identification by the adapted DNN model.

Finally, the processed data—comprising the unit-sphere normalized points, the associated class labels, and the normalized surface normals—is serialized into Hierarchical Data Format 5 (HDF5) files. This binary format is optimized for high-throughput I/O operations, serving as the direct feed for the Deep Neural Network training and validation phases.

4.3.2.5 Quality assurance analysis model

Finally, the last data transformation process relates to the analysis model used to classify manufactured parts and compare the 3D models. Those QA assessment sessions use the transformed, segmented, and vectorized 3D models' data as inputs, having collections of classes and features comparison results as outputs.

Simulated and real parts' photogrammetry-based 3D point clouds are processed through the entire transformation pipeline to prepare them for AI model training. The analytical AI models are based on Deep Neural Networks (DNN) and machine learning models, which will be further discussed in section 4.4. In the following data load subsection, the storage and general structure of the documents that compose the output of said analytical models will be discussed.

4.3.3 Data load

This subsection is aimed at explaining the load of transformed data through the data pipeline and files that are generated regarding the extracted and simulated 3D models of manufactured parts for both the training of AI models and the validation of the analysis process.

3D models that have its origin referential, density and are transformed into triangle .STL meshes to be stored in local collections, explained further in 4.3.3.1.

Following the referential correction, segmentation, and vectorization of the point cloud data, the framework is designed to extend beyond immediate classification to include data persistence. While the current study utilized the acquired datasets primarily for the training and validation of the Deep Neural Network, future operational iterations of the pipeline will automatically archive the inference results in the Google Cloud Firestore Database. This integration aims to create a permanent history of quality analyses, the structure of which is detailed in Subsection 4.3.3.2.

4.3.3.1 Local repository of 3D point cloud models

The 3D point cloud models, which were obtained via photogrammetry, DT-based positional data or CAD files and had their superficial point distribution density corrected, with noise filtering and corrected referential are then stored in a local repository of collections of files based on training, testing, validation labels and with their origin also labeled (simulated, generated from positional data, validation scanning of real parts).

Those files are composed of large triangle mesh objects codified into .STL files (CHEN, Y. H.; NG; WANG, Y. Z., 1999). Having a history of generated 3D meshes helps assist decision-making throughout the process of refining deposition parameters. It also supports future analysis of comparisons among many different deposited parts through deposition sessions, being a complement to the collected and stored variables.

Also, since the manipulation of 3D meshes may vary according to the developed AI model (and possibly future) ones and proposed analyses, storing original and corrected 3D models of deposited and simulated parts is useful to extract features from such point clouds and test the impact of different meshes refining and preparing pipelines on analytical results.

4.3.3.2 Proposed collection of documents stored in Firebase Firestore

After the analysis of the 3D models obtained from CAD referential objects, photogrammetry, and positional data acquired from the Digital Twin pipeline, including passing through the whole ETL data pipeline, analysis documents are generated based on the AI model's perspective on that data.

Upon the completion of the full ETL pipeline—encompassing the processing of CAD reference objects, photogrammetry scans, and Digital Twin positional data—the system generates comprehensive analysis reports based on the AI model's inference. In the proposed production workflow, these outputs are serialized into structured documents rather than remaining transient data.

These documents, whether derived from the quality assessment of fabricated parts or from comparative metrics between "as-built", "as-monitored", and "as-planned" models, will be structured in a standard JSON format. This approach establishes a historical repository of evaluated parts for the ABB robotic GTAW AM cell, ensuring QA analysis traceability. This schema for the Google Cloud Firestore database is essential for future work, as it enables the refinement of manufacturing parameters based on the accumulated history of quality level assessments and superficial meso-scale feature variations among evaluated models.

4.4 Proposed DNN-based analyses and models training

In this section, the training of AI models on simulated data will be discussed from the perspective of specific planned QA analyses. Such analytical models should be able to support decision-making on the manufacturing process and fine-tuning of welding parameters and deposition strategies.

Starting from this contextualization, a DNN-based quality evaluation model is proposed, with its implementation details being presented in subsection 4.4.1.

Firstly, the adapted DNN model is applied to a scenario of simulated 3D models with their respective segments' classification being based on the coverage of simulated superficial defects throughout the surface, as explained in subsection 4.4.2.

Then, the same DNN model is applied to a second and more robust training pipeline, with the classification system being based on the normalized Fiedler number (WANG, C.; SAMARI; SIDDIQI, 2018)) of the surface of the acquired 3D models, with the previously presented radial segmentation (as explained in subsubsection 4.3.2.2). This second implementation will be discussed through the subsection 4.4.3.

ICP superposition of 3D models from the Digital Twin positional data with the photogrammetry-based one is used as a basis of registration/representational analysis of said models, as the convergence from the Digital Twin ones and the photogrammetry-based ones should achieve a certain threshold to be considered sufficiently equal, with the DT solution being evaluated on its efficiency of representation of the real deposited part. This analysis is further commented on in subsection 4.5.5.

Lastly, the ZEISS 3D Inspection software version 2026.2.0.1091 is used for industrial-grade alignment metrics validation on captured 3D mesh models, using 3D laser scanner meshes for validation of captured photogrammetry-based ones, along with CAD files for nominal reference.

Those implementations were trained, tested, and validated using a laptop equipped with an 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz (2.30 GHz) processor, 32 GB of RAM, and an NVIDIA GeForce RTX 3080 Laptop GPU.

4.4.1 PointNet for 3D point cloud superficial roughness classification

The selection of a Deep Neural Network architecture was driven by the limitations of classical machine learning approaches in this specific domain. Preliminary investigations utilizing baseline classifiers (SVM, Random Forest) trained on manually extracted geometric features—such as height variance, mean thickness, and centroid distance—proved insufficient, failing to generalize across the dataset. These global statistical features could not adequately resolve the subtle, localized topological defects (meso-scale roughness) inherent

to the WAAM process. Consequently, PointNet (QI et al., 2016) was selected specifically for its capability to process raw, unordered point sets directly. This allows the model to learn complex, non-linear spatial features through shared Multi-Layer Perceptrons (MLPs) without the information loss associated with manual feature engineering. Other point cloud-based DNN models were evaluated for the presented framework, such as PointNet++ and PointNext, these models weren't able to generalize across all 3 quality classes proposed according to the generated dataset.

The DNN model for classification was created based on the PointNet classification model (QI et al., 2016), adapting the model for TensorFlow 2 using CUDA cores for training optimization and altering the shape of the input nodes to be compatible with normals + positional data input, using vectors of 6 positions to include the normal vector for each point instead of the original positional points, with only 3 positions input. The simplified architecture and layer visualization of the adapted PointNet model can be seen in Figure 31.

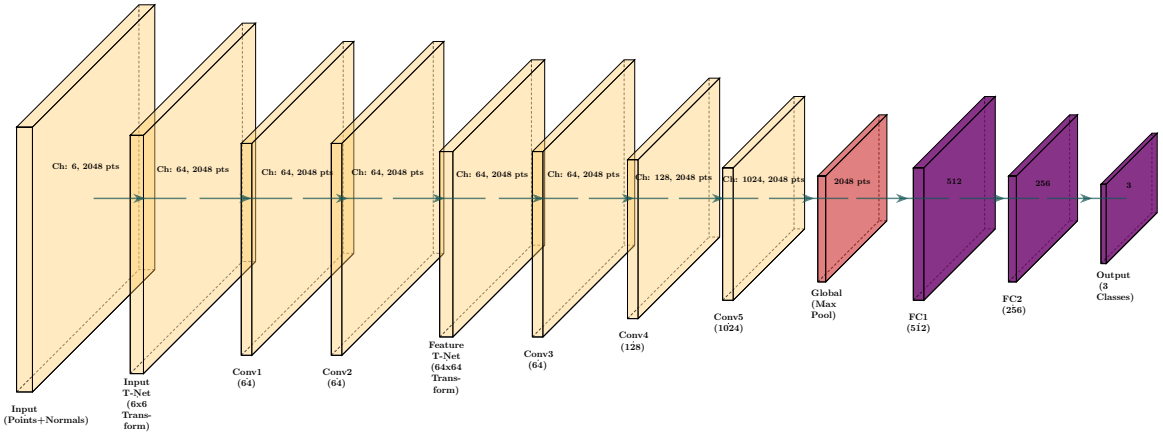


Figure 31 – Adapted PointNet classification network simplified model architecture. Adapted from (QI et al., 2016).

Input 3D parts were segmented into 2048-point point clouds compatible with the neural network input nodes. To enhance the model's geometric awareness, each input point cloud is normalized to fit within a unit sphere, and its associated surface normals are normalized to unit length. These two components—spatial coordinates and normals—are concatenated into a six-dimensional tensor, which is then passed through an input transformation network that learns to align the data spatially. The transformed input undergoes a series of shared multi-layer perceptrons (MLPs), implemented as 1D convolutional layers, to extract local features from each point. A second transformation network refines the feature space by enforcing invariance to geometric distortions, and the resulting features are aggregated into a global descriptor using max pooling.

The global feature vector is passed through a series of fully connected layers, each incorporating batch normalization and dropout to enhance generalization and reduce overfit-

ting. This sequence culminates in a final classification layer that produces raw, unnormalized scores representing the model's confidence for each of the three surface quality classes. These logits are later transformed into probabilities using activation functions such as softmax. To mitigate class imbalance and support stable training, the model leverages a focal loss function with conservative parameters, which places greater emphasis on challenging examples that are harder to classify. Furthermore, a regularization term is applied to the transformation matrices to promote orthogonality and maintain geometric consistency. Together, these components enable the model to effectively learn both local and global spatial features, making it well-suited for assessing surface roughness in metal additive manufacturing parts.

The application of this proposed adapted DNN is discussed through two approaches of training and evaluation, which will be presented in the following subsections.

4.4.2 Simulated 3D point cloud models and superficial roughness classes

In order to establish a threshold for comparison between models, training, and overall validation of the AI models based on the classification of manufactured 3D parts' meshes, it is necessary to define the classes' features and overall requirements among different quality levels.

The first approach on the training part of the proposed Neural Network is based on generating 3D models with five different levels of surface roughness, overbuilding on top, and wall thickness. Such a classification model is conceived to be able to address common WAAM defects such as geometrical deformations and surface waviness (TOMAR; SHIVA; NATH, 2022).

Simulated parts are created based on estimating the input model surface normal vectors and translating surface points by a fraction (up to 4/5) of those normals' distances, accumulating thickness and porosity throughout the quality levels from best to worst. The modifications to surface points are made based on the change chance percentages.

The five quality classification labels applied to both simulated (with their respective punctual defect chances) and real parts are detailed as follows:

- **Excellent**: parts with a tolerance of up to 1/4 of the original models' thickness and a few punctual surface porosity defects. 5% punctual defect chance (20% if the previous point was modified), 1/4 of normal vector length;
- **Good**: parts with a tolerance of up to 1/2 of the original models' thickness. 20% and 30%, 1/4;
- **Fair**: parts with a tolerance of up to 3/4 of the original models' thickness, up to half of the surface area covered with porosity defects. 30% and 50%, 1/4;

- **Poor**: parts with a thickness up to 2.5 times the original models', up to 3/5 of the surface area covered with deformations, visible overbuilding on top. 40% and 60%, 3/5;
- **Bad**: parts up to 3 times the original models' thickness, a rough surface full of deformations and a significant overbuilding on top. 60% and 80%, 4/5.

A balanced batch of 1000 (80% train and 20% test) generated 3D meshes based on the said classes was created from each input CAD. The generated models that served as inputs to the training phase can be visualized in Figure 32, highlighting different quality levels. All 3D models used in the pipeline had their point density corrected based on SPSR density correction, with 49152 points for each input model. The ICP point-to-point algorithm available in the open3D Python library (BESL; MCKAY, 1992) was used to align the generated 3D meshes and point clouds.

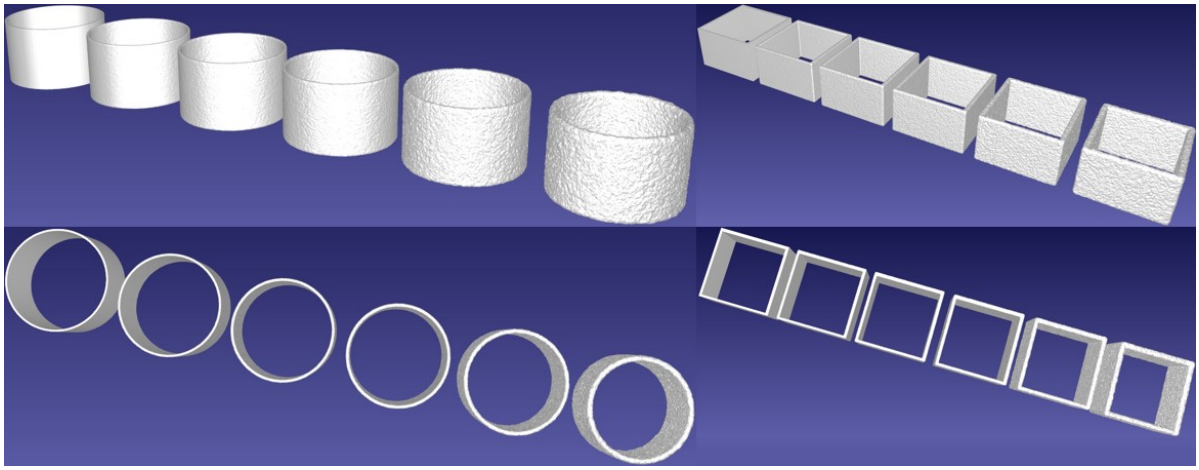


Figure 32 – 3D meshes simulated from original 3D CAD models of a 1.5 mm thick cylinder hull and 3 mm thick cube hull (top left of each image); passing through excellent simulated quality level until bad quality simulated parts (decreasing quality diagonally to the bottom right, top left are the original non-defective CAD models).

Although models with 5 levels of quality were created, only **Excellent**, **Fair**, and **Bad** quality simulated 3D models were used to train the developed grading evaluation models due to training time and accuracy results. Additionally, adjacent classes were too close in their features, such as the calculated superficial Fiedler number (RAO; KONG; DUTY; SMITH, 2016), for the model to distinguish them; some overlapping was also observed. The said evaluation model will be discussed in the following section 4.5.

4.4.3 Second approach, real 3D point cloud models for training and superficial roughness classes

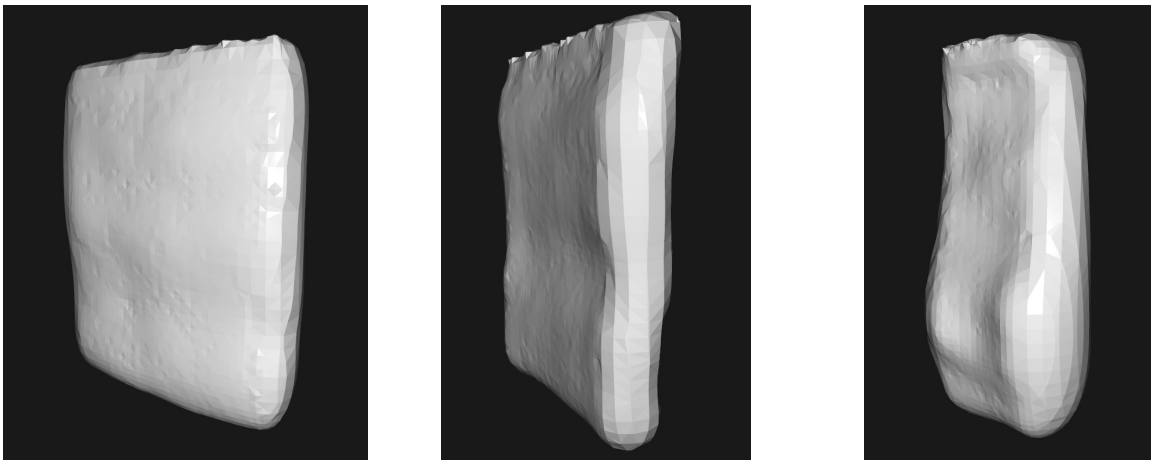
After the radial segmentation commented on in sub-subsection 4.3.2.2 is done, the classification of the real parts' segments is performed using an unsupervised learning approach based on the Fiedler number, a spectral graph metric that quantifies surface connec-

tivity. For each 3D point cloud segment, a k-nearest neighbors (k-NN) graph is constructed, setting $k = 6$ to define local connectivity. The graph's normalized Laplacian is then computed to extract the second-smallest eigenvalue (Fiedler number) (WANG, C.; SAMARI; SIDDIQI, 2018), as calculated from the formula in Equation 2.3. These scalar values serve as the sole input feature for clustering via the K-means algorithm, which groups segments into three predefined quality classes. Segments are then assigned to quality levels (i.e., Excellent, Good, Poor) based on their cluster's mean Fiedler value, and the results are stored and summarized for further analysis.

The classification results of all the segments obtained via the Fiedler number-based KNN clusters' classification are presented in Table 13. Also, examples from classified segments from each quality level can be observed in Figure 33.

Table 13 – **Segment classification summary based on Fiedler Number surface connectivity metric.**

Experiment	Total Segments	Good	Fair	Poor	Fiedler Mean
First Experiment	24	3 (12.5%)	19 (79.2%)	2 (8.3%)	0.002386
Second Experiment	24	3 (12.5%)	14 (58.3%)	7 (29.2%)	0.007046
Third Experiment	24	6 (25.0%)	18 (75.0%)	0 (0.0%)	0.001359
Fourth Experiment	24	4 (16.7%)	20 (83.3%)	0 (0.0%)	0.000752
Overall	**96**	**16 (16.7%)**	**71 (74.0%)**	**9 (9.4%)**	**0.002886**



(a) (b) (c)

Figure 33 – (a) Example of a segment classified as "Good" quality, (b) "Fair" quality segment example, and (c) "Poor" quality segment example.

4.5 Analysis results

In this section, the analytical results of the application of the QA-oriented DNN-based classification framework are discussed. The proposed PointNet architecture serves as the primary mechanism for superficial roughness classification, establishing the baseline for

the quality assurance of manufactured parts. The validation and performance of this model are evaluated across two distinct domains: first, its application on a controlled dataset of simulated 3D models, followed by its evaluation on photogrammetry-based meshes acquired from real-world manufactured parts, as presented in the following subsections.

4.5.1 Evaluation Criteria

To assess the performance of the surface roughness classification model in a production setting, five key metrics are considered: Accuracy, Precision, Recall, mean Average Precision (mAP), and F1-score. These indicators help evaluate the system's ability to evaluate levels of surface quality reliably and consistently.

Finally, a separate Confidence Interval (CI) metric for the uncertainty around the observed accuracy on the training process is presented, which is calculated separately after the conclusion of the DNN model's training.

4.5.1.1 Accuracy

Accuracy measures the overall correctness of the model's predictions. It represents the proportion of correctly classified instances to the total number of samples.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Where:

- TP: True Positives — quality class correctly identified.
- TN: True Negatives — other classes of quality' segments correctly identified.
- FP: False Positives — quality class incorrectly identified as belonging to the class in question.
- FN: False Negatives — quality class of a segment of the analyzed class in question being incorrectly labeled as another class.

4.5.1.2 Precision

Precision evaluates the reliability of positive predictions. It measures the proportion of correctly classified segments among all segments classified in a certain class.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.2}$$

Where:

- TP: True Positives.
- FP: False Positives.

4.5.1.3 Recall

Recall, also known as Sensitivity or True Positive Rate, measures the proportion of segments that were successfully detected in a certain class.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

Where:

- FN: False Negatives.

4.5.1.4 Mean Average Precision (mAP)

mAP is commonly used in object detection tasks. It calculates the average precision across all classes and thresholds, providing a comprehensive view of the model's detection performance.

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{4.4}$$

Where:

- AP_i: Average Precision for class i.
- N: Total number of classes.

4.5.1.5 F1-score

The F1-score combines Precision and Recall into a single metric by calculating their harmonic mean. It is especially useful when the dataset is imbalanced.

$$\text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.5}$$

This score balances the trade-off between Precision and Recall, offering a more nuanced view of the model's effectiveness.

