Universidade do Minho
Escola de Engenharia

Diogo Aires Gonçalves Ribeiro

An Intelligent Decision Support System for Industrial Screw Tightening

October, 2023

An Intelligent Decision Support System for Industrial Screw Tightening
Diogo Ribeiro

UMinho | 2023

Universidade do Minho
Escola de Engenharia

Diogo Aires Gonçalves Ribeiro

An Intelligent Decision Support System for Industrial Screw Tightening

Doctorate Thesis
Doctorate in Advanced Engineering Systems for Industry (AESI)

Work developed under the supervision of:

Professor Paulo Cortez

October, 2023

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.1.7) (Lourenço, 2021).

Acknowledgements

As I reach the conclusion of this momentous and protracted expedition in my life, I extend my profound gratitude to all those individuals who have contributed in some way to the completion of this doctoral thesis. I convey my heartfelt thanks to each and every one of them.

Above all, I offer my deepest appreciation to my supervisor, Paulo Alexandre Ribeiro Cortez, for your motivation, dedication, invaluable guidance, boundless patience, and unwavering support throughout these years. Working with you as a Doctor of Philosophy (PhD) student has been an absolute delight, and I am truly thankful for the immense wisdom and enriching experiences gained under your tutelage.

I extend my heartfelt gratitude to Bosch Automotive Electronics, Portugal (AE/P) for providing the necessary resources and environment for conducting this thesis. A truly exceptional appreciation goes to Jaerder Sousa, Guilherme Moreira, Rui Costa for their availability, invaluable assistance, and support. I would like to express my gratitude to all the professors and colleagues of the PhD Program in Advanced Engineering Systems for Industry (AESI). Their contributions, insights, and collaboration have been instrumental in shaping my research and academic growth.

On a personal level, I extend my heartfelt and genuine appreciation to my family, particularly my parents Elisa Gonçalves and Aires Ribeiro, for their trust and unconditional support. Their belief in me has been a constant source of motivation and strength throughout this journey. To my sister Ana Ribeiro, I offer a special thanks for igniting the competitive spirit within me, pushing me to strive for excellence and always endeavor to be the best version of myself. I would also like to extend my gratitude to Ana Soares and Lourenço Alves. I would also like to express my deepest gratitude to my parents-in-law Ana Trigo and Emidio Trigo, as well as my sister-in-law Joana Trigo. Their presence in my life has been a constant source of inspiration, urging me to explore uncharted territories and reach for ever greater heights.

To my beloved wife, Margarida Trigo, who has been by my side for the past 16 years, I extend my deepest gratitude. Without you, none of the achievements and milestones in this journey would have been possible or even meaningful. Your abiding support, care, and patience throughout this immense endeavor have been immeasurable. Thank you for understanding the demands and challenges that came with pursuing this path and for being my rock during moments of uncertainty and fatigue. Your love and encouragement have fueled my determination to persevere. Now, as I reach this significant milestone, I am committed to reciprocating the immense support and sacrifices you have made. It is my turn to be there for you, to support your dreams, and to ensure that our shared future is filled with joy, love, and fulfillment.

I am deeply grateful to my incredible circle of friends who have steadfastly believed in my capacity to embark on this endeavor. In no particular order, I want to express my appreciation to Ana Leitão, Carla Gomes, Carlos Costa, Carina Andrade, Catarina Moura, Catarina Trindade, César Cerdeira, Cristiana Gomes, Elisabete Antunes, Francisco Dias, Hossein Babaee, Joel Gonçalves, João Galvão, Julio Barros, Luis Fernandes, Miguel Silva, Nuno Silva, Pedro Martins, Raquel Ferreira, Rita Costa, Rita Ferreira, Roberto Laranjeira, Sandra Fernandes, Sahar Taki, Sofia Moreira, and Susana Santos. To each and every one of you, thank you for your devoted support and love.

Last but certainly not least, I want to express my sincere appreciation to Deeper Insights, the company where I have had the privilege of working as a Machine Learning Software Engineer for the past two years. Each passing day has brought new learnings and insights from the brilliant minds that make up this organization. However, I must give special thanks to Catarina Carvalho, Clare Walsh, Cláudio Sá, Jack Hampson, Letícia Fernandes, Rafael Almeida, Simon Smith and Sónia Marques for their unconditional support throughout this transformative journey. Your continuous guidance, encouragement, and camaraderie have played a vital role in my growth and success. Working alongside you has been an absolute pleasure, and I am grateful for the collaborative and inspiring environment we have created.

This thesis is dedicated to my grandfather, whose premature departure leaves a void in my life. I am certain that he would have taken great pride in my achievements.

This work has been supported by **FCT** – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020 and the doctoral scholarship grant: PD/BDE/135105/2017. Moreover, I would like to thank to Lourenço, 2021 for providing the *novathesis* LaTeX template for free.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the Universidade do Minho.

______________________, ______________________________

(Place) (Date)

____

(Diogo Aires Gonçalves Ribeiro)

“*It does not matter how slowly you go as long as you do not stop.*”

— Confucius

“*Surround yourself with people who believe in your dreams, encourage your ideas, support your ambitions, and bring out the best in you.*”

— Roy T. Bennett, The Light in the Heart

Resumo

Um sistema inteligente de apoio à decisão para aparafusamentos industriais

No contexto da Industry 4.0 (I4.0), o aumento exponencial de dados coletados pelas indústrias potencia a utilização de técnicas baseadas em dados para inspeção de qualidade dos seus processos produtivos. Este estudo foca-se na deteção de processos de aparafusamento anómalos, uma tarefa crucial neste contexto. Devido ao alto custo de etiquetagem manual destes dados, exploramos unicamente abordagens não supervisionadas. O objetivo principal deste trabalho é o desenvolvimento de um Intelligent Decision Support System (IDSS) baseado em técnicas de *Machine Learning (ML)* capazes de prever o sucesso de um processo de aparafusamento, aproveitando dados extraídos de um *Big Data Warehouse (BDW)*, alimentado por máquinas de assemblagem presentes no chão de fábrica de uma renomeada fábrica de peças automotivas. Realizou-se um estudo exploratório para extrair características distintas do conjunto de dados inicial e avaliou-se a sua aplicabilidade em vários modelos de *Machine Learning (ML)*, tipicamente utilizados para deteção de anomalias. Devido à natureza tipicamente desbalanceada dos dados, esta investigação foca-se em métodos não supervisionados de *One-Class Classification (OCC)*. Para validar o desempenho dos modelos inicialmente obtidos, alargou-se a avaliação a um conjunto de dados mais extenso e diversificado. Os resultados conseguidos foram altamente convincentes e demonstram um desempenho competitivo quando comparados com abordagens tradicionais, tais como catálogos de defeitos. De modo a providenciar Explainable Artificial Intelligence (XAI) nestes processos, desenvolveu-se uma ferramenta de visualização que possibilita aos operadores observar regiões anómalas com maior detalhe e estabelecer limites de deteção para estas anomalias. Os melhores modelos foram integrados num ambiente que conecta processos de *ML* e o *Big Data Warehouse (BDW)* da fábrica. Esta integração facilita o armazenamento de previsões dos modelos de *ML*, sendo ainda possível avaliar a sua evolução em tempo real.

Palavras-chave: Anomaly Detection, Big Data (BD), Industry 4.0 (I4.0), Intelligent Decision Support System (IDSS), One-Class Classification (OCC), Unsupervised Learning, Explainable Artificial Intelligence

(XAI)

Abstract

An Intelligent Decision Support System for Industrial Screw Tightening

Within the context of Industry 4.0 (I4.0), the demand for quality assessment procedures utilizing data-driven techniques has intensified due to the exponential growth in production data. This study focuses on addressing the detection of abnormal screw tightening processes, a crucial task within the industrial domain. Given the high cost associated with manual labeling efforts, the study concentrates on unsupervised approaches. The main objective of this study is the development of a Machine Learning (ML)-based Intelligent Decision Support System (IDSS), capable of predicting the success or failure of an assembly process. This work revolves around constructing an Intelligent Decision Support System (IDSS) by leveraging the Big Data (BD) extracted from the Big Data Warehouse (BDW). This valuable data originates from the assembly machines located on the ground floor of a renowned automotive parts factory plant. Initially, an exploratory study was undertaken to evaluate the possible features that could be extracted from the extensive dataset being collected. The objective was to examine their correlation with various types of ML models, with the aim of addressing the anomaly detection problem. Given the specific type of data available, which exhibits an unbalanced nature, the exploration involved several unsupervised One-Class Classification (OCC) methods. To validate the performance of the best-performing models, we then conducted an evaluation using a larger and more diverse dataset. The results obtained were competitive with traditional approaches, including defect catalogs. In order to provide Explainable Artificial Intelligence (XAI) screw tightening knowledge, we developed, we developed a visualization tool. This tool empowers operators to visualize anomalous regions more effectively and establish distinct thresholds for anomaly detection. Furthermore, the best-performing models were seamlessly integrated into a unified environment that bridges ML processes and the BDW. This integration facilitates data storage and enables the real-time monitoring of ML models within an industrial context.

Keywords: Anomaly Detection, Big Data (BD), Industry 4.0 (I4.0), Intelligent Decision Support System (IDSS), One-Class Classification (OCC), Unsupervised Learning, Explainable Artificial Intelligence (XAI)

Contents

List of Figures

List of Tables

Acronyms

AE	Autoencoder *(pp. xii, xiii, 4, 5, 27, 31, 37, 38, 43, 48–52, 54–58, 66, 68, 69)*
AE/P	Automotive Electronics, Portugal *(pp. iii, 2, 9, 10, 21, 36)*
AI	Artificial Inteligence *(pp. 10, 11, 15)*
ANN	Artificial Neural Network *(p. 38)*
ANSI	American National Standards Institute *(p. 18)*
AO	Analytical Object *(pp. 59–62)*
API	Application Programming Interface *(pp. 37, 62)*
ARD	Anomaly Region Detection *(p. 2)*
AUC	Area Under the ROC Curve *(pp. xiii, 27, 31–33, 53–58)*
BA	Business Analytics *(p. 14)*
BD	Big Data *(pp. vii, ix, xii, 1, 4, 10, 33–37, 43, 45, 57–59, 68, 69)*
BDA	Big Data Analytics *(pp. 3, 58, 69)*
BDW	Big Data Warehouse *(pp. vii, ix, 3, 58–61, 66, 68, 69)*
BI	Business Intelligence *(p. 14)*
BI-A	Business Intelligence and Analytics *(p. 14)*
BN	Batch Normalization *(pp. xii, 51, 52)*
BP	Business Process *(pp. 60–62)*
CAO	Complementary Analytical Objects *(p. 60)*
CART	Classification and Regression Tree *(p. 30)*
CAZ	Clamp-Affected Zone *(p. 18)*
CPS	Cyber-Physical Systems *(pp. 10–13, 39)*

CPU Central Processing Unit *(p. 36)*
CRISP-DM Cross-Industry Standard Process for Data Mining *(pp. xii, 25, 27, 28)*

DaL Data Lake *(pp. 59, 60)*
DGI Decrease Gini Impurity *(p. 30)*
DOF Degrees Of Freedom *(pp. 20, 21)*
DSRM Design Science Research Methodology *(pp. xii, 10)*
DSRM-IS Design Science Research Methodology for Information Systems *(pp. 7, 26, 39)*
DSS Decision Support System *(pp. xii, 13–15)*
DT Decision Tree *(p. 28)*

EDA Exploratory Data Analysis *(p. 26)*
EER Equal Error Rate *(pp. 27, 53, 57)*
EIS Executive Information Systems *(p. 14)*
ETL Extract, Transform and Load *(pp. 44, 68)*

FN False Negatives *(pp. 32, 33)*
FP False Positives *(p. 32)*
FPGA Field Programmable Gate Arrays *(p. 36)*
FPR False Positive Rate *(p. 32)*

GoF Good or Fail *(pp. xiii, 2, 27, 44, 45, 62, 63, 65, 68)*
GPU Graphics Processing Unit *(pp. 36–38, 53)*

I-CPS Industrial Cyber-Physical Systems *(pp. 11, 13)*
I-IoT Industrial Internet of Things *(pp. xii, 10–13)*
I4.0 Industry 4.0 *(pp. vii, ix, 4, 10, 11, 39, 40, 42, 68)*
IDSS Intelligent Decision Support System *(pp. vii, ix, 3, 4, 13–15, 27, 39, 68)*
IF Isolation Forest *(pp. xiii, 4, 5, 27, 29, 37, 43, 48–50, 54–58, 66, 68, 69)*
IoT Internet of Things *(pp. 11, 33)*
IS Information Systems *(pp. 7, 13, 39)*
IT Information Technology *(pp. 1, 7, 68)*
iTree Isolation Tree *(pp. 29, 50)*

KNN K-Nearest Neighbors *(p. 50)*

LOF Local Outlier Factor *(pp. 27, 29, 37, 48–50, 54, 66, 68)*

LR	Logistic Regression *(p. 28)*
LSTM	Long Short-Term Memory *(pp. 38, 69)*
M2M	Machine-to-Machine *(p. 12)*
MAE	Mean Absolute Error *(pp. 27, 51)*
ML	Machine Learning *(pp. vii, ix, xii, 1–4, 6, 8, 9, 15, 27–31, 34–37, 42, 43, 48, 53, 58–62, 68, 69)*
MO	Materialized Object *(pp. xiii, 62)*
OC-SVM	One-class Support Vector Machine *(pp. 27, 29, 30, 37)*
OCC	One-Class Classification *(pp. vii, ix, 3, 28, 29, 68)*
PCA	Principal Component Analysis *(pp. 48, 49)*
PCB	Printed Circuit Board *(pp. 43, 68)*
PhD	Doctor of Philosophy *(pp. iii, 3, 15, 24, 28, 36, 39, 68)*
PPV	Positive Predictive Value *(p. 32)*
RBF	Radial Basis Function *(p. 38)*
RDD	Resilient Distributed Dataset *(p. 37)*
ReLU	Rectified Linear Unit *(p. 51)*
RF	Random Forest *(pp. 27–30, 49, 52, 54, 66, 68)*
RFID	Radio Frequency-based Identification *(p. 12)*
RO	Research Objectives *(pp. 2–4, 24, 27, 29)*
ROC	Receiver Operating Characteristic *(pp. xiii, 27, 31–33, 48, 53, 55)*
SCARA	Selective Compliance Assembly Robot Arm *(pp. 20, 21)*
SDN	Software-Defined Networking *(p. 12)*
SGD	Stochastic Gradient Descent *(p. 51)*
SVM	Support Vector Machine *(pp. 28–30, 38)*
TN	True Negatives *(p. 32)*
TNR	True Negative Rate *(p. 32)*
TP	True Positives *(p. 31)*
TPR	True Positive Rate *(p. 32)*
WPAN	Wireless Personal Area Networks *(p. 12)*
WSAN	Wireless Sensor and Actuator Network *(p. 12)*

XAI Explainable Artificial Intelligence *(pp. vii, ix, 43, 48, 58, 66, 69)*

1 Introduction

1.1 Motivation

The forth industrial revolution, has been forcing companies into modernizing their productive processes (e.g., increasing efficiency and reducing costs) to gain competitive advantages over their competitors. While in some areas fully autonomous robots can perform most repetitive tasks, in others, human intervention is still required. The reduction of assembly errors during the production of goods is an area of major focus for companies looking to optimize their processes. One of these processes is the assembly of individual parts which compose the final product. By handheld screwdrivers with torque and angle sensing capabilities, detecting anomalous processes can be accomplished faster and more accurately. During the screw tightening process, data is collected in real-time, generating several instances with multiple features, including angle-torque pairs. In effect, developments in Information Technology (IT) have led to improved approaches and tools to store and process sensory data (e.g., Big Data (BD), Machine Learning (ML)), which can potentially provide gains in the field of anomaly detection in an industrial environment.

In this domain, defects are infrequent. However, each time a new failure mode is identified by domain experts, it must be characterized and added to a defect catalog. This expert system-based approach is relatively static and requires manual effort for catalog maintenance, making it unsuitable for the high volumes of screwing operations in an industrial setting. Often, defects are identified late in the production chain, resulting in additional production time and costs. Furthermore, the limited research available in this area, underscores the significance to enhance and automate the inspection task by adopting a data-driven approach based on ML algorithms. Such an approach can lead to substantial improvements by leveraging the power of automated analysis and prediction.

In this context, there are two primary prediction objectives that are commonly represented as binary classification tasks:

- Good or Fail (GoF) - ascertain the success or failure of an assembly process.
- Anomaly Region Detection (ARD) - provide the user with the specific region(s) in the tightening curve where the anomaly has been detected.

1.2 Research Objectives

The primary objective of this thesis is to present, devise, execute, and confirm a methodology for accurately categorizing an industrial fastening process as GoF, without relying on an explicit defect catalog for comparing tightening curves and identifying potential failure modes. The key aim is to develop ML-driven techniques that can seamlessly integrate into Bosch's AE/P shop floor machinery, effectively reducing the misclassification of units as "good" that should have been identified as faulty and preventing them from progressing to the next assembly stage. To fulfill the required objectives associated with this goal, the following research objectives Research Objectives (RO) were formulated:

- **RO1** - Develop, implement, and validate a ML-based framework capable of predicting the success or failure of an assembly process.
- **RO2** - Design, develop, and validate a tool capable of visualizing the anomalous zones within a faulty assembly process. This tool should enable the identification and display of areas where abnormalities or errors occur during the assembly process. The visualization capability is intended to enhance the analysis and understanding of faulty assembly processes, facilitating the prompt identification and resolution of problematic areas.
- **RO3** - Integrate the developed system into shop floor machinery with limited computational power while ensuring its efficiency, lightweight nature, and effectiveness. This integration aims to maintain the system's efficiency and effectiveness, enabling it to function smoothly within the constraints of the shop floor environment.

1.3 Contributions

The primary achievements and contributions of this doctoral dissertation include:

- A comprehensive study was conducted in order to investigate and compare various ML approaches for anomaly detection in industrial fastening processes (Ribeiro, Matos, Cortez, et al., 2021).We began by exploring the most significant data categories extracted from Bosch AE/P internal data collection systems. This selection was based on previous studies conducted by field experts at the plant and supported by relevant literature, which identified angle-torque pairs as potential indicators for anomaly detection in assembly processes. Given the nature and diversity of the provided data, multiple OCC algorithms were explored and evaluated. This initial exploratory study yielded valuable insights into the performance of various ML methods when applied to the analyzed anomaly detection cases. The best performing models identified were then subjected to rigorous testing on a larger and more diverse dataset (Ribeiro, Matos, Moreira, et al., 2022). This extended testing phase aimed to evaluate the generalizability and robustness of the models beyond the initial study dataset.

- An ML-based IDSS capable of predicting the success or failure of an assembly process has been developed, implemented, and validated. This artefact encompasses advanced ML algorithms and techniques, ensuring accurate and reliable predictions. Through rigorous testing and validation, the framework has demonstrated its effectiveness in proactive decision-making within assembly processes. By leveraging this artefact, manufacturing industries can optimize efficiency, reduce errors, and enhance overall productivity.

- An interactive visualization tool that has revolutionized the identification and display of areas affected by abnormalities or errors during the assembly process. This tool empowers operators to swiftly recognize and address problematic areas, leading to more efficient operations. It also offers human operators a valuable capability to set the final threshold and facilitates the explainability and interpretability of ML anomaly detection decisions. As a result, this contribution significantly improves the overall performance and comprehension of anomaly detection systems in screw cases, thereby paving the way for promising opportunities in further research and practical applications.

- An integration test study of scientific and technological contributions from both Big Data Analytics (BDA) and ML fields, enhancing a BDW within an advanced data analytics environment (Galvão et al., 2022). The proposed architecture establishes a unified environment between ML processes and the BDW, enabling data storage and monitoring of ML models and the BDW itself. This integration expands the analytical scope beyond business-level decision-making, allowing the establishment of performance metrics and continuous monitoring over time. Ultimately, this integration enhances decision-making and the overall effectiveness of the data analytics environment.

This PhD resulted in the publication of three scientific papers, one of which received the prestigious Best Paper Award:

1. **Title:** A Comparison of Anomaly Detection Methods for Industrial Screw Tightening.
   Authors: Diogo Ribeiro, Luís Miguel Matos, Paulo Cortez, Guilherme Moreira, André Luiz Pilastri
   Conference (Best Paper Award): International Conferences on Computational Science and Its Applications (ICCSA)
   DOI: https://doi.org/10.1007/978-3-030-86960-1_34
   RO: RO1

2. **Title:** Isolation Forests and Deep Autoencoders for Industrial Screw Tightening Anomaly Detection.
   Authors: Diogo Ribeiro, Luís Miguel Matos, Guilherme Moreira, André Luiz Pilastri, Paulo Cortez
   Journal: Computers
   DOI: https://doi.org/10.3390/computers11040054
   RO: RO2

3. **Title:** Bosch's Industry 4.0 Advanced Data Analytics: Historical and Predictive Data Integration for Decision Support
   Authors: João Galvão, Diogo Ribeiro, Inês Araújo Machado, Filipa Ferreira, Júlio Gonçalves, Rui Faria, Guilherme Moreira, Carlos Costa, Paulo Cortez, Maribel Yasmina Santos
   Conference: Research Challenges in Information Science (RCIS)
   DOI: https://doi.org/10.1007/978-3-031-05760-1_34
   RO: RO3

1.4 Thesis Organization

This thesis is divided into four main chapters, structured as follows:

- **Chapter 1 - Introduction**
  This introductory chapter describes the main motivations behind this work, the formulated research objectives identified from the research opportunities, the research methodology adopted and the structure of this document.

- **Chapter 2 - Background**
  In Chapter 2, fundamental concepts essential to this thesis are introduced, including Industry 4.0 (I4.0), BD, IDSS and Industrial Fastening Processes. The chapter also encompasses a comprehensive range of core concepts related to ML, including selected algorithms, metrics, and relevant research in this particular field of study.

- **Chapter 3 - Methods, Experiments and Results**
  Chapter 3 presents a comprehensive overview of the assembly industrial sector in the era of the I4.0 revolution. It examines the challenges and opportunities associated with this revolution and explores the utilization of smart sensors and semi-autonomous robots to assist human operators in production tasks that are not easily automated. The chapter emphasizes the potential of employing Machine Learning (ML) algorithms to develop an automated screw tightening inspection system, focusing on scalability and continuous model updating. Furthermore, this chapter investigates a novel approach for anomaly detection using low-dimensional input, specifically angle-torque pairs. The exploration involves the application of unsupervised ML algorithms such as Isolation Forest (IF) and deep dense Autoencoder (AE), which are discussed in detail in 1. Encouraging results

are achieved, indicating the effectiveness of these methods. The subsequent introduction of 2 extends the evaluation of the proposed anomaly detection techniques for screw tightening. This evaluation encompasses a broader range of assembled products and a larger dataset collected over an extended time period. To assess the performance and computational effort of the IF and AE models, various computational experiments are conducted. These experiments serve to measure their efficiency and effectiveness.