4.5.1.6 Confidence Interval

To evaluate the statistical robustness of the reported performance metrics, specifically the overall accuracy, a 95% Confidence Interval (CI) is calculated. This metric quantifies the uncertainty associated with the point estimate, providing a probabilistic range within which the true model performance is expected to lie. Given the fixed size of the testing dataset, the normal approximation method (Wald Interval) is applied:

$$\text{CI} = \hat{p} \pm Z \times \sqrt{\frac{\hat{p}(1-\hat{p})}{N}} \tag{4.6}$$

Where:

- $\hat{p}$: The observed performance metric (e.g., overall accuracy).
- Z: The Z-score corresponding to the desired confidence level (1.96 for a 95% confidence level).
- N: The total number of samples in the testing dataset.

This interval estimates the uncertainty around the observed accuracy $\hat{p}$ by giving a 95% range of plausible values for the true accuracy that accounts for sampling variability in the test set.

4.5.2 Superficial quality classification on simulated 3D models

In this subsection, the results of the training of the adapted PointNet model on simulated superficial roughness 3D models will be presented, along with the validation results on point clouds of the real parts scanned via photogrammetry.

4.5.2.1 Training and testing on simulated superficial roughness

The hyperparameters of the training session that presented the best results using simulated 3D meshes with different superficial defect levels for quality evaluation are presented in Table 14. Hyperparameters were tuned manually between training sessions based on the accuracy of generalization across classes and general performance.

Training graphs of on-training evaluation metrics are presented in Figure 34. The evaluation graphs were generated from training of the adapted PointNet model throughout 200 epochs, with the best overall model on testing being achieved at epoch 199.

The training analysis of the simulated dataset reveals a characteristic behavior of idealized optimization. The loss evolution graph presents a sharp, precipitous decline during the initial epochs, rapidly stabilizing into a near-zero plateau with minimal oscillation. This

Table 14 – Training hyperparameters for 3-class adapted PointNet model.

Hyperparameter	Value	Description
Training		
Batch Size	32	Samples per batch
Max Epochs	201	Maximum training epochs
Learning Rate (Initial)	0.001	Base learning rate
Decay Step	200000	Steps for learning rate decay
Decay Rate	0.7	Learning rate decay factor
Optimizer (Adam)		
Beta1	0.9	First moment decay rate
Beta2	0.999	Second moment decay rate
Epsilon	1×10^{-8}	Numerical stability term

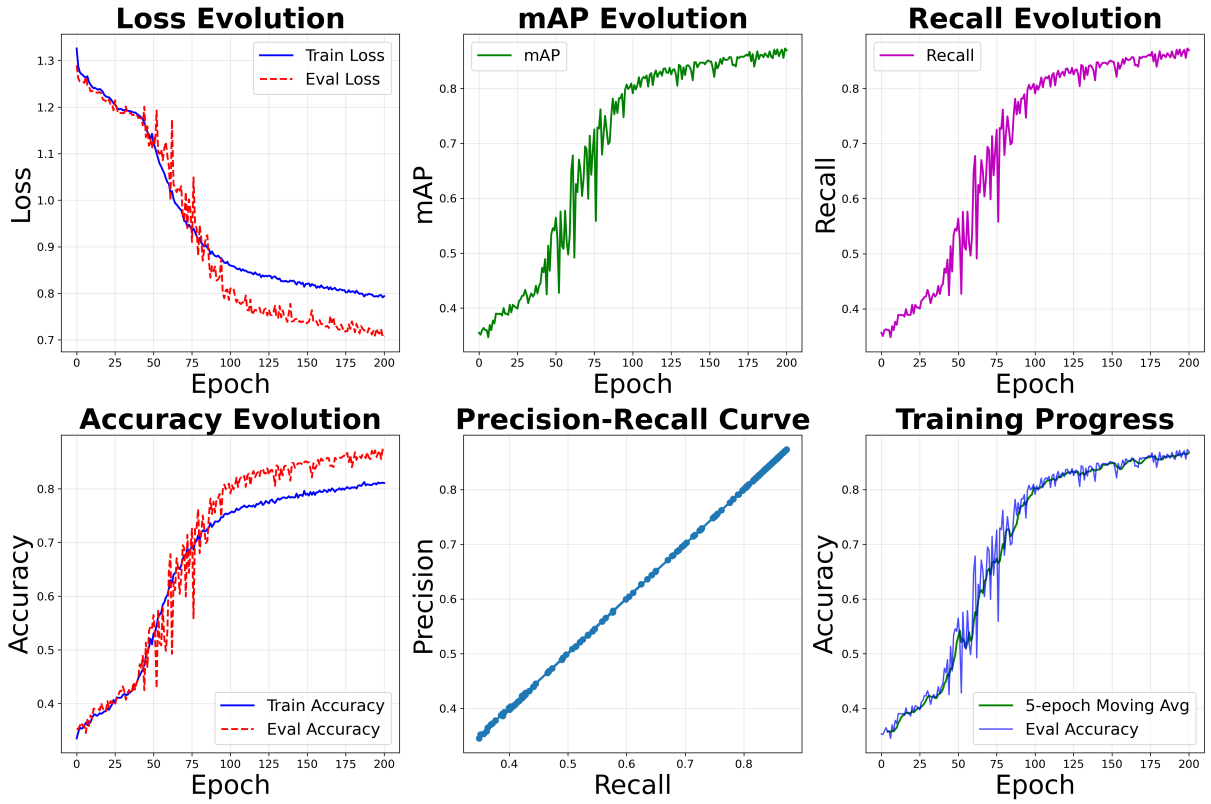


Figure 34 – Training metrics from the adapted PointNet model on the simulated 3D models.

smooth convergence indicates that the PointNet architecture effectively identified the deterministic geometric patterns—specifically the mathematical sine-wave distortions—used to generate the synthetic anomalies. Unlike the real-world dataset, the absence of topological noise and surface irregularities allowed the optimizer to minimize the error gradient efficiently, resulting in a highly stable training run where the validation loss tracks the training loss with exceptional consistency.

Corroborating this behavior, the performance metrics demonstrate a “closed-world” mastery of the defined geometry. Both training and testing accuracy evolve synchronously, reaching a high performance ceiling with a peak accuracy on training+test (all 24000 input

segments, 24 segments per model) of 87.30%. Based on this achieved accuracy on training, a 95% confidence interval based on Equation 4.6 was calculated from the combined sample size ($N = 24000$), yielding a margin of error of the training process's accuracy of $\pm 0.42\%$. The Precision-Recall and mAP curves remain consistently high and stable throughout the latter half of the training, reflecting the model's confidence in distinguishing the discrete, mathematically defined features of the simulated "Good" versus "Poor" classes.

However, the implications of these "perfect" graphs highlight the critical challenge of domain transfer in Digital Twin applications. The extreme stability and lack of variance in the validation metrics suggest that the model has overfitted to the clean logic of the simulation. Because the training data lacked the stochastic surface irregularities inherent to the CW-GTAW process, the network was not forced to learn robust features that account for ambiguity.

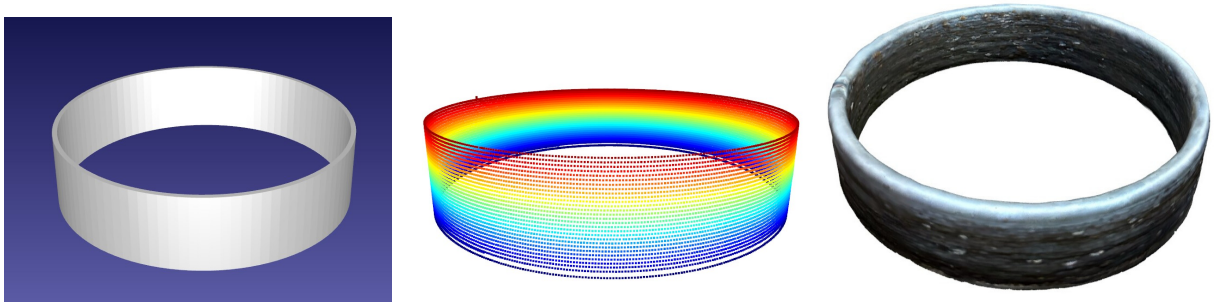
Table 15 – PointNet-based neural network model performance metrics on simulated 3D models' surfaces (test batch, 4800 segments).

PointNet Testing Performance	
Accuracy	0.8599
F1-Score per Class	
Excellent	0.84
Fair	0.79
Bad	0.94
Classification Report	
Excellent	Precision: 0.83, Recall: 0.85
Fair	Precision: 0.82, Recall: 0.77
Bad	Precision: 0.92, Recall: 0.96

Results from testing of the PointNet-based model are presented in Table 15. The PointNet-based neural network has a high 90.62% accuracy, keeping in mind that the 3 classes that serve as input have very strict tolerances. Also taking into account that the PointNet model originally served as a shape classification of 3D point clouds, being a surface roughness classification model more strict and complex considering the large amount of points for each model and superficial details that have to be considered in the evaluation process.

4.5.2.2 Validation on real scanned parts

Initial tests with LiDAR scanning didn't produce results as good as generating 3D models for validation of the real deposited parts via grids of images using PolyCam software on an iPhone 16 Pro Max. For the validation scenario, then, 3 parts were deposited and scanned via photos. An example of the expected CAD model, deposition point cloud seen by the Digital Twin and the real part (scanned mesh via pictures, around 160 photos per part) are shown in Figure 35.



(a) (b) (c)

Figure 35 – (a) CAD model used as a reference for the deposition, (b) Deposited part as seen by the Digital Twin, and (c) Point cloud generated by pictures taken of the real part.

Using the same pipeline of data transformation and analysis/classification described in the methodology, it is possible to compare the obtained 3D models (DT positions and real scanned part) and classify the real parts. Validation results on the real parts are presented in Table 16. No excellent segments were found on the scanned models' surfaces for the real deposited parts based on the criteria proposed in subsection 4.4.2; most of them were classified as bad, which is understandable when considering that the deposition process is still undergoing parameter and fine control tuning. For the evaluation of the classification model, those segments were manually classified based on the criteria of simulated 3D models discussed in sub-subsection 4.3.2.2.

Table 16 – PointNet model performance metrics on real data validation, no excellent segments were detected on deposited parts' surfaces.

PointNet Validation Performance	
Accuracy	0.9062
F1-Score per Class	
Fair	0.73
Bad	0.94
Classification Report	
Fair	Precision: 0.73, Recall: 0.73
Bad	Precision: 0.94, Recall: 0.94

The classification of positional data 3D models captured from the Digital Twin and via images contributes to fault detection over sent commands and stored data when compared to the real results of the part obtained. Imperfections concentrated in specific areas of the real part might contribute to parameter tuning - such as current, wire feed speed, and height offset from the molten pool - when analyzed with the data stored via the DT. Scanning and classifying those 3D models of real parts also helps in having a labeled history of data, further used to improve the deposition process. Code for simulating 3D models, training, and testing the first approach's proposed neural network and pipeline is available on GitHub (CABRAL, J. V. A., 2025a). Since generalization couldn't be proved across all the 3 classes evaluated in this solution, this first approach ended up being rejected.

4.5.3 Superficial quality classification on scanned 3D models from real parts

In this subsection, the results of the training of the adapted PointNet model on quality classification of real photogrammetry-scanned 3D models will be presented, along with the validation results on a separate subset of segments of the same parts.

4.5.3.1 Data augmentation on real scanned parts' segments

For training and testing sets, randomized data augmentation is applied to increase sample diversity and improve model generalization. Augmentation includes rotation (primarily around the Z-axis), scaling, and Gaussian jittering, with augmentation factors tailored to each class to balance representation. Augmentation factors for each class, along with the intervals of parameters for augmentation on each technique, are presented in Table 17.

Table 17 – Data augmentation factors and augmentation technique parameters for 3-class point cloud classification.

Class	**Class Name**	**Augmentation Factor**
Class 0	Good	36
Class 1	Fair	8
Class 2	Poor	90
Total training segments: 1328. **Test:** 310.		

Augmentation Technique	**Parameter**	**Value**	**Application Probability**
Rotation (Z-axis)	Angle Range	±30°	80%
Rotation (X/Y-axis)	Angle Range	±10°	30% (if Z-axis applied)
Scaling	Scale Factor Range	0.9–1.1	60%
Jittering	Standard Deviation	0.01	70%

4.5.3.2 Training process on real scanned parts' segments

Each segment is resampled to a fixed number of points (e.g., 2048 points) and normalized to fit within a unit sphere, ensuring consistency across samples. The final output is stored in compressed HDF5 format, enabling efficient loading and training within the deep learning pipeline.

The training of the adapted PointNet surface roughness classification model was done on real segments of parts that were treated with the described ETL pipeline, using mixed segments from all the 4 real parts deposited described in subsection 4.2, splitting into 70% train and 20% test datasets with data augmentation, and a third balanced dataset for post-validation with 10% of the segments being used without data augmentation for post-training separate evaluation based solely on mixed segments of the real deposited parts. The hyperparameters that resulted in the best overall generalization of the adapted PointNet model and accuracy are presented in Table 18.

Table 18 – Training hyperparameters for 3-class adapted PointNet model.

Hyperparameter	Value	Description
Training		
Batch Size	16	Samples per batch
Max Epochs	500	Maximum training epochs
Learning Rate (Initial)	0.0005	Base learning rate
Learning Rate (Minimum)	0.000001	Minimum learning rate
Decay Step	20000	Steps for learning rate decay
Decay Rate	0.95	Learning rate decay factor
Minimum Class Accuracy	0.65	Threshold for model saving
Optimizer (Adam)		
Beta1	0.9	First moment decay rate
Beta2	0.999	Second moment decay rate
Epsilon	1×10^{-8}	Numerical stability term

Training routines were rejected/accepted according to their generalization and overall balanced accuracy throughout the 3 classes. Training routines and epochs that resulted in a balanced overall accuracy of over 65% were separated for further validation on the separate validation dataset. In this second approach, hyperparameters were also tuned manually between training sessions based on the accuracy of generalization across classes and general performance.

Metrics evaluation graphs were generated from training of the adapted PointNet model throughout 500 epochs, with the best overall model on evaluation being achieved at epoch 402. Said graphs can be observed in Figure 36.

The training analysis of the model demonstrates a consistent and well-behaved learning process across multiple performance metrics. The loss evolution graph shows a gradual decline in both training and evaluation loss, indicating effective convergence without signs of overfitting. The mAP and recall curves steadily rise over epochs, reflecting improved precision and sensitivity in classification. Accuracy metrics for both training and evaluation follow an upward trajectory, with evaluation accuracy slightly trailing — a healthy sign of generalization. The precision-recall scatter plot reveals a balanced trade-off, with points clustering along the diagonal, while the training progress graph confirms stable learning through a rising epoch average and evaluation accuracy. Together, these visualizations validate the robustness of the model and its suitability for surface roughness classification tasks in metal additive manufacturing.

However, volatility in the loss, along with noticeable noise across all other metrics (mAP, Recall, and Accuracy), indicates an unstable training run, potentially due to a high diversity of data with a small sample size of similar segments with similar superficial roughness characteristics. This instability makes it difficult to reliably assess the model's true performance at any single epoch, but the general evaluation shows an evolution in general

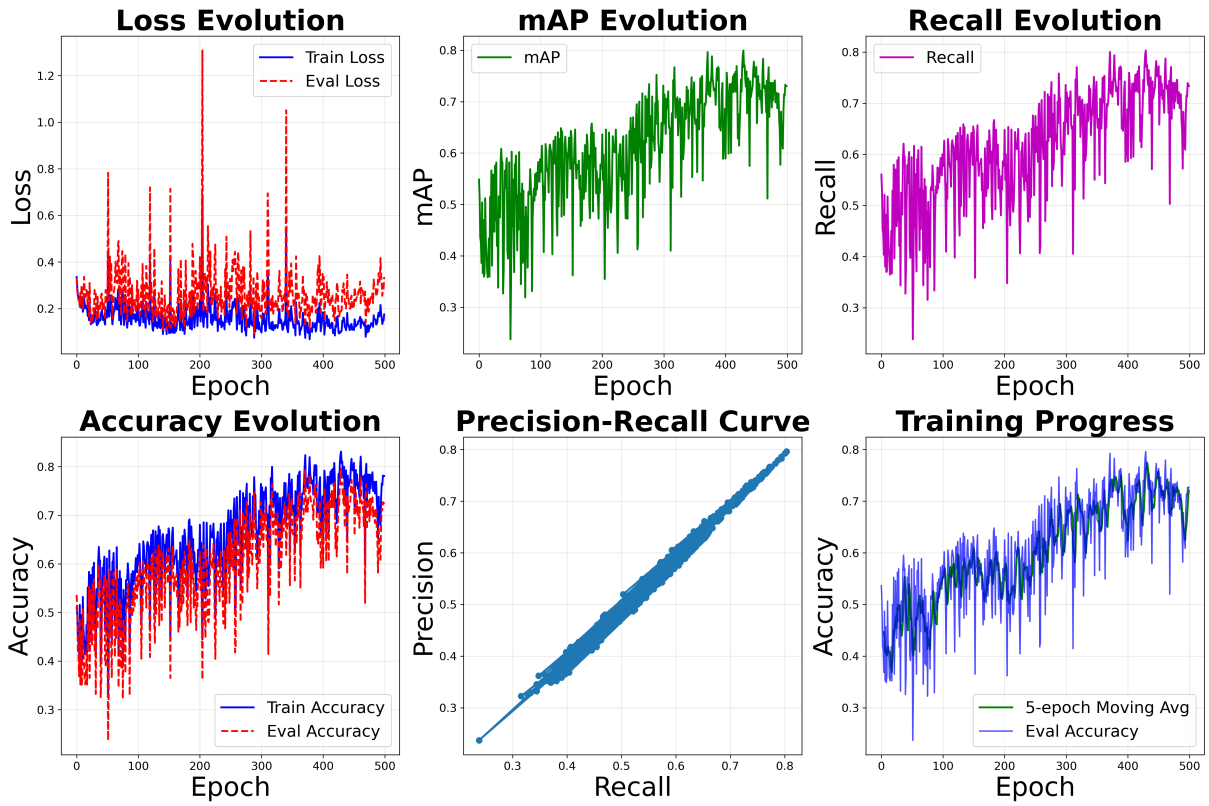


Figure 36 – Training metrics from the adapted PointNet model on the mixed data-augmented random segments from all the experiments of the real deposited parts.

accuracy.

4.5.3.3 Testing and validation results

Results from testing and post-training unaugmented segments' data validation of the PointNet-based model are presented in Table 19, with their respective confusion matrices presented in Figure 37.

The PointNet-based neural network has a fair accuracy of 75.64%, keeping in mind that the 3 classes that serve as input have very strict tolerances for different classifications among input classes, with high variability on input data, which contrasts with the small dataset available when considering the very difficult task of evaluating surface roughness quality of real manufactured parts using 3D point clouds. The achieved accuracy is considered substantial, particularly when acknowledging that PointNet's original function was 3D shape classification. The shift to surface roughness demands a higher sensitivity to localized superficial details, making this quality assessment task considerably more challenging.