- **Chapter 4 - Conclusions**
  The last chapter of the document (Chapter 4) summarizes all scientific contributions of this work, uncovers its limitations and suggests possible future improvements to address such pitfalls.

2 Background

2.1 Bibliographic Search Strategy

2.1.1 Search Strategy

In the research search strategy process for the topic under investigation, relevant keywords and combinations such as ”anomaly detection”, ”screw tightening”, ”deep learning”, ”ML”, ”neural networks”, ”artificial intelligence”, ”manufacturing processes”, and ”quality control” were identified. The search was conducted in multiple databases including Google Scholar, Scopus, and Science Direct using these keywords. The search was refined by limiting the publication date to within the past 10 years and only including reputable sources such as peer-reviewed journals and conference proceedings.

An initial filtering stage of the search process involved sweeping through the titles and abstracts of the search results to ensure that they met the inclusion criteria and were relevant for the investigation. This resulted in a few potentially relevant articles from Scopus and Science Direct. It was found that despite a comprehensive search across all databases, there was very little literature investigating the topic, underscoring the need for future research in this area.

Google Scholar provided a large number of results, but many of them were either not relevant or did not meet the inclusion criteria, and a number of duplicates were also identified in the search results. On the other hand, Scopus provided a more comprehensive collection of literature, including articles, conference proceedings, and books. Advanced search options, such as boolean operators, were used to combine search terms and limit the search to specific fields.

Science Direct was another valuable source of information for this topic as it included a large collection of peer-reviewed articles and conference proceedings related to engineering, computer science, and other related fields.

After analyzing the obtained bibliography references, using all mentioned databases, a final manual inspection was conducted to select the relevant related works. In total, this resulted in 18 papers that were cited in this work, with some recurring authors.

2.1.2 Design Science Research Methodology for Information Systems (DSRM-IS)

Research in Information Systems (IS) can be conducted by adopting multiple paradigms, but the most frequent ones are design science and behavioral science. The behavioral science paradigm focuses on developing and verifying theories that explain or predict human or organizational behavior. In contrast, the design science paradigm aims to create and evaluate Information Technology (IT) artifacts intended to solve identified organizational problems. This paradigm is characterized by the building and application of IT artifacts to achieve knowledge and understanding of the problem and its solutions.

According to Hevner et al. (2004), the Design Science Research Methodology for Information Systems (DSRM-IS) produces two design processes and four types of artifacts: constructs, models, methods, and instantiations. The two design processes are the build and evaluate processes. In the build design process, artifacts are developed to address a real-world problem. In the evaluate design process, artifacts are assessed for their usefulness in solving the problem. These artifacts serve as tangible outcomes of the research process and contribute to advancing knowledge and practice in the field of information systems.

- **Constructs**: Abstract concepts or components of a problem domain that can be used to develop models or methods. Constructs are typically defined through a literature review or through the identification of key concepts within the problem domain.

- **Models**: Representations of constructs or systems that can be used to test hypotheses or theories, or to develop new methods. Models can take various forms, such as mathematical or conceptual models, and can be used to simulate or visualize a problem domain.

- **Methods**: Prescriptions for solving problems or achieving objectives within a problem domain. Methods can be developed through the use of constructs and models and can take the form of algorithms, guidelines, or procedures.

- **Instantiations**: Physical artifacts or systems that are created to solve a specific problem within a problem domain. Instantiations can be prototypes or final products that are evaluated in the context of the problem domain.

This methodology aims to be consistent with prior literature and provides a nominal process model for conducting design science research in IS, as well as a mental model for presenting and evaluating such research (Peffers et al., 2008). The DSRM-IS process model consists of six steps or activities:

1. **Problem identification and motivation:** The first step involves defining the specific research problem and explaining the significance of finding a solution (Peffers et al., 2008). In order to develop an effective solution, it may be helpful to break down the problem conceptually to better capture its complexity. Additionally, justifying the value of a solution serves two purposes: it inspires both the researcher and the audience to pursue the solution and accept the results, and it helps to explain the reasoning behind the researcher's understanding of the problem. To successfully complete this activity, one must have a thorough understanding of the problem and the importance of finding a solution. An extensive literature review work was conducted to identify anomaly detection methods for improving tightening processes. The review revealed a significant research opportunity in this area, as only a limited number of studies have been conducted.

2. **Objectives of the solution:** By analyzing the problem definition and considering what is possible and feasible, one can deduce the objectives of a solution. These objectives can take a quantitative form, such as setting specific targets for improvement over current solutions, or a qualitative form, such as predicting how a new artifact will facilitate solving previously unaddressed problems. The objectives must be derived logically from the problem specification. To achieve this, one must have an understanding of the current state of the problems and solutions, including their effectiveness. This requires access to relevant resources and knowledge. As previously introduced, the objectives of this doctoral thesis are driven by an organizational need. Furthermore, the thesis is a part of an innovation project that demands a strong alignment between the business objectives and scientific goals. The specific objectives are outlined in Section 1.2.

3. **Design and development:** The next step is designing and developing the artifact. This artifact can take the form of a model, method, instantiation, or construct, or it can embody new properties of technical, social, and/or informational resources. Broadly defined, a design research artifact is any object that incorporates a research contribution within its design. This process involves identifying the desired functionality and architecture of the artifact, and then creating the physical artifact itself. To successfully move from the objectives to the design and development phase, it is important to have a good understanding of relevant theories that can be utilized to develop an effective solution. During this stage, several Machine Learning (ML) models were developed as artifacts, which were subjected to rigorous testing. The best-performing models were selected for production, following a comprehensive evaluation process.

4. **Demonstration:** In order to validate the effectiveness of the artifact, it is crucial to demonstrate its successful use in solving one or more instances of the problem. This can involve various activities such as experimentation, simulation, case study, or proof. To effectively carry out the demonstration, it is essential to possess thorough knowledge on how to utilize the artifact in solving the problem. To assess the usefulness of the model in detecting anomalies in screw tightening processes for multiple products, various product families were utilized as case studies. This was necessary

due to the nature of the problem at hand. The results of these case studies are elaborated in Section 3.6.

5. **Evaluation:** The evaluation of the artifact's effectiveness in solving the problem is a critical step in the research process. This requires the observation and measurement of the artifact's performance, which involves comparing the objectives of the solution with the actual results obtained during the demonstration phase. It is essential to have knowledge of relevant metrics and analysis techniques to conduct a comprehensive evaluation. The evaluation process can take several forms depending on the nature of the problem venue and the artifact used. For instance, it can include a comparison of the artifact's functionality with the objectives defined in activity 2, objective quantitative measures such as budgets or items produced, client feedback or satisfaction surveys, simulations, or quantifiable measures of system performance such as response time or availability. The evaluation can also incorporate any relevant empirical evidence or logical proof. After completing this activity, researchers can decide whether to improve the artifact's effectiveness by iterating back to activity 3, or continue with communication and leave further enhancement to future projects. Nonetheless, the feasibility of this iteration may vary depending on the research venue. The purpose of evaluating the demonstration cases is to assess the effectiveness of the proposed ML-based approaches in addressing the problem in hand. Apart from the comparison of the proposed artifacts with the solution objectives, Section 3.5 describes the application of diverse evaluation metrics and techniques to evaluate the overall predictive performance of the ML models.

6. **Communication:** When appropriate, researchers should communicate the problem and its significance, the artifact's utility and novelty, the rigor of its design, and its effectiveness to relevant audiences such as practicing professionals. It is essential to have knowledge of the disciplinary culture to effectively convey the information. In scholarly research publications, researchers can use the structure of the design process to structure the paper. The nominal structure of an empirical research process, which includes problem definition, literature review, hypothesis development, data collection, analysis, results, discussion, and conclusion, is commonly used for empirical research papers. Therefore, the same approach can be taken when communicating the design process. The final activity involves disseminating the research findings through various channels, including the writing and publication of the doctoral thesis as well as scientific publications in relevant journals (as described in Section 1.3). Additionally, the progress and status of the research were regularly presented to the management (direction board) of Bosch AE/P through annual presentations.

Additionally, there are four different research entry points (see Fig. 1): problem-centered initiation, objective-centered solution, design and development-centered initiation, and client/context-initiated research (Peffers et al., 2008). While the process is structured sequentially, it is not mandatory to follow the steps in a sequential way, and it can start at any step. For example, problem-centered initiation research starts with step 1 because the idea for research resulted from problem observation or future research suggested in a paper from a prior project. The objective-centered solution research starts with step 2, and the research is triggered in the industry due to the problem they face or is oriented for artifact development. The design and development-centered initiation start with step 3, and the research results from the existence of an artifact that was developed to solve other problems to identify functionalities and performance requirements for a new artifact. Lastly, client/context-initiated research starts with step 4 and is based on practical observation of the final solution, identifying the impact of the solution. In this work, the research entry point is considered to be problem-centered. This doctoral thesis was conducted at Bosch AE/P to address a real-world problem faced by the organization. As such, the objectives were already pre-defined and aligned with the needs of the organization. However, prior literature was reviewed to align with existing gaps in the literature and research opportunities, which falls under the problem identification and motivation step of the DSRM process (Hevner et al., 2004).

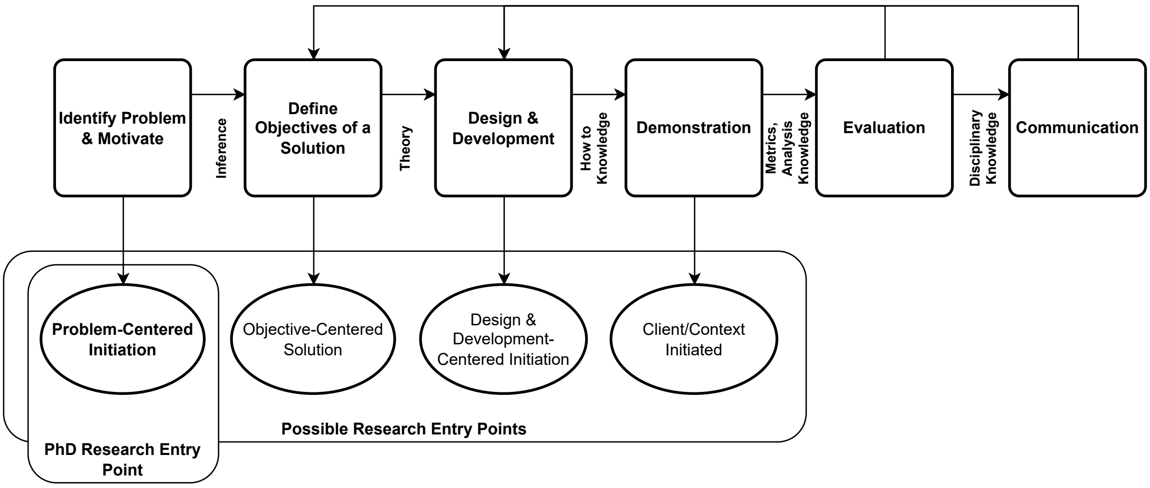


Figure 1: DSRM Process Model, adapted from Peffers et al. (2008).

2.2 Industry 4.0

Industry 4.0 (I4.0), also known as the Fourth Industrial Revolution, is a term that was first coined at the industrial Hannover Fair in 2011 (Lasi et al., 2014; Liao et al., 2017; Boyes et al., 2018; Dalenogare et al., 2018). It refers to the application of generic concepts of Cyber-Physical Systems (CPS) to industrial production systems (Lasi et al., 2014; Dalenogare et al., 2018), and represents the introduction of information technologies into the industry to achieve a higher level of operational efficiency, productivity, and automatization (Lu, 2017; Dalenogare et al., 2018; Xu et al., 2018). I4.0 brings a set of disruptive technologies that are transforming industrial production, business models, and business processes, such as autonomous robots, simulation, cybersecurity, cloud computing, augmented reality, Artificial Inteligence (AI), BD and analytics, and other technologies.

The design principles of I4.0 are interoperability, virtualization, decentralization, real-time capability, service orientation, and modularity (Lu, 2017). Two key concepts emerged with I4.0: Industrial Internet of

Things (I-IoT) and Industrial Cyber-Physical Systems (I-CPS). I-IoT and Industrial Cyber-Physical Systems (I-CPS) are extensions of the traditional Internet of Things (IoT) and CPS, respectively. I-IoT consists of the interconnection of intelligent industrial devices to improve the operational efficiency and productivity of the industrial system, while I-CPS supports manufacturing and industrial production applications Xu et al. (2018).

The result of I4.0 is a smart, connected, and highly automated manufacturing environment that has the potential to revolutionize the way products are manufactured, managed, and delivered (Lasi et al., 2014; Liao et al., 2017; Boyes et al., 2018; Dalenogare et al., 2018). It is characterized by the integration of physical and digital technologies, such as IoT, AI, and CPS (Lasi et al., 2014; Liao et al., 2017; Boyes et al., 2018; Dalenogare et al., 2018).

2.2.1 Internet of Things

There are numerous definitions for the IoT in the literature, with three of the most relevant definitions being detailed by Boyes et al. (2018):

1. *"A definition for the IoT would be a group of infrastructures, interconnecting connected objects and allowing their management, data mining and the access to data they generate where connected are sensor(s) and/or actuator(s) carrying out a specific function that is able to communicate with other equipment"*;

2. *"The terms Internet of Things and IoT refer broadly to the extension of network connectivity and computing capability to objects, devices, sensors, and items not ordinarily considered to be computers. These smart objects require minimal human intervention to generate, exchange, and consume data; they often feature connectivity to remote data collection, analysis, and management capabilities"*;

3. *"The IoT represents a scenario in which every object or thing is embedded with a sensor and is capable of automatically communicating its state with other objects and automated systems within the environment. Each object represents a node in a virtual network, continuously transmitting a large volume of data about itself and its surroundings..."*;

Based on these definitions of IoT, the I-IoT can be defined as a global network infrastructure that allows interconnection of industrial devices and equipment's through sensory, communication, networking, and information processing technologies, so that they can share information between them and coordinate decisions (Al-Fuqaha et al., 2015; Boyes et al., 2018; Xu et al., 2018). This means that I-IoT involves using IoT technologies in industrial environments to interconnect industrial assets, such as smart objects and cyber-physical assets (Boyes et al., 2018; Xu et al., 2018). This network infrastructure can be used to monitor and control physical objects in CPS (Xu et al., 2018).

The generic I-IoT architecture consists of three layers, as depicted in Fig. 2. These three layers are: the application layer, communication/network layer, and physical layer (Al-Fuqaha et al., 2015; Xu et al., 2018). The application layer refers to different industrial applications, such as smart factories, smart plants, smart supply chains, and other applications. The main responsibility of this layer is to bridge the gap between the user and applications by providing timely monitoring, accurate control, and efficient management through numerous sensors and actuators in those smart industrial applications (Al-Fuqaha et al., 2015; Xu et al., 2018). The communication/network layer connects all "*things*" through numerous communication networks and technologies, such as Wireless Sensor and Actuator Network (WSAN), 5G, Wireless Personal Area Networks (WPAN), Machine-to-Machine (M2M), and Software-Defined Networking (SDN), to share information between them. This layer provides networking support and data transfer for communication between "things" (Xu et al., 2018). Finally, the physical layer is composed of smart objects and cyber-physical assets, such as sensors, actuators, Radio Frequency-based Identification (RFID), manufacturing equipment's, and other industrial objects (Al-Fuqaha et al., 2015; Xu et al., 2018). These smart objects and cyber-physical assets are responsible for acquiring and processing information (Al-Fuqaha et al., 2015).

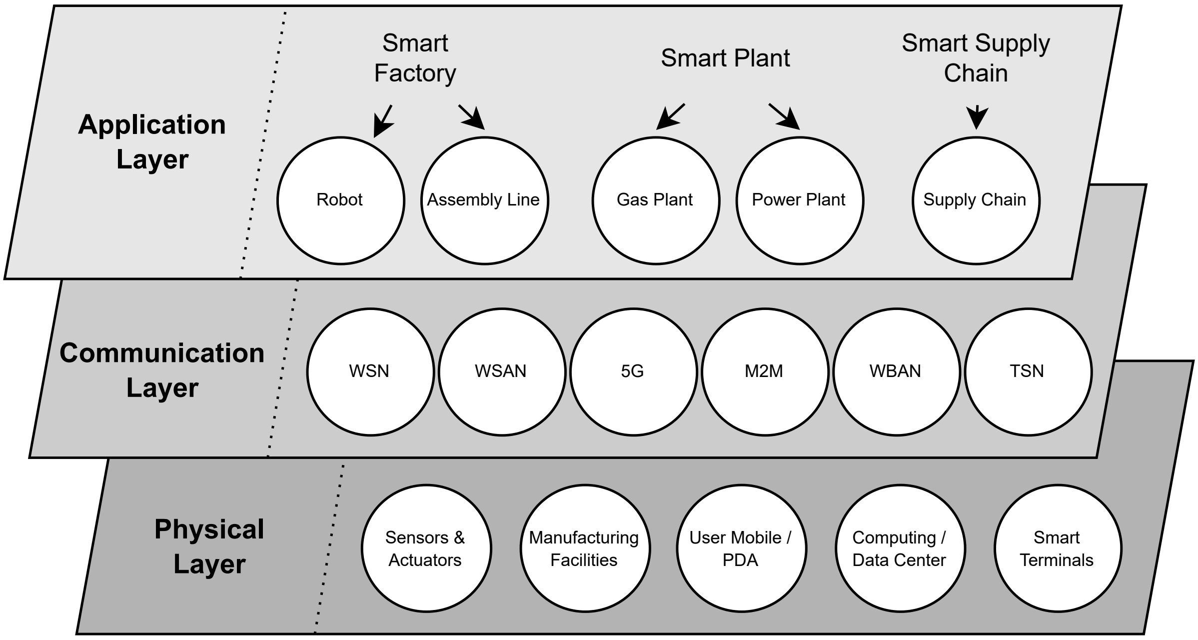


Figure 2: I-IoT architecture, adapted from Xu et al. (2018).

2.2.2 (Industrial) Cyber-Physical Systems

Helen Gill proposed the term Cyber-Physical Systems (CPS) during the National Science Foundation (NSF) CPS workshop in 2006. Since then, numerous definitions of CPS have been proposed in the literature (Alguliyev, Imamverdiyev, and Sukhostat, 2018). Some examples of these definitions are:

1. *"CPS is a system that can effectively integrate cyber and physical components using modern sensor, computing, and network technologies"* - Alguliyev, Imamverdiyev, and Sukhostat (2018).

2. *"CPS is the integration of computing and physical processes. They include embedded computers, network monitors, and controllers, usually with feedback, where physical processes affect computations and vice versa"* - Colombo et al. (2017) and Alguliyev, Imamverdiyev, and Sukhostat (2018).

3. *"A system comprising a set of interacting physical and digital components, which may be centralized or distributed, that provides a combination of sensing, control, computation, and networking functions, to influence outcomes in the real world through physical processes"* - Boyes et al. (2018).

I-CPS is defined as a vertical industrial system based on cyber and physical systems (Xu et al., 2018). Every real physical object has at least one cyber representation, and each cyber system can be associated with a physical representation (Colombo et al., 2017). I-CPS provides productive and efficient manufacturing and automation, and enables the monitoring and control of industrial physical processes (Colombo et al., 2017; Xu et al., 2018). The I-IoT represents the integration of communication/network layer of the I-CPS Xu et al. (2018). According to Xu et al. (2018), the I-IoT corresponds to the communication/network layer integration of the I-CPS.

This leads to the creation of a manufacturing environment that is highly automated, connected, and intelligent. Real-time monitoring and control of the production process are made possible by I-CPS, enabling early detection of failures in assembly processes and preventing their propagation down the manufacturing chain.

2.3 (Intelligent) Decision Support Systems

According to Arnott and Pervan (2014), Decision Support System (DSS) is a field within the IS discipline that aims to assist and enhance decision-making processes. The concept of DSS was initially introduced by Gorry and Morton (1971) in the early 1970s. They defined DSS as: "interactive computer-based systems, which help decision makers utilize data and models to solve unstructured problems". Later on, in the same decade, Keen and Morton (1978) refined the definition of DSS by describing them as : "Decision support systems couple the intellectual resources of individuals with the capabilities of the computer to improve the quality of decisions. It is a computer-based support system for management decision-makers who deal with semi-structured problems".

In the early 1980s, the concept of Intelligent Decision Support System (IDSS) emerged with the intention of integrating artificial intelligence and expert system tools into DSS (Arnott and Pervan, 2014; Kaklauskas, 2015). IDSS is an interactive tool that assists with decision-making in well-structured decision and planning scenarios. It employs expert system techniques and specialized decision models to support the decision-making process (Arnott and Pervan, 2014).

Arnott and Pervan (2014) reported that research in DSS accounts for approximately 10% of the overall IS field. In recent years, this research has predominantly focused on data-driven approaches, particularly

Business Intelligence (BI). The authors also highlighted the evolution of DSS where the latest trend in DSS is the adoption of Business Analytics (BA), which encompasses BI, Predictive Analytics, and Optimization techniques.

Howard Dresner first proposed the term Business Intelligence (BI) in 1989, which gained widespread attention by the early 2000s (Turban, Sharda, and Delen, 2011; H. Chen, Chiang, and Storey, 2012; Arnott and Pervan, 2014). The shift from Executive Information Systems (EIS) to BI was driven by the introduction of dimensional modeling and data warehousing concepts (Turban, Sharda, and Delen, 2011; Arnott and Pervan, 2014). BI combines architectures, tools, databases, analytical tools, applications, and methodologies to enable effective decision-making (Turban, Sharda, and Delen, 2011).

The concept of Business Analytics (BA) emerged from the fusion of BI with new capabilities such as optimization, forecasting, predictive modeling, and statistical analysis. Thomas H. Davenport popularized the term BA in his article for the Harvard Business Review in 2006, defining it as ”the extensive use of data, statistical and quantitative analysis, exploratory and predictive models, and fact-based management to drive decisions and actions” (Arnott and Pervan, 2014). While there is some overlap between the definitions of BI and BA, the latter is primarily seen as the key analytical component within BI (H. Chen, Chiang, and Storey, 2012). As a result, the unified concept of Business Intelligence and Analytics (BI-A) has been proposed (H. Chen, Chiang, and Storey, 2012). All of these terms - IDSS, BI, BA - have emerged from the evolution of DSS and are illustrated in Fig. 3.

Analytics can be classified into four distinct levels: **Descriptive**, **Diagnostic**, **Predictive**, and **Prescriptive** analytics. They can be described as followed:

- **Descriptive analytics** - is used to answer the question ”What happened?” and involves understanding what is currently happening within an organization (Delen and Demirkan, 2013; Sharda et al., 2015).
- **Diagnostic analytics** - addresses the question ”Why did it happen?” and seeks to identify the root cause of a problem (Sharda et al., 2015).
- **Predictive analytics** - attempts to answer the question ”What will happen?” and utilizes mathematical algorithms and programming to find patterns within data that can be used to make predictions about the future (Delen and Demirkan, 2013; Sharda et al., 2015).
- **Prescriptive analytics** - aims to answer the question ”How can we make it happen?” and focuses on decision-making to achieve the best possible performance (Delen and Demirkan, 2013; Sharda et al., 2015). It includes techniques such as multi-criteria decision-making, optimization, and simulation.

Analytics of all kinds can offer significant benefits to organizations seeking to gain a better understanding of their operations. Nonetheless, both diagnostic and predictive analytics are crucial in facilitating comprehension of the root causes of a specific failure and forecasting the result of a particular fastening process. In effect, in this PhD work we assume the development of IDSS that makes usage of modern AI techniques, namely ML algorithms, to generate data-driven models that allow to perform descriptive and predictive analytics related with the screw tightening task.

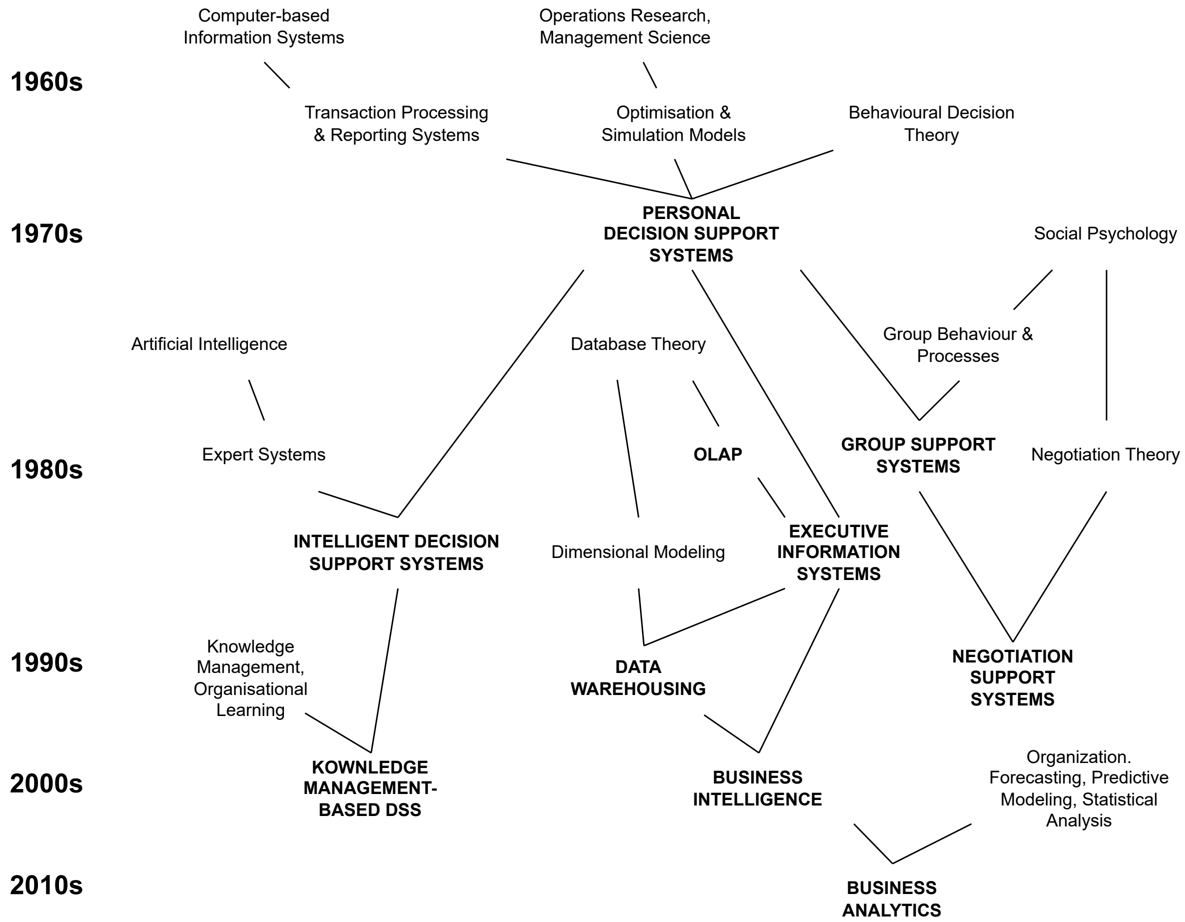


Figure 3: Genealogy of DSS, adapted from Arnott and Pervan (2014).

2.4 Industrial Fastening Processes

Fasteners are taken into consideration to fulfill diverse purposes in applications, encompassing strength, appearance, and reusability. Reusability holds particular significance as disassembly is often required for maintenance, service, or repair of many assemblies (J. Speck, 2018). Even in moderate-volume assemblies, the desire to disassemble can arise from either a portion of the target market or the product manufacturer, if not both. Consequently, the reusability of fasteners becomes a crucial design requirement, even for products with a remote possibility of requiring disassembly. Since it is impossible to predict which products will necessitate disassembly, the requirement for reusability extends to all assemblies produced during the production run.

The role of fastener appearance is vital in the market acceptance of products. Despite purchasers often claiming that economic or technical features are their primary decision factors, the undeniable influence of style and appearance on our purchasing choices cannot be overlooked. Fastener appearance can symbolize robustness, exemplified by exposed, high-strength fasteners in industrial machinery, instilling visual assurance of product strength, durability, and value in customers and prospects. Conversely, fasteners can be discreetly designed, reflecting the prevailing trend in consumer durable goods such as automobiles and appliances. This sleek and well-tailored appearance communicates thoughtful design, seamless assembly, and a modern, up-to-date aesthetic. The popularity of these diverse fastener "looks" fluctuates, and neglecting their significance poses a risk for assembly designers and manufacturers.

At their most fundamental level, fasteners are utilized for their strength, particularly their holding strength in assemblies. Fasteners, directly "out of the box," offer potential holding power, making holding ability the key element of their overall strength. Alongside appearance and reusability, holding power constitutes the mechanical foundation of all fasteners.

2.4.1 Fastener Types and Selection

Within the realm of fastener selection, an overwhelming array of designs exists within the industrial fastener market. If one is entrusted with the responsibility of choosing appropriate fasteners for products and assemblies, they encounter a complex landscape. Similarly, when tasked with developing joining techniques and procedures, the effectiveness of their work can profoundly impact the market success of their company's products in comparison to competitors, while also carrying legal ramifications (Park and Okudan Kremer, 2015). Furthermore, when undertaking the layout of an assembly process, whether involving the design of a new assembly or the modification of an existing one, the decisions made can have enduring implications for the growth or decline of the process. The efficiency of the assembly process and the quality of the fastening hardware are pivotal factors that determine the outcome.

To simplify these decision-making processes, it is beneficial to categorize them as hardware and software components. The software encompasses the overall assembly and joining engineering approach, including assembly drawings, procedures, instructions, as well as inspection and testing documentation. On the other hand, the hardware comprises the actual fasteners, installation tooling, and all the necessary equipment at the assembly site. By differentiating between the hardware and software aspects of assembly layout, one can approach hardware selection based on the predetermined approach outlined by the software. Expanding on this analogy, possessing a comprehensive understanding of the available hardware in the fastener industry empowers one to make more informed decisions concerning the optimal assembly approach for their assemblies.

The categorization of commonly used fasteners can be outlined as follows, based on their type (Amancio-Filho and J. d. Santos, 2009; J. Speck, 2018):

- **Tension**: In the industrial world, tension fasteners emerge as the most frequently employed fastening design (Amancio-Filho and J. d. Santos, 2009; J. Speck, 2018). As a design strategy, the optimization of assembly solutions can often be achieved by directing service loads concentrically towards a rigid, minimum member joint and clamping it with a well-preloaded tension fastener. Within the automotive industry, various types of tension fasteners are commonly employed. These include:

  1. **Pan Head**: Is a commonly used tension fastener with a wide range of drive types, suitable for high-speed production. Its design minimizes the risk of head failure by ensuring failure occurs in the threaded area, making it a desirable choice for tension applications.
  2. **Hex Head**: Hex head screws and bolts are popular choices in industrial applications, offering advantages such as widespread availability, compatibility with standard tightening tools, and strong load-carrying capabilities. Additionally, hex head cap screws are produced with varying strength levels, tailored through specific manufacturing and heat-treating processes.
  3. **Socket Screws**: Are compact yet robust tension fasteners with an internal wrenching tightening drive. They meet rigorous industrial standards, exhibiting superior tolerances and control. While initially designed for tool and die work, their high strength and compact size have made them versatile components across diverse applications.
  4. **Blind Rivets**: Serve as tension fasteners, suited for lower magnitude loads in comparison to socket products. Their key benefit lies in swift installation, devoid of threading and torquing requirements. Consisting of a rivet body and inner mandrel, blind rivets can be efficiently tightened using manual or power actuated riveting tools. During installation, the mandrel separates and breaks off as intended. Material options for blind rivets encompass steel and aluminum, with diverse combinations available.
  5. **Weld Studs**: Are fasteners that are affixed to metal sheets through fusion welding using welding current. This welding technique ensures a sturdy mounting point for subassemblies and components. Generally, weld studs are manufactured using cold heading and roll threading methods
  6. **Self-Threading Screws**: Self-threading fasteners offer cost-saving advantages in applications where the reusability of threaded fasteners is needed without the requirement for separately generated internal threads. However, their implementation should be planned meticulously to ensure optimal fastener performance and minimize assembly challenges. These fasteners can be categorized as thread cutting, thread forming, or thread rolling.

- **Compression**: Compression fasteners are a category of fasteners designed to apply and distribute clamping forces to securely join components or assemblies. They work by generating compressive forces that hold the parts together, ensuring stability and preventing relative movement (Amancio-Filho and J. d. Santos, 2009; J. Speck, 2018). Some examples of compression fasteners include:

1. **Set Screws**: Headless threaded fasteners that undergo shortening under preload, serving as torque-transmitting elements. American National Standards Institute (ANSI) standard hex socket screws, commonly made from alloy steel or 300 series stainless steel, are frequently used. Internal hex set screws, known as "safety" set screws, are advantageous due to their reduced risk of catching loose objects in rotating machinery. These screws feature formed or machined points, with ANSI standards encompassing variations like external cone points, half and full dog points, flat points, and oval points, each offering unique application advantages.
2. **Washers**: Washers function as compression fasteners, distributing clamp load and reducing bearing stresses by acting in a "snowshoe" manner. Using well-made washers can prevent indention and increase clamp load by reducing underhead friction. Washers also extend the clamped grip length, offering performance advantages. However, they can lead to issues such as increased parts count and joint faces, potentially causing vibrational loosening. Locking washers, like split lock washers, help prevent loosening, albeit with a limited clamp load increase. Belleville washers offer higher efficiency. The Clamp-Affected Zone (CAZ) represents the volume of material where the fastener's clamping load is applied, shaped by factors such as assembly member ductility, clamping load magnitude, and washer area.

- **Shear**: Are specifically designed to bear loads perpendicular to their axis. Dowel pins and roll pins are excellent examples of shear fasteners. Dowel pins, an early form of fasteners for carrying shear loads, continue to be highly efficient in assembly methods (J. Speck, 2018). Dowel pins exhibit a wide range of variations, from simple low-carbon wire cutoffs to alloy steel and hardened, ground tool-and-die quality dowels featuring end chamfers and radii. In contrast, roll pins are made from coiled strip steel, offering spring-like characteristics upon cutting into pin shapes. While dowel pins provide superior shear strength, roll pins offer installation economies in assemblies. It is worth noting that dowel pins should be pressed into the assembly rather than hammered, as is occasionally done.

- **Adhesives**: Adhesives for fasteners encompass a diverse array of types, providing versatility in their applications (Amancio-Filho and J. d. Santos, 2009; J. Speck, 2018). These adhesives can serve as independent fastening systems or complement mechanical fasteners. When used as supplementary elements, adhesives can be selected based on desired strength and permanence. Preapplied adhesives or those applied on-site can be specified accordingly. In both cases, the absence of air during installation is crucial to allow the adhesives to cure.

2.4.2 Joining Mechanics

The engineering mechanics associated with fastening and joining exhibit similarities throughout various assembly work conducted on a daily basis. Fasteners or fastening systems apply forces that, when executed correctly, contain the service loads within the fastening design and installation. The range of assembly component materials, the number of joint interfaces, and the intended service conditions highlight the diverse mechanics involved in assembly work and performance. To comprehensively explore the field of fastener mechanics, it is beneficial to categorize procedures based on the materials of the assembly components being joined. A logical approach involves distinguishing between metallic and nonmetallic assemblies, despite acknowledging that many all-metal assemblies possess qualities of springiness and plasticity, while numerous engineered thermoplastic products exhibit high rigidity and compressive strength. Furthermore, hybrid assemblies composed of both metal and plastic components exhibit properties of both materials. Therefore, making mechanical distinctions between joint materials facilitates the understanding and application of mechanical techniques and models, enabling students and observers of assembly engineering to enhance their comprehension and optimize assembly performance.

2.4.2.1 Rigid Metal Joints

Rigid metal joints are a type of connection between metallic components that offer high levels of stiffness and immobility. These joints are designed to minimize or eliminate any relative movement between the connected parts, ensuring a rigid and stable structure. Common examples of rigid metal joints include welded connections, bolted connections with high clamping forces, and adhesive bonding with high-strength adhesives. Rigid metal joints are often employed in applications where maintaining precise alignment, resisting deformation, and minimizing vibrations are crucial. These joints provide excellent load transfer capabilities, distribute forces evenly, and enhance the overall strength and stability of the assembled structure.

2.4.2.2 Nonmetal Joints

In the realm of nonmetal assembly components, there exists a certain degree of flexibility that differs from metal joints. However, the specifications for nonmetallic fasteners lack well-established and proven rules to guide their selection. Nonetheless, a considerable base of suppliers caters to the demand for fasteners in nonmetallic component applications. To elucidate the mechanics involved, it is helpful to examine the example of a hypothetical product consisting of a contoured structural member made of composite material. The objective is to fasten multiple injection-molded plastic brackets onto the panel. Examples of these types of nonmetal joints include (J. Speck, 2018):

- **Snap Fit Assembly**: In this joining technique, snap fit assembly is employed. Factors to consider include the geometry, tolerances, and the retention of fastening load. A rectangular snap with a barbed end and a rectangular receiver in the panel is designed, with considerations given to the shape and angle of entry of the snap fit ramps and the force required for snapping. Removal requirements may involve modifying the snap, ensuring that only one barb is engaged with clearance in the opposite barb zone for release. The tolerance of molded snap geometries impacts fastening load retention, and environmental conditions such as temperature and ultraviolet radiation can affect the strength of the snap fastening.

- **Adhesive Assembly**: The adhesive fastening approach necessitates considering surface area, surface preparation, adhesive thickness and condition, and compatibility with the intended service. The strength of the cured joint can be tested by applying forces perpendicular and parallel to the adhesive-held surfaces. Evaluating the joint's strength under different time and temperature conditions further allows for assessing its performance.

- **Blind Rivet Assembly**: Mechanical considerations for blind rivet assembly encompass the compressive strengths of the bracket and panel material, as well as the force required for cinching the joint and breaking off the mandrel. If the forces exceed the compressive strength, the mandrel may pull through the rivet body, leading to an incomplete rivet installation. Retention of rivet clamp load is another important factor, and comprehensive testing under various conditions aids in addressing these concerns.

- **Self-Threading Screw Assembly**: The self-tapping screw installation entails considerations of the strip-to-drive ratio and the compatibility of screw thread dimensions with the hole site in both the bracket and panel. The thread-forming threads capture the panel material between the major and root diameters, exerting a relatively perpendicular load. In plastic materials, thread angles narrower than those of machine screws are often beneficial.

2.4.3 Automated Assembly Machinery

To integrate automated assembly processes effectively and accommodate the rapid escalation of assembly build quantities resulting from commercial success and increasing demand, the implementation of an automatic assembly machine becomes a logical and crucial step. The success and satisfaction of achieving timely ramp-up in assembly production hinges upon meticulous planning, decision-making, and execution of the automatic assembly machine. Inadequate decision-making in this regard renders subsequent assembly operations incapable of fully restoring the potential gains in assembly efficiency. Conversely, when planned and executed with precision, an automatic assembly machine or machines represent a rational and profitable approach to assembly management, allowing for successful response to increased assembly demands.

Presently, a multitude of suppliers offer automated threaded fastening systems encompassing robots, fastening tools, and various accessories such as feeders (refer to Fig. 4a), pickup devices (see Fig. 4b), and machine vision modules. Different types of robots are available, including gantry-type and Cartesian robots known for their high positioning accuracy and larger footprint. Selective Compliance Assembly Robot Arm (SCARA) robots are commonly utilized in electronics assembly, while 6-Degrees Of Freedom (DOF) articulated arms (Lara, Althoefer, and L. Seneviratne, 1998; Matsuno, Huang, and Fukuda, 2013) are capable of driving screws from various angles to handle complex part geometries. Parallel Delta robots facilitate rapid pick-and-place movements. Passive compliance units, such as springs (Hwang et al., 2012), are frequently employed to connect these robots with fastening tools, enabling easier initial thread mating while maintaining the appropriate insertion force.

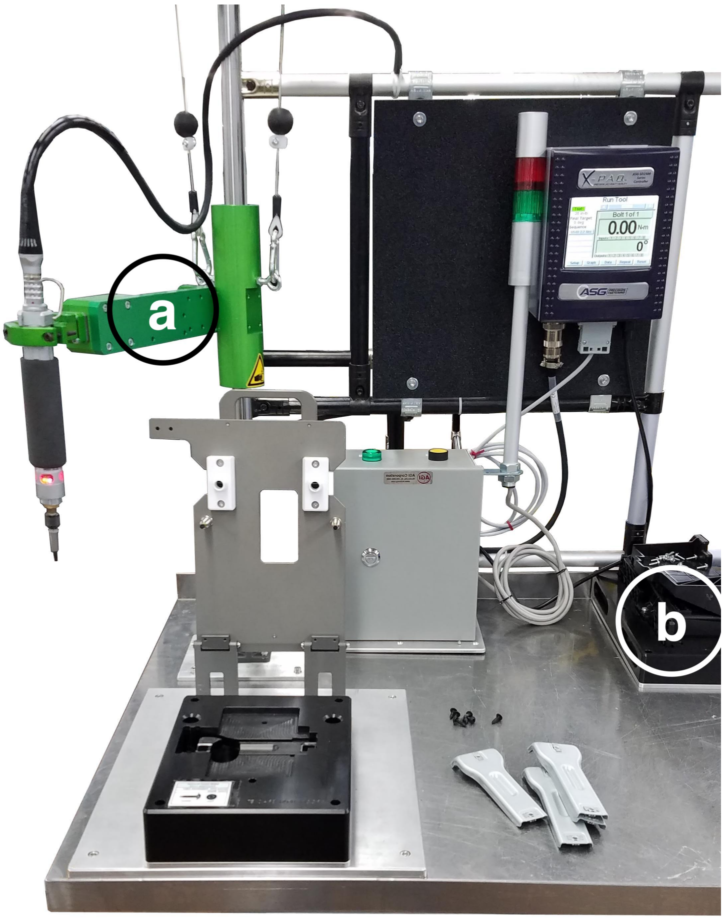


Figure 4: Vibratory bowl feeder (a) and Pickup device (b), similar to the ones under study, from AGI Corp.

The selection and sizing of robots are influenced by factors such as assembly speed, positioning accuracy, insertion axis, reaction moments generated by the fastening tool, part geometries, system integration complexities, and other application-specific considerations. Typically, small and lightweight robots are suitable for small screws, Cartesian and SCARA systems are suitable for midsize fasteners, while larger multiaxis robots are utilized for midsize and larger fastener sizes. Among these options, 6-DOF articulated robots offer agility, extended reach, adaptability for future use, and the ability to handle complex part geometries that may require multiple screw insertion orientations.

2.4.4 The Assembly Process

At Bosch AE/P, threaded fasteners serve as the primary method for joining mechanical assemblies. Among external tension-type threaded fasteners, screws and bolts are the two main types utilized. Bolts are specifically designed to be paired with nuts or threaded holes to achieve a robust clamping force. Conversely, screws are primarily intended for use with preformed internal threads, commonly known as machine screws, although they can also create their own threads, as observed in self-tapping screws. It is worth noting that despite their technical distinctions, the terms bolts and screws are frequently used interchangeably. In the context of fastening, a thread refers to a helical ridge with a uniform cross-sectional shape present on the inner or outer surface of a cylinder. External threads are featured on bolts, studs, or screws, while internal threads are found on nuts and tapped holes.

Automated screw fastening encompasses several sequential stages of operation, typically involving feeding, alignment, screwdriving, and post-fastening processes (Järvenpää, Lanz, and Tuokko, 2013), Fig. 5).

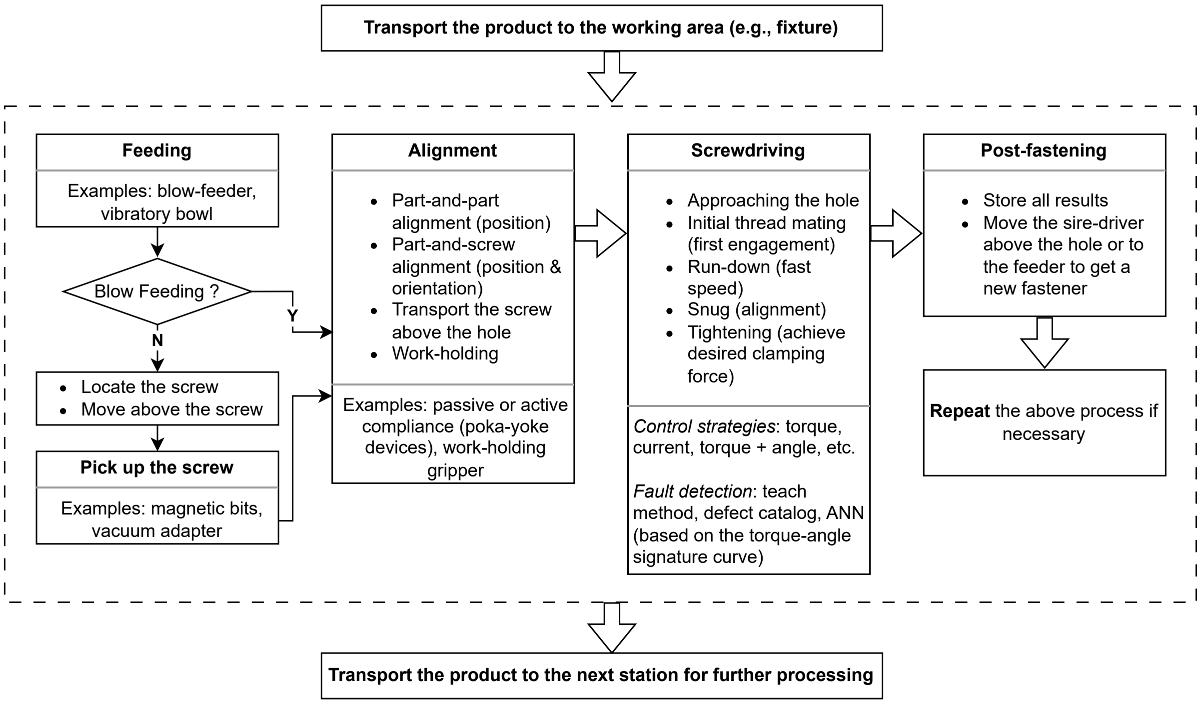


Figure 5: Typical screw fastening procedure in an automatic assembly line, adapted from Jia et al. (2019).