Regarding the statistical robustness of the 75.64% accuracy, a 95% confidence interval (based on Equation 4.6) was calculated based on the combined evaluation sample size ($N = 322$), yielding a margin of error of $\pm$4.69%. While the overall accuracy reflects the challenge of distinguishing between the subtle topological boundaries of "Good" and "Fair"

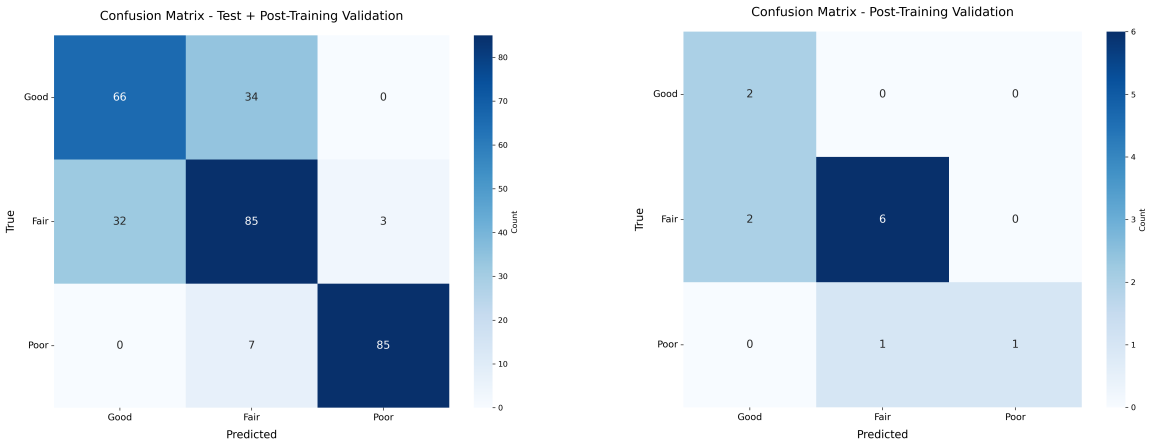
Table 19 – (a) Evaluation metrics of the adapted PointNet model on Test + Post-training Validation segments' data (b) Evaluation metrics on Post-training Validation segments only.

(a)

PointNet - Test + Post-training validation metrics	
Accuracy	0.7564
F1-Score per Class	
Good	0.67
Fair	0.69
Poor	0.94
Classification Report	
Good	Precision: 0.67, Recall: 0.66
Fair	Precision: 0.67, Recall: 0.71
Poor	Precision: 0.97, Recall: 0.92

(b)

PointNet - Post-training validation metrics	
Accuracy	0.7500
F1-Score per Class	
Good	0.67
Fair	0.80
Poor	0.67
Classification Report	
Good	Precision: 0.50, Recall: 1.00
Fair	Precision: 0.86, Recall: 0.75
Poor	Precision: 1.00, Recall: 0.50



(a) (b)

Figure 37 – (a) Confusion matrix of the application of the adapted PointNet model on Test + Post-training Validation segments' data, (b) Confusion matrix on Post-training Validation segments' only.

classes using only geometric data, the model demonstrates exceptional performance where it matters most for Quality Assurance: defective parts' detection. The classification report (Table 19) indicates a Recall of 0.92 and Precision of 0.97 for the "Poor" class. This confirms that while the model may occasionally misclassify acceptable surface variations, it is highly reliable at identifying significant deviations, minimizing the risk of false negatives (Type II errors) in an industrial inspection context.

4.5.4 Specific execution order of applied algorithms

This project leverages the PointNet classification model to assess the superficial quality of metal deposited hollow parts based on their roughness, utilizing TensorFlow 2.10 on Windows 11 to bridge legacy functions for 3D surface processing. The data simulation, transformation, and preparation pipeline operates through a defined sequence of Python scripts designed to convert raw geometric inputs into segmented, classified data suitable for neural network training.

Aiming for the correct functioning of the whole surface quality classification pipeline based on the adapted PointNet DNN model, for the specific workflow for simulating 3D models, transforming data, and training the proposed adapted neural network, the following order of algorithm execution is necessary, as stated on the project's GitHub (CABRAL, J. V. A., 2025a):

- Run `create_rugosity_from_mesh.py` in order to generate superficial rugosity on the base 3D models in the original directory based on pre-defined classes; if utilizing a simulated dataset, *N* batch size 3D models and .csv files for positional and normals data will be created;
- Run `create_mesh_from_xyz_bottom.py` to generate correct .STL meshes from validation models and to extract positional data and normals to be stored in .csv files to be further processed, if utilizing a validation dataset or real parts for training;
- Run `segmentation_icp_point_reduction_pie.py` in order to create 2048-point segments from the simulated or validation positional data to prepare the input for the neural network;
- Run `classify_validation_segments.py` for the k-means clustering of segments to classify their superficial roughness quality level according to their calculated Fiedler number;
- Run `pre_process_data_unified_mixed_3_4_classes.py` to create H5 files that will be used as input to the neural network, with or without data augmentation according to the configuration;
- Finally, run `train_with_normals.py` to train the neural network on the segmented simulated 3D models.

4.5.5 Comparisons on 3D point clouds of real deposited parts (DT x Photogrammetry x CAD, and Photogrammetry x 3D laser scanner)

As mentioned in the methodology section 3.5, to validate the photogrammetry method for 3D point cloud extraction and mesh generation, photogrammetry-based point

clouds are compared with DT-generated ones and with the referential CAD models. The objective is to verify tolerances based on RMSE and Hausdorff as mentioned in the same referenced section in the methodology chapter, and also to establish the initial fuzzy-based pipeline problem detection as explained in the methodology section 3.7.

Also, for a final validation of the captured point clouds made via photogrammetry, manufactured parts were 3D laser scanned as mentioned in the methodology section 3.2, an example of the scan setup is shown in Figure 38.

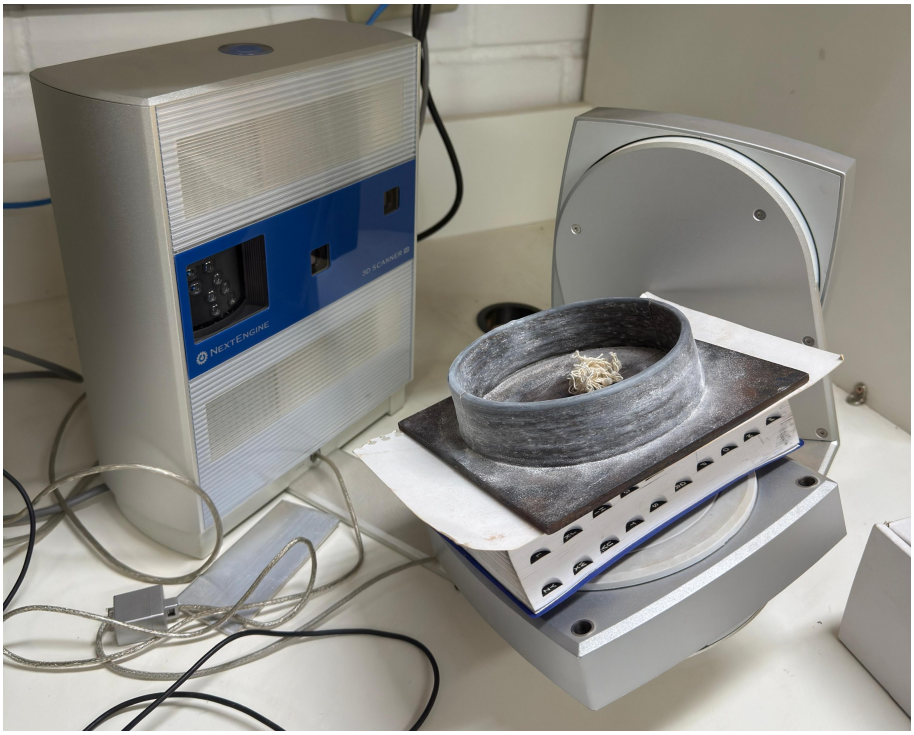


Figure 38 – Scan setup for the NextEngine 2020i 3D laser scanner. Height was adjusted for the lenses' focus, and a piece of fiber was added for referential surface reconstruction.

However, because the parts were scanned on the desktop 3D laser scanner after all parts contemplated in this study were deposited - since the 2nd and 3rd experiments used the same baseplate - the 2nd deposited part couldn't have its outer wall scanned, as it was overlapped by the outer wall of the 3rd part in the vision angle of the camera of the NextEngine 3D scanner. Also, one of the main problems when scanning the parts with the 3D laser scanner was the lack of a reference for surface reconstruction, since they are essentially uniform cylinders; thus, the baseplate served as a reference for the scanning. Hence, it was decided not to cut those parts, considering the risk of introducing new geometrical errors and discrepancies when compared to the previously acquired photogrammetry-based 3D scans. Therefore, unfortunately, the 2nd part's laser 3D scan is excluded from the analysis of this subsection and the subsequent one.

The results on comparing the 3D point clouds after ICP alignment using Python's Open3D library are presented in the sub-tables in Table 20.

Table 20 – Quantitative alignment metrics extracted from the ICP comparison output. Each mini-table shows per-experiment Fitness, RMSE and Hausdorff distance for the named comparison.

Digital Twin vs. Photogrammetry			
Experiment	**Fitness**	**RMSE (mm)**	**Hausdorff (mm)**
1	1.00	1.77	7.64
2	1.00	1.39	4.49
3	1.00	2.97	8.39
4	1.00	1.98	5.60
Mean ± Std. Dev.	**1.00 ± 0.00**	**2.03 ± 0.58**	**6.53 ± 1.56**

Digital Twin vs. CAD			
Experiment	**Fitness**	**RMSE (mm)**	**Hausdorff (mm)**
1	1.00	1.33	2.00
2	1.00	1.35	4.16
3	1.00	1.34	4.10
4	1.00	1.36	3.72
Mean ± Std. Dev.	**1.00 ± 0.00**	**1.35 ± 0.01**	**3.50 ± 0.88**

Photogrammetry vs. CAD			
Experiment	**Fitness**	**RMSE (mm)**	**Hausdorff (mm)**
1	1.00	1.09	5.47
2	1.00	0.94	2.44
3	1.00	4.09	9.55
4	1.00	1.42	3.99
Mean ± Std. Dev.	**1.00 ± 0.00**	**1.88 ± 1.28**	**5.36 ± 2.64**

Photogrammetry vs. 3D Scanner			
Experiment	**Fitness**	**RMSE (mm)**	**Hausdorff (mm)**
1	1.00	1.52	7.34
2	N/A	N/A	N/A
3	1.00	3.83	10.80
4	1.00	1.84	5.29
Mean ± Std. Dev.	**1.00 ± 0.00**	**2.40 ± 1.02**	**7.81 ± 2.27**

Comparing the data of Hausdorff and RMSE metrics on the Photogrammetry model and CAD (considering the expected scale drift of 0–3 mm on the measurement), it is noticeable that most of the experiments have an RMSE value lower than 1.5 mm. However, this value, when aligned with high observed Hausdorff values in experiments 3 and 4, evidences a high value of the scale drift discussed.

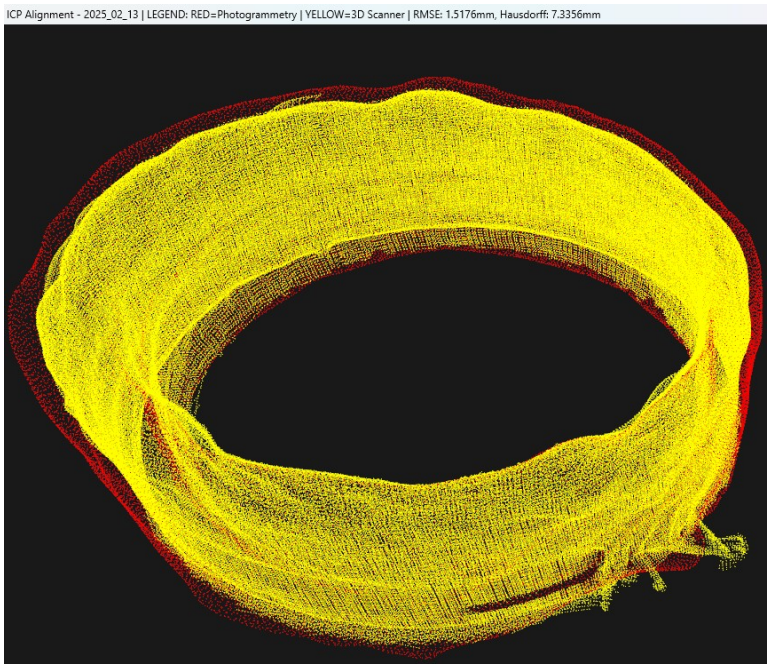
On the other hand, when results from alignment metrics are compared between the photogrammetry and the 3D laser scanner, the observed increase in the RMSE and Hausdorff values is probably due to the better resolution of the scans from the 3D laser scanner, detecting smaller superficial features. But also might indicate the lack of alignment due to possible stacking of errors in the scanning process, both from the 3D laser scanner and photogrammetry, resulting in a poorer alignment when compared to the CAD reference model ones. This hypothesis that higher resolution results in better detection of surface roughness is further validated by comparing the results of CAD and the 3D laser scanner, as shown in Table 21.

Table 21 – ICP comparison metrics — 3D Scanner vs. CAD

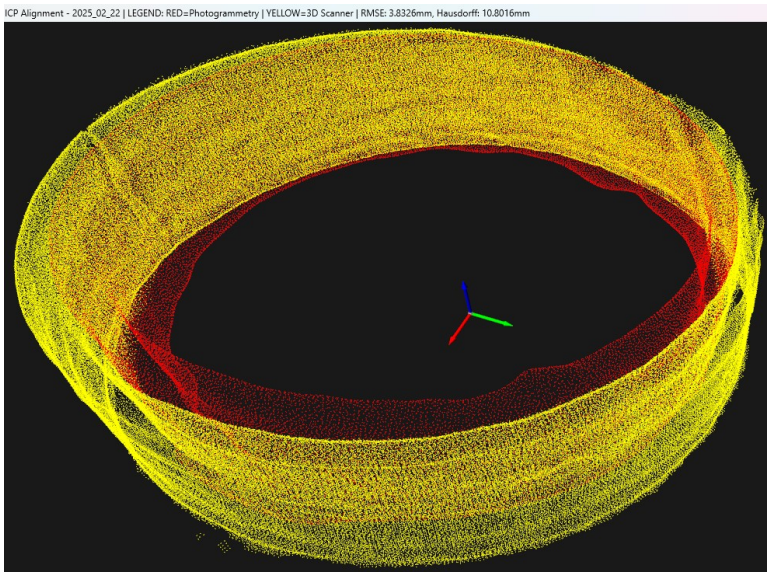
Experiment	**Fitness**	**RMSE (mm)**	**Hausdorff (mm)**
First	1.00	1.44	9.47
Second	N/A	N/A	N/A
Third	1.00	1.31	5.13
Fourth	1.00	0.86	3.27
Mean ± Std. Dev.	**1.00 ± 0.00**	**1.20 ± 0.25**	**5.96 ± 2.60**

Compared to the results of photogrammetry vs. CAD, it is noticeable that the RMSE values of the 3D laser scanner, when put against the CAD reference, present considerably different values. In the first experiment's part, the 3D laser scanner better captured superficial defects whilst it presented lower RMSE values, which are probably higher in the photogrammetry ones because of drifts in scale, which is considerably higher in the third experiment's part in the photogrammetry scan vs. CAD metrics.

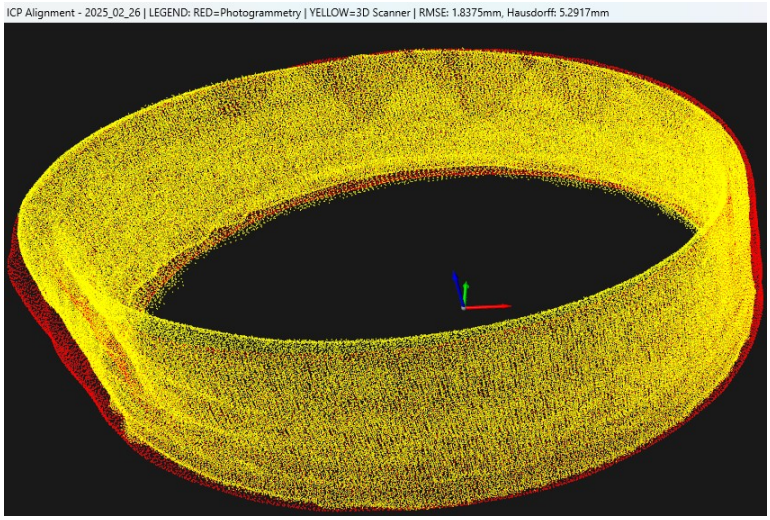
The enhanced definition and higher resolution of observed defects of the 3D laser scanner can be observed in the superimposed point clouds from photogrammetry and laser scans in Figure 39.



(a)



(b)



(c)

Figure 39 – (a) First, (b) third, and (c) fourth experiments' point clouds of photogrammetry (in red) superposed by 3D laser scanner (in yellow).

It is important to note, though, that the transformations of the proposed pipeline that culminate in the evaluation of segments of deposited parts by the adapted PointNet DNN should not be deeply affected by such scale deviations on diameter and possibly height, because surfaces' sampling is uniformized and all segments are normalized through unit sphere normalization before being fed into the DNN model. However, in the particular case of experiment 3, it is probable that, because experiment 2 was beforehand deposited in the same baseplate, the second part's shadow caused a large-scale drift and interfered with the superficial points detection by the photogrammetry-based PolyCam scan, resulting in a smoother surface than what would be expected. This problem can be minimized by correcting the light in the scene during scanning, and also acquiring a scanner with better nominal maximum accuracy.

Those hypotheses on the scale errors and magnitude of defects are further investigated in subsection 4.5.6.

4.5.5.1 Fuzzy inspection analysis on ICP metrics

Based on the comparative analysis of CAD, Digital Twin (DT), and Photogrammetry point clouds presented in Table 20, potential deviations in the photogrammetry generation, DT pipeline, and the robotic CW-GTAW AM cell configuration are diagnosed using the fuzzy decision-making logic referenced in Table 13. Table 22 summarizes the binary assessment of these comparisons, where "O" indicates sufficient similarity (RMSE $\leq$ 1.65 mm and Hausdorff $\leq$ 4.40 mm) and "X" indicates a deviation exceeding these thresholds.

Table 22 – Binary sufficiently equivalence point clouds assessment (O = both RMSE $\leq$ 1.65 mm and Hausdorff $\leq$ 4.40 mm; X = one or both metrics exceed thresholds).

Experiment	DT vs. Photogrammetry	DT vs. CAD	Photogrammetry vs. CAD
First	X	O	X
Second	X	O	O
Third	X	O	X
Fourth	X	O	O

The consistent "O" result in the DT vs. CAD comparison across all experiments indicates that there was no catastrophic failure in the path planning or robot kinematics. It confirms that the programmed RAPID trajectory was correctly executed and aligned with the ideal design specifications. Consequently, it can be inferred that the robotic cell configuration and sensor calibration were functioning within nominal parameters regarding positioning.