Feeding refers to the organization and orientation of threaded fasteners before they are used with screwdrivers. To achieve this, screw feeders are commonly employed. Various feeding mechanisms such as vibratory feeders, flex feeders, tape feeders, shaker trays, and blow feeders are utilized for this purpose. The choice of feeder type depends on factors like the type of screws, their materials and sizes, system cost, and speed requirements.

Prior to initiating the screwdriving process, it is essential for the machine to possess the capability of picking up the designated screw to be used. There are several methods available for picking up the fastener to be used. These include blow feeding, where screws are oriented using a bowl feeder and then blown through a tube to reach the screwdriver tip; magnetic attraction, where the driver bit is magnetized to attract the screws; and vacuum grippers, which offer precise orientation control and are commonly preferred for automated screwdriving due to their versatility across different materials. Once the screw and driver bit are paired together, the alignment process can be initiated. After successfully picking up and aligning the screw, the screwdriving process can commence. Ensuring proper alignment is vital for successful screwdriving, which involves aligning the fixtured part with the assembled part (known as *part-part alignment*) and aligning the fixtured part with the screw itself (*part-screw alignment*) to prevent any potential failures. Typically, a fixture or a compliance device is employed to restrict the user's range of motion and aid in achieving the necessary alignment.

Once the screw is picked up and aligned, the screwdriving process can begin. The process can be divided into several stages, which can be visualized through a torque-angle curve (Jia et al., 2019). Each of these stages can be roughly associated with the steps described in Fig. 10 and Table 1, where certain failures may occur:

- ***Starting the thread* or *finding the thread* (0)**: entails mating the bolt thread with the nut thread. referred to as *starting the thread* or *finding the thread* depending on whether a self-tapping screw is utilized or not. Examples of failures in this stage include *cross-threading* in which the first (full) external thread crosses the internal thread in such a way that the thread engaged on one side of the internal thread is not on the same revolution as the thread engaged in the opposite side"(Nicolson and Fearing, 1993); and jamming.

- ***Prevailing torque zone* (or *rundown zone*) (1)**: stage where the screw steadily drives through the hole until the screw head contacts the work surface. *Mis-starts* (J. A. Speck, 2015) frequently occur as a common failure mode within this fastening zone. These mis-starts refer to instances where the initial alignment or engagement of screws or bolts with their respective threads or holes is improperly executed. *Stripped Threads* are also very common and occur when excessive force or incorrect alignment during the rundown can strip the threads of the fastener or the mating component, causing a weakened or ineffective connection.

- ***Snug zone* (or *alignment zone*) (2)**: zone where the fastener and joint mating surfaces align. The snug zone is a complex combination of macroeffects and microeffects due to surface/thread deformations. *Insufficient Thread Engagement* is a prevalent occurrence in this context, characterized by the fastener not fully engaging with the threads in the mating component during the snug phase. This failure leads to diminished clamping force and a weakened joint. Another issue stemming from these similar characteristics is the *floating screw*, where the screw fails to fully seat within the work surface.

- ***Elastic clamping zone* (3)**: where the torque-angle curve has a constant slope. In some safety-critical applications, the screw may need to be tightened beyond the elastic clamping zone to the postyield zone, where plastic deformation occurs. Understanding the mechanics and failure modes of this process is crucial for ensuring successful screwdriving. Potential failures within the elastic clamping zone can arise due to *over-tightening*, where excessive torque surpasses the elastic limit of the fastener or connected components, resulting in deformation, cracks, or even fractures, leading to structural failures. Additionally, *fatigue failure* may occur as a consequence of repeated cycles of over-tightening and relaxation in this zone, leading to the development of cracks, fractures, or a gradual loss of clamping force over time.

The purpose of both RO1 and RO2 in the present study is to detect potential failures throughout all stages of the fastening process, regardless of having comprehensive knowledge about the specific underlying processes that encompass these common failure modes. In conjunction with the fastening process itself, the quality assessment stage takes place to determine the success or failure of the fastening.

2.4.5 Monitoring the quality and detecting faults

In modern automated assembly machines, the incorporation of microprocessors enables the analysis of torque on the transducer and the degrees of rotation of each connecting screw during installation. This analytical capability allows for mathematical comparison with the torque-rotation function or signature of a properly preloaded connecting joint (Peterson, Niznik, and Chan, 1988; Nieoczym and Longwic, 2016). Consequently, the smart assembly system can effectively differentiate between normal process variation and exceptional causes such as bolts with insufficient yield strength or instances of cross threading during initial stages. By engaging in this analysis of assembly operating conditions, the assembly machine assumes the role of a cognitive, decision-making unit, capable of identifying assembled units that fall outside the preload specification and require rework or repair. One notable advantage of such a system is the ability to store data pertaining to each assembly for future reference. By utilizing correctly programmed assembly machines, the performance of the fasteners supplied by connecting screw manufacturers can be evaluated using data obtained directly from the assembly process. Assemblies that do not meet engineering specifications can be retrospectively reviewed, allowing for effective control over the factors contributing to noncompliance. Numerous algorithms and monitoring strategies have been developed for detecting faults in the screw insertion process. One commonly utilized approach is the teach method (Smith, 1980), often incorporating limit checking. This method operates under the assumption that each screw insertion process exhibits a distinctive torque-angle fastening signature curve, with faults typically manifesting as significant deviations from normal signals (refer to Fig. 10). Consequently, as a preliminary setup step before assembly, the signature signal corresponding to the specific screwdriving operation is taught and recorded, utilizing the average of accurate insertion examples (Althoefer, Lara, Zweiri, et al., 2008). This is further explored in Section 2.8. During assembly, this method compares the real-time insertion signals with the stored correct signature signals. In the context of this PhD thesis, Bosch's quality assessment process adheres precisely to this procedure, which also includes further detailing of failure modes by comparing the results against a defect catalog, as outlined in Section 3.1.

2.5 Data Mining and Machine Learning

2.5.1 Cross-Industry Standard Process for Data Mining

Cross-Industry Standard Process for Data Mining (CRISP-DM) is a well-established methodology used in the field of data mining and analysis. It provides a structured framework for designing and executing projects, ensuring that they are approached in a systematic and organized manner, making it a popular choice for organizations looking to better understand their data. It was first developed in the 1990s by a group of experts in the field of data mining. Coming from well established companies like Daimler Chrysler, SPSS and NCR Corporation they recognized the need for a standardized approach to data mining projects and worked together to create a methodology that could be used across industries. This resulted in the development of the CRISP-DM first published by Chapman et al. (2000).

The CRISP-DM methodology consists of six main stages (Fig. 6), each of which is designed to build on the previous stage and move the project towards a successful outcome. The six stages of CRISP-DM are: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment.

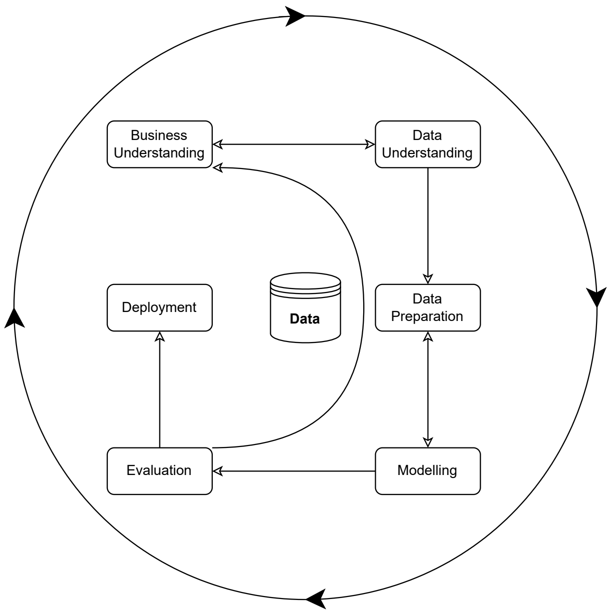


Figure 6: The CRISP-DM methodology, adapted from Chapman et al. (2000).

1. **Business Understanding**: is the starting point for any data mining project and involves defining the objectives and goals of the project from a business perspective. It is initiated by the gathering of background information about the current business situation, followed by a documentation stage where the specific business objectives are decided upon by key decision makers. It must also include a clear definition of the success criteria which will later be used to assess on the quality of the final solution. The main output of this work is typically a project plan / roadmap which answers

all business questions asked thus far and both the business and data mining goals for the project. This stage closely relates to the first and second steps of the DSRM-IS (see Section 2.1.2).

2. **Data Understanding**: involves taking a closer look at the data available for mining. It includes activities such as data cleaning and transformation, data visualization, and identification of relationships and patterns in the data. The objective of this stage is to gain a comprehensive understanding of the data and its quality (which can be assessed from a multitude of vectors like missing data, data errors, etc). This understanding is critical for making informed decisions about which data to include in the project and how to prepare it ahead of modeling. To acquire such comprehensive knowledge of the data at hand two reports are usually produced : a data quality report and a data exploration report. While the prior focus on aspects such as data cardinally, correlation between variables and overall data quality, the latter is closely related to a more traditional Exploratory Data Analysis (EDA). This step is critical in avoiding unexpected problems during the next phase - *data preparation* - which is typically the longest part of a project.

3. **Data Preparation**: it is one of the most important and often time-consuming aspects of any data mining project. Whilst the overhead of such task be minimized by devoting adequate energy to the previously introduced phases, preparing and packaging data is a complex task. As a process it involves a plethora of data preparation tasks such as the merging of data sets, aggregation of records, selection of subsets of data, data cleansing and guarantying adequate distribution of data across the training and test data sets. This stage is critical as the quality of the data will greatly impact the accuracy of the models created (Stojanovic et al., 2016). This stage proved particularly important for this project. The inclusion and contribution of the domain experts in the feature selection process (the process of reducing the number of input variables from the initial set) of this project greatly decreased the number of variables at hand. From the initial forty-four variables collated from multiple systems present in the factory, only two proved to be relevant to the problem we were addressing. This de-compounding effect helped ease the computing resources required to process such vast quantities of data like the ones we have to handle on this project.

4. **Modeling**: is the stage were the results begin to shed some light on the business problem posed during the business understanding phase. Modelling is commonly conducted in multiple iterations and involves a refinement process where models are initially trained using their default parameters and then fine-tuned in order to obtain more robust variations of themselves. If after this refinement the results are still not satisfactory one can revert back to the data preparation phase where manipulations to the data can be made to accommodate the particularities required by the preferred model. The essence of a model transcends the mere implementation of an algorithm on a given dataset; it encompasses a collection of statistical measures and discernible patterns that can be utilized to generate predictions and derive meaningful insights regarding interdependencies when applied to novel data. During the development phase of this thesis work, several

ML algorithms were used to predict both the anomalous processes and regions of anomaly on a fastening process. Given the nature of the problem at hand, three unsupervised and one-class learning algorithms were selected for empirical comparison: Local Outlier Factor (LOF), IF and AE. For bench-marking purposes they were compared against the popular Random Forest (RF). Unlike the the tree core models, the RF was trained on both normal and abnormal records given its supervised learning characteristic. From the initial set of explored models, the One-class Support Vector Machine (OC-SVM) was excluded due to its inability to accommodate large volumes of data.

5. **Evaluation**: From the modeling phase, we determined that the models built are technically correct and effective according to the data mining success criteria defined earlier. However, its results must be compared against the business success criteria established at the beginning of the project. This stage usually comprises the selection of the final model and putting together conclusions drawn from the models themselves. The latter are also known as findings. During the development of this work two main metrics were adopted for the evaluation stage: Receiver Operating Characteristic (ROC) (assessed from an Area Under the ROC Curve (AUC) perspective) and Equal Error Rate (EER) were selected for RO1 and the Mean Absolute Error (MAE) describes the reconstruction error for RO2. Given that each record fed to the model receives a GoF classification, predictions are aggregated per tightening cycle before any evaluation is performed. Additionally a realistic rolling window strategy was employed which proved to be a robust method for simulating an anomaly detection process through time. As this method produces multiple AUC values per iteration, results are accumulated by computing their median values and their paired median differences are statistically compared using the Wilcoxon non parametric test (Hollander, Wolfe, and Chicken, 2013).

6. **Deployment**: involves putting the chosen models into production and making them available to interested parties. This stage may also involve training stakeholders on how to use the models and providing ongoing support and maintenance. The deployment stage is critical for ensuring that the models are effectively used to deliver business value. Besides these activities, the deployment stage typically includes the generation of a report, devising a monitoring and maintenance plan and a review of the project as per described by Chapman et al. (2000). The RO3 of this work encompasses the initial steps towards the deployment of the solution under an IDSS umbrella. Given the difference in pace between industry and research it was not possible to fully assess the quality of the final solution in a extended environment. However, the preliminary results suggest that the strategic decisions made throughout the development cycle and the produced artifact efficiently address the problematic being studied.

The CRISP-DM methodology has become the *de facto* standard for data mining projects and is used by organizations around the world. Fig. 7 summarizes all CRISP-DM stages and the generic tasks and outputs for each phase.

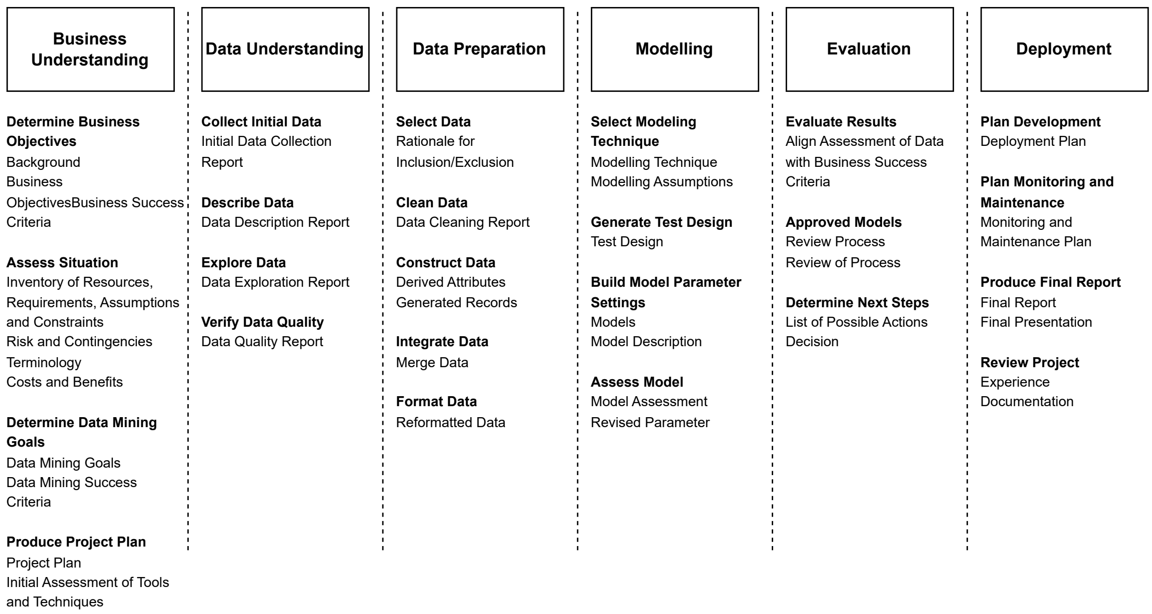


Figure 7: Overview of the CRISP-DM phases, generic tasks (**bold**) and outputs, adapted from Chapman et al. (2000).

2.5.2 Machine Learning Algorithms

ML encompasses a diverse range of types, such as supervised, unsupervised, semi-supervised, and reinforcement learning. Supervised learning relies on labeled datasets, where inputs are mapped to corresponding outputs. Algorithms are trained using this labeled data to make accurate predictions or classifications. It is commonly used in problems where the output variable is categorical (classification) or continuous (regression). Popular supervised learning algorithms include Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), and Support Vector Machine (SVM). Unsupervised learning operates on unlabeled datasets, without specific output labels. The goal is to discover patterns, similarities, and differences within the data. Clustering and association are two categories of unsupervised learning. Clustering algorithms group data points based on their similarities, while association algorithms identify relationships and dependencies among variables. In this work, we particularly address OCC tasks, which can be assumed as an unsupervised form of learning, since the data-driven models are trained only with normal (the most common) instances, thus the labeled attribute is not used to generate the ML models. The adopted OCC ML algorithms are described in the next subsection. Depending on the availability and nature of the training data, it is essential to select the appropriate ML technique. As detailed in Chapter 3, this PhD work exclusively focused on the development and implementation of unsupervised learning models. This decision was driven by two key factors. Firstly, quality labeled data is often costly, requiring a manual inspection that is prohibitive when handling big data. Secondly, within the screw tightening domain, the learning examples are highly unbalanced, with most cases being related with normal assembly conditions. To establish a benchmark for performance comparison, a RF model was trained and used as a baseline for evaluating the models developed in this work. Further details on the methodology and experimental results can be found in Chapter 3. Furthermore, as illustrated by RO2, the primary objective of applying ML to the problem at hand is to classify fastening processes as either good or bad. Therefore, the selection of models needs to be narrowed down to those capable of performing such tasks. Typically , ML encompasses two distinct approaches: classical ML algorithms and deep learning algorithms.

Given the importance of OCC tasks, there are several ML algorithms that can produce data-driven models using only normal training examples. This section lists the OCC non deep learning methods that were explored in this work and the already mentioned RF baseline:

- **IF**: The IF algorithm (F. T. Liu, Ting, and Z. Zhou, 2008; Ding and Fei, 2013) is an unsupervised anomaly detection method that separates anomalous data points from normal ones. It achieves this by creating a collection of smaller decision trees, known as Isolation Tree (iTree)s, which isolate anomalous instances by splitting the data using random parameters. Anomalies, being few and different from normal data, are more easily separated and positioned closer to the root of the tree. The algorithm forms an ensemble of these iTrees, called an IF, to collectively identify and discriminate between good and bad data. By evaluating the average path lengths within the forest, anomalies are identified as data points with shorter path lengths, while normal data points have longer path lengths. Data falling in between these extremes are considered potential anomalies.

- **LOF**: Introduced by Breunig et al. (2000) and widely used for unsupervised anomaly detection (i.e. Cheng, Zou, and Dong (2019) and Alghushairy et al. (2021)), operates by calculating the local density deviation of a specific data point in relation to its neighboring points. LOF identifies outliers as samples that exhibit significantly lower density compared to their neighbors. If a point is deemed an outlier based on its local neighborhood, it is considered a local outlier. This approach leverages the density of the surrounding data to identify anomalies effectively. LOF is particularly advantageous in scenarios where the data density varies across the dataset. Consequently, LOF assigns an outlier factor to each point, quantifying its deviation from the dataset. To ensure a fair comparison, in this study, the LOF model is trained solely on normal examples, employing the outlier factor score directly as the anomaly decision score. While the IF algorithm is renowned for its proficiency in identifying global outliers, it may not perform as effectively in detecting local outliers. In contrast, the LOF algorithm excels in local outlier detection, successfully capturing anomalies characterized by low density compared to their neighboring data points. However, it should be noted that LOF is associated with high time complexity due to its intricate calculations and neighborhood-based approach (Cheng, Zou, and Dong, 2019).

- **OC-SVM**: SVM have proven to be highly successful in ML, originating from the work of Vapnik (1963). SVMs use linear models to achieve nonlinear boundaries by transforming input data into higher dimensional spaces. This allows them to represent complex boundaries in the original space. For classification, SVMs find the hyperplane that maximally separates the dataset, aiming to minimize generalization error (Witten, Frank, and Hall, 2011). In anomaly detection, the OC-SVM offers a semi-supervised approach suggested by Schölkopf et al. (2001). By adapting SVMs to the one-class classification problem, the OC-SVM treats the origin as the sole representative of the second class. Using relaxation parameters, the OC-SVM separates the one class from the origin and applies standard OC-SVM techniques. This method requires only normal data for training, enabling the detection of anomalies without prior knowledge. However, the OC-SVM is sensitive to outliers and may pose computational challenges when applied to large datasets, limiting its suitability for certain scenarios (Schölkopf et al., 2001).

- **RF**: First introduced by Breiman (2001) and extensively studied by, Genuer, Poggi, and Tuleau-Malot (2010) and Biau and Scornet (2015). Is a highly popular and efficient algorithm for classification and regression tasks. It leverages the aggregation of a large number of de-correlated decision trees, combining bootstrap aggregation (random sampling with replacement) and random feature selection. These techniques mitigate variance, reduce the risk of overfitting, and enhance prediction performance (Witten, Frank, and Hall, 2011). Each tree in the RF model is constructed using several bootstrap samples randomly selected from the original dataset, employing the Classification and Regression Tree (CART) method and the Decrease Gini Impurity (DGI) as the splitting criterion. During the tree building process, splits are made to maximize the CART criterion, considering a specific number of randomly selected candidate features (often referred to as *mtry*) (Genuer, Poggi, and Tuleau-Malot, 2010; Biau and Scornet, 2015). The combination of these techniques in the RF algorithm enhances its robustness and prediction capabilities.

2.5.3 Deep Learning

Deep learning algorithms, particularly deep neural networks, have gained immense popularity in recent years for their ability to learn hierarchical representations of data through interconnected layers of neurons (Bengio, 2009). This breakthrough approach eliminates the need for manual feature engineering, as deep learning models extracts relevant features from raw data. This flexibility allows them to effectively capture complex patterns and dependencies in various domains, including image and speech recognition, natural language processing, and reinforcement learning. While deep learning has revolutionized ML, it does come with certain requirements and limitations. Deep learning algorithms typically demand large amounts of labeled data and significant computational resources for effective training. Additionally, the high complexity and black-box nature of deep neural networks can make them less interpretable compared to classical ML models. However, despite these challenges, deep learning continues to push the boundaries of the field, particularly in domains where raw, high-dimensional data is abundant, and automatic learning of intricate patterns is crucial. In the realm of deep learning, researchers have identified two common phases in model development. The first phase involves initializing the model with unsupervised methods, allowing it to learn the underlying data distribution (Srivastava et al., 2014). The second phase focuses on enhancing the model's performance through supervised learning algorithms, such as back-propagation (Vaswani et al., 2017).