Conversely, the DT vs. Photogrammetry comparison yielded a consistent "X" (mismatch) across all datasets. Under the fuzzy inference rules, combined with the successful DT-CAD alignment, this isolates the error to a lack of volumetric fidelity in the Digital Twin. Currently, the DT records the Tool Center Point (TCP) trajectory—effectively a 1D line in 3D space—but lacks a physics-based simulation to estimate the deposited bead width and

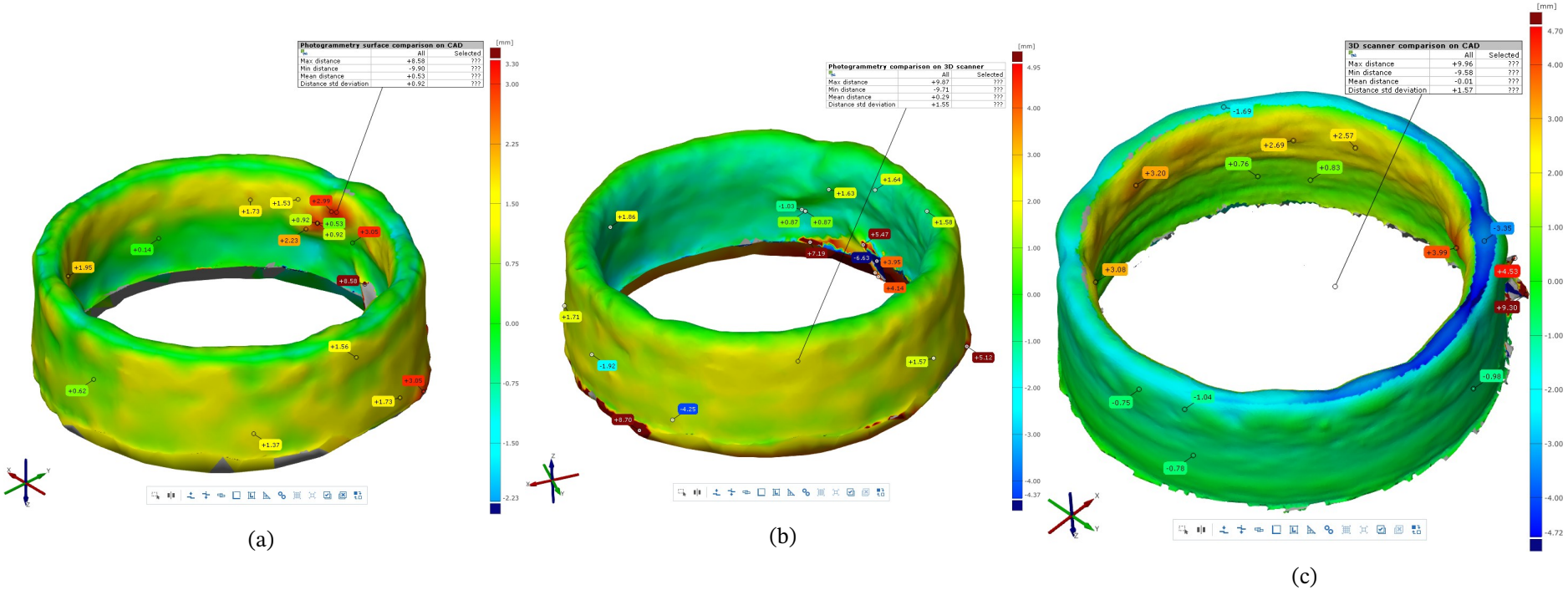
layer height. Thus, the estimated 5 mm wall thickness from the CAD models is applied to the DT-based point clouds for comparison, currently lacking real variability in a way that errors from DT-CAD comparisons are actually from the deviations and layering variations of acquired TCP Z position during the deposition sessions. Since the photogrammetry captures the actual volumetric geometry (hollow tubes with wall thickness), the ICP algorithm registers a high residual error when attempting to align a "thin" trajectory wireframe with a "thick" physical mesh. This highlights a necessity for future work: integrating thermal or visual sensors (e.g., UV-filtered cameras) to estimate bead geometry in real-time and inflate the DT model accordingly.

Finally, the photogrammetry vs. CAD comparison shows mixed results. Since the manufacturing process was stable, as evidenced by the consistent visual quality of parts 2 and 4, the intermittent "X" results are attributed to the scale drift inherent in the Structure-from-Motion (SfM) algorithm present in PolyCam photogrammetry discussed in sub-subsection 4.3.1.3, which, in the case of part 3 comparison, the scale drift was way too large, which also contributed to a bad alignment and RMSE results. While the geometry is topologically correct, minor scale deviations in the scanning process occasionally push the Hausdorff distance beyond the strict 4.40 mm threshold, even when the physical part is acceptable.

4.5.6 3D quality control inspection

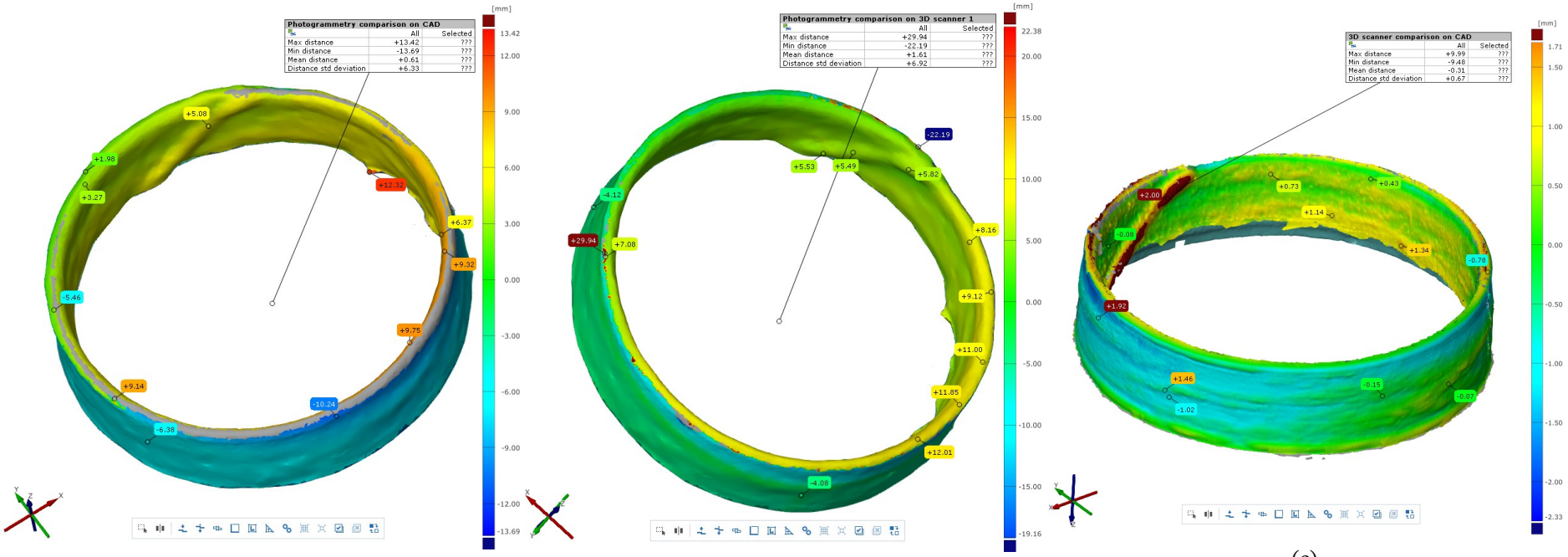
Finally, regarding the general QA 3D inspection of fabricated parts, the ZEISS 3D Inspection software version 2026.2.0.1091 is used to generate heatmaps on surface comparisons to evaluate distances between surface points of 3D models.

For this step of analysis, CAD models, photogrammetry-based, and 3D scanner .stl files are used for superposed surface evaluation deviation metrics. Visualizations of the generated heatmaps are available in the next pages' Figures [40–43].



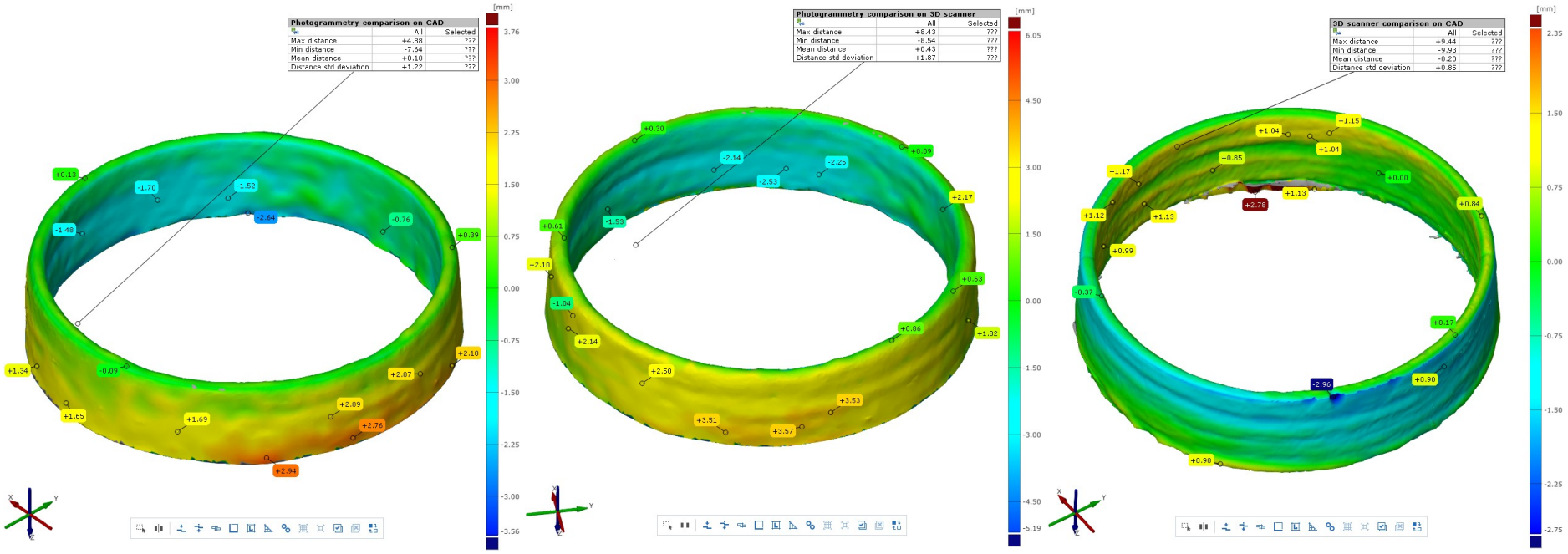
(a) (b) (c)

Figure 40 – Zeiss surface heatmap differences inspection of the first deposited part. (a) Photogrammetry against CAD, (b) Photogrammetry against 3D scanner, and (c) 3D scanner against CAD.



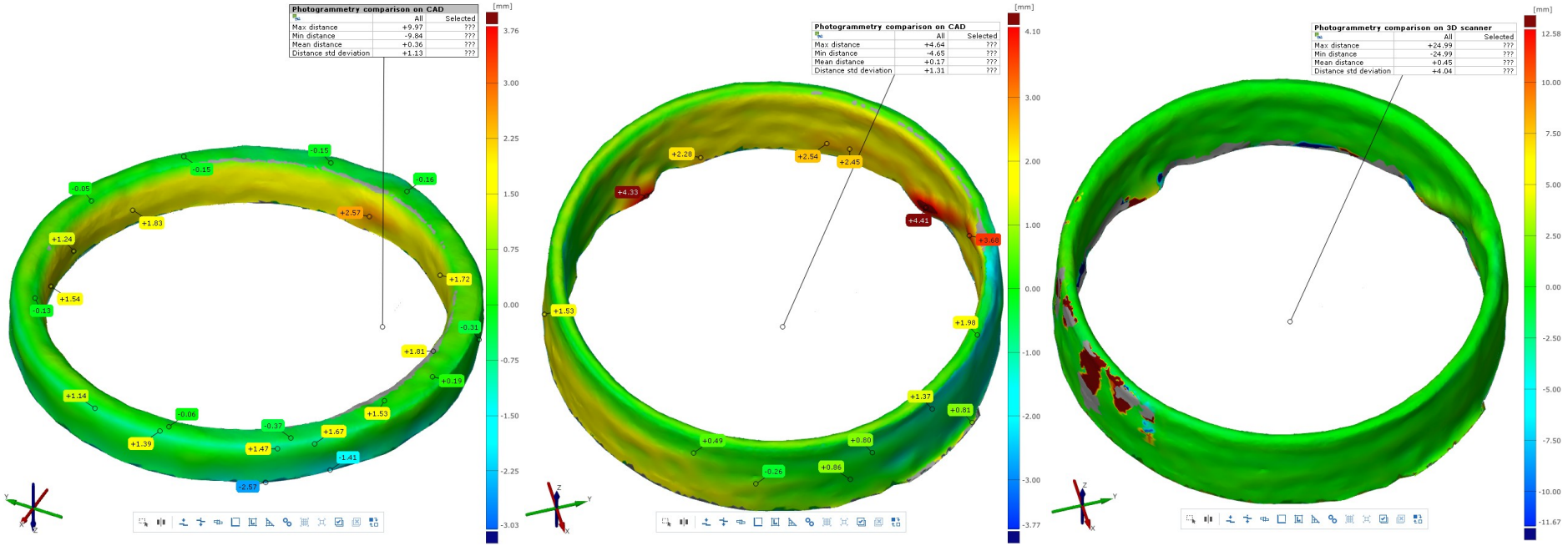
(a) (b) (c)

Figure 41 – Zeiss surface heatmap differences inspection of the third deposited part. (a) Photogrammetry against CAD, (b) Photogrammetry against 3D scanner, and (c) 3D scanner against CAD.



(a) (b) (c)

Figure 42 – Zeiss surface heatmap differences inspection of the fourth deposited part. (a) Photogrammetry against CAD, (b) Photogrammetry against 3D scanner, and (c) 3D scanner against CAD.



(a) (b) (c)

Figure 43 – (a) Zeiss surface heatmap differences inspection of the second deposited part against CAD. Zeiss surface heatmap differences inspection of the third deposited part with 1.1 * height and diameter scale correction on photogrammetry .stl input. (b) Photogrammetry against CAD, and (c) Photogrammetry against 3D scanner.

Comparing the results of the surface comparisons on Zeiss GOM analysis against the ICP metrics obtained from the alignment of point clouds, the following differences are obtained, presented in Tables [23–25].

Table 23 – Zeiss GOM vs. Point Cloud ICP - Photogrammetry vs. CAD.

Photogrammetry vs. CAD				
Experiment	**Metric**	**Zeiss GOM Values (mm)**	**Point Cloud ICP Values (mm)**	**Difference (mm)**
First	RMSE	0.92	1.09	-0.17
	Hausdorff	9.90	5.47	+4.43
Third	RMSE	6.33	4.09	+2.24
	Hausdorff	13.69	9.55	+4.14
Fourth	RMSE	1.22	1.42	-0.20
	Hausdorff	7.64	3.99	+3.65

Table 24 – Zeiss GOM vs. Point Cloud ICP - Photogrammetry vs. 3D scanner.

Photogrammetry vs. 3D scanner				
Experiment	**Metric**	**Zeiss GOM Values (mm)**	**Point Cloud ICP Values (mm)**	**Difference (mm)**
First	RMSE	1.55	1.52	+0.03
	Hausdorff	9.87	7.34	+2.53
Third	RMSE	6.73	3.83	+2.90
	Hausdorff	29.94	10.80	+19.14
Fourth	RMSE	1.87	1.84	+0.03
	Hausdorff	8.54	5.29	+3.25

Table 25 – Zeiss GOM vs. Point Cloud ICP - 3D scanner vs. CAD.

3D scanner vs. CAD				
Experiment	**Metric**	**Zeiss GOM Values (mm)**	**Point Cloud ICP Values (mm)**	**Difference (mm)**
First	RMSE	1.57	1.44	+0.13
	Hausdorff	9.96	9.47	+0.49
Third	RMSE	0.67	1.31	-0.64
	Hausdorff	9.99	5.13	+4.86
Fourth	RMSE	0.85	0.86	-0.01
	Hausdorff	9.93	3.27	+6.66

The analysis of the geometric comparison data reveals a consistent discrepancy in the Hausdorff distance metrics, where the values reported by GOM Inspect are significantly

higher than those generated by Open3D. This variance is primarily driven by the fundamental difference in how scanning artifacts are treated; the elevated Hausdorff values peaking at 19.68 mm in the third experiment are indicative of localized reconstruction errors, such as holes or noisy boundary edges manifested in the 3D models. While Open3D's algorithms may filter or smooth these outliers during the Iterative Closest Point (ICP) calculation, GOM Inspect identifies these specific topological defects as maximum deviations, highlighting the sensitivity of the metric to model completeness rather than just surface alignment. In our workflow, RMSE is computed after ICP alignment using the nearest-neighbor approach: the aligned source points are queried against the dense reference (target) using a KD-tree, the Euclidean distance to each point's closest neighbor is recorded, and the RMSE is the square root of the mean of those squared nearest-neighbor distances. This nearest-neighbor RMSE avoids the incorrect assumption of index-wise correspondence (point[i] ↔ point[i]) that would arise if clouds were subtracted directly, and it reflects the practical matching used to evaluate geometric agreement after registration. Also, in the watertight models used in GOM for the comparisons, holes and detection defects in the scans are prone to cause high deviations, resulting in a higher Hausdorff metric. The Python functions for the ICP metrics calculation are available in the appendix section A.2.

The consistent observation of elevated RMSE values in the 3D scanner models compared to their photogrammetry counterparts is, counter-intuitively, attributable to the scanner's superior resolution rather than a lack of precision. The 3D scanner captures a significantly higher density of surface details, including micro-textures and roughness inherent to the WAAM process, which the photogrammetry reconstruction tends to smooth over. While photogrammetry algorithms often interpolate "cleaner" surfaces by averaging pixel data, the scanner faithfully records the true peaks and valleys of the deposited layers. This fidelity results in a higher mathematical deviation (RMSE) from the idealized CAD surface, even though the scanner data is a more accurate representation of the physical reality.

Competitiveness between the two acquisition methods should fundamentally be evaluated through their respective deviations from the CAD reference, with a distinct preference given to the 3D scanner for metrology applications. As demonstrated in the first experiment (Table 23 vs. Table 25), while photogrammetry achieved a lower RMSE (0.92 mm vs. 1.57 mm), the 3D scanner is constrained by a much tighter hardware accuracy limit (0.127 mm) compared to the estimated uncertainty of the photogrammetry setup (0.3 mm). This implies that the scanner's deviations are reliable measurements of the part's actual form error, whereas the photogrammetry's "better" scores may result from the smoothing of real geometric features. Consequently, despite higher nominal RMSE values in early trials, the 3D scanner remains the preferred instrument for validating the Digital Twin, offering a traceable accuracy that ensures quality control decisions are based on the part's true topology rather than an algorithmic approximation.

However, the significant deviation observed in the third experiment—yielding an RMSE of 6.33 mm against the CAD—should not be interpreted as a degradation of surface topology, but rather as a systematic scale drift of approximately 10% in both diameter and height. It is probable that because the second experiment's part remained deposited on the same baseplate, the shadow cast by the second part interfered with the photogrammetry-based PolyCam scan of the third part. This optical occlusion likely disrupted the depth estimation, resulting in a reconstructed model that was geometrically smoother than expected yet dimensionally undersized. Conversely, the 3D scanner, serving as the assessment baseline, exhibited tighter adherence to the reference geometry, with higher RMSE values (e.g., 1.57 mm) directly correlating with surface roughness rather than scaling errors.

The quantitative comparisons for the second experiment's photogrammetry against the CAD model are presented in Table 26. This table also includes the results for the third experiment against both the CAD and 3D laser scanner, following a required scale correction. Due to a large-scale drift in the original scan, a factor of 1.1 was applied to the height and diameter of the third experiment's photogrammetric 3D model, as supported by Autodesk Fusion estimates.

Table 26 – Comparison of photogrammetry metrics for the second experiment and the photogrammetry scale-corrected third experiment.

Metric	**Photogrammetry x CAD (Second experiment)**	**Photogrammetry x CAD (Third exp. scale corrected)**	**Photogrammetry x 3D Scanner (Third exp. scale corrected)**
RMSE (mm)	1.13	1.31	4.04
Hausdorff (mm)	9.97	4.64	24.99

The validity of the photogrammetry data is further confirmed when a linear scale correction factor of 1.1 is applied to the third experiment's dataset. Upon correction, the RMSE against the CAD model drops drastically to 1.31 mm, a value remarkably consistent with the second experiment (1.13 mm). Additionally, the RMSE against the 3D scanner reduced to 4.04 mm, corroborating the hypothesis of a "geometrically correct but scaled" model. The residual high Hausdorff distance (24.99 mm) in the scanner comparison is attributable to specific surface reconstruction errors—such as holes generated by the ScanStudio PRO software—rather than a generalized alignment failure. Consequently, the proposed framework remains robust for quality assurance; crucially, the observed scale drift poses no issue for the downstream DNN quality evaluation, as surfaces are segmented and normalized prior to vectorization, rendering the network input effectively scale-invariant.

When evaluating these results against the manufacturing quality tolerance of RMSE 1.5 mm $\pm$ 10% (an acceptable range of 1.35 mm to 1.65 mm), the photogrammetry pipeline exhibits a risk profile contingent on environmental control rather than intrinsic capability.