Various deep learning structures find applications in anomaly detection, image recognition, and voice recognition, among others. These structures pave the way for further advancements in deep learning, enabling the development of more powerful and versatile models. One of the most popular deep learning algorithms in the context of anomaly detection classification is the AE. Deep AEs have emerged as powerful tools for modeling high-dimensional data in unsupervised settings. They consist of an encoder and a decoder (Bengio et al., 2006; Kingma and Welling, 2014). The encoder compresses the input data into a latent encoding, while the decoder reconstructs the data from the encoding. By minimizing the reconstruction error on normal data, AEs are trained for anomaly detection (Zhao et al., 2017; Zong et al., 2018). However, it is not always the case that anomalies result in higher reconstruction errors, as observed in previous studies (Zong et al., 2018). Anomalies may be well-reconstructed if they share compositional patterns with normal training data or if the decoder is highly effective in decoding abnormal encodings. AEs are therefore neural networks trained through unsupervised learning to minimize the discrepancy between the original input and its reconstruction (Vincent et al., 2010). Comprising an encoder and a decoder, AEs map the input to a hidden representation and back to the original input space. Deep AEs can be created by stacking multiple AEs, enabling the modeling of complex data representations (Vincent et al., 2010). AE-based anomaly detection is a deviation-based approach that employs unsupervised learning (C. Zhou and Paffenroth, 2017; Z. Chen et al., 2018; Gong et al., 2019). By using the reconstruction error as an anomaly score, AEs identify data points with high reconstruction errors as anomalies. These AEs are trained only on normal instances, making them proficient at reconstructing normal data while struggling with unseen anomaly data.

2.5.4 Machine Learning Evaluation Metrics

When developing ML models, it is crucial to evaluate and assess their performance using a range of metrics. These metrics provide valuable insights into how effectively the model performs tasks such as classification, time-series prediction, and regression. By utilizing specific metrics, informed decisions can be made, enabling the selection of preferred models. Evaluating the model's performance is an essential aspect of quantifying its effectiveness (Witten, Frank, and Hall, 2011; Galar et al., 2012). In the case of classification models, their overall performance is often measured using the AUC obtained from ROC analysis. AUC-ROC, or simply AUC, serves as an indicator of how well the model distinguishes between different scenarios. In ROC analysis, the model's predictions are treated as probabilities (p) for a binary class. A decision threshold (*D*) is set, and if $p > D$, the class is considered true. By using a fixed threshold (see Section 3.6 for details on a proposed alternative method), the predicted class labels can be used to compute the well-known confusion matrix. Fig. 8 provides an example of such a matrix, which aligns predicted outcomes with actual values and presents four main statistics for binary classification tasks, where:

1. **True Positives (TP)** : number of outcomes where the model correctly predicts the positive class;

2. **True Negatives (TN)** : number of outcomes where the model correctly predicts the negative class;

3. **False Positives (FP)** : number of outcomes where the model incorrectly predicts the positive class;

4. **False Negatives (FN)** : number of outcomes where the model incorrectly predicts the negative class;

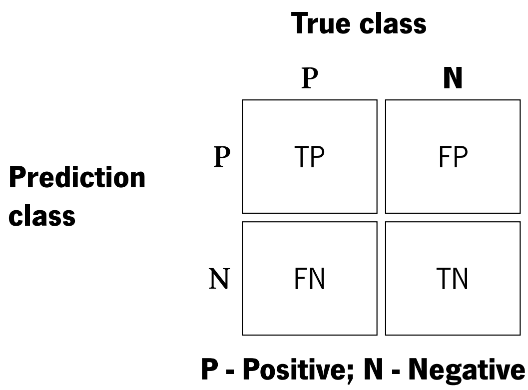


Figure 8: Confusion matrix for a binary classification task.

The confusion matrix provides valuable statistics and insights that can be derived to evaluate model performance. Metrics like True Positive Rate (TPR), True Negative Rate (TNR), Positive Predictive Value (PPV), F1 score, and accuracy are defined using the following formulas (II, 2005; Sun, Wong, and Kamel, 2009):

- $TPR = \frac{TP}{TP+TN}$, also known as recall or sensitivity;
- $TNR = \frac{TN}{TN+FP}$, also known as specificity;
- $PPV = \frac{TP}{TP+FP}$, also known as precision;
- $FPR = \frac{FP}{FP+TN}$;
- $ACC = \frac{TP+TN}{TP+TN+FP+FN}$;
- $F1 = 2 * \frac{PPV*TPR}{PPV+TPR}$.

The ROC curve is a graphical representation technique used to visualize, organize, and select classifiers based on their performance (Sun, Wong, and Kamel, 2009; Fawcett, 2006b). It presents a curve that summarizes the trade-off between TPR (*y*-axis) and False Positive Rate (FPR) (*x*-axis) at threshold points *D* ranging from 0.0 to 1.0. The AUC-ROC measures the quality of the probabilistic classifier and can be calculated using Equation (2.1). An AUC-ROC value of 0.5 corresponds to a random classifier, while a value of 1 represents a perfect classifier.

$$AUC - ROC = \int_0^1 \frac{TP}{TP + FN} d \frac{FP}{FP + TN} d = \int_0^1 \frac{TP}{P} d \left(\frac{FP}{N} \right) \tag{2.1}$$

These metrics allow us to gain a comprehensive understanding of the model's performance and effectiveness in various classification tasks. Considering the circumstances in which this study was conducted, the objective was to optimize two aspects: the maximization of the AUC and the minimization of FN. Reducing the number of FN is crucial because it ensures that the model doesn't mistakenly classify a faulty screwing process as good. Such misclassifications would result in defective parts progressing along the assembly line, which is undesirable and most importantly, expensive.

2.6 Big Data

In recent years, the volume of data being generated has grown exponentially, and this trend is expected to continue in the future. This surge in data has led to the emergence of Big Data (BD), which refers to the large and complex data sets that are difficult to process and analyze using traditional data management and analysis tools (Lee, Kao, and Yang, 2014). With the advancement of technology, organizations now have access to a wide range of tools and technologies that enable them to manage, process, and analyze massive amounts of data.

BD is frequently characterized by its dimensions, known as its "Vs." Previous explanations of BD (Narasimhan and Bhuvaneshwari, 2014) emphasized three Vs (volume, velocity, and variety), but a more widely acknowledged definition currently incorporates the following four Vs (volume, velocity, variety, and veracity). It should be emphasized that additional Vs can also be identified in existing literature (W. Fan and Bifet, 2012; Demchenko et al., 2013). For instance, value is frequently included as a fifth V. However, value is defined as the intended result of BD analysis rather than an inherent attribute of BD (Khan, Uddin, and Gupta, 2014). Due to this rationale, the present thesis exclusively focuses on the four dimensions that define BD (Kune et al., 2016).

In this context, the emergence of the "4Vs of BD" framework offers valuable insights into comprehending the fundamental attributes of large-scale data. The 4Vs encompass:

- **Volume**: The amount of data generated and collected by organizations is vast and ranges from terabytes to exabytes. This data is sourced from various channels, such as social media, e-commerce, and IoT devices (Beyer and Laney, 2012; Parker, 2012).

- **Velocity**: Real-time or near real-time processing of data is a crucial aspect of BD. The high speed at which data is generated and updated can make it challenging for organizations to keep up with (Beyer and Laney, 2012).

- **Variety**: The concept of variety refers to the wide range of data types and formats that exist, including structured, semi-structured, and unstructured data. While structured data is relatively easy to manage and analyze, semi-structured data requires more sophisticated analysis methods. Unstructured data, such as text, images, and videos, is the most challenging to manage and analyze, and often requires advanced ML and artificial intelligence techniques (Beyer and Laney, 2012; Jagadish et al., 2014).

- **Veracity**: Pertains to the dependability and credibility of data, which may vary depending on the source and quality. The abundance of data in BD entails a risk of generating inaccurate or unreliable data, which may result in incorrect insights and suboptimal decision making (Beyer and Laney, 2012; Grolinger et al., 2014; L'Heureux et al., 2017).

- **Value**: Analyzing and utilizing BD can provide significant business value to organizations. Effective management and analysis of BD can provide valuable insights into operations and customers, leading to better decision making and improved competitiveness (Jagadish et al., 2014; Khan, Uddin, and Gupta, 2014).

2.7 Challenges of Machine Learning in the Context of Big Data

In general, BD brings forth a range of difficulties and prospects for organizations. Its significance is expected to keep expanding in the future as the generation of data proceeds at an accelerated pace. Nevertheless, employing ML in BD situations also comes with its own set of challenges.

It is a prevailing assumption within the field of ML that an increase in data volume should inherently lead to improved learning capabilities for algorithms, and in turn, produce more accurate results (Grolinger et al., 2014; L'Heureux et al., 2017). However, the advent of BD has precipitated a variety of challenges. More traditional algorithms, by their design, were not intended to accommodate such immense volumes of data. Many ML algorithms were developed under the assumption that datasets would be of manageable size, allowing for the entirety of the data to be accommodated within memory. Further, it was assumed that all data would be readily available at the time of training. These assumptions are fundamentally challenged by BD, resulting in traditional algorithms becoming either non-functional or suffering significant performance deficits. Fig. 9 depicts the facets of BD and their corresponding relation to ML.

The subsequent summary provides a succinct outline of the challenges associated with implementing ML in the realm of BD, organized according to the V dimensions of BD:

1. **Volume:** As data size increases, computations are increasingly costly and potentially infeasible. The curse of modularity (Parker, 2012) affects algorithms that assume data can fit entirely in memory or a single file. Class imbalance (Japkowicz and Stephen, 2002; Ghanavati et al., 2014;

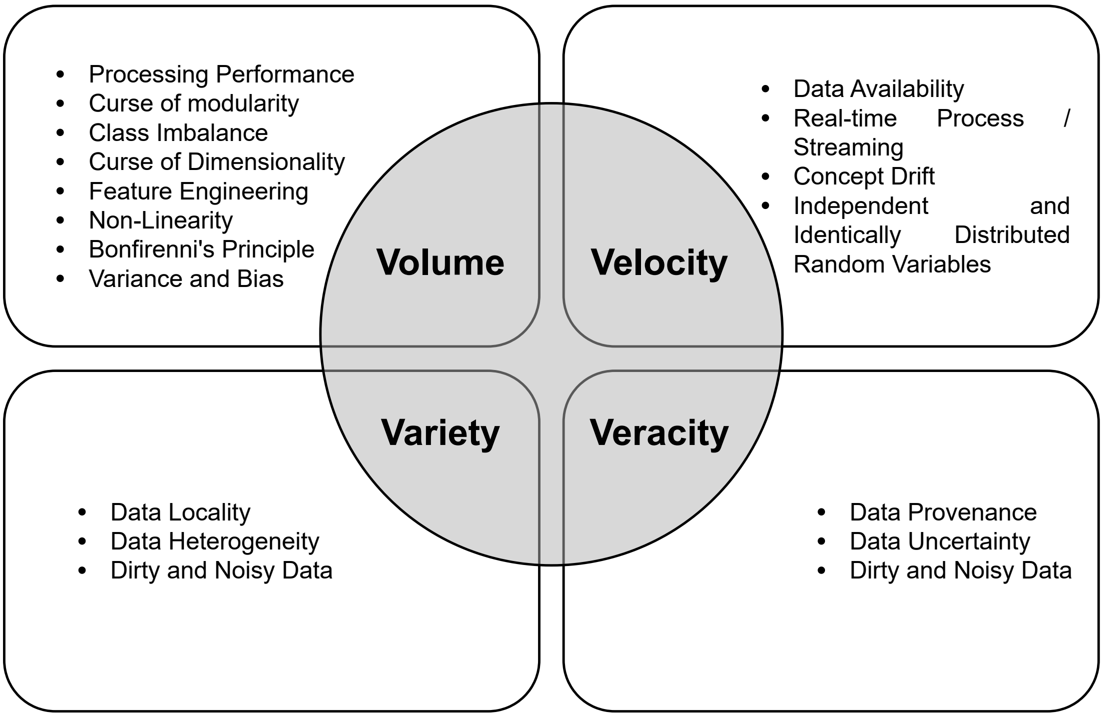


Figure 9: BD characteristics with associated ML challenges, adapted from L'Heureux et al. (2017).

Krawczyk, 2016), the curse of dimensionality (P. M. Domingos, 2012), feature engineering (J. Fan, Han, and H. Liu, 2014; Najafabadi et al., 2015), non-linearity (Kiang, 2003; Lipovetsky, 2022), Bonferonni's principle (Leskovec, Rajaraman, and Ullman, 2014; Calude and Longo, 2017), and variance and bias (P. Domingos, 2000) are additional challenges associated with data volume.

2. **Variety:** Arise when large datasets are distributed across multiple files and locations (Parker, 2012; Leskovec, Rajaraman, and Ullman, 2014). Data heterogeneity poses difficulties in integrating diverse data from various sources (Zheng, 2015). Dirty and noisy data, which may contain errors and inconsistencies, are also challenges in dealing with variety (Swan, 2013; J. Fan, Han, and H. Liu, 2014).

3. **Velocity:** Challenges in data availability arise when data arrive continuously or at non-real-time intervals (Geng and Smith-Miles, 2009). Real-time processing refers to the need for processing fast-arriving data in real-time or near-real-time (Gandomi and Haider, 2015). Concept drift occurs as new data continuously arrive, leading to changes in the distribution of the target output (Dongre and Malik, 2014). The assumption of independent and identically distributed random variables may be broken with BD, posing challenges in ML (Parker, 2012).

4. **Veracity:** Involve tracing and recording the origin and movement of data, which becomes complex and computationally costly with BD (Wang et al., 2015). Data uncertainty arises due to imprecise

data sources and methods of data collection (Jagadish et al., 2014). Dirty and noisy data, which contain inaccuracies and inconsistencies, also affect the veracity of BD (Lovelace et al., 2016).

Addressing these challenges requires innovative approaches and methodologies tailored to BD. Strategies such as parallel computing, distributed processing frameworks, incremental learning, real-time processing systems, concept drift detection, and advanced techniques for handling noisy and uncertain data are crucial in overcoming these challenges and ensuring the reliability and quality of data for ML.

Distributed computing refers to a computing system in which software components are shared among multiple computers or processors, enabling them to work together as a unified system (Coulouris, Dollimore, and Kindberg, 2002). This approach can improve the efficiency and performance of the software by utilizing the resources of multiple computers simultaneously, even when they are geographically distributed (Coulouris, Dollimore, and Kindberg, 2002; Steen and Tanenbaum, 2016). While distributed computing can be challenging to implement, it offers many benefits over centralized systems, including scalability and redundancy. A distributed system can consist of many possible configurations, such as mainframes, personal computers, workstations, and even minicomputers.

Distributed computing plays a crucial role in ML, enabling the training of large-scale models with vast amounts of data. ML models require significant computational resources, and distributed computing allows for parallel processing of data across multiple machines. This approach significantly reduces training time and improves model accuracy. In the context of handling BD, distributed computing strategies can be employed. These strategies include:

1. **Vertical Scaling:** Enhancing the resources of a single node, such as multi-core Central Processing Unit (CPU)s, supercomputers, Graphics Processing Unit (GPU)s, and Field Programmable Gate Arrays (FPGA)s. Vertical scaling has limitations, including memory constraints and limited applicability (Nvidia, 2016).

2. **Horizontal Scaling:** Distributing processing across networked nodes. Horizontal scaling can be further divided into two categories:

   a) **Batch-Oriented Systems:** Prioritizing throughput over latency, batch-oriented systems include solutions such as Apache Hadoop and NIMBLE (Ghoting et al., 2011). They address issues such as the curse of modularity and data locality but only partially resolve the curse of dimensionality. In the context of the current work, this was the selected approach due to the constraints associated with performing this PhD thesis at Bosch AE/P.

   b) **Stream-Oriented Systems:** Operating on real-time or near real-time data, stream-oriented systems include examples like Apache Storm and Spark Streaming. They mitigate processing performance and real-time processing issues but are suitable only for simple ML tasks.

Frameworks like Apache Spark and TensorFlow are widely used in the industry for distributed ML. As data scales, the underlying data storage and transfer architectures become increasingly critical for algorithm performance. The emergence of Resilient Distributed Dataset (RDD)s has revolutionized in-memory computations on large clusters (Zaharia, Chowdhury, Das, et al., 2012). RDDs have been successfully implemented within the Spark cluster computing framework (Zaharia, Chowdhury, Franklin, et al., 2010; Zaharia, Chowdhury, Das, et al., 2012). One popular tool for distributed computing in ML is PySpark, a Python-based open-source framework. PySpark provides a comprehensive set of Application Programming Interface (API)s for complex data processing tasks, including data ingestion, transformation, analysis, and ML algorithms for classification, regression, clustering, and more. PySpark is particularly advantageous for BD applications as it efficiently scales to handle large datasets. It seamlessly integrates with other data processing and ML tools such as Apache Hadoop, Apache Hive, and Apache Mahout.

The significance of distributed computing in ML is evidenced by various applications. Farber (2011) detailed how to program and perform distributed GPU computations that exceeded the capabilities of a single-node GPU. Sterling, Anderson, and Brodowicz (2018) implemented the widely known MapReduce (Dean and Ghemawat, 2008) framework using distributed computing techniques.

In conclusion, distributed computing is essential for ML, enabling the training of large-scale models with extensive datasets. Strategies such as vertical and horizontal scaling offer ways to distribute the workload across multiple nodes. Frameworks like Apache Spark and PySpark provide powerful tools for implementing distributed computing in ML applications. The advancements in distributed computing have opened up new possibilities for handling and analyzing large-scale data in various domains, including ML.

2.8 Related Work

Screwdriving is among the common assembly methods used in industry that have been particularly difficult to fully automate despite several continual efforts (Chandola, Banerjee, and Kumar, 2009). A simpler characterization of the screwing curve using the angle-torque pairs was early proposed by Bickford (1998) and in our previous work (Ribeiro, Matos, Cortez, et al., 2021). Moreover, the angle-torque curve was also adopted by the ISO rotary tool evaluation standards *ISO 5393:2017 Rotary tools for threaded fasteners — Performance test method* (2017). With the lower availability of anomalous data, anomaly detection is often regarded as a one-class problem, which is a form of unsupervised learning where fault detection systems are trained only on normal data. Classic algorithms such as the LOF, OC-SVM and the more recent tree ensemble IF method seem to perform relatively well on many of the uses cases, as reported by Breunig et al. (2000) and Alla and Adari (2019). The main drawbacks of these algorithms is that they tend to require more computational effort, also offering no online update capability, thus being less recommended to handle bigger volumes of data. Deep learning AEs are less affected by such computational limitations. For instance, a linear increase in the training set size typically results in an exponential computational cost growth in terms of the OC-SVM training, while for the AE model the cost increase is also linear. Moreover, the training of AEs can be speedup by adopting GPUs. Thus, AEs are being increasingly adopted for anomaly detection tasks (C. Zhou and Paffenroth, 2017). To the best of our knowledge, only a small portion of research studies have employed AEs in this domain.

Considering the industrial screw tightening domain, the work of Ferhat et al. (2021) demonstrated the use of Bayesian rules and distance rejection heuristics to incrementally discover new manufacturing defects, which were then clustered for interpretation purposes by a manufacturing expert. Using a similar clustering principle, Diez-Olivan et al. (2017) proposed a kernel density-based pattern architecture that can successfully identify patterns in data and correctly categorize them as good or bad. However, the proposed kernel density-based architecture is too computationally expensive when handling larger datasets, which occurs in our analyzed screw tightening industrial domain. Two supervised learning methods, namely a linear SVM classifier and an Artificial Neural Network (ANN), were compared on Matsuno, Huang, and Fukuda (2013) on their capability to discriminate between four task states: ”Successful”, ”Empty”, ”Failed” and ”Jammed”. Although their discrimination abilities was roughly the same, the linear SVM was slightly faster. While interesting results were achieved, this work diverts from our approach under two important aspects. Firstly, it assumes a supervised learning approach, which requires labeled data that is often costly to collect (e.g., requiring human effort and time). Secondly, the work of Matsuno, Huang, and Fukuda (2013) is more focused on root-cause analysis rather than providing a feasible solution to the problem. Another SVM experiment was conducted by Ponpitakchai (2016), which tried different kernel functions (e.g., linear and polynomial) to monitor screwdriving processes on the cover of hard disks. Unlike our use case, this experiment relies on the driver motor data instead of the data being produced by the handheld machine. Yet, the proposed SVM approach is infeasible for our domain due to the required high computational cost. For instance, when adopting a similar SVM algorithm using a preliminary (and smaller) screw tightening dataset, the learning process halted after 40 hours of execution time. More recently, Cao et al. (2019) designed a supervised learning experiment using pairwise data (angle-torque) and a Long Short-Term Memory (LSTM) deep learning architecture for the purpose of discriminating whether a fastening process is successful or not. While good results were obtained, and similarly to Matsuno, Huang, and Fukuda (2013), the experiments relied on labeled data (with both normal and abnormal cases) that can be particularly difficult to be obtained (e.g., requiring a manual data curation effort).

The capability of a Radial Basis Function (RBF) ANN to differentiate between successful and unsuccessful insertions, including different failure types, is explored in the studies of Klingajay and Giannoccaro (2004), Althoefer, Lara, and L. Seneviratne (2005), and Althoefer, Lara, Zweiri, et al. (2008). This work builds upon their previous research presented on Lara, Althoefer, and L. D. Seneviratne (1999). The authors conduct a comprehensive investigation employing model-based simulations and experimental studies to monitor self-tapping screw insertions. The optimization of network architecture through computer simulations reveals that normalizing the insertion signature signals prior to training yields optimal results. Furthermore, the network demonstrates its ability to classify real signals obtained from screw insertions into plastic plates using an electric screwdriver equipped with a torque sensor and an optical encoder. The chosen network structure and training procedure, based on the computer study, effectively classify all signals, even those unseen during training. Notably, after few hundred training cycles, the activation of output nodes distinctly differentiates between successful insertions and failure types, with modest training requirements. Nonetheless, it is important to note that this study relies on the torque signature of a tightening cycle, which may not always be accessible in the machines under investigation.

The existing literature exhibits a scarcity of articles that specifically address the same problem being investigated in this PhD document, highlighting the significant gap in research regarding this particular issue. This underscores the importance of the current study, as it aims to contribute novel insights and fill this notable void in the academic discourse, thereby advancing knowledge and understanding in the field.