While the scanner provided a reliable baseline, stabilizing at a highly accurate 0.85 mm in later trials, the photogrammetry method also successfully fell within tolerance limits in the first (0.92 mm), second (1.13 mm), and fourth (1.22 mm) experiments. Even the third experiment, once corrected for scale drift, satisfied the quality criteria (1.31 mm). Consequently, it is concluded that photogrammetry is a viable quality assurance tool, provided that scale drift, often signaled by suspiciously high deviations, is identified and minimized through proper lighting calibration or post-process scaling.

5 Discussion

In this chapter, the general evaluation of the whole analytical process developed and the experiments conducted will be discussed. The limitations of the feature detection algorithms and methods applied will also be addressed, comparing them with similar applications from the state-of-the-art of analytical methods on metal AM deposition processes in the context of Industry 4.0.

5.1 Performance of the applied PointNet on surface roughness evaluation

Although the approved model's overall accuracy is 75.64%, its practical utility in a manufacturing setting is better defined by its robust capability to detect the most critical failures. As detailed in the confusion matrix and testing results (Table 19 (a) and Figure 37 (a)), the model demonstrated high-confidence detection for the 'Poor' quality class, achieving a 0.97 Precision and a 0.92 Recall. This reliability in identifying and isolating unacceptable components fulfills a key requirement for industrial quality control. It enables automated decision-making, such as flagging parts for immediate rework or initiating a process shutdown, which is vital for optimizing production cycles and reducing material waste.

If the ETL pipeline is integrated as-is, as proposed in this work, to a metal AM manufacturing cell on the factory floor, it is expected to take around 20-22 minutes from the start of scanning the complete deposited part with the PolyCam software (taking 160-180 pictures) to the complete surface quality evaluation of the segments of the deposited part. However, if a dedicated camera is fixed at the manufacturing cell to take those pictures in a pre-programmed routine that consumes from the PolyCam API, this processing time could be reduced by half.

The proposed QA framework, which classifies surface roughness, proved effective for assessing and tuning process parameters. This was validated by the progressive improvement in manufacturing quality observed in each subsequent experiment, which leveraged insights from the Digital Twin dataflow.

A key contribution of this pipeline is its ability to compare the 3D models generated from DT positional data against the 3D models from scanned real-world parts. This comparison facilitates fault detection by highlighting discrepancies between commanded actions and actual results. Furthermore, by analyzing this data from the DT, imperfections concentrated in specific areas of a part can be correlated with process parameters (like current, wire feed

speed, or molten pool height offset), providing a direct feedback loop for parameter tuning. This process of scanning and classifying real parts also generates a valuable labeled historical dataset, which can be used to continually refine the deposition process.

In the context of other AI and machine learning models for quality inspection in manufacturing (RYDZI et al., 2024; SANKHYE; HU, 2020), the achieved results are comparable, with typical evaluation accuracies in the 80 to 99 percent range. A specific example, (ALVARES, A. J. et al., 2025), utilized DNNs for quality inspection, employing YOLOv5s and Faster R-CNN to classify LMD-wire printed part defects, achieving detection accuracies of 0.932 and 0.872, respectively. That model was trained to identify five distinct defect classes (Balling, Dripping, Necking, Overbuilding, Stubbing). However, the pipeline proposed in this paper differs significantly from these applications, as they typically do not integrate DT data for comparative analysis. Moreover, the evaluation of metal AM parts using 3D point cloud surface analysis is not a widely established method in the current state-of-the-art for metal AM QA.

As with any analytical framework, the one proposed here has scope limitations. Although the application of the framework in other geometries is completely valid, as the data transformation pipeline minimizes the influence of global surface properties on the evaluation of segments, the validation was conducted exclusively on tubular hollow parts. Consequently, to achieve better generalization and results on different shapes and materials, the model would require training on a more diverse dataset, incorporating segments from various deposition methods and part topologies. Future work will be limited to hollow parts but will aim to expand the Fiedler number-based classification framework by training the model on hollow cubes, slopes, and pyramid trunks, which will also allow for the exploration of different DNN backbones.

A further limitation is that the framework was validated only using the CW-GTAW process. Although it is intended as a generalized framework for evaluating the surface roughness of metal AM parts, its applicability must be confirmed on parts manufactured with other processes, such as DED LMD-wire or the more common GMAW process.

5.2 General evaluation of surface quality

In order to properly evaluate the developed methods of surface roughness classification and their application to metal AM deposited parts and their scanned 3D models, it is necessary to apply external metrics and evaluations outside of the proposed framework itself, comparing those to the feature extraction methods of the Digital Twin-based ETL pipeline. Also, another plausible discussion is the limitations of the photogrammetry-based 3D scanning and the sampling of the Digital Twin itself when compared to the real deposited part and the 3D models.

On the evaluation process of the deposited parts themselves, establishing a reliable "ground truth" for training a deep neural network like PointNet is a primary challenge in quality assurance, as the model's performance is entirely dependent on the quality of its training labels. Relying solely on human visual inspection to label surface quality is inherently subjective and prone to bias. This qualitative approach can introduce significant analytical errors and inconsistencies into the training dataset, as different inspectors may grade the same defect differently, or their perception may shift over time. This subjectivity makes it difficult to validate the model's performance robustly or apply it to new scenarios, as the baseline for "good" or "poor" quality is not standardized. The proposed methodology, therefore, adopts a more quantitative and objective approach by using an unsupervised learning step. This method uses the Fiedler number, a spectral graph metric that quantifies surface connectivity, to create a numerical, objective feature for each 3D segment. These features are then grouped into quality classes using a K-means clustering algorithm, providing a repeatable and data-driven ground truth that mitigates the human bias inherent in visual-only labeling.

While this Fiedler number-based clustering provides a more objective and scalable foundation for training, it does not entirely eliminate the need for preliminary human oversight. The effectiveness of the K-means clustering algorithm is dependent on the variety of data provided; it must have a sufficient number of segments from each potential quality class to form meaningful, distinct clusters. If all deposited parts were of "fair" quality, the algorithm would still attempt to create "good" and "poor" labels, but they would not be representative. Therefore, a preliminary visual inspection is still valuable, not for labeling individual segments, but for ensuring that the overall batch of deposited parts (e.g., the four experimental cylinders) contains a diverse range of surface qualities. This combined approach mitigates the most significant biases of visual inspection, improves the accuracy and distribution of the training classes, and enables the Fiedler approach to standardize the classification process as new, diverse parts are added to the dataset.

5.2.1 Application of PointNet on simulated x real data augmented segments for classification of surface roughness

A core challenge in this research was the establishment of a reliable "ground truth" for training the PointNet model, a problem that is evident when comparing the project's initial and evolved methodologies. The first approach, documented in the FAIM conference paper (CABRAL, J. V. A.; CUNHA FACCIOLLI, et al., 2026), served as an initial proof-of-concept but was built on a significant potential bias: the use of simulated data for training. This model was trained on 3D meshes with five levels of artificially generated defects and achieved a high accuracy of 85.99% on its simulated test set. However, when validated against three real deposited parts, this model could not identify any segments as "Excellent" and was limited

to classifying them as "Fair" or "Bad". This strongly suggests that the features learned from the "perfect" simulated environment were not robust and did not generalize to the noise and variability of real-world parts, biasing the model against higher-quality classifications.

This critical limitation led to the improved methodology presented in the second approach published at the Sensors MDPI journal (CABRAL, J. V. A.; ALVARES, A. J., et al., 2026), which was designed to remove this simulation bias by training the model on real experimental data. The most significant methodological advancement was the creation of an objective, data-driven ground truth. Instead of relying on subjective manual labels or artificial defects, this new approach introduced an unsupervised learning step. All 96 segments from the four real deposited parts were first analyzed using a spectral graph metric, the Fiedler number, to numerically quantify their surface connectivity. A K-means clustering algorithm then grouped these segments into "good," "fair," and "poor" classes. This unsupervised approach provided an objective, repeatable, and non-subjective basis for labeling the real-world data, directly correcting the primary flaw of the initial work.

The methodological evolution also extended to the data preparation itself. The initial study utilized a segmentation based on height, which is less effective at analyzing vertical defects. The second methodology replaced this with a radial (azimuthal) slicing. This new method, which creates segments at a 15-degree angle, capturing the part's entire height, proved more effective for evaluating continuity and defects across multiple deposited layers. Furthermore, to create a more robust model from the 96 available real segments, the new pipeline incorporated randomized data augmentation (rotation, scaling, and jittering), a step not present in the initial study, to improve the model's ability to generalize and improve overall fitness on the real deposited parts, as the simulated ones weren't able to fully represent the subtle variations on surface features and geometrical characteristics observed on the real parts. Further improvements on the methodology of surface roughness generation aligned with other methods of simulated defects placements could solve this, but as of the limitations of the scope and methodology presented throughout this work, the simulation of parts wasn't able to generate a good generalization and overall fitness of the real models, specially on better quality parts, which indicates a heavy bias on classification of poorer quality parts.

The results of these two approaches highlight the trade-off between superficial accuracy and practical utility. The first approach's 90.62% accuracy on "real data" was high but brittle, as it was only solving a limited, two-class problem ("Fair" vs. "Bad"). In contrast, the second approach, with an overall accuracy of 75.64%, appears lower but represents a more challenging and realistic three-class problem trained on noisy, diverse, and objectively-labeled real-world data. The volatility observed during the second training model is an honest reflection of this task's complexity, which the simulated data had artificially hidden.

Ultimately, this comparison demonstrates that the initial, simulation-based model was an insufficient solution. It performed well in its artificial environment but failed to

capture the nuances of the real CW-GTAW process. The second model, while having a lower aggregate accuracy, is paradoxically far more useful. It is a robust, practical tool trained on the actual variability of the manufacturing process. Its exceptional performance in detecting critical failures—achieving a Precision of 0.97 and a Recall of 0.92 for the "Poor" quality class—is the key requirement for any industrial QA system. This demonstrates that the Fiedler number-based ground truth, while leading to a more complex training process, produces a final model that is verifiably more reliable for its intended industrial application.

5.3 Limitations of the implemented framework

Like every analysis framework, the one proposed in this Master's dissertation has its own scope limitations. In this section, those limitations are discussed from the perspective of the analytical process, the experimental apparatus/parameters, and the deposition process covered.

A first limitation concerns the definition of ground truth on the PointNet-based classification pipeline. The classification pipeline relied on spectral graph connectivity metrics (Fiedler number) and unsupervised clustering to generate proxy labels for surface quality, which cannot always be directly correlated to surface roughness quality levels on deposited parts. The resolution of the photogrammetry scanning process also imposes constraints. With a spatial resolution on the order of millimeters, the scans are well-suited for capturing global morphology and connectivity defects, but they cannot resolve fine-scale roughness features at the micrometer level.

Dataset scale and diversity represent another limitation. The experimental campaign was limited to four hollow parts fabricated under CW-GTAW WAAM conditions, resulting in 96 radial segments. While sufficient for a proof of concept, this dataset is too small to guarantee generalization across different geometries, deposition strategies, or materials. The volatility observed in training metrics reflects this limited sample diversity. Expanding the dataset to include more parts, a wider range of deposition parameters, and different AM processes would be necessary to strengthen statistical confidence and model robustness.

Another limitation is the model's inability to evaluate key deposition efficiency metrics, such as dimensional accuracy or material usage. This is an intentional design choice, as the radial segmentation process, along with vectorization, normalization, and compression, was developed to mitigate the influence of general geometric properties, like the part's overall diameter or curvature, in order to isolate the surface roughness texture. As a result, the pipeline in its current form cannot be used to measure the effective deposited wall thickness or the precise cross-sectional area of each layer. This "deposition efficiency" is a crucial metric in WAAM for analyzing process stability, calculating operational costs, and predicting material consumption. Furthermore, the photogrammetry scanning method, while excellent

for external surfaces, does not capture the internal cross-sectional geometry, which is where these efficiency metrics are most accurately and usually measured (ALDALUR et al., 2020).

Contrary to the entire framework developed through this work, it is known that the most common applications of AI on the evaluation of superficial roughness quality levels and the detection of manufacturing defects on metal AM process analysis are based on image inputs to CNNs (LEE et al., 2021). Having that in mind, a scope-related limitation of the proposed methodology is its specific focus on quantifying surface roughness via topological connectivity. The Fiedler number-based approach excels at assessing the overall continuity of the 3D surface graph but is not designed to identify or classify other significant, primarily visual defects. Superficial anomalies such as discoloration, minor porosity, or spatter, which may not substantially alter the 3D graph connectivity, are therefore not captured by this analysis. These types of defects are often better suited for 2D image-based inspection methods. For example, approaches using Convolutional Neural Networks (CNNs) like YOLO can be trained to detect and classify a wide range of specific visual anomalies from photographs of the deposited part, as demonstrated in related work on defect classification (ALVARES, A. J. et al., 2025). The current pipeline should therefore be seen as a complement to, rather than a replacement for, these image-based QA systems.

A further initial limitation of the proposed framework is the usage of a simple geometry to validate it; it is possible that complex geometries, such as gears or parts with tight tolerances with many edges close to one another (or tight curvatures) might require another previous subsegmentation, depending on the case or separation of region of interest using CNN-based algorithms such as YOLO for preprocessing of the input images.

5.4 Photogrammetry on iPhone 16 Pro Max x NextEngine 2020i 3D laser scanner

The NextEngine 2020i desktop 3D laser scanner has a better resolution (0.1 mm) when compared to the high-resolution photogrammetry scanning from the PolyCam app on an iPhone 16 Pro Max (0.3 mm); however, results shown in subsection 4.5.6 suggest that there is not a substantial difference on quality of scanning and point clouds/meshes generation from the processing softwares when considering the obtained evaluation metrics and 3D inspection made. Hence, many advantages of the usage of the smartphone for the application of metal AM manufactured parts scanning and for the proposed pipeline framework of analysis are concluded.

Firstly, being a day-to-day carry-on device, from a factory floor to a laboratory, the highly mobile and easy-to-use device can be applied to many manufacturing cells without the necessity of adapting them to accommodate new hardware. One can simply drag it out of their pocket to scan the freshly manufactured part; in fact, the image acquisition can be made

as the part is cooling down, without the necessity of taking it out from the manufacturing table or support. Besides waiting to cool it down, one would have to transport the part to the 3d laser scanner base, and accommodate it, besides the dimensional limitations of the small base, which can't accommodate objects larger than 12 centimeters in diameter in macro mode.

Another clear advantage is the price. NextEngine 2020i has an original price of 2995 dollars (ANIWAA, 2026), whereas the iPhone 16 Pro Max 16 GB had an original retailer price of 1199 dollars (source: https://www.gsmarena.com/here_are_the_apple_iphone_16_and_iphone_16_pro_prices_around_the_world-news-64467.php), along with the yearly subscription price of PolyCam software of 99 dollars (source: https://poly.cam/pricing); therefore, NextEngine scanner's price is more than double that of the proposed solution. Hence, the low-cost requirement for the proposed framework specified in the general objectives subsection 1.2.2 is fulfilled. That is, without considering the fact that many other smartphones can offer the same camera resolution (48 MP) with a lower price, such as the Samsung Galaxy A51 (source: https://www.businessinsider.com/guides/tech/samsung-galaxy-a51-price-release-date-specs).

Furthermore, the more detailed point clouds from 3D laser scanning have more than 10 times the number of points (around 538000 points) as the 49152 points uniformly sampled from photogrammetry scans. Processing that amount of data through the proposed pipeline without downsampling would probably require a processing time of roughly 100 times (since most matrices multiplications and matrices transformation algorithms have a complexity of $O(n^2)$ and larger batch sizes would have to be used, increasing also Video RAM usage and RAM usage through the transformation part of the pipeline. In fact, using that amount of points for training could take up to 3 weeks per training routine (using the same specifications), since training routines had an average of 5 hours in duration for 500 epochs, rendering it infeasible to apply the framework on the current setup, probably medium scale in-cloud server rigs would have to be used for training of such dimensions. Thus, the drawback of resolution vs. feasibility of the project has to be evaluated when considering using raw data from hardware with such resolution; downsampling would probably be inevitable.

Having that in mind, to support a higher resolution on input, the framework must have further segmentation beforehand, probably identifying superficial defects and potential rougher regions of the surface using image-based CNN models (such as YOLO) and separating them for further evaluation on the proposed framework in this current work.

6 Conclusions

In this work, an implementation of a QA pipeline based on a Digital Twin for additive manufacturing was presented. The proposed pipeline consists of the acquisition of positional data via DT-based positional data or meshes generated through images using PolyCam 3D. Photogrammetry meshes are then aligned via the ICP method and have their density corrected via the Screened Poisson Surface Reconstruction method. After such corrections, the point clouds are segmented and classified according to a k-nearest neighbors graph constructed from the calculated Fiedler number of the segments to serve as input for training an adapted DNN model based on PointNet.

The adapted PointNet neural network model has shown an accuracy of 75.64% in classifying parts from 3D models scanned from real metal deposited parts. This demonstrates the model's ability to validate the deposition process and support the tuning of parameters and overall improvement of the Digital Twin (DT). Though it is important to highlight the model's current limitation to evaluate contiguous low superficial roughness oriented hollow parts, more variations of deposited 3D steel parts are being considered to be evaluated by the model, giving the necessary adjustments to refine its training.

Moreover, the use of PointNet in robotic GTAW AM cells aligns with the broader trend of intelligent manufacturing, where AI-driven models enhance automation and precision. By leveraging Digital Twin simulations, manufacturers can optimize deposition strategies using historical data, ensuring consistent surface quality across builds. This approach can minimize post-processing requirements, such as grinding or polishing, which are typically needed to correct surface irregularities. The ability to classify meso-scale roughness at the surface level also supports quality assurance protocols, ensuring compliance with industry standards.

The generated 3D models are then analyzed for the framework validation via superposition of planes to measure positional correspondence levels among models generated via the DT, CAD models, and photogrammetry-scanned real manufactured parts. Those correspondence levels are then evaluated against a threshold and used in the Fuzzy decision-making table presented in Figure 13 to try and determine causes for possible faults in the DT and robotic cell and errors in tolerances of manufactured parts.

Furthermore, the efficiency and cost-effectiveness of the photogrammetry 3D scans against the traditional fixed desktop alternative of a 3D laser scanner were evaluated and discussed, demonstrating that a practical application for a swift pipeline for evaluating 3D point clouds is far more plausible when using the proposed framework of photogrammetry 3D scans for training and testing.

Although the developed pipeline establishes a robust methodology for geometrical and surface quality assessment, the full potential of this Digital Twin-enabled system lies in its scalability and integration with complementary technologies. The current validation of the decision-making models paves the way for more complex, multimodal analysis and real-time interventions. Therefore, to fully bridge the gap between static post-process quality assurance and dynamic process control, several specific advancements are envisioned to expand the system's capabilities and responsiveness.

6.1 Future work

Whilst the general pipeline of evaluation of deposited parts can be improved in many ways using state-of-the-art techniques for QA classification, the following steps are proposed to further develop the general pipeline of QA for the metal AM manufacturing cell:

- Implement a YOLO-based method of defects cataloging and identification on the surface of deposited parts, complementing the already existing approach of point cloud-based surface roughness evaluation, resulting in a multimodal pipeline of evaluation of deposited parts;
- Incorporate tactile or optical profilometry to generate ISO-compliant surface roughness labels, enabling validation of the correspondence between connectivity-based quality classes and standardized roughness parameters;
- Generate 3D scans of real manufactured parts using a higher-precision handheld scanner capable of tens of micrometer-level resolution.

Finally, a significant evolution of the current project would be to transition the framework from a post-process quality assurance (QA) tool to a near real-time process monitoring and control system. This extension would involve integrating an in-situ vision sensor, such as a camera equipped with a UV filter, mounted directly on the robotic manufacturing cell. While the current pipeline relies on commanded positional data from the Digital Twin (DT) and post-process photogrammetry scans, this new approach would capture the actual deposition geometry during the process. As demonstrated in studies on vision-based sensing for arc welding (XIONG; PI; CHEN, H., 2019), such a camera setup can effectively filter the intense arc glare and allow for the real-time measurement of key bead characteristics. This would provide a high-frequency stream of data on the true molten pool width and deposited layer height, which would be stored in the DT to create a much higher-fidelity representation of the as-built part.