2.9 Summary

This chapter serves as an introduction to the fundamental knowledge supporting the development of this PhD. It begins by discussing the paradigms of research in IS, namely the behavioral science and design science paradigms. The chapter then introduces the DSRM-IS methodology, which outlines the investigative process designed to address research gaps and meet the research objectives outlined in the previous section. This method aligns with existing literature, providing a process model for conducting design science research in IS and a framework for presenting and evaluating such research. The context in which the investigation takes place is then detailed, focusing on the I4.0 Revolution. This revolution involves applying CPS concepts to industrial production systems, aiming to enhance operational efficiency, productivity, and automation. The integration of information technologies into traditional industries becomes crucial, as it paves the way for automating various processes on the shop floor. The introduction of IDSS is emphasized, highlighting their potential to revolutionize traditional industry processes by utilizing vast amounts of existing industry data. Given that the core of this PhD thesis revolves around accurately and automatically detecting anomalies in fastening processes, this chapter also delves into the intricacies of these processes. Fasteners serve multiple purposes in applications, including strength, appearance, and reusability. The market acceptance of products heavily relies on the appearance of fasteners. The different types of fasteners and their selection process are explained, considering factors such as availability, end goal, material type, and jointing method. Automation of assembly processes in industries is also explored.

Furthermore, the chapter addresses the implementation of machine learning as a means to overcome challenges faced by traditional assessment approaches. It identifies the main types of machine learning algorithms and pre-selects suitable algorithms based on the available data types and the ability to process large volumes of data. Finally, the related work section highlights the previous contributions made in this particular field of study. It emphasizes the significance of the current work by demonstrating how it builds upon and expands the limited existing research in the area.

Overall, this chapter sets the stage for the subsequent development of the PhD by providing essential background information, outlining the research method, discussing the context of the I4.0, explaining fastening processes, introducing machine learning, and establishing the significance of the current work in relation to previous research.

3 Methods, Experiments and Results

Chapter Table of Contents:

3.1 Introduction

The I4.0 revolution forced most companies to adapt to a more fierce and competitive market where having optimized productive processes can dictate whether a company is successful or not. Within this context and in particular for the assembly industrial sector, there is a pressure to reduce of costs whilst increasing efficiency. Indeed, with the introduction of smart sensors and actuators, several modern factories use autonomous or semi-autonomous robots capable of assisting human operators during production processes. This occurs because some assembly tasks are hard to fully automate and thus it is important to have a human-in-the-loop. An example of such task is the bonding process between plastic parts or between plastic parts and electronic components. These are usually achieved via welding, riveting, glue, or by simply screwing parts together (which is the focus of this paper).

Mechanically combining parts using screws is usually achieved with the help of human-operated handheld screwdrivers. Over time, these machines collect thousands of real-time data points which are later used for quality assessment. Although the collected data might vary from factory to factory, or from handheld screwdriver manufacturers, most assume the minimum collection of angle-torque pairs. On process completion, these pairs are used to plot a tightening curve, which is presented to the operator on a monitor above the station. The operator is then prompted to infer on the process correctness using the available data and a quality estimation produced by the handheld screwdriver machine software. Despite defects being rare in this domain, some units require a more thorough inspection to prevent faulty units from moving to the next assembly stations. Examples of faulty processes are crossed or damaged threads, cracked or split parts and others. To better understand the main reasons why a screwing cycle is deemed faulty, factories make use of the collected data to build a defect catalog.

Due to the dynamic nature of the screw tightening process, which includes the introduction of different models to be assembled over time, there is a constant need to update defect catalogs. Although defect catalogs are empirically proven to work, they use a rather static and meticulous data curation process that involves manual labor and screwdriver software updates. Such rigid defect catalogs are unsuitable for industries that produce millions of fastening movements each day. Thus, there is an opportunity to develop a highly automated screw tightening inspection system by adopting ML algorithms. While there is vast body of knowledge covering anomaly detection in several domains, the advancements in the application of ML to fastening processes is rather scarce. To the best of our knowledge, there are only a few studies which can relate to our approach (as shown in Section 2.8).

In this work, we assume an anomaly detection task that consists in the automatic detection of screw tightening process failures by employing unsupervised ML algorithms. The goal is to feed these algorithms with only normal examples, creating data-driven models that tend to trigger high anomaly scores when faced with "abnormal" data (i.e, outside the learned normal input space). This task is non trivial due to several reasons. Firstly, as already mentioned, the screw tightening process is dynamic (e.g., assembly of new products), thus the ML models need to be continuously updated. Secondly, the data is highly imbalanced, where only a tiny fraction of the examples correspond to failures (e.g., 0.2%). Thirdly, data is generated with a high velocity, thus resulting in BD that requires a high computational effort by some ML algorithms.

In our previous work (Ribeiro, Matos, Cortez, et al., 2021), we performed an initial exploration of a low-dimensional input approach (angle-torque pairs) for anomaly detection for a small subset of the available industrial data (corresponding to two days of production). Promising results were obtained by two unsupervised learning ML algorithms: IF and deep denseAE. In this, we further evaluate the robustness of the proposed screw tightening anomaly detection methods by considering a wider range of assembled products (three distinct products) and a larger number of industrial records, collected during a longer time period (two months). By exploring more recent data, new and different fastening curves were introduced in our system. Using these more richer screw tightening datasets, we conduct several computational experiments, measuring the anomaly detection performance and the required computational effort of the proposed IF and AE models. Moreover, we adopt a realistic and robust rolling window, with several training and testing updates that are simulated over time. We also present a novel dynamic anomaly detection threshold visualization tool that provides XAI knowledge the human operators, supporting the identification of specific angle-torque regions related with faulty screw processes.

Finally, the integration with the shop floor of the factory is effectively demonstrated, underscoring the robust capabilities of the machine learning model in accurately evaluating the quality of the input screw tightening processes. Additionally, the development of a comprehensive, feature-rich dashboard has significantly facilitated ongoing system monitoring for domain experts. This achievement represents a significant contribution and highlights the successful implementation of advanced technologies in industrial settings.

3.2 Industrial Data

This study was conducted within a large electronics manufacturer that supplies assembled components (mainly automotive instrument clusters) to some of the most recognized brands in the automobile industry. The assembly of a cluster is an extensive process with multiple assembly stages. In this work, we focus on one of the final phases in the process, where the plastic housing is combined with either the Printed Circuit Board (PCB)s or other plastic parts. Bonding plastics or electronics to plastics can be achieved via a multitude of techniques that involve glue, welding or the use of screw fasteners. For the remainder of the article we optimize the validation process of screw tightening processes, which makes use of threaded fasteners as the bonding mechanism for the units.

Validating a fastening procedure can be challenging, since it involves several real-time variables. The manufacturing experts take their knowledge into account and leverage the information provided by the handheld driver manufacturers to overcome most of the problems inherent to the usage of this bonding technique. Despite the efforts to mitigate most failure modes with approaches such as sequence based fastening, some deviations still occur on the shop floor and the control mechanisms must be able to signal them fast and accurately.

The plant standardizes most production lines to follow the most efficient layout in terms of assembled clusters throughput per hour. Individually, each assembly station follows a strict set of rules that check for violations in the correctness of the fastening sequence. The assembly process starts with the operator inserting the raw plastic housing in a assembly jig. This jig is designed with the "poka yoke" principle (Shimbun, 1989), which prevents wrong or misaligned parts from being loaded into the assembly jig. A scanner installed in a favorable position reads the imprinted identification code and verifies if the product successfully passed the previous checkpoints and if the station is capable of executing this assembly stage. The correct screw tightening program is then loaded onto the handheld tool. This stage is fundamental because different variants of the same part can be produced on the same machine where the threaded holes can be of different dimensions or be located in different places. The settings specified in this program are also used as a baseline to compare against the real values produced by the machine. The operator then guides the handheld driver to a feeder which is always on and not controlled by the program. Once a screw is loaded on the screwdriver bit, the operator is instructed to follow a predefined sequence with the aid of instructions carefully illustrated on a monitor located above the assembly station. Each inserted screw results in a GoF status message, presented on the screen which indicates whether the fastening succeeded or not. Simultaneously, the produced data is stored locally on a *.csv* file. Depending on the result, two different courses of action can be performed: on failure – the operator is instructed to stop the procedure; on success – the data is uploaded to a remote server where it will be thoroughly analyzed by an expert tool that compares the produced data against a defect catalog. If no defect is detected, the operator is instructed to proceed to the next unit. This new control flow differs from the one presented in our past work (Ribeiro, Matos, Cortez, et al., 2021), as the manufacturing experts deemed it more important to detect false negatives (assembly cycles considered good but are not) than further analysis of faulty units that would have to restart the assembly process.

Regardless of the outcome, new files are made available on a remote data-server (separate from the one running the expert tool) and are then subjected to Extract, Transform and Load (ETL) processes to make data easier to be worked on. Up until this stage, we have no control over the generated data stream. We are currently reading this pre-processed data using *.parquet* files partitioned by date but this process is undergoing some structural changes which will fundamentally change the way we access new data, making it available through Solace, an event broker tool (Solace, 2021).

Table 1 outlines some of the variables collected for each assembly process. The granularity of the data collated depends on a diverse factors, such as the type of the machine in use or the type of product being produced. Some machines do not even support the computation of the variables collected once per fastening (e.g., DTM based attributes). Nevertheless, all machines support the collection of angle and torque values. For each i process, there is a real-time generation of several $k \in \{1, 2, 3, ..., K_i\}$ values, where K_i denotes the total number of observations where each part number can produce a different K_i value. For each unique combination of i and k values, the machine collects hundreds of angle ($\alpha_{i,k}$)

and torque ($\tau_{i,k}$) measurements. It should be highlighted that while there is no direct temporal variable in the collected attributes, the angle ($\alpha_{i,k}$) attribute can be used as a sequential temporal measure of the fastening cycle. In effect, the angle represents the rotation made by the screw during the fastening process. Thus, as the angle value increases, the closer is the screwing cycle to reach its end, which is often marked by an abrupt increase/decrease of the torque value (e.g., screw head fastened to the surface).

A typical screwdriving cycle is characterized by four stages as per described in Table 2. Should a process be successful (Fig. 10), the unit needs to undergo all steps and meet the transition conditions of each sequentially. During the initial stage (step 0), the screwdriving machine rotates counterclockwise in an attempt to latch to the screw head. Although this step is already part of the assembly process it is not relevant for the current analysis and some machines do not even report it. The remaining three steps represent mechanical milestones for the correct fixation of a screw to a plastic housing. There are some specific situations where a successful procedure skips or adds an additional step. The main reason for this behavior is related to the mechanical properties of the bonding units and the capabilities of each assembly station. For example, the DTM attribute is not calculated if the handheld driver does not support such a feature. For all others, the tool estimates the clamping angle and torque for the fastening and then compares them to the actual values (represented by *screw_dtm_clamp_angle and* screw_dtm_clamp_torque). These are some of the computations the assembly machine executes to assess whether a process was finalized successfully or not, returning a GoF label (attribute *screw_gof*).

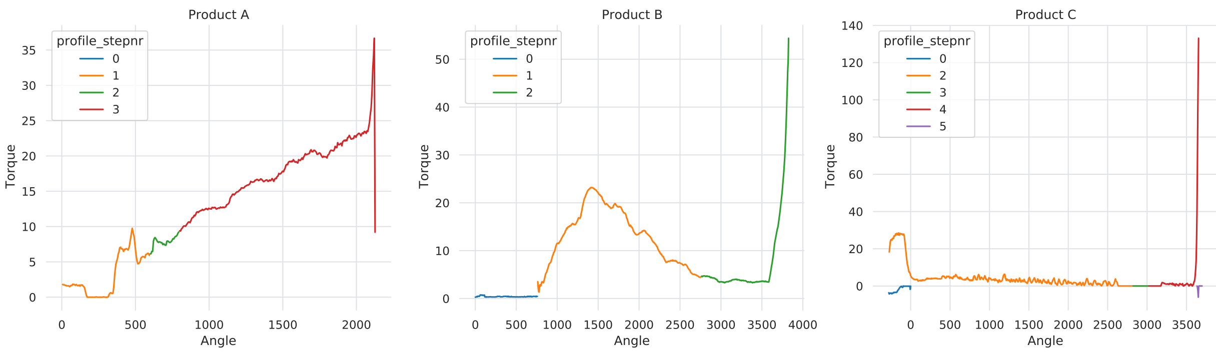


Figure 10: Example of normal ("Good") screw tightening cycles for three different products.

Despite using one-class algorithms and selecting the angle-torque pairs as our input features, the *screw_gof* variable, denoted as y_i for the i-th screw tightening process, is required during the data preparation step. Not only is it used to separate good from bad fastening cycles, it also serves as our target variable during the model evaluation stage. These labels are provided by the screwdriver and are computed by comparing the curve signature against a predefined set of static rules. Although these rules are created and tested by manufacturing experts, some cycles end up being misclassified. These rare occurrences are usually reported and handled as fast as possible. However, and considering the nature of this BD problem, it is not time and computationally feasible to verify all assembly cycles that are part

Table 1: Description of the screw industrial data variables.

Variable	**Description**
	Hundreds of values per screw fastening ():
profile_angle ($\alpha_{i,k}$)	Value for the angle (e.g., -208.1 degrees)
profile_torque ($\tau_{i,k}$)	Value for the torque (e.g., 35.6 Ncm)
profile_gradient	Torque gradient for two consecutive angle values (e.g., 20)
	Four values per fastening (one for each step):
profile_stepnr	Screwing step number (∈{1,2,3,4})
screw_total_angle	Total step angle (e.g., 990 degrees)
screw_total_torque	Total step torque (e.g., 39.3)
	One value per fastening (k):
screw_dtm_clamp_angle	Value for actual DTM Clamp angle(e.g., 79.1)
screw_dtm_clamp_torque	Value for actual DTM Clamp torque (e.g., 30.2)
screw_timestamp	Timestamp (e.g., "2020-10-08 06:30:51")
part_number	Product family identifier (e.g., "2222111")
serial_number	Product serial number (e.g., "11111")
screw_number	Screw number (∈{1,2,...,8})
screw_energy	Total energy required (e.g., 19959)
screw_total_angle	Total angle (e.g., 2993 degrees)
screw_total_torque	Total torque (e.g., 30000 Ncm)
screw_gof (y_i)	Screwing process "Good" (1) or "Fail" (0)

Table 2: Fastening Process Steps.

Step	**Description**	**Transition condition**
Simple Torque Step (1)	Serves as a transition to the next step when the thread starts to be formed	Achieve either the transition torque or angle within the parameterized time range.
Angle Step (2)	Fixed number of turns conducted	Min Angle < Angle Target < Max Angle within stipulated time.
Torque Step (3)	Apply a fixed torque value	Min Torque < Torque Target < Max Torque within stipulated time.
Seating Control Step (4)	Apply a torque value from the screw seating value to the part	Clamping Torque < Max Seating Torque; Min Total Angle < Total Angle < Max Total Angle; Min Total Torque < Total Torque < Max Total Torque.

Table 3: Statistics of the most recently collected datasets.

	Unique Part Numbers	**Unique Serial Numbers**	**Num. Good Screws**	**Num. Bad Screws**	**Ratio**
Product A	52	6,189	8,512	19	0.002
Product B	5	13,160	45,662	99	0.002
Product C	3	4,441	13,163	102	0.008
Total	60	23,790	67,337	220	

of our dataset. Instead, we assume that these occurrences are very rare and thus that they do not impact on our anomaly detection training and evaluation procedures.

A product family is composed of multiple *part_numbers* and is comprised of multiple fastening cycles that share similar characteristics (e.g., angle-torque signature). As such, we grouped our datasets by (*part_number*), serial number (*serial_number*) and screw number (*screw_number*, maximum of 8 screws per individual unit). This grouping allows a fast identification of all k values related with a tightening process (i), which is useful for anomaly detection evaluation purposes.

An initial dataset of around two days worth of screw tightening processes, collected in November of 2020, was first made available. It includes a total of 2,853,967 entries related with N =6,162 fastening cycles. On average, there is around $\overline{K_i}$ =463 records (angle-torque pairs) per cycle.This dataset is highly unbalanced. In effect, there are 74 defects that correspond to only 2% of the screw tightening cycles.

Table 3 describes the more recent and larger dataset that is explored in this paper. In the table, the product names were anonymized for confidentiality reasons (termed here as A, B and C). These were selected based on their availability and in an attempt to cover a diversity within the universe of products fabricated by the analyzed manufacturer. For each individual family, we collected two months worth of data (from February to March 2021), which result in a total of 23,790 unique serial numbers and roughly 26.9 million individual observations (67,337 fastening cycles times 400 data points per screw). As the research literature suggests, and like the initial data, datasets in this domain are usually highly unbalanced. In this case, there is only a tiny percentage of faulty processes (e.g., 0.2% for Product A). Moreover, on average, there are around $\overline{K_i}$ =400 records (angle-torque pairs) per cycle.

3.3 Anomaly Detection Angle-Torque Approach

Given the important of the screw tightening anomaly detection task, several experiments were previously conducted by the manufacturing experts. A large range of experiments were held, including the application of batch process monitoring procedures (MacGregor and Nomikos, 1996). However, this industrial anomaly detection task is non trivial. Unlike other types of processes (e.g., chemical reactions), there can be “normal” abrupt changes in the torque values, as shown in Fig. 10 in terms of the final angle-torque curve for Product A. In effect, inefficient results were achieved when adopting the batch process monitoring attempts. Nevertheless, in one of their attempts, the experts conducted a Principal Component Analysis (PCA) that supported our assumption that angle-torque pairs are fundamental to evaluate screwing processes. Moreover, when we started our research, we performed an initial set of experiments using a larger number of input variables (from Table 1). Yet, the ML models (e.g., LOF, IF, AE) obtained much worst class discrimination results while requiring an expensive computational effort. In some cases, such as when using LOF, the computational training process even halted due to an out-of-memory issue. Based on these results, we then opted to focus on the low-dimensional angle-torque input data approach, which provided better results with a much lower computational effort. Thus, in this work we developed two ML algorithms (IF and AE) that make use of two data features (angle and torque values) as the input values, producing then an anomaly decision score (d). It should be noted that during training, the ML algorithms are only fed with normal instances.

Considering that each fastening cycle is comprised of $k \in \{1, 2, ...K_i\}$ angle-torque observations, the output of each detection model will contain $d_{i,k}$ anomaly decision scores, where i denotes the i-th screw being evaluated. The overall decision score (d_i) is computed by averaging each individual score such that $d_i = \sum_i d_{i,k}/K_i$. For each unique i value, the resulting score (d_i) can be compared to a target label of a test set (y_i), making the computation of the ROC curve (Fawcett, 2006b) possible. Given that human operators require a class label, a fixed Th threshold value is adopted such that for the i-th output is considered anomalous when $d_i > Th$. The overall procedure used to compute the final screw classification score (d_i) is shown in Fig. 11.

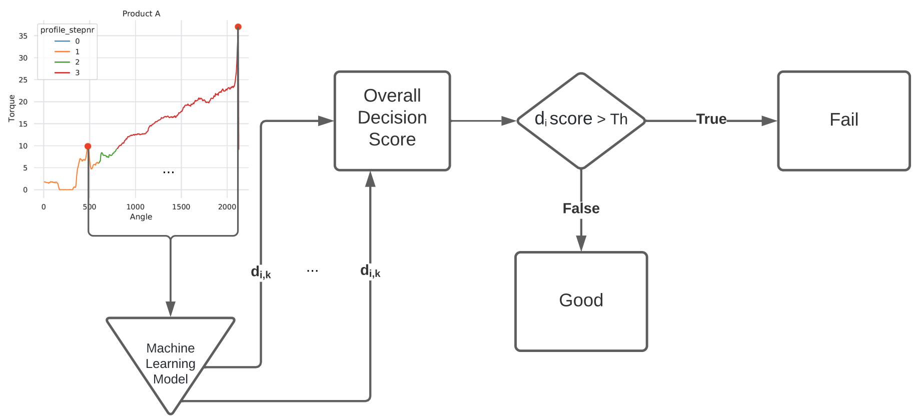


Figure 11: Schematic of the overall screw tightening anomaly detection process.

Selecting the most appropriate Th requires either domain knowledge or a semi-automatic selection of the best specificity-sensitivity trade-off of a ROC curve generated using a validation set (with labeled data). In this work, we assume the usage of domain knowledge, as provided by the screw tightening human operators. To support the expert Th value selection, we propose a specialized XAI interactive visual tool that includes a threshold selection mechanism. The tool allows a finer control over the threshold selection process, allowing experts to manually specify individual thresholds for different families of products and to easily identify problematic angle-torque regions within the screw tightening profile curves.

The data collected during each individual assembly process contains 44 unique attributes spanning 3 granularity levels. Table 1 summarizes the 16 attributes related to the actual fastening process, while the remaining 28 features are used by the assembly management systems. As described in Section 3.2, each process is composed of multiple observations of angle-torque pairs ($\alpha_{i,k}$ - $\tau_{i,k}$) and the torque gradient between two consecutive angle values is grouped by step number. Each step is comprised of multiple fine-grained observations, the total angle and torque for the given step. Subsequently, each step is grouped under a specific fastening identifier which, alongside the aforementioned data, reflects the whole industrial screw tightening process.

While there is a large number of attributes, several experiments were conducted beforehand by manufacturing experts to try and address the problem currently being studied. Experts designed a multitude of experiments using multivariate techniques which proved to be inefficient and incapable of discriminating good from bad screwing cycles. For confidentiality reasons the results of such experiments can not be made public. In one of their runs, the conducted PCA further supported our hypothesis that angle-torque pairs contain the bulk of the information necessary to evaluate screwing processes. Additionally, during the set up of our previous work (Ribeiro, Matos, Cortez, et al., 2021), a group of experiments with multiple combinations of variables and encoding techniques (such as one-hot encoding) which not only increased the overall model complexity but proved to add no extra capacity to our models. As such, and given the previously obtained results using older screw tightening data it was found that the low-dimensional angle-torque input data provided a better anomaly detection performance while requiring much less computational effort. Thus, all screw anomaly methods described in this work assume just two input values: the simpler angle-torque pairs: $(\alpha_{i,k}, \tau_{i,k})$ for each (i, k) example.

3.4 Learning Models

Three unsupervised (one-class) learning algorithms were selected for the empirical comparison: LOF, IF and a AE. These methods are are only trained with normal cases (one-class approach). For benchmarking purposes, we selected the popular RF method. However, we note that RF is a supervised learning method that, contrary to the one-class methods, requires labeled data for the training procedure. Thus, RF was trained with both normal and abnormal angle-torque instances.

To implement the methods we used the *scikit-learn* (for RF, LOF and IF) (Pedregosa et al., 2011) and *TensorFlow* (for AE) (Gulli and Pal, 2017) Python modules. At an initial stage, we also explored the OC-SVM implementation of *scikit-learn*. However, our training sets are too large for the algorithm, which did not compute results in an affordable time (e.g., the first rolling window iteration execution was halted after 40 hours of execution time). In order to provide a fair comparison, and also reduce the computational effort, in the experiments we assumed in general the default *scikit-learn* and *keras* hyperparameter values. Before feeding the models, all angle-torque data values were first normalized, where the numeric values were transformed to fit into a [0,1] scale by using a max-min normalization, via the *MinMaxScaler* procedure of the *sklearn* module.