This augmented Digital Twin, enriched with real-time geometric data, would serve as the foundation for a closed-loop control architecture. By comparing the measured molten

pool width and layer height against the desired setpoints from the CAD model, the system could make adaptive decisions, a concept central to advanced robotic manufacturing control systems. For instance, if the vision system detects that the molten pool width is expanding due to heat accumulation, the controller could autonomously (XIONG; YU, et al., 2022) decrease the arc current or increase the travel speed. This would not only enable the system to correct for defects like overbuilding or necking as they occur — improving material efficiency and final quality — but would also provide invaluable data for the PointNet model, allowing it to draw direct correlations between near real-time geometric deviations and the final, post-process surface roughness.

In conclusion, this dissertation contributes a novel analytical pipeline for quality perception in CW-GTAW robotic additive manufacturing cells, enabled by Digital Twin integration. By combining spectral graph theory, deep learning, and fuzzy decision-making, the framework establishes a foundation for intelligent QA in metal AM. The identified limitations and the proposed future work together chart a clear research trajectory: from proof-of-concept connectivity analysis toward comprehensive, multimodal, and real-time quality assurance systems for advanced manufacturing.

References

ALDALUR, E.; VEIGA, F.; SUÁREZ, A.; BILBAO, J.; LAMIKIZ, A. Analysis of the Wall Geometry with Different Strategies for High Deposition Wire Arc Additive Manufacturing of Mild Steel. **Metals**, v. 10, n. 7, 2020. ISSN 2075-4701. DOI: 10.3390/met10070892. Available from: <https://www.mdpi.com/2075-4701/10/7/892>. Cit. on p. 125.

ALVARES, A. J.; BETANCOURTH, B. S. F.; CABRAL, J. V. A.; LACROIX, I. Automated defect classification in additive manufacturing LMD-wire using deep learning. **Journal of the Brazilian Society of Mechanical Sciences and Engineering**, v. 47, n. 9, p. 432, July 2025. DOI: 10.1007/s40430-025-05740-5. Available from: <https://doi.org/10.1007/s40430-025-05740-5>. Cit. on pp. 41, 42, 45, 121, 125.

ANIWAA. **NextEngine 3D Scanner Ultra HD**. 2026. Model: 2020i. Available from: <https://www.aniwaa.com/product/3d-scanners/nextengine-3d-scanner-ultra-hd/>. Visited on: 18 Jan. 2026. Cit. on pp. 34, 55, 126.

ANTONINI, J. 8.04 - Health Effects Associated with Welding. In: HASHMI, S.; BATALHA, G. F.; VAN TYNE, C. J.; YILBAS, B. (Eds.). **Comprehensive Materials Processing**. Oxford: Elsevier, 2014. P. 49–70. DOI: https://doi.org/10.1016/B978-0-08-096532-1.00807-4. Cit. on p. 37.

BELLINI, C.; BERTO, F.; COCCO, V. D.; IACOVIELLO, F.; MOCANU, L. P.; RAZAVI, N. Additive manufacturing processes for metals and effects of defects on mechanical strength: a review. **Procedia Structural Integrity**, v. 33, p. 498–508, 2021. 26th International Conference on Fracture and Structural Integrity. ISSN 2452-3216. DOI: https://doi.org/10.1016/j.prostr.2021.10.057. Available from: <https://www.sciencedirect.com/science/article/pii/S2452321621001529>. Cit. on p. 44.

BESL, P.; MCKAY, N. A method for registration of 3-D shapes. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 14, n. 2, p. 239–256, 1992. DOI: 10.1109/34.121791. Cit. on pp. 56, 93.

BONG KIM, D.; SHAO, G.; JO, G. A digital twin implementation architecture for wire + arc additive manufacturing based on ISO 23247. **Manufacturing Letters**, v. 34, p. 1–5, 2022. ISSN 2213-8463. DOI: https://doi.org/10.1016/j.mfglet.2022.08.008. Available from: <https://www.sciencedirect.com/science/article/pii/S2213846322001808>. Cit. on p. 19.

BOTKINA, D.; HEDLIND, M.; OLSSON, B.; HENSER, J.; LUNDHOLM, T. Digital Twin of a Cutting Tool. **Procedia CIRP**, v. 72, p. 215–218, Jan. 2018. DOI: 10.1016/j.procir.2018.03.178. Cit. on pp. 26, 31.

BOULCH, A.; MARLET, R. **POCO: Point Convolution for Surface Reconstruction**. 2022. arXiv: 2201.01831. Cit. on p. 35.

BRADLEY, P.; FAYYAD, U.; REINA, C. Scaling EM (Expectation-Maximization) Clustering to Large Databases, Apr. 2000. Cit. on p. 35.

CABRAL, J.; RODRIGUEZ, E.; ALVARES, A. Digital Twin Implementation for Machining Center Based on ISO 23247 Standard. **IEEE Latin America Transactions**, v. 21, p. 628–635, May 2023. DOI: 10.1109/TLA.2023.10130834. Cit. on p. 43.

CABRAL, J. V. A.; ALVARES, A. J.; FACCIOLLI, A. C. d. C.; CARVALHO, G. C. d. A Case Study on DNN-Based Surface Roughness QA Analysis of Hollow Metal AM Fabricated Parts in a DT-Enabled CW-GTAW Robotic Manufacturing Cell. **Sensors**, v. 26, n. 1, p. 4, 2026. ISSN 1424-8220. DOI: https://doi.org/10.3390/s26010004. Available from: <https://www.mdpi.com/1424-8220/26/1/4>. Cit. on pp. 57, 63, 123.

CABRAL, J. V. A. **Pointnet-based superficial classification of metal AM parts**. https://github.com/MASCAM/Pointnet_based_superficial_classification_of_metal_AM_parts.git. Visited on: 25 Feb. 2025. Cit. on pp. 100, 105.

CABRAL, J. V. A. **PointNet_Superficial_Roughness_Classifier_Hollow_Metal_AM_Parts**. IEEE Dataport, 2025b. DOI: 10.21227/pp5k-st77. Available from: <https://dx.doi.org/10.21227/pp5k-st77>. Cit. on p. 82.

CABRAL, J. V. A. **Desenvolvimento de um gêmeo digital para simulação, monitoramento e armazenamento de dados de processo em uma célula robótica de manufatura aditiva por deposição de metal a arco e arame**. 2023. S. 88. Monography (Trabalho de Conclusão de Curso (Bacharelado em Engenharia Mecatrônica)) – Universidade de Brasília, Brasília. Orientador: Guilherme Caribé de Carvalho. Coorientador: Alberto José Álvares. Available from: <https://bdm.unb.br/handle/10483/39750>. Cit. on pp. 43, 48, 50, 52, 63.

CABRAL, J. V. A.; ÁLVARES, A. J.; CARVALHO, G. C. de. Digital Twin Implementation for an Additive Manufacturing Robotic Cell based on the ISO 23247 Standard. **IEEE Latin America Transactions**, v. 22, p. 651–658, 8 Aug. 2024. ISSN 1548-0992. DOI: 10.1109/TLA.2024.10620386. Available from: <https://doi.org/10.1109/TLA.2024.10620386>. Cit. on p. 43.

CABRAL, J. V. A.; CUNHA FACCIOLLI, A. C. da; CARVALHO, G. C. de; ÁLVARES, A. J. Digital Twin-Enabled Quality Assurance Analysis of Metal Manufactured Parts Based on Neural Networks Applied to 3D Meshes. In: SRIHARI, K.; KHASAWNEH, M. T.; YOON, S.; WON, D. (Eds.). **Flexible Automation and Intelligent Manufacturing: The Future of Automation and Manufacturing: Intelligence, Agility, and Sustainability**. Cham: Springer Nature Switzerland, 2026. P. 82–92. ISBN 978-3-032-07675-5. Cit. on p. 122.

CAVALCANTE, I.; FRAZZON, E.; FORCELLINI, F.; IVANOV, D. A supervised machine learning approach to data-driven simulation of resilient supplier selection in digital manufacturing. **International Journal of Information Management**, v. 49, p. 86–97, Mar. 2019. DOI: 10.1016/j.ijinfomgt.2019.03.004. Cit. on pp. 25, 31.

CHEN, Y. H.; NG, C. T.; WANG, Y. Z. Generation of an STL File from 3D Measurement Data with User-Controlled Data Reduction. **The International Journal of Advanced Manufacturing Technology**, v. 15, n. 2, p. 127–131, Feb. 1999. ISSN 1433-3015. DOI: 10.1007/s001700050049. Available from: <https://doi.org/10.1007/s001700050049>. Cit. on p. 89.

CHUA, Z. Y.; AHN, I. H.; MOON, S. K. Process monitoring and inspection systems in metal additive manufacturing: Status and applications. **International Journal of Precision Engineering and Manufacturing-Green Technology**, v. 4, n. 2, p. 235–245, Apr. 2017. ISSN 2198-0810. DOI: 10.1007/s40684-017-0029-7. Available from: <https://doi.org/10.1007/s40684-017-0029-7>. Cit. on p. 45.

COSTELLO, S. C.; CUNNINGHAM, C. R.; XU, F.; SHOKRANI, A.; DHOKIA, V.; AND, S. T. N. The state-of-the-art of wire arc directed energy deposition (WA-DED) as an additive manufacturing process for large metallic component manufacture. **International Journal of Computer Integrated Manufacturing**, Taylor & Francis, v. 36, n. 3, p. 469–510, 2023. DOI: 10.1080/0951192X.2022.2162597. eprint: https://doi.org/10.1080/0951192X.2022.2162597. Available from: <https://doi.org/10.1080/0951192X.2022.2162597>. Cit. on p. 40.

DECKER, N.; WANG, Y.; HUANG, Q. Efficiently registering scan point clouds of 3D printed parts for shape accuracy assessment and modeling. **Journal of Manufacturing Systems**, v. 56, p. 587–597, 2020. ISSN 0278-6125. DOI: https://doi.org/10.1016/j.jmsy.2020.04.001. Available from: <https://www.sciencedirect.com/science/article/pii/S0278612520300455>. Cit. on pp. 27, 31, 36, 57.

DOUNGTAP, S.; PETCHHAN, J.; PHANICHRAKSAPHONG, V.; WANG, J.-H. Towards Digital Twins of 3D Reconstructed Apparel Models with an End-to-End Mobile Visualization. **Applied Sciences**, v. 13, n. 15, 2023. ISSN 2076-3417. DOI: 10.3390/app13158571. Available from: <https://www.mdpi.com/2076-3417/13/15/8571>. Cit. on p. 42.

DUTTA PRAMANIK, P.; MUKHERJEE, B.; PAL, S.; UPADHYAYA, B.; DUTTA, S. Ubiquitous Manufacturing in the Age of Industry 4.0: A State-of-the-Art Primer. In: Nov. 2019. P. 73–112. Available from: <https://doi.org/10.1007/978-3-030-14544-6_5>. Cit. on p. 19.

EAGAR, T. W. Energy Sources Used for Fusion Welding[1]. In: WELDING, BRAZING, AND SOLDERING. ASM International, Jan. 1993. ISBN 978-1-62708-173-3. DOI: 10.31399/asm.hb.v06.a0001332. eprint: https://dl.asminternational.org/

book/chapter-pdf/475430/a0001332.pdf. Available from: <https://doi.org/10.31399/asm.hb.v06.a0001332>. Cit. on p. 38.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: KNOWLEDGE Discovery and Data Mining. 1996. Available from: <https://api.semanticscholar.org/CorpusID:355163>. Cit. on pp. 35, 54.

FERNANDES, T. L.; BALDO, C. R.; DONATELLI, G. D. The concept of digital twin used to investigate geometrical variations in the production of pipe spools. **Advances in Industrial and Manufacturing Engineering**, v. 3, p. 100054, 2021. ISSN 2666-9129. DOI: https://doi.org/10.1016/j.aime.2021.100054. Available from: <https://www.sciencedirect.com/science/article/pii/S2666912921000246>. Cit. on pp. 29, 31.

FOUNTAS, N. A.; KECHAGIAS, J. D.; VAXEVANIDIS, N. M. Optimization of Selective Laser Sintering/Melting Operations by Using a Virus-Evolutionary Genetic Algorithm. **Machines**, v. 11, n. 1, 2023. ISSN 2075-1702. DOI: 10.3390/machines11010095. Available from: <https://www.mdpi.com/2075-1702/11/1/95>. Cit. on p. 44.

FU, T.; LIU, S.; LI, P. Digital twin-driven smelting process management method for converter steelmaking. **Journal of Intelligent Manufacturing**, v. 36, n. 4, p. 2749–2765, 2025. Cited by: 23; All Open Access, Green Open Access, Hybrid Gold Open Access. DOI: 10.1007/s10845-024-02366-7. Available from: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-105002913601&doi=10.1007%2fs10845-024-02366-7&partnerID=40&md5=6587d77faaeeba6292f5e9eeb849e872>. Cit. on pp. 30, 32.

GENG, J. Structured-light 3D surface imaging: a tutorial. **Advances in Optics and Photonics**, v. 3, p. 128–160, Mar. 2011. DOI: 10.1364/AOP.3.000128. Cit. on p. 82.

GHOSH, N. Comparative Analysis of Microstructural Assessment and Weld Defects in GMAW and GTAW Techniques. In: RUDAWSKA, A.; AL-NAIB, U. M. B. (Eds.). **Advances in Materials Processing - Recent Trends and Applications in Welding, Grinding, and Surface Treatment Processes**. London: IntechOpen, 2024. chap. 3. DOI: 10.5772/intechopen.1005881. Available from: <https://doi.org/10.5772/intechopen.1005881>. Cit. on p. 38.

GUIDI, G.; RUSSO, M.; MAGRASSI, G.; BORDEGONI, M. **Performance Evaluation of Triangulation Based Range Sensors**. **Sensors**, v. 10, n. 8, p. 7192–7215, Aug. 2010. DOI: \url{10.3390/s100807192}. Cit. on p. 34.

HANSARD, M.; LEE, S.; CHOI, O.; HORAUD, R. **Time-of-Flight Cameras: Principles, Methods and Applications**. London: Springer Science & Business Media, 2012. ISBN 978-1-4471-4658-2. Cit. on p. 34.

HOU, M.; LI, P.; CHENG, S.; YV, J. CNN-based defect detection in manufacturing. **Advanced Control for Applications**, n/a, n/a, e196. e196 ADC2-23-0202.R1. DOI: https://doi.org/10.1002/adc2.196. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/adc2.196. Available from: <https://onlinelibrary.wiley.com/doi/abs/10.1002/adc2.196>. Cit. on p. 44.

HUANG, J.; GOJCIC, Z.; ATZMON, M.; LITANY, O.; FIDLER, S.; WILLIAMS, F. **Neural Kernel Surface Reconstruction**. 2023. arXiv: 2305.19590. Cit. on p. 34.

ISO 23247-1. Automation systems and integration — Digital twin framework for manufacturing — Part 1: Overview and general principles, 2021. Cit. on p. 21.

JAIN, K. How Photogrammetric Software Works: A Perspective Based on UAV's Exterior Orientation Parameters. **Journal of the Indian Society of Remote Sensing**, v. 49, Nov. 2020. DOI: 10.1007/s12524-020-01256-8. Cit. on p. 33.

JIN, L.; ZHAI, X.; KANG, W.; ZHANG, K.; WU, D.; NAZIR, A.; JIANG, J.; LIAO, W.-H. Big data, machine learning, and digital twin assisted additive manufacturing: A review. **Materials & Design**, v. 244, p. 113086, June 2024. DOI: 10.1016/j.matdes.2024.113086. Cit. on p. 44.

KAZHDAN, M. M.; BOLITHO, M.; HOPPE, H. Poisson surface reconstruction. In: EUROGRAPHICS Symposium on Geometry Processing. 2006. Available from: <https://api.semanticscholar.org/CorpusID:14224>. Cit. on p. 84.

KAZHDAN, M. M.; HOPPE, H. Screened poisson surface reconstruction. **ACM Trans. Graph.**, v. 32, 29:1–29:13, 2013. Available from: <https://api.semanticscholar.org/CorpusID:1371704>. Cit. on p. 54.

KRÜCKEMEIER, S.; ANDERL, R. Concept for Digital Twin Based Virtual Part Inspection for Additive Manufacturing. **Procedia CIRP**, v. 107, p. 458–462, 2022. Leading manufacturing systems transformation – Proceedings of the 55th CIRP Conference on Manufacturing Systems 2022. ISSN 2212-8271. DOI: https://doi.org/10.1016/j.procir.2022.05.008. Available from: <https://www.sciencedirect.com/science/article/pii/S2212827122002918>. Cit. on pp. 26, 31.

LEE, J.; PARK, H. J.; CHAI, S.; KIM, G. R.; YONG, H.; BAE, S. J.; KWON, D. Review on Quality Control Methods in Metal Additive Manufacturing. **Applied Sciences**, v. 11, n. 4, 2021. ISSN 2076-3417. DOI: 10.3390/app11041966. Available from: <https://www.mdpi.com/2076-3417/11/4/1966>. Cit. on pp. 45, 125.

LI, J.; TAN, H.; ZHANG, K.; XU, Z.; LUAN, F.; XU, Y.; HONG, Y.; SUNKAVALLI, K.; SHAKHNAROVICH, G.; BI, S. **Instant3D: Fast Text-to-3D with Sparse-View Generation and Large Reconstruction Model**. 2023. arXiv: 2311.06214. Cit. on p. 35.

LI, X.; JIA, X.; YANG, Q.; LEE, J. Quality analysis in metal additive manufacturing with deep learning. **Journal of Intelligent Manufacturing**, v. 31, Dec. 2020. DOI: 10.1007/s10845-020-01549-2. Cit. on p. 45.

LIU, C.; LE ROUX, L.; KÖRNER, C.; TABASTE, O.; LACAN, F.; BIGOT, S. Digital Twin-enabled Collaborative Data Management for Metal Additive Manufacturing Systems. **Journal of Manufacturing Systems**, v. 62, p. 857–874, 2022. ISSN 0278-6125. DOI: https://doi.org/10.1016/j.jmsy.2020.05.010. Cit. on p. 42.

LIU, M.; KUMAR, A.; BUKKAPATNAM, S.; KUTTOLAMADOM, M. A Review of the Anomalies in Directed Energy Deposition (DED) Processes & Potential Solutions - Part Quality & Defects. **Procedia Manufacturing**, v. 53, p. 507–518, 2021. 49th SME North American Manufacturing Research Conference (NAMRC 49, 2021). ISSN 2351-9789. DOI: https://doi.org/10.1016/j.promfg.2021.06.093. Available from: <https://www.sciencedirect.com/science/article/pii/S2351978921001189>. Cit. on pp. 40, 65.

LIU, Q.; ZHANG, H.; LENG, J.; CHEN, X. Digital twin-driven rapid individualised designing of automated flow-shop manufacturing system. **International Journal of Production Research**, v. 57, p. 3903–3919, May 2018. DOI: 10.1080/00207543.2018.1471243. Cit. on pp. 26, 31.

MANDOLLA, C.; PETRUZZELLI, A.; PERCOCO, G.; URBINATI, A. Building a Digital Twin for Additive Manufacturing through the Exploitation of Blockchain: A case analysis of the aircraft industry. **Computers in Industry**, v. 109, p. 134–152, Apr. 2019. DOI: 10.1016/j.compind.2019.04.011. Cit. on pp. 27, 32.

MARTINA, F.; MEHNEN, J.; WILLIAMS, S.; COLEGROVE, P.; WANG, F. Investigation of the benefits of plasma deposition for the additive layer manufacture of Ti–6Al–4V. **Journal of Materials Processing Technology**, Elsevier, v. 212, n. 6, p. 1377–1386, 2012. Cit. on p. 39.