Unlike the other anomaly detection methods, LOF is a density-based algorithm (Breunig et al., 2000) that heavily depends on K-Nearest Neighbors (KNN) (Fix and Hodges, 1989). It is designed as local because it depends on how well isolated an object is from the surrounding neighborhood. Instead of classifying an object as being an outlier or an inlier, an outlier factor is assigned describing up to which degree the object differs from the rest of the dataset. It should be noted that LOF can be used to detect both local and global outliers. Moreover, it can be trained with multi-class instances. In this paper, in order to provide a fair comparison, we only use normal examples to fit the model, where the outlier factor score is directly used as the anomaly decision score ($d_{i,k}$).

IF is an anomaly detection algorithm which, as the name indicates, uses the isolation principle to classify data as anomalous rather than modeling normality. It leverages the power of smaller decision trees combining them into a single, more capable architecture. As it makes no use of labeled data, it is an unsupervised model. It operates based on the principle that anomalies are few and numerically different from normal instances. This forces anomalous instances to be isolated easier (separated from all other instances) than normal data. Using these principles as foundation, a tree structure is created in an attempt to isolate every single instance by applying multiple splits with random parameters and then assessing on their normality. Anomalous observations with fewer attributes capable of describing them tend to be positioned closer to the root of the tree. This structure is usually regarded as an *iTree* and is the main component of anomaly detection of this algorithm. As such, an IF (F. T. Liu, Ting, and Z. Zhou, 2008) is an ensemble of iTrees that when combined can discriminate between good and bad data. Fig. 12 summarizes the working principle of this algorithm, where the data points with shorter average path lengths are indicated as anomalies. On the other side of the spectrum, data points with bigger average path lengths are considered normal. Data points which fall in between these too extremities are classified as potential anomalies. These distances are computed and expressed by *scikit-learn* as an anomaly score ranging from $\hat{y}_{i,k}$ =-1 (highest abnormal score) to $\hat{y}_{i,k}$ =1 (highest normal score), where $\hat{y}_{i,k}$ denotes the IF output for the k-th angle-torque pair of the i-th screw tightening instance. In order to compute an anomaly probility, we rescale the IF score by computing $d_{i,k} = (1 - \hat{y}_{i,k})/2$.

Unlike the IF, an AE is a type of artificial neural network that learns normality rather than computing abnormality scores. The AE is comprised of two main stages: an initial stage where it learns a representation for a set of data (encoding), typically by reducing the input data (the number of features describing the input data) and is usually unaffected by noise; and a decoding stage, in which the model tries to reconstruct the input signal. This neural network architecture imposes a bottleneck out of the encoding stage (which forces dimensionality reduction), resulting in a compact knowledge image of the original input, usually referred to as latent space. When applied to the specific case of anomaly detection, the architecture accepts normal data as input and then attempts to produce $(\hat{\alpha}_{i,k}, \hat{\tau}_{i,k})$ outputs which should be identical to the input pair $(\alpha_{i,k}, \tau_{i,k})$, where $\alpha_{i,k}$ denotes the angle and $\tau_{i,k}$ the torque for the k-th

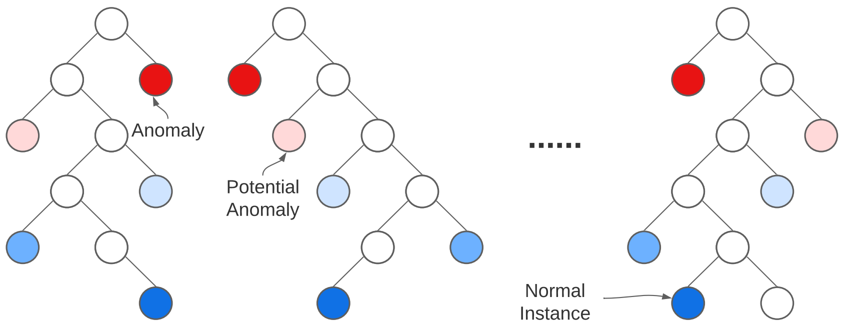


Figure 12: The Isolation Forest working principle, adapted from Regaya, Fadli, and Amira (2021).

measurement of the i-th screw tightening instance. To ascertain the quality of this reconstruction, on each new (i, k) instance we compute its MAE (Alla and Adari, 2019), formulated as follows:

$$MAE_{i,k} = (|\alpha_{i,k} - \hat{\alpha}_{i,k}| + |\tau_{i,k} - \hat{\tau}_{i,k}|)/2 \tag{3.1}$$

The reconstruction MAE is used as the anomaly decision score $d_{i,k} = MAE_{i,k}$ for each input instance, where higher reconstruction errors stand for a higher abnormality probability. As previously mentioned, the architecture we developed assumes $(\alpha_{i,k}, \tau_{i,k})$ pairs as input and produces two output nodes which form a fully-connected structure including a stack of layers for both stages. All intermediate hidden layers are activated using the Rectified Linear Unit (ReLU) function, contrasting with the output nodes which assumes a logistic function (all $\alpha_{i,k}$ and $\tau_{i,k}$ values are normalized ∈[0,1] as previously described). To address and reduce the internal covariate shift, which occurs when the input distribution of the training and test set differ but the output label distribution remains intact, we apply a *BN* layer, discarding the need for dropout layers. In fact, BN also normalizes the layer inputs for each batch of data that passes through it (Ioffe and Szegedy, 2015). The development of this network architecture, all its assumptions and decisions resulted from several trials, conducting during preliminary experiments using older screw tightening data. Each experiment was conducted by imposing one bottleneck layer with 1 hidden node and a varying range of additional hidden layers. The best performing architecture was achieved when the AE was composed of 20 layers (input, *BN* and output layers) as show in Fig. 13. Additionally, all models were trained using the Adam optimization algorithm (which proved to be slightly better than Stochastic Gradient Descent (SGD) in our test case), using the previously described loss function MAE, a total of 100 epochs and early stopping (with 5% of the training data being used as the validation set).

As explained in Section 2.8, the AE model can be adapted to dynamic environments when there are frequent data updates. In Matos et al. (2019), two neural network learning modes were compared: reset and reuse. When new data arrives and the neural network is retrained, the former assumes a random weight initialization, while the latter uses the previously trained weights as the initial set of weights.

As explained in the AE model can be adapted to dynamic environments when there are frequent data updates. In Matos et al. (2019), two neural network learning modes were compared: reset and

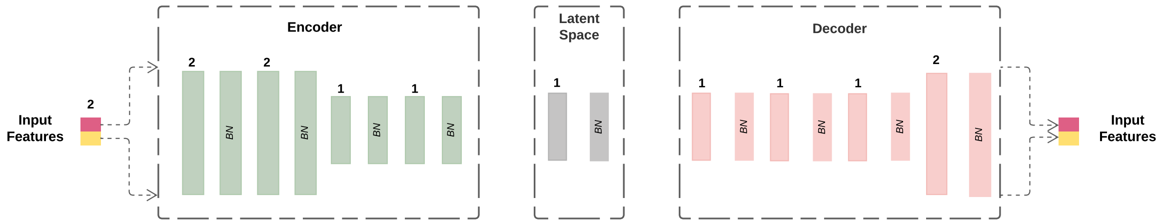


Figure 13: The adopted deep dense AE structure (the numbers denote the number of hidden units per layer and the term BN represents a Batch Normalization layer).

reuse. When new data arrives and the neural network is retrained, the former assumes a random weight initialization, while the latter uses the previously trained weights as the initial set of weights. In this work, we compare both learning modes when executing the rolling window evaluation.

Finally, RF is a supervised learning algorithm that works as an ensemble of decision trees that form a "forest". Each tree depends on the values of a randomly sampled input vector and a bagging selection of training samples (Breiman, 2001). RF can be used for both regression and classification tasks. In this work, we used RF to output anomaly class probabilities ($\in$[0.0,1.0]), which are used as the decision score ($d_{i,k}$).

3.5 Evaluation

To compare the anomaly detection models, we execute the realistic rolling window procedure (Tashman, 2000; Matos et al., 2019). This procedure is more robust than the popular random holdout train-test split, since it realistically simulates an anomaly detection through time, producing several training and test updates, as exemplified in Fig. 14. A fixed training window with W examples is first set. Then, in the first iteration (u =1), the model is trained with the oldest W instances. Once the model is fit, it performs T ahead predictions. Then, it simulates the passage of time by "rolling" the training and test sets by assuming a temporal jump step (S). The oldest S instances are discarded from the training set, being replaced by S more recent ones. A new model is then fit, allowing to perform T new subsequent predictions, and so on. In total, there are $U = (D - (W + T))/S$ model updates (training and test iterations), where D is the total dataset size.

For the initial dataset we adopted the screw tightening granularity, where all k instances for a particular i tightening process are always kept together, thus $D = 6,162$. After consulting the industrial experts, we adopted the values $W = 5,000$, $T = 500$ and $S = 33$, which produces a total of $U = 20$ model updates. Besides the predictive decision scores, for each anomaly detection method we recorded the computational effort, measured in terms of the average (over all u iterations) total training and and anomaly screw tightening inference times (both in s). All experiments were executed using a 2.3 GHz Intel Core i9 processor.

Using the most recently collected data (regarding the three analyzed products), and also with the help

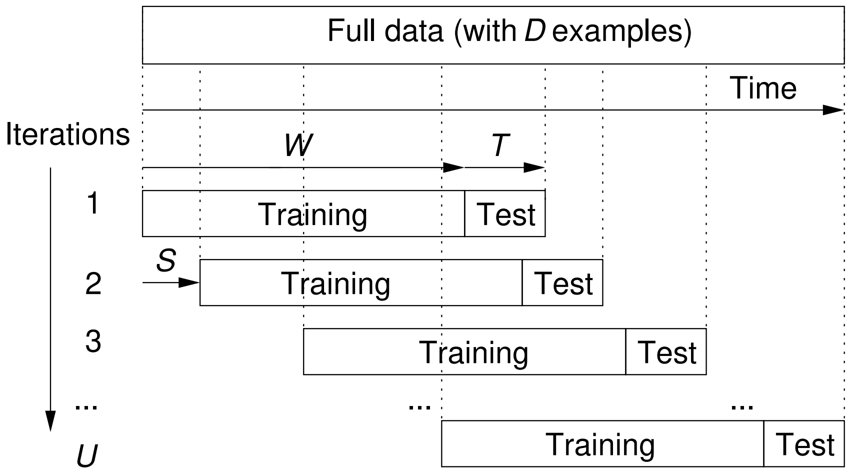


Figure 14: Schematic of the adopted rolling window procedure.

of the manufacturing experts, we assume a total of $U = 8$ standard rolling window train and test iterations by fixing the following values: Product A – W=6,809, T=851 and S=106; Product B – W=36,529, T=4,566 and S=570; Product C – W=10,530, T=1,316 and S=164 (all values related to numbers of screw fastening processes). We particularly note that these values are much higher than the ones employed in the initial dataset. This occurs because previously we have collected only two days of screw industrial data, while the most recent datasets have a substantially higher number of data records (67,337 fastening cycles related with two months). For each ML model, we collected the computational effort, measured in terms of the rolling window average preprocessing, training and testing (inference) time (all in seconds *s*), per anomaly detection method. This is of particular interest to the manufacturing experts that need to deploy the developed systems in an industrial context with near real-time requirements. All experiments were executed on a M1 Pro processor with integrated GPU.

To evaluate the anomaly detection performance, we adopt the ROC curve (Fawcett, 2006a), which produces a visual representation of the ability of a binary classifier to distinguish between normal and anomalous data as its discrimination threshold varies. In the case of both the standard and static training rolling window procedures, the ROC curves are computed by using the target labels and the anomaly scores ($d_{i,k}$) for the T tightening test examples of each rolling window iteration. The overall discrimination ability is then measured by the AUC ($AUC = \int_0^1 ROC\, dTh$) and the EER. As argued in Pereira, Cortez, and Mendes (2021), the AUC measure has several advantages. For instance, the measurement of quality is is unaffected by balancing issues in the dataset (as previously explained, our datasets are highly unbalanced). Additionally, interpreting its results is fairly easy and can be categorized as follows: 50% – performance of a random classifier; 60% - reasonable; 70% - good; 80% - very good; 90% - excellent; and 100% - perfect. The EER is used to predetermine the threshold value for which the false acceptance rate and false rejection rate is equal (Reich and Vijaykumar, 2021). Under this criterion, the lower the value, the more accurate is the classification. Given that multiple AUC and EER values are generated for each rolling window iteration, the experimentation results are aggregated by computing their median values and using the Wilcoxon non parametric statistic test (Hollander, Wolfe, and Chicken, 2013) to check whether the paired median differences are statistically significant.

3.6 Results

3.6.1 Experiments on the initial dataset

Table 4 summarizes the rolling window evaluation results, while Fig. 15 shows the evolution of AUC values through the rolling window iterations. The supervised method (RF) provides an almost perfect median AUC score (99%), which sets a high comparison standard for the unsupervised one-class methods. Regarding LOF, it corresponds to the fastest training method, requiring an average of 18.3 s for each rolling window iteration. However, this method also provides the worst median discrimination capability (AUC of 80%). When comparing both AE learning modes, reuse and reset, the former provides slightly better median AUC results and lower average computational training times (explained by the usage of the early stopping procedure). The best one-class median AUC result is obtained by the IF (99%), which is equivalent to the RF performance and that slightly outperforms the AE reuse method (by 4 percentage points). As shown in Table 4, the median AUC differences are only significant when comparing RF, IF and AEs with LOF. Fig. 15 confirms that the two best one-class performing models continuously provide an excellent (almost perfect) discrimination along the distinct evaluation iterations. Given these results, we recommend the two unsupervised methods IF and AE, since they do not require labeled data (as RF). In particular, IF obtains the highest discrimination capability. As for AE reuse, the method produces an almost similar performance while requiring much less computational effort. In effect, when compared with IF, the AE reuse training is around 7 times faster. Moreover, the AE reuse anomaly inference time is half of the one required by IF and one third of the response time required by RF.

Table 4: Rolling window screw tightening anomaly detection results (best values in **boldface** font; selected models in *italic* font).

	AUC	**Train Time** (s)	**Inference Time** (s)
LOF	0.80	**18.30**	0.12
RF	**0.99***	25.33	0.12
IF	**0.99***	168.18	0.08
AE reset	0.93*	29.64	**0.04**
AE reuse	0.95*	23.00	**0.04**

$\star$ – Statistically significant under a paired comparison with LOF.

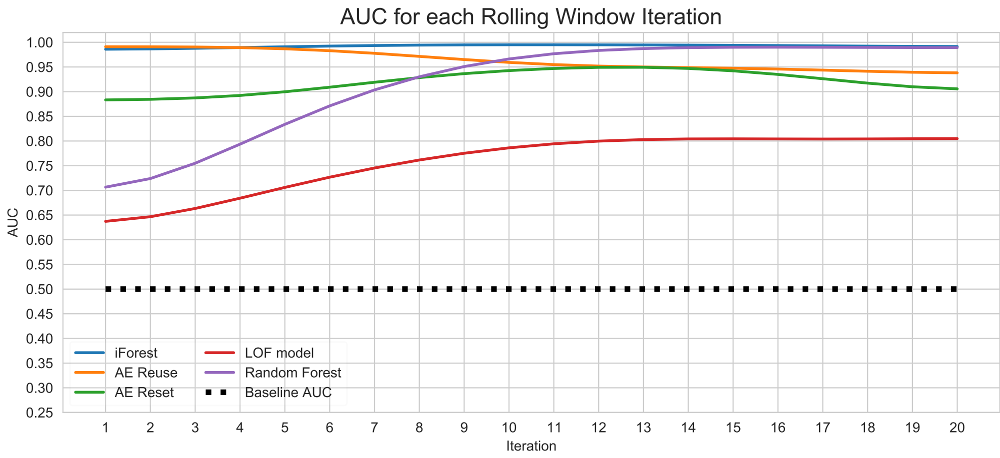


Figure 15: Rolling window AUC evolution for the screw tightening anomaly detection methods (dashed black line denotes the performance of a random baseline classifier).

As an example, Fig. 16 plots the individual ROC curves for the selected models during the third rolling window iteration. The curves overlap in the sensitive (top right) region, with the AE reuse model slightly presenting a better specificity (bottom left region) in this example.

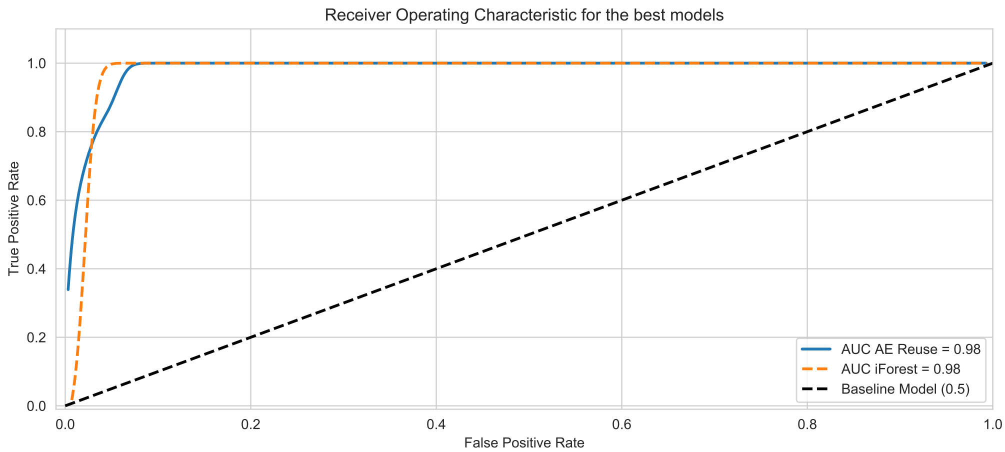


Figure 16: ROC curves for the best two models for the rolling window iteration u=3 (dashed black line denotes the performance of a random baseline classifier).

To demonstrate the secondary goal (identification of the anomaly process angle-torque regions), we selected the IF and AE reuse models that were obtained in the same rolling window iteration and a defect example from the test set. Then, we defined a threshold value (Th) that corresponds to the mean plus one standard deviation of the training decision scores (a different Th value was used for each method). Fig. 17 shows the obtained anomaly points that correspond to the highest decision scores, such that $d_{i,k} > Th$. It is interesting to note that both models signal several identical abnormal regions, such as the three abrupt torque decays (set at the normalized x-axis angle values of 0.30, 0.46 and 0.86).

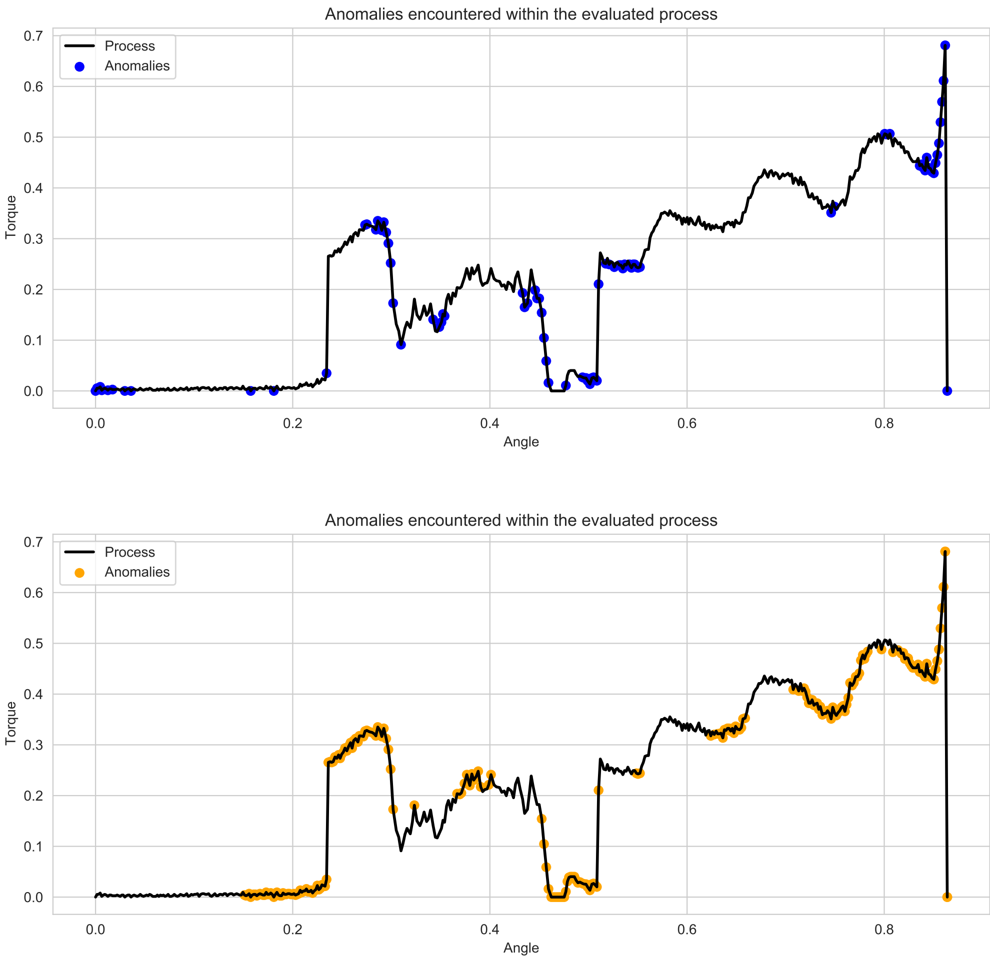


Figure 17: Anomaly points identified by IF (top) and AEreuse (bottom) for a screw fastening defect process.

The obtained results were shown to the industrial experts, who considered them very positive. Besides the high quality AUC results, the IF and AE reuse models allowed to detect one real defect example that was labeled as "normal" by the screw assembly expert system tool. Thus, there is a potential for a better screw tightening defect detection by the proposed data-driven models. Moreover, the secondary goal demonstration example (Fig. 17) was valued as interesting and useful, particularly in the end part of the process (right $x-$axis region) when there is an abrupt torque decay. The experts highlighted that this is one of the defect behaviors. Nevertheless, they also mentioned that there are other types of screw tightening failures (at more initial stages), which will be analysed in future work.

3.6.2 Experiments with Larger and More Recent Data

In this section, we further compare the performance of the two unsupervised anomaly detection methods that previously provided the best results (Section 3.6.1): IF and AE reuse. For such purpose, we adopt more recently collected data and that corresponds to a larger time period (two months). Moreover, the analyzed datasets are associated with three distinct assembled product families, which tend to produce different screw tightening profile curves (as shown in Fig. 10).