MOHAMED, O. A.; MASOOD, S. H.; BHOWMIK, J. L. Modeling, analysis, and optimization of dimensional accuracy of FDM-fabricated parts using definitive screening design and deep learning feedforward artificial neural network. **Advances in Manufacturing**, v. 9, n. 1, p. 115–129, Mar. 2021. Cit. on p. 45.

NEWCOMBE, R.; IZADI, S.; HILLIGES, O.; MOLYNEAUX, D.; KIM, D.; DAVISON, A.; KOHLI, P.; SHOTTON, J.; HODGES, S.; FITZGIBBON, A. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In: p. 127–136. DOI: 10.1109/ISMAR.2011.6092378. Cit. on p. 36.

OPEN3D TEAM. **Open3D: A Modern Library for 3D Data Processing**. 2025. Available from: <https://github.com/isl-org/Open3D>. Cit. on pp. 56, 59.

OZYESIL, O.; VORONINSKI, V.; BASRI, R.; SINGER, A. A survey of structure from motion. **Acta Numerica**, Cambridge University Press, v. 26, p. 305–364, 2017. Cit. on pp. 33, 56.

PANTELIDAKIS, M.; MYKONIATIS, K. Extending the Digital Twin Ecosystem: A real-time Digital Twin of a LinuxCNC-controlled subtractive manufacturing machine. **Journal of Manufacturing Systems**, v. 74, p. 1057–1066, 2024. ISSN 0278-6125. DOI: https://doi.org/10.1016/j.jmsy.2024.05.012. Available from: <https://www.sciencedirect.com/science/article/pii/S0278612524001080>. Cit. on pp. 27, 31.

PLATT, B. F.; HASIOTIS, S. T.; HIRMAS, D. R. **Use of Low-Cost Multistripe Laser Triangulation (MLT) Scanning Technology for Three-Dimensional, Quantitative Paleoichnological and Neoichnological Studies**. **Journal of Sedimentary Research**, v. 80, n. 7, p. 590–610, July 2010. DOI: \url{10.2110/jsr.2010.059}. Cit. on p. 34.

POLYCAM. **Polyform - A 3D Scanning and Reconstruction Software**. 2025. Accessed: 24-May-2025. Available from: <https://github.com/PolyCam/polyform>. Cit. on p. 82.

QI, C. R.; SU, H.; MO, K.; GUIBAS, L. J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. **arXiv preprint arXiv:1612.00593**, 2016. Cit. on pp. 58, 91.

RAO, P.; KONG, Z.; DUTY, C.; SMITH, R. Three Dimensional Point Cloud Measurement Based Dimensional Integrity Assessment for Additive Manufactured Parts Using Spectral Graph Theory. In: v002t04a048. DOI: 10.1115/MSEC2016-8516. Cit. on pp. 43, 93.

RAO, P.; KONG, Z.; DUTY, C.; SMITH, R.; KUNC, V.; LOVE, L. Assessment of Dimensional Integrity and Spatial Defect Localization in Additive Manufacturing (AM) using Spectral Graph Theory (SGT). **Journal of Manufacturing Science and Engineering**, Sept. 2015. DOI: 10.1115/1.4031574. Cit. on pp. 26, 32.

REISGEN, U.; SHARMA, R.; MANN, S.; OSTER, L. Increasing the manufacturing efficiency of WAAM by advanced cooling strategies. **Welding in the World**, v. 64, n. 8, p. 1409–1416, Aug. 2020. ISSN 1878-6669. DOI: 10.1007/s40194-020-00930-2. Available from: <https://doi.org/10.1007/s40194-020-00930-2>. Cit. on p. 79.

RODRIGUEZ, M. Z.; COMIN, C. H.; CASANOVA, D.; BRUNO, O. M.; AMANCIO, D. R.; RODRIGUES, F. A.; F. COSTA, L. da. **Clustering Algorithms: A Comparative Approach**. 2016. arXiv: 1612.08388. Cit. on p. 35.

RYDZI, S.; ZAHRADNIKOVA, B.; SUTOVA, Z.; RAVAS, M.; HORNACEK, D.; TANUSKA, P. A predictive quality inspection framework for the manufacturing process in the context of industry 4.0. en. **Sensors (Basel)**, MDPI AG, v. 24, n. 17, p. 5644, Aug. 2024. Cit. on p. 121.

SALES DE MATOS, B. **Desenvolvimento de software de geração de trajetórias para construção de modelos metálicos tipo casca por meio do processo GTAW com adição de arame frio**. 2022. S. 102. Projeto Final de Curso (Engenharia de Controle e Automação) – Universidade de Brasília, Brasília. Cit. on p. 64.

SAMPEDRO, G. A. R.; PUTRA, M. A. P.; ABISADO, M. 3D-AmplifAI: An Ensemble Machine Learning Approach to Digital Twin Fault Monitoring for Additive Manufacturing in Smart Factories. **IEEE Access**, v. 11, p. 64128–64140, 2023. DOI: 10.1109/ACCESS.2023.3289536. Cit. on pp. 25, 31.

SANKHYE, S.; HU, G. Machine learning methods for quality prediction in production. en. **Logistics**, MDPI AG, v. 4, n. 4, p. 35, Dec. 2020. Cit. on p. 121.

SANTOS, B. Industry 4.0: an overview. In. Cit. on p. 19.

SARMA, R.; KAPIL, S.; JOSHI, S. N. A comparative analysis of trochoidal toolpath with traditional toolpaths used in wire arc-based directed energy deposition process. **Progress in Additive Manufacturing**, Springer, v. 10, n. 2, p. 1587–1604, 2025. Cit. on p. 75.

SHENZHEN CREALITY 3D TECHNOLOGY CO., LTD. **Creality CR-Scan Otter: High-Precision Handheld 3D Scanner**. 2024. Accessed: 2026-01-16. Available from: <https://www.creality.com/products/creality-cr-scan-otter>. Cit. on p. 82.

SINGH, J.; IL'YENKO, V.; OLIVEIRA, J. P. Comparative Study of GTAW and GMAW for WAAM: Microstructural and Mechanical Insights. **Science and Technology of Welding and Joining**, v. 30, n. 4, p. 215–230, 2025. DOI: 10.1177/13621718251353015. Cit. on p. 38.

SINGH, R. 3 - Welding and joining processes. In: SINGH, R. (Ed.). **Applied Welding Engineering (Third Edition)**. Third Edition: Butterworth-Heinemann, 2020. P. 157–186. DOI: https://doi.org/10.1016/B978-0-12-821348-3.00015-X. Cit. on p. 36.

TABAR, R. S.; WÄRMEFJORD, K.; SÖDERBERG, R.; LINDKVIST, L. Efficient Spot Welding Sequence Optimization in a Geometry Assurance Digital Twin. **Journal of Mechanical Design**, v. 142, n. 10, p. 102001, Apr. 2020. ISSN 1050-0472. DOI: 10.1115/1.4046436. eprint: https://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/142/10/102001/6528380/md_142_10_102001.pdf. Available from: <https://doi.org/10.1115/1.4046436>. Cit. on pp. 29, 31.

TARIQ, U.; JOY, R.; WU, S.-H.; MAHMOOD, M. A.; MALIK, A. W.; LIOU, F. A state-of-the-art digital factory integrating digital twin for laser additive and subtractive manufacturing processes. **Rapid Prototyping Journal**, v. 29, n. 10, p. 2061–2097, Aug. 2023. ISSN 1355-2546. DOI: 10.1108/RPJ-03-2023-0113. eprint: https://www.emerald.com/rpj/article-pdf/29/10/2061/2887691/rpj-03-2023-0113.pdf. Available from: <https://doi.org/10.1108/RPJ-03-2023-0113>. Cit. on pp. 29, 31.

TASHI; ULLAH, A. S.; WATANABE, M.; KUBO, A. Analytical Point-Cloud Based Geometric Modeling for Additive Manufacturing and Its Application to Cultural Heritage Preservation. **Applied Sciences**, v. 8, n. 5, 2018. ISSN 2076-3417. DOI: 10.3390/app8050656. Available from: <https://www.mdpi.com/2076-3417/8/5/656>. Cit. on pp. 28, 42.

TOMAR, B.; SHIVA, S. Cold metal transfer-based wire arc additive manufacturing. **Journal of the Brazilian Society of Mechanical Sciences and Engineering**, v. 45, n. 3, p. 157, Feb. 2023. ISSN 1806-3691. DOI: 10.1007/s40430-023-04084-2. Available from: <https://doi.org/10.1007/s40430-023-04084-2>. Cit. on p. 37.

TOMAR, B.; SHIVA, S.; NATH, T. A review on wire arc additive manufacturing: Processing parameters, defects, quality improvement and recent advances. **Materials Today Communications**, v. 31, p. 103739, 2022. ISSN 2352-4928. DOI: https://doi.org/10.1016/j.mtcomm.2022.103739. Available from: <https://www.sciencedirect.com/science/article/pii/S2352492822005992>. Cit. on p. 92.

UNIVERSITY OF FLORIDA. **NextEngine 3D Scanning Tutorial**. https://web.mae.ufl.edu/rapidpro/pages/rapid_proto/3D_Scanner_Tutorial.pdf. Accessed: 2026-01-16. Cit. on p. 56.

VENTURINI, G.; MONTEVECCHI, F.; BANDINI, F.; SCIPPA, A.; CAMPATELLI, G. Feature based three axes computer aided manufacturing software for wire arc additive manufacturing dedicated to thin walled components. **Additive Manufacturing**, v. 22, p. 643–657, 2018. ISSN 2214-8604. DOI: 10.1016/j.addma.2018.06.013. Cit. on p. 75.

WANG, C.; SAMARI, B.; SIDDIQI, K. **Local Spectral Graph Convolution for Point Set Feature Learning**. 2018. arXiv: 1803.05827 [cs.CV]. Available from: <https://arxiv.org/abs/1803.05827>. Cit. on pp. 90, 94.

WANG, H.; KOVACEVIC, R. Rapid prototyping based on variable polarity gas tungsten arc welding for a 5356 aluminium alloy. **Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture**, Sage Publications Sage UK: London, England, v. 215, n. 11, p. 1519–1527, 2001. Cit. on p. 76.

WANG, H.; CAO, L.; LI, Y.; SCHNEIDER, M.; DETEMPLE, E.; EGGELER, G. Effect of cooling rate on the microstructure and mechanical properties of a low-carbon low-alloyed steel. **Journal of Materials Science**, v. 56, n. 18, p. 11098–11113, June 2021. ISSN 1573-4803. DOI: 10.1007/s10853-021-05974-3. Available from: <https://doi.org/10.1007/s10853-021-05974-3>. Cit. on p. 79.

WANG, J.; YE, L.; GAO, R.; LI, C.; ZHANG, L. Digital Twin for rotating machinery fault diagnosis in smart manufacturing. **International Journal of Production Research**, v. 57, p. 3920–3934, July 2019. DOI: 10.1080/00207543.2018.1552032. Cit. on pp. 25, 31.

WILLIAMS, F.; GOJCIC, Z.; KHAMIS, S.; ZORIN, D.; BRUNA, J.; FIDLER, S.; LITANY, O. **Neural Fields as Learnable Kernels for 3D Reconstruction**. 2021. arXiv: 2111.13674. Cit. on p. 34.

WU, W.; XUE, J.; XU, W.; LIN, H.; TANG, H.; YAO, P. Parameters Optimization of Auxiliary Gas Process for Double-Wire SS316L Stainless Steel Arc Additive Manufacturing. **Metals**, v. 11, n. 2, 2021. ISSN 2075-4701. DOI: 10.3390/met11020190. Available from: <https://www.mdpi.com/2075-4701/11/2/190>. Cit. on p. 44.

XIONG, J.; PI, Y.; CHEN, H. Deposition height detection and feature point extraction in robotic GTA-based additive manufacturing using passive vision sensing. **Robotics and Computer-Integrated Manufacturing**, v. 59, p. 326–334, 2019. ISSN 0736-5845. DOI: https://doi.org/10.1016/j.rcim.2019.05.006. Available from: <https://www.sciencedirect.com/science/article/pii/S0736584517304635>. Cit. on p. 128.

XIONG, J.; YU, Y.; ZHENG, S.; ZHANG, G. Arc voltage measurements for height control in pulsed arc additive manufacturing. **Measurement**, v. 191, p. 110867, 2022. ISSN 0263-2241. DOI: https://doi.org/10.1016/j.measurement.2022.110867. Available from: <https://www.sciencedirect.com/science/article/pii/S0263224122001531>. Cit. on p. 129.

XU, J.; CHENG, W.; GAO, Y.; WANG, X.; GAO, S.; SHAN, Y. **InstantMesh: Efficient 3D Mesh Generation from a Single Image with Sparse-view Large Reconstruction Models**. 2024. arXiv: 2404.07191. Cit. on p. 35.

XU, Y.; SUN, Y.; LIU, X.; ZHENG, Y. A Digital-Twin-Assisted Fault Diagnosis Using Deep Transfer Learning. **IEEE Access**, PP, p. 1–1, Jan. 2019. DOI: 10.1109/ACCESS.2018.2890566. Cit. on p. 28.

ZHANG, H.; LIU, Q.; CHEN, X.; ZHANG, D.; LENG, J. A Digital Twin-Based Approach for Designing and Multi-Objective Optimization of Hollow Glass Production Line. **IEEE Access**, v. 5, p. 26901–26911, Oct. 2017. DOI: 10.1109/ACCESS.2017.2766453. Cit. on pp. 28, 31.

ZHANG, J.; YAO, Y.; DENG, B. Fast and Robust Iterative Closest Point. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 44, p. 3450–3466, 2020. Available from: <https://api.semanticscholar.org/CorpusID:229934524>. Cit. on p. 36.

Appendix

APPENDIX A – Highlighted algorithms

A.1 Segmentation algorithms

A.1.1 Layering slicing segmentation

Code A.1 – Segmentation of positional data algorithm as the main module of the transformation part of the ETL data pipeline.

```
import os
import numpy as np
import pandas as pd
import open3d as o3d

FILENAME = 'Real_Hollow_Cylinder_30mm_2025_02_26_v2'
VALIDATION = True
if VALIDATION :
    VALIDATION_BASE_PATH = 'Validation/'
    INPUT_PATH =
       'Positional_data/Validation/Simulated/{}_simulated_positions
       ({}).csv'
    OUTPUT_PATH = 'Segmented_data/Validation/Original/{} ({}).csv'
else:
    INPUT_PATH = 'Positional_data/Simulated/{}_simulated_positions
       ({}).csv'
    OUTPUT_PATH = 'Segmented_data/Simulated/{} ({}).csv'

OUTPUT_FILENAME = FILENAME + '_segmented_positions'

def load_point_cloud(file_path):
    data = pd.read_csv(file_path).values
    return data

def save_point_cloud(data, file_path):
    df = pd.DataFrame(data, columns=['pos_x', 'pos_y', 'pos_z',
       'normal_x', 'normal_y', 'normal_z', 'predicted_class'])
    #print(df)
    df.to_csv(file_path, header=True, index=False)

def normalize_bottom_points(segment, k_neighbors=5,
   density_radius=0.005):
    """
    Normalize bottom points based on point density and adjust
       outliers.
    Preserves all columns including labels/predicted_class.
```

```
    """
    if len(segment) == 0:
        return np.zeros((2048, segment.shape[1]))

    # Create point cloud and KDTree
    pcd = o3d.geometry.PointCloud()
    pcd.points = o3d.utility.Vector3dVector(segment[:, :3])
    pcd.normals = o3d.utility.Vector3dVector(segment[:, 3:6])
    kdtree = o3d.geometry.KDTreeFlann(pcd)

    # Find points density and Z distribution
    z_values = segment[:, 2]
    min_z = np.min(z_values)
    max_z = np.max(z_values)
    z_range = max_z - min_z

    # Calculate density for bottom region
    bottom_region_height = z_range * 0.1
    bottom_points_mask = z_values <= (min_z + bottom_region_height)
    bottom_points = segment[bottom_points_mask]

    if len(bottom_points) > 0:
        # Process bottom points
        bottom_pcd = o3d.geometry.PointCloud()
        bottom_pcd.points =
           o3d.utility.Vector3dVector(bottom_points[:, :3])
        bottom_pcd.normals =
           o3d.utility.Vector3dVector(bottom_points[:, 3:6])

        # Voxel downsample bottom points for uniform density
        voxel_size = density_radius
        bottom_pcd_down =
           bottom_pcd.voxel_down_sample(voxel_size=voxel_size)

        # Get target Z from downsampled bottom points
        bottom_points_down = np.asarray(bottom_pcd_down.points)
        target_z = np.median(bottom_points_down[:, 2])

        # Adjust sparse bottom points to target Z
        for i in range(len(segment)):
            if z_values[i] <= (min_z + bottom_region_height):
                [k, _, _] =
                   kdtree.search_radius_vector_3d(segment[i, :3],
                   density_radius)
                if k < k_neighbors:
                    segment[i, 2] = target_z

    # Create point cloud from adjusted segment
    adjusted_pcd = o3d.geometry.PointCloud()
    adjusted_pcd.points = o3d.utility.Vector3dVector(segment[:,
       :3])
```

```
76      adjusted_pcd.normals = o3d.utility.Vector3dVector(segment[:,
            3:6])
77
78      # Voxel downsample the entire segment
79      voxel_size = density_radius * 2
80      downsampled_pcd =
            adjusted_pcd.voxel_down_sample(voxel_size=voxel_size)
81
82      # Convert back to numpy arrays
83      points = np.asarray(downsampled_pcd.points)
84      normals = np.asarray(downsampled_pcd.normals)
85
86      # Get labels from original points (nearest neighbor)
87      if segment.shape[1] > 6:  # If we have labels
88          labels = segment[:, 6:]  # Get all columns after position
                and normals
89
90          # Find nearest neighbors for label assignment
91          tree = o3d.geometry.KDTreeFlann(adjusted_pcd)
92          new_labels = []
93          for point in points:
94              [_, idx, _] = tree.search_knn_vector_3d(point, 1)
95              new_labels.append(labels[idx[0]])
96          new_labels = np.array(new_labels)
97
98          # Combine points, normals and labels
99          combined = np.hstack([points, normals, new_labels])
100     else:
101         combined = np.hstack([points, normals])
102
103     # Ensure exactly 2048 points
104     if len(combined) > 2048:
105         indices = np.random.choice(len(combined), 2048,
                replace=False)
106         combined = combined[indices]
107     elif len(combined) < 2048:
108         extra_needed = 2048 - len(combined)
109         extra_indices = np.random.choice(len(combined),
                extra_needed, replace=True)
110         extra_points = combined[extra_indices]
111         combined = np.vstack([combined, extra_points])
112
113     return combined
114
115 def segment_point_cloud(data, num_segments=24,
        points_per_segment=2048):
116     z_values = data[:, 2]
117     min_z, max_z = np.min(z_values), np.max(z_values)
118     segment_height = (max_z - min_z) / num_segments
119
120     segmented_data = []
121     for i in range(num_segments):
```

```
122         z_min = min_z + i * segment_height
123         z_max = z_min + segment_height
124         segment = data[(z_values >= z_min) & (z_values <
               z_max)].copy()
125 
126         if segment.shape[0] > 0:
127             # Normalize bottom points based on density
128             segment = normalize_bottom_points(segment)
129 
130             # Normalize Z positions within segment
131             segment_z_min = np.min(segment[:, 2])
132             segment[:, 2] = segment[:, 2] - segment_z_min
133 
134         segmented_data.append(segment)
135 
136     return segmented_data
137 
138 def apply_icp(source, target):
139     source_pcd = o3d.geometry.PointCloud()
140     source_pcd.points = o3d.utility.Vector3dVector(source[:, :3])
141     target_pcd = o3d.geometry.PointCloud()
142     target_pcd.points = o3d.utility.Vector3dVector(target[:, :3])
143 
144     threshold = 0.02
145     trans_init = np.eye(4)
146     reg_p2p = o3d.pipelines.registration.registration_icp(
147         source_pcd, target_pcd, threshold, trans_init,
148         o3d.pipelines.registration.
               TransformationEstimationPointToPoint())
149 
150     transformation = reg_p2p.transformation
151     source[:, :3] =
           np.asarray(source_pcd.transform(transformation).points)
152     return source
153 
154 def process_file(input_file, output_base, num_segments=24,
       points_per_segment=2048, file_number=0):
155     data = load_point_cloud(input_file)
156     segmented_data = segment_point_cloud(data, num_segments,
           points_per_segment)
157 
158     # Apply ICP alignment based on the first segmented part
159     reference_segment = segmented_data[0]
160     for i in range(1, num_segments):
161         segmented_data[i] = apply_icp(segmented_data[i],
               reference_segment)
162 
163     # Save segmented parts to separate CSV files
164     for i, segment in enumerate(segmented_data):
165         if (file_number == 0 and i == 0):
166             output_file =
                   f'Segmented_data/{VALIDATION_BASE_PATH}Original/
```

```
                {FILENAME}_segmented_positions.csv'
167         else:
168             output_file = output_base.format(OUTPUT_FILENAME,
                    file_number * num_segments + i)
169         #print(i)
170         save_point_cloud(segment, output_file)
171
172 def main():
173     os.makedirs('Segmented_data/Simulated', exist_ok=True)
174     os.makedirs('Segmented_data/Validation/Simulated',
            exist_ok=True)
175     for i in range(1):
176         #print(i)
177
178         if i == 0:
179
180             input_file =
                    f'Positional_data/{VALIDATION_BASE_PATH}Simulated/
                    {FILENAME}_simulated_positions.csv'
181
182         else:
183
184             input_file = INPUT_PATH.format(FILENAME, i)
185
186         process_file(input_file, OUTPUT_PATH, 24, 2048, i)
187
188 if __name__ == "__main__":
189     main()
```

A.1.2 Radial slicing segmentation

Code A.2 – Segmentation of positional data algorithm as the main module of the transformation part of the ETL data pipeline.

```
1  import os
2  import numpy as np
3  import pandas as pd
4  import open3d as o3d
5  import math
6
7  FILENAME = 'Real_Hollow_Cylinder_30mm_2025_02_20_v2'
8  #FILENAME = 'Basic_Hollow_Cube_main'
9  #FILENAME = 'Basic_Hollow_Cylinder_main'
10 #FILENAME = 'Basic_Hollow_Cylinder_12cmDD_30mmHH_main'
11 #FILENAME = 'Basic_Hollow_Cylinder_80mm_main'
12
13 VALIDATION = True
14 if VALIDATION :
15     VALIDATION_BASE_PATH = 'Validation/'
```

```
    INPUT_PATH =
        'Positional_data/Validation/Simulated/{}_simulated_positions
        ({}).csv'
    OUTPUT_PATH = 'Segmented_data/Validation/Original/{} ({}).csv'
else:
    INPUT_PATH = 'Positional_data/Simulated/{}_simulated_positions
        ({}).csv'
    OUTPUT_PATH = 'Segmented_data/Simulated/{} ({}).csv'
    VALIDATION_BASE_PATH = ''

OUTPUT_FILENAME = FILENAME + '_segmented_positions'

def load_point_cloud(file_path):
    data = pd.read_csv(file_path).values
    return data

def save_point_cloud(data, file_path):
    df = pd.DataFrame(data, columns=['pos_x', 'pos_y', 'pos_z',
        'normal_x', 'normal_y', 'normal_z', 'predicted_class'])
    #print(df)
    df.to_csv(file_path, header=True, index=False)

def normalize_bottom_points(segment, k_neighbors=5,
    density_radius=0.005):
    """
    Normalize bottom points based on point density and adjust
        outliers.
    Preserves all columns including labels/predicted_class.
    """
    if len(segment) == 0:
        return np.zeros((2048, segment.shape[1]))

    # Create point cloud and KDTree
    pcd = o3d.geometry.PointCloud()
    pcd.points = o3d.utility.Vector3dVector(segment[:, :3])
    pcd.normals = o3d.utility.Vector3dVector(segment[:, 3:6])
    kdtree = o3d.geometry.KDTreeFlann(pcd)

    # Find points density and Z distribution
    z_values = segment[:, 2]
    min_z = np.min(z_values)
    max_z = np.max(z_values)
    z_range = max_z - min_z

    # Calculate density for bottom region
    bottom_region_height = z_range * 0.1
    bottom_points_mask = z_values <= (min_z + bottom_region_height)
    bottom_points = segment[bottom_points_mask]

    if len(bottom_points) > 0:
        # Process bottom points
        bottom_pcd = o3d.geometry.PointCloud()
```

```
62          bottom_pcd.points =
               o3d.utility.Vector3dVector(bottom_points[:, :3])
63          bottom_pcd.normals =
               o3d.utility.Vector3dVector(bottom_points[:, 3:6])
64
65          # Voxel downsample bottom points for uniform density
66          voxel_size = density_radius
67          bottom_pcd_down =
               bottom_pcd.voxel_down_sample(voxel_size=voxel_size)
68
69          # Get target Z from downsampled bottom points
70          bottom_points_down = np.asarray(bottom_pcd_down.points)
71          target_z = np.median(bottom_points_down[:, 2])
72
73          # Adjust sparse bottom points to target Z
74          for i in range(len(segment)):
75              if z_values[i] <= (min_z + bottom_region_height):
76                  [k, _, _] =
                       kdtree.search_radius_vector_3d(segment[i, :3],
                       density_radius)
77                  if k < k_neighbors:
78                      segment[i, 2] = target_z
79
80      # Create point cloud from adjusted segment
81      adjusted_pcd = o3d.geometry.PointCloud()
82      adjusted_pcd.points = o3d.utility.Vector3dVector(segment[:,
           :3])
83      adjusted_pcd.normals = o3d.utility.Vector3dVector(segment[:,
           3:6])
84
85      # Voxel downsample the entire segment
86      voxel_size = density_radius * 2
87      downsampled_pcd =
           adjusted_pcd.voxel_down_sample(voxel_size=voxel_size)
88
89      # Convert back to numpy arrays
90      points = np.asarray(downsampled_pcd.points)
91      normals = np.asarray(downsampled_pcd.normals)
92
93      # Get labels from original points (nearest neighbor)
94      if segment.shape[1] > 6:  # If we have labels
95          labels = segment[:, 6:]  # Get all columns after position
               and normals
96
97          # Find nearest neighbors for label assignment
98          tree = o3d.geometry.KDTreeFlann(adjusted_pcd)
99          new_labels = []
100         for point in points:
101             [_, idx, _] = tree.search_knn_vector_3d(point, 1)
102             new_labels.append(labels[idx[0]])
103         new_labels = np.array(new_labels)
104
```

```
105         # Combine points, normals and labels
106         combined = np.hstack([points, normals, new_labels])
107     else:
108         combined = np.hstack([points, normals])
109 
110     # Ensure exactly 2048 points
111     if len(combined) > 2048:
112         indices = np.random.choice(len(combined), 2048,
                replace=False)
113         combined = combined[indices]
114     elif len(combined) < 2048:
115         extra_needed = 2048 - len(combined)
116         extra_indices = np.random.choice(len(combined),
                extra_needed, replace=True)
117         extra_points = combined[extra_indices]
118         combined = np.vstack([combined, extra_points])
119 
120     return combined
121 
122 def segment_point_cloud(data, num_segments=24,
    points_per_segment=2048):
123     """
124     Segment the point cloud into pie-like slices based on angular
        segments.
125     Each slice contains all points within a specific angular range
        around the Z-axis.
126     """
127     # Calculate angles for each point relative to the center (0,0)
128     x_coords = data[:, 0]
129     y_coords = data[:, 1]
130 
131     # Calculate angles in radians (atan2 gives angle from positive
        X-axis)
132     angles = np.arctan2(y_coords, x_coords)
133 
134     # Convert angles to degrees and normalize to [0, 360)
135     angles_deg = np.degrees(angles)
136     angles_deg = (angles_deg + 360) % 360
137 
138     # Calculate angular segment size
139     segment_angle = 360.0 / num_segments
140 
141     segmented_data = []
142     for i in range(num_segments):
143         # Define angular bounds for this segment
144         start_angle = i * segment_angle
145         end_angle = (i + 1) * segment_angle
146 
147         # Handle the case where the segment crosses 0 degrees
148         if start_angle == 0:
149             # For the first segment, include points at exactly 0
                degrees
```

```
            segment_mask = (angles_deg >= start_angle) &
                (angles_deg < end_angle)
        else:
            # For other segments, include points at the start
                angle but not at the end angle
            segment_mask = (angles_deg >= start_angle) &
                (angles_deg < end_angle)

        segment = data[segment_mask].copy()

        if segment.shape[0] > 0:
            # Normalize bottom points based on density
            segment = normalize_bottom_points(segment)

            # Normalize Z positions within segment
            segment_z_min = np.min(segment[:, 2])
            segment[:, 2] = segment[:, 2] - segment_z_min

        segmented_data.append(segment)

    return segmented_data

def apply_icp(source, target):
    source_pcd = o3d.geometry.PointCloud()
    source_pcd.points = o3d.utility.Vector3dVector(source[:, :3])
    target_pcd = o3d.geometry.PointCloud()
    target_pcd.points = o3d.utility.Vector3dVector(target[:, :3])

    threshold = 0.02
    trans_init = np.eye(4)
    reg_p2p = o3d.pipelines.registration.registration_icp(
        source_pcd, target_pcd, threshold, trans_init,
        o3d.pipelines.registration.
        TransformationEstimationPointToPoint())

    transformation = reg_p2p.transformation
    source[:, :3] =
       np.asarray(source_pcd.transform(transformation).points)
    return source

def process_file(input_file, output_base, num_segments=24,
   points_per_segment=2048, file_number=0):
    data = load_point_cloud(input_file)
    #print(data.shape)
    segmented_data = segment_point_cloud(data, num_segments,
       points_per_segment)

    # Apply ICP alignment based on the first segmented part
    reference_segment = segmented_data[0]
    for i in range(1, num_segments):
        segmented_data[i] = apply_icp(segmented_data[i],
           reference_segment)
```

```
195
196     # Save segmented parts to separate CSV files
197     for i, segment in enumerate(segmented_data):
198         if (file_number == 0 and i == 0):
199             output_file =
                    f'Segmented_data/{VALIDATION_BASE_PATH}Original/
200             {FILENAME}_segmented_positions.csv'
201             #output_file =
                    f'Segmented_data/{VALIDATION_BASE_PATH}Simulated/
202             # {FILENAME}_segmented_positions.csv'
203         else:
204             output_file = output_base.format(OUTPUT_FILENAME,
                    file_number * num_segments + i)
205         #print(i)
206         save_point_cloud(segment, output_file)
207
208 def main():
209     os.makedirs('Segmented_data/Simulated', exist_ok=True)
210     os.makedirs('Segmented_data/Validation/Simulated',
           exist_ok=True)
211     for i in range(1):
212         #print(i)
213
214         if i == 0:
215
216             input_file =
                    f'Positional_data/{VALIDATION_BASE_PATH}Simulated/
217             {FILENAME}_simulated_positions.csv'
218
219         else:
220
221             input_file = INPUT_PATH.format(FILENAME, i)
222
223         process_file(input_file, OUTPUT_PATH, 48, 2048, i)
224
225 if __name__ == "__main__":
226     main()
```

A.2 ICP metrics algorithm Python functions

Code A.3 – Point-to-point ICP metrics estimation Python functions for point clouds' superposition comparisons.

```
1 # Dependencies: numpy, scipy, open3d
2 import numpy as np
3 from scipy.spatial import cKDTree
4 from scipy.spatial.distance import cdist
5 import open3d as o3d
6
7
```

```
8  def calculate_rmse_nearest_neighbor(source_points: np.ndarray,
       target_ref_points: np.ndarray) -> float:
9      """RMSE between two point clouds of different sizes using
           nearest-neighbor matching.
10     Returns the RMS of distances from each source point to its
           nearest target point.
11     """
12     tree = cKDTree(target_ref_points)
13     distances, _ = tree.query(source_points, k=1)
14     return float(np.sqrt(np.mean(distances**2)))
15 
16 def calculate_hausdorff_distance(points1: np.ndarray, points2:
       np.ndarray, max_points: int = 49152) -> float:
17     """Discrete Hausdorff distance (symmetric) between two point
           clouds.
18     If a cloud exceeds max_points it is uniformly downsampled to
           that limit.
19     Note: for mesh-aware Hausdorff use point->triangle distances
           (trimesh or similar).
20     """
21     def uniform_downsample(pts, limit):
22         if len(pts) <= limit:
23             return pts
24         step = len(pts) / limit
25         idx = np.round(np.arange(0, len(pts), step)).astype(int)
26         idx = np.clip(idx, 0, len(pts) - 1)
27         idx = np.unique(idx)[:limit]
28         return pts[idx]
29 
30     p1 = uniform_downsample(points1, max_points)
31     p2 = uniform_downsample(points2, max_points)
32 
33     # pairwise distances (may be memory heavy if both are large;
           downsampling mitigates)
34     D = cdist(p1, p2, metric='euclidean')
35     min_1_to_2 = np.min(D, axis=1)
36     min_2_to_1 = np.min(D, axis=0)
37     return float(max(np.max(min_1_to_2), np.max(min_2_to_1)))
38 
39 def run_icp_and_metrics(source_points: np.ndarray, target_points:
       np.ndarray,
40                         max_iterations: int = 50, tolerance: float
                               = 1e-6) -> dict:
41     """Run point-to-point ICP (Open3D) and compute metrics.
42     Returns a dict with transformation, ICP fitness/inlier_rmse,
           nearest-neighbor RMSE,
43     Hausdorff distance, iteration count, and aligned point clouds
           (Open3D objects).
44     Important: RMSE is computed via nearest-neighbor after
           alignment (KD-tree).
45     """
46     # Build Open3D point clouds
```

```
47      src_pcd = o3d.geometry.PointCloud()
48      src_pcd.points = o3d.utility.Vector3dVector(source_points)
49      tgt_pcd = o3d.geometry.PointCloud()
50      tgt_pcd.points = o3d.utility.Vector3dVector(target_points)
51
52      # adaptive correspondence distance (10% of average scale or at
           least 1.0)
53      src_scale = np.max(np.ptp(source_points, axis=0))
54      tgt_scale = np.max(np.ptp(target_points, axis=0))
55      avg_scale = (src_scale + tgt_scale) / 2.0
56      max_corr = max(avg_scale * 0.1, 1.0)
57
58      # iterative single-step ICP loop to track convergence
59      current = o3d.geometry.PointCloud(src_pcd)
60      cumulative_T = np.eye(4)
61      prev_fitness = -1.0
62      prev_rmse = float('inf')
63      step_result = None
64      iterations = 0
65
66      for _ in range(max_iterations):
67          step_result = o3d.pipelines.registration.registration_icp(
68              current, tgt_pcd, max_corr,
69              o3d.pipelines.registration.Transformation
70  EstimationPointToPoint(),
71              o3d.pipelines.registration.ICPConvergenceCriteria(
72                  relative_fitness=1e-6, relative_rmse=1e-6,
                       max_iteration=1
73              )
74          )
75          iterations += 1
76          cumulative_T = step_result.transformation @ cumulative_T
77          current.transform(step_result.transformation)
78
79          fitness_change = abs(step_result.fitness - prev_fitness)
80          rmse_change = abs(step_result.inlier_rmse - prev_rmse)
81          if fitness_change < tolerance and rmse_change < tolerance:
82              break
83          prev_fitness = step_result.fitness
84          prev_rmse = step_result.inlier_rmse
85
86      if step_result is None:
87          raise RuntimeError("ICP failed to produce a result")
88
89      # apply final transform to original source and compute
           nearest-neighbor RMSE
90      src_pcd.transform(cumulative_T)
91      aligned = np.asarray(src_pcd.points)
92
93      # nearest-neighbor RMSE (aligned source -> dense target)
94      tree = cKDTree(target_points)
95      dists, _ = tree.query(aligned, k=1)
```

```
    nn_rmse = float(np.sqrt(np.mean(dists**2)))

    # Hausdorff (discrete, symmetric)
    haus = calculate_hausdorff_distance(aligned, target_points)

    return {
        "transformation": cumulative_T,
        "fitness": step_result.fitness,
        "inlier_rmse": step_result.inlier_rmse,
        "rmse": nn_rmse,
        "hausdorff_distance": haus,
        "iterations": iterations,
        "aligned_source_pcd": src_pcd,
        "target_pcd": tgt_pcd
    }
```

APPENDIX B – Research Dissemination Through Peer-Reviewed Publications During the Master's Program

B.1 Journal Articles

J. V. A. Cabral, A. J. Alvares, A. C. da C. Facciolli and G. C. de Carvalho, **"A Case Study on DNN-Based Surface Roughness QA Analysis of Hollow Metal AM Fabricated Parts in a DT-Enabled CW-GTAW Robotic Manufacturing Cell"**, Sensors, vol. 26, no. 1, p. 4, 2026. DOI: 10.3390/s26010004 .

A. J. Alvares, B. S. F. Betancourth, J. V. A. Cabral and I. Lacroix, **"Automated defect classification in additive manufacturing LMD-wire using deep learning"**, Journal of the Brazilian Society of Mechanical Sciences and Engineering, vol. 47, no. 9, p. 432, 2025. DOI: 10.1007/s40430-025-05740-5 .

J. V. A. Cabral, A. J. Álvares and G. C. de Carvalho, **"Digital Twin Implementation for an Additive Manufacturing Robotic Cell based on the ISO 23247 Standard"**, IEEE Latin America Transactions, vol. 22, no. 8, pp. 651–658, 2024. DOI:10.1109/TLA.2024.10620386 .

B.2 Book Chapters

J. V. A. Cabral, A. C. da C. Facciolli, G. C. de Carvalho and A. J. Álvares, **"Digital Twin-Enabled Quality Assurance Analysis of Metal Manufactured Parts Based on Neural Networks Applied to 3D Meshes"**, in Flexible Automation and Intelligent Manufacturing: The Future of Automation and Manufacturing: Intelligence, Agility, and Sustainability, K. Srihari, M. T. Khasawneh, S. Yoon, and D. Won, Eds., Cham: Springer Nature Switzerland, 2026, pp. 82–92. DOI: 10.1007/978-3-032-07675-5_9 .

APPENDIX C – Undergraduate and Scientific Initiation Program Scientific Production and Achievements

C.1 Journal Articles

J. V. A. Cabral, E. A. R. Gasca and A. J. Álvares, **“Digital Twin Implementation for Machining Center Based on ISO 23247 Standard”**, IEEE Latin America Transactions, vol. 21, no. 5, pp. 628–635, 2023. DOI: 10.1109/TLA.2023.10130834.

C.2 Conference Papers

J. V. A. Cabral, L. Oliveira and A. J. Álvares, **“Desenvolvimento Digital Twin para Simulação e Otimização Processo de uma Plataforma de Manufatura Aditiva Robotizada”**, in 12th Brazilian Congress on Manufacturing Engineering (COBEF 2023), Brasília: ABCM, 2023. DOI: 10.26678/ABCM.COBEF2023.COF23-0180 .

J. V. A. Cabral, E. A. R. Gasca and A. J. Álvares, **“Desenvolvimento de um Sistema Digital Twin para o Centro de Usinagem Haas Mini Mill”**, in 74ª Reunião Anual da SBPC, Brasília: SBPC, 2022, p. 1. URL: https://reunioes.sbpcnet.org.br/74RA/inscritos/resumos/1126_11a567fa4809642059401b2e3fbf424e9.pdf .

C.3 Achievements

Nominee for the Scientific Initiation Highlight Award at the 27th UnB Scientific Initiation Congress and 18th DF Scientific Initiation Congress. Invited and accepted into "74ª Reunião Anual da Sociedade Brasileira para o Progresso da Ciência (SBPC)".