Table 5 summarizes the rolling window evaluation results for the three products and the two selected learning architectures (IF and AE reuse), while the individual AUC and EER values (obtained for each rolling window iteration) are shown in Fig. 18. The results attest an excellent discrimination level that was achieved by the two learning methods for all three products, almost reaching the perfect AUC value (in effect, we had to increase the AUC precision values to three digits in order to distinguish them from the maximum AUC=1.0 value Table 5). In terms of the discrimination power, both methods present very similar performances. For instance, the maximum median AUC difference is just 0.7 percentage points). Also, as shown in Fig. 18, the IF and AE reuse curves are very close (e.g., the maximum individual difference for the same iteration is just 3.8 percentage points (second iteration for Product A). Turning to the EER criterion, it clearly favors the AE models, which produce lower values for all three products when compared with the IF model. Moreover, the AE is computationally much faster when compared with IF. In effect, the latter method requires on average around 2.7 times more computational effort for training and around 3.0 times more inference time (to detect if a screwing procedure is anomalous). Given that the screw tightening domain often generates BD, this computationally light property of the AE reuse method is highly valuable.

Table 5: Rolling window screw tightening anomaly detection results (best values in **boldface** font).

		AUC	**EER**	**Train Time** (s)	**Inference Time** (s)
IF	Product A	**0.998**	0.840	115.933	0.018
	Product B	**0.999**	0.899	1664.067	0.029
	Product C	**0.999**	0.917	321.862	0.019
AE	Product A	0.996	**0.793**	**61.250**	**0.005**
	Product B	0.996	**0.832**	**842.869**	**0.012**
	Product C	**0.999**	**0.593**	**76.467**	**0.006**

While the EER criterion can be used to set classification score thresholds, the assumed balance between false acceptance and rejection rate is not applicable to the analyzed industrial screw tightening context, where a higher false acceptance rate is less costly than failing to accept a valid unit. To better explain the anomaly detection results to the human operators, we have developed an interactive tool that assumes a dynamic threshold selection. (Figs. 19a and 19b). The tool works by first loading a trained anomaly detection model. Then, each time a new fastening cycle is analyzed (the inference process), the operator can manually perform a real-time experiment with different threshold values (Th) by using a slider button (shown at the bottom of the figures). Each time a new threshold is selected, angle-torque regions with high anomaly scores (i.e., $d_{i,k} > Th$) are highlighted (yellow colored points in Figs. 19a and 19b). As shown in the graphs, a lower threshold produces a higher angle-torque region selection

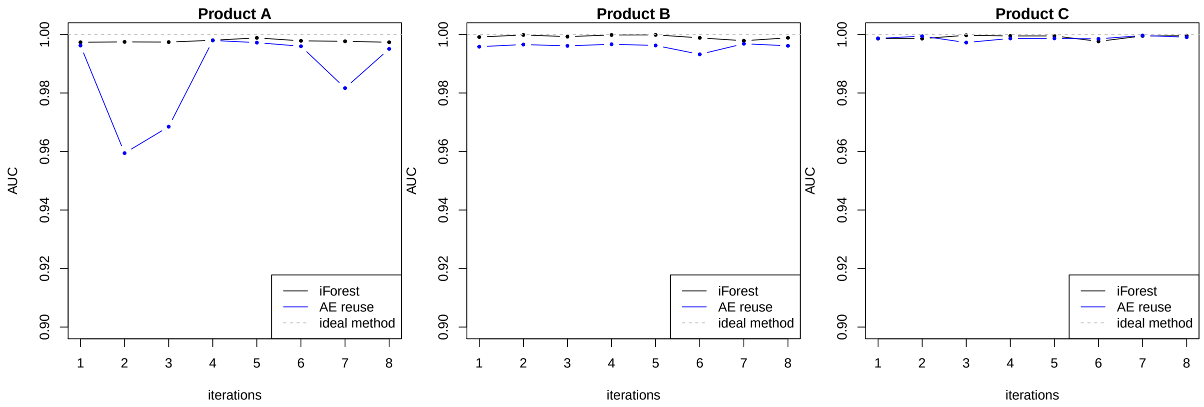


Figure 18: AUC for each rolling window iteration.

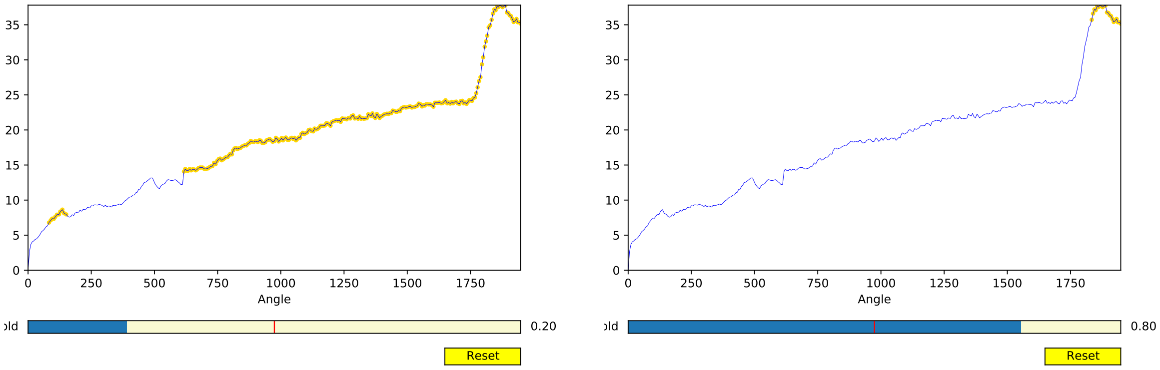
area.

The purpose of the interactive tool is twofold. Firstly, it can be used by the human operators to fix the final threshold, which is adopted to signal all abnormal screw cases. Secondly, it can be used to better reason about the ML anomaly detection decisions. In effect, the interactive threshold graphs works as an XAI tool, helping in the identification of the angle-torque regions that were responsible for the anomaly, which can potentially support the identification of screw anomaly causes.

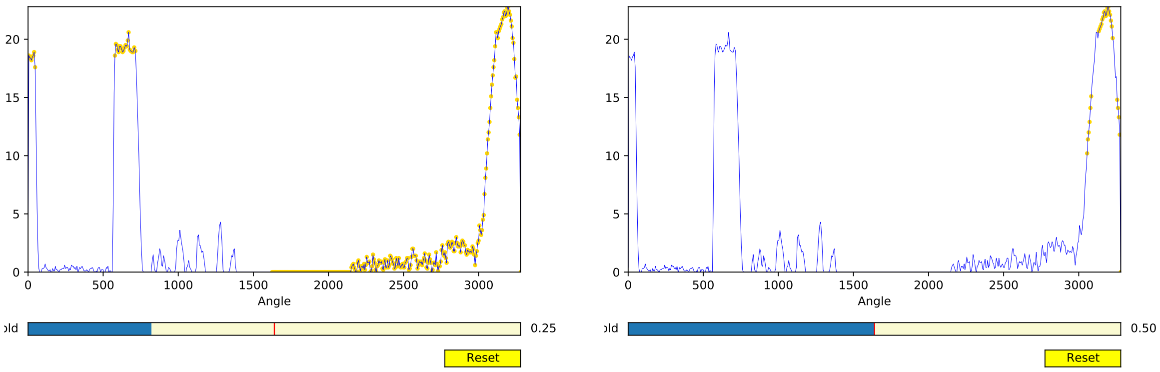
We experimented the tool with both IF and AE resuse methods. In general, the same threshold level produced the same anomaly angle-torque region selection. To simplify the visualization, the two selected XAI screw anomaly examples are related with the AE reuse model. Fig. 19a demonstrates a torque reached too soon error. Although the tightening curve resembles the one presented in Fig. 10, the model correctly detected that the torque values for the corresponding angle were reached too soon (which usually indicates a stripped plastic thread fault). On Fig. 19b a different type of failure mode was detected by the AE model. The unexpectedly steep torque values at the beginning of the process point at a mating problem between the screw and plastic threads. This seems to be confirmed by the expected values of torque being reached at a higher angle (meaning the screwdriver had to perform a few more rotations). The apparent loss of torque in the middle of the curve signature is normal for some parts, given that the tolerances for the screw cavities are rather high on this product.

3.6.3 System Integration on the shop floor

Research in BDA and ML is usually done by different teams, who independently work in providing analytical means to analyse the available data. This work aims to integrate the scientific and technological contributions from both fields, supporting the integration of predictions to enrich a BDW that is used in an advanced data analytics environment that assists decision-makers for better decisions. The proposed architecture (Fig. 20) creates a unified environment between ML processes and the BDW, as the main storage component, in BD contexts. Besides the storage itself, this architecture allows the monitoring of the ML models and the BDW, expanding the analytical scope beyond the decision-making at the business



(a) Example of the dynamic threshold tool usage for a Product A torque reached too soon error.



(b) Example of the dynamic threshold tool usage for a Product B stripped screw error.

level - it is now possible to establish performance metrics for the BDW and the ML models and monitor them over time.

3.6.3.1 Components and Supporting Technologies

The advanced data analytics architecture (Fig. 20) is composed of three main components: Data Sources, BD Cluster, and Visualization Tools. The Data Sources can be of different types, depending on the data they produce/handle: data can be structured, semi-structured, or unstructured. Besides this classification, the data sources may present data that is produced at different speeds, with different sizes and formats, thus justifying the context of BD (Cai and Zhu, 2015). The BD Cluster component integrates two areas, which are Data Lake (DaL) and BDW. A distinction in data storage was made to accommodate analytical data and non-analytical data. The DaL is used to support the storage of any kind of data/processes/models such as Raw Data, *Data Pipelines*, ML Models, among others. The BDW is a storage ecosystem supporting the storage of data modeled as Analytical Object (AO)s, representing highly independent and autonomous entities with focus on analytical subjects in terms of decision support

(M. Y. Santos and Costa, 2020). In the proposed architecture, the DaL has three subareas, the *Standard Raw Data*, the *Data Pipelines Repository*, and the ML *Models Repository* and its interfaces. The purpose of the *Standard Raw Data* subarea is to standardize the data and its access, so that throughout the system data follows the same naming structure, making it easier for all users to understand and use. To be efficient and coherent, this standardization needs the definition of a set of basic rules that must be applied to all the data that is here stored. These rules should follow the DATSIS principles, which enable data to become *Discoverable*, *Addressable*, *Trustworthy*, *Self-describing*, *Interoperable*, and *Secure*. For example, this proposal includes rules for standardization of the attributes' name, sharing the information about the BDW and DaL, and ownership in the Organization Wiki, among others included in organization directives. The *Standard Raw Data* is used to feed two distinct, interrelated subareas, namely the *Data Pipelines Repository* and the ML Models Repository and its Interfaces. More specifically, data is stored in different sets linked to their Business Process (BP)s, where can be accessed in a Jupyter Notebook, which allows it to be read in the form of a Spark Dataframe. These technologies are examples and were the ones used in this implementation. Nevertheless, other technologies with similar purposes can be used, as the technological landscape is quite diverse in this field. The same applies for the other technologies mentioned throughout this paper. *Data Pipelines* in the *Data Pipelines Repository* prepare, transform and enrich data according to the defined data model, creating one or several tables in the BDW. These tables are the physical implementation of the modeled objects. BPs have their data, but they also use data that can be shared between them. This data that can be shared by several BPs is stored inside the Complementary Analytical Objects (CAO) folder of each BP. The ML models in the ML Models Repository provide predictions of events in the data, which brings a competitive advantage for the decision-making process. The modeled AOs integrate historical attributes with predictive attributes obtained from trained ML models. To access these predictions, an interface needs to be established between the *Data Pipelines* and the ML models. This interface is based on a class, developed in PySpark, which encodes a series of functions that allow a ML model to run on the Spark Dataframe. Thus, in the *Data Pipelines* development environment (in this case in Jupyter Notebooks), another notebook containing the ML Class is invoked. After invoking and correctly importing it, the functions in it are applied to the Spark Dataframe that contains the *Standard Raw Data*. The output is a Spark Dataframe, which contains the predictions for the processed events. Once the output of the ML model is obtained, it is integrated into the Data Pipeline. After integrating the predictive outputs, the AO including the historical and predictive attributes is stored in the BDW as a Hive table. The BDW is organized according to the purpose of the data and integrates two distinct subareas: Business Warehouse and Monitoring. This division is important to efficiently store data regarding the BPs and the performance of the BDW and the ML models. Besides training ML models and using them, or creating AOs and storing them in the BDW, their evolution must be monitored over time so that these components can be improved, updated, and maintained. Otherwise, the system can become obsolete or not address performance requirements in an industrial context. For the ML models, for example, due to the volatile nature of the data in a business activity, the data that is used for training the models can change and the models need to evolve to meet the new data needs, thus obtaining more accurate predictions. Once all the data has been integrated and properly stored into the BDW, it is possible to analyse it in the *Visualization Tools component*. This includes data visualizations that support analytical tasks with associate indicators regarding ML (such as accuracy, for instance), Data Processing (such as processing time, for instance), and AOs (such as the number of records, for instance). This component foresees analytical dashboards for the analysis of the different BPs, and for the monitoring of the *Data Storage Processing and Analysis* and the ML models. In the work here presented, the visualizations were implemented in PowerBI . Although the architecture presents the *Monitoring* subarea and the related visualizations, they are considered future work and for this reason are not described in this paper.

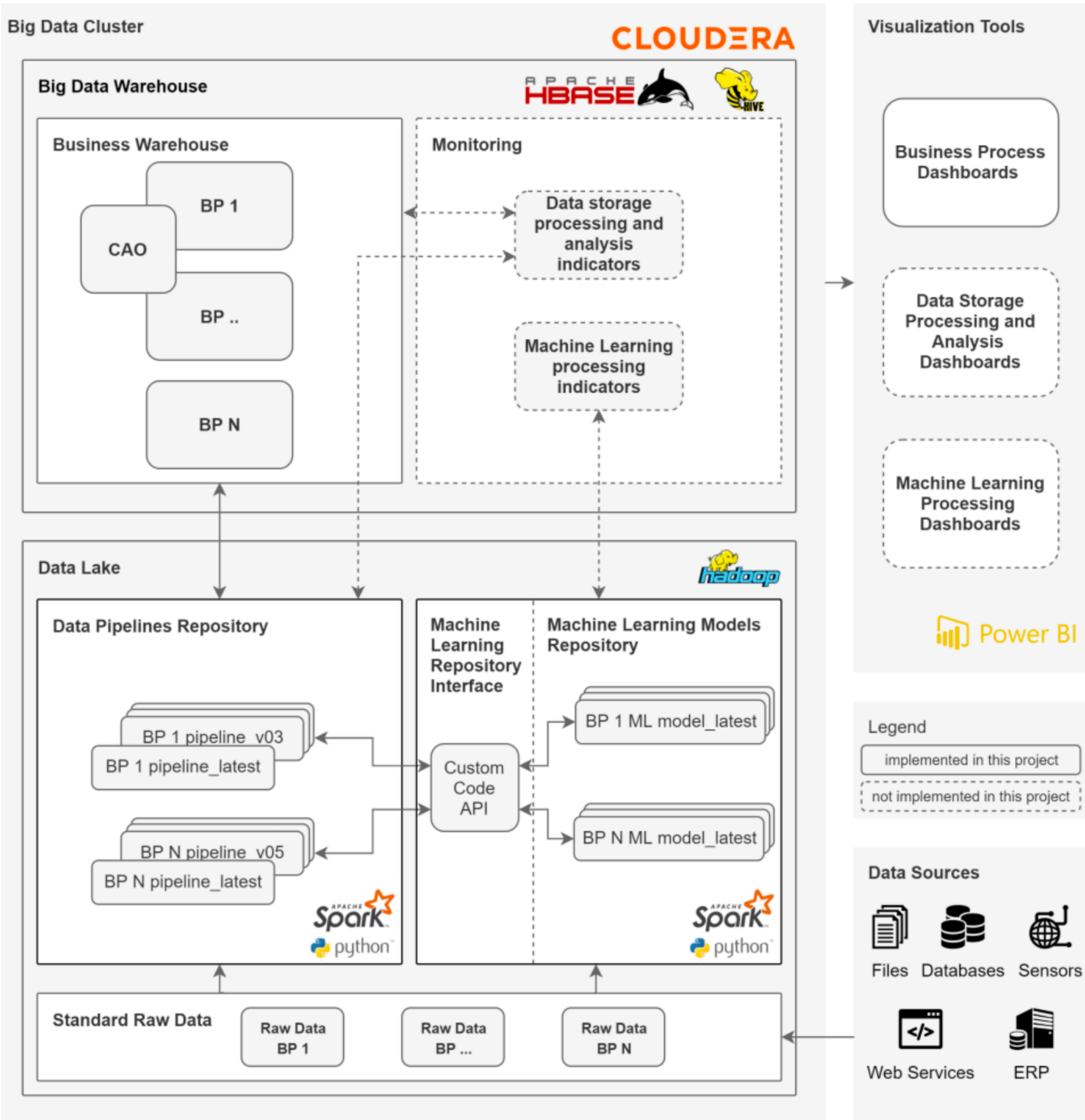


Figure 20: Advanced Data Analytics Architecture, from Galvão et al. (2022).

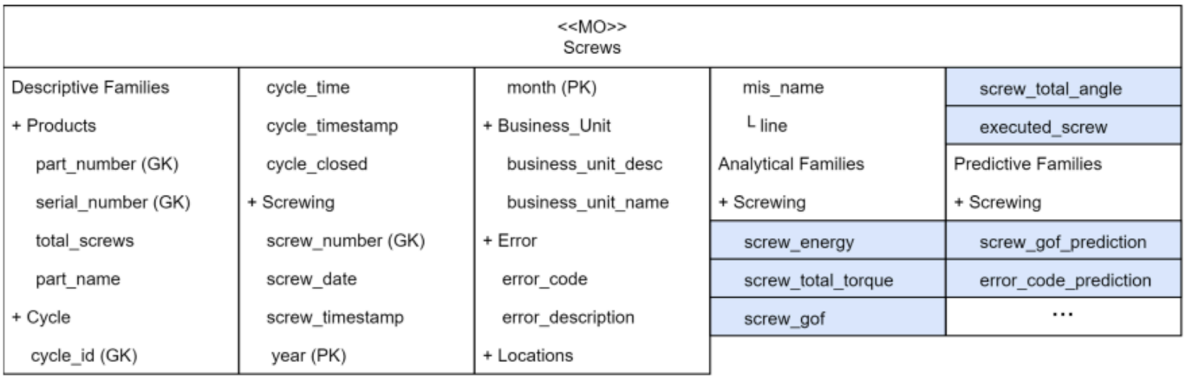


Figure 21: *MO Screws*, from Galvão et al. (2022).

3.6.3.2 Data Model

In this demonstration case, the screw data model was identified following the steps presented in Galvão et al. (2022) subsection 3.2 and integrates one *AO* (*AO Screws*), two *Special Objects* (Dates and Locations), and one materialized object (*MO Screws*). Due to confidentiality reasons, Fig. 21 only presents *MO Screws* as only this object is used to feed the dashboards here presented.

AO Screws will provide the necessary analytical information to *MO Screws* so that predictions can be made in the backend, and these predictions are stored in *MO Screws* along with other relevant attributes. It is important to mention that *AO Screws* has, for each screw, more than 400 records, and the *MO Screws* only has one record for each screw with the aggregated data (as described in Section 3.2). In the proposed model, the attributes highlighted in blue are considered NA in the *AO Screws*, but not in the *MO Screws*.

3.6.3.3 Integration and Flows

All the available data sources were integrated in the *Data Pipelines* that load the historical data of the screws. Once the necessary standardization has been made to the data, it is stored in the *Standard Raw Data Screw* folder. The screw data stored in the *Standard Raw Data* was used to train and optimize the prediction models. Those prediction models are stored in the ML Models Repository and mapped in the corresponding API. To feed the *Business Warehouse* another *Data Pipeline* is created. This pipeline loads the screw data from the *Standard Raw Data*, and for each screw, the ML API is called to predict the result of the GoF test. With the prediction results, a Dataframe is created containing the screw data from the *Standard Raw Data* grouped by *part_number*, *serial_number*, *cycle_id* (only available on business level databases), and *screw_number*, along with the GoF prediction. After that, the *Dataframe* is stored in a Hive Table that matches the AO Screw (*AO Screws*), an AO modeled for this BP and that integrates all the data relevant to support the decision support needs in this industrial plant. The decision process is supported by several analytical dashboards available in the Visualization Tools. As proposed in the architecture, all pipelines are stored in the *Data Pipelines Repository*.

3.6.3.4 Decision Support Dashboards

Dashboards are key elements in the daily work of the decision-makers, so their design was achieved with the engagement and validation of the final users. As a requirement for this demonstration case, decision-makers must have a set of dashboards that present macro visualizations of the screwing process, as well as more detailed ones capable to show the results of the GoF test of each screw. All the dashboards developed for this demonstration case have two main areas, an L shape bar along top and left side is dedicated to filters and a more central area with all the graphical/table elements that integrate the dashboard. In the filters area, the user can select from a wide range of options, such as temporal options, production line, or equipment, among others. Regarding all the examples presented in this paper, it is worth mentioning that all data was anonymized for confidentiality reasons. The first dashboard example (Fig. 22) is a general dashboard, with a holistic view of the screwing process. The purpose of this dashboard is to allow the user to consult potentially important data of the business process in a fast and effective way. Starting with the first element (Fig. 22, part 1), it is possible to analyse data regarding the quantities by equipment. These quantities are related to the successful or unsuccessful production of each equipment (Screw GoF 1 and Screw GoF 0, respectively). The available data is presented in a descending order considering the produced quantity. It is possible to apply top and side filters to this same visualization, allowing, for example, a visualization of the equipment with an unsuccessful production, with the Screw GoF at 0 (Fig. 22, part 6), or filter this data by a specific equipment or production line (Fig. 22, part 4). It is important to see in the dashboards the temporal attributes, year, month, week, and day filters (Fig. 22, part 5), allowing the user to filter the data by a specific date, thus increasing the level of detail and specificity of these visualizations. It is important to see in the dashboards the temporal attributes, year, month, week, and day filters (Fig. 22, part 5), allowing the user to filter the data by a specific date, thus increasing the level of detail and specificity of these visualizations.

Fig. 22, part 2, provides information about the most common errors that cause problems in a production equipment. It is possible to apply again the filters to a specific equipment, to a specific production line, to detail a specific date, search by error code and error description (Fig. 22, part 7), resulting in information regarding the equipment that most tend to suffer that specific error during the production process. In the last element (Fig. 22, part 3), a table calculates the percentage of failures relative to the production cycles (cycle GoF), for each production line. This table shows the count of the total cycle GoF values per production line, the count of the lines with cycle GoF at 0 and calculates the failure percentage. Again, the user can filter a specific production line to return the failure percentage for that same line in the chart. It is also possible to quickly clear all the filters selected by pressing the clear button (Fig. 22, part 8) which resets all the previous settings. The dashboard with more detailed data (Fig. 23) takes as input the *part_number* and the serial number of a product (Fig. 23, part 1). For that part and serial number, the user has an overview of the GoF test results in a bar graphic (Fig. 23, part 5). Also, the user can see the stations where the product pass considering the screws GoF test results in each station (Fig. 23, part 6). If the user selects a station, the bar graphic in Fig. 23, part 7, will highlight the screw cycles of that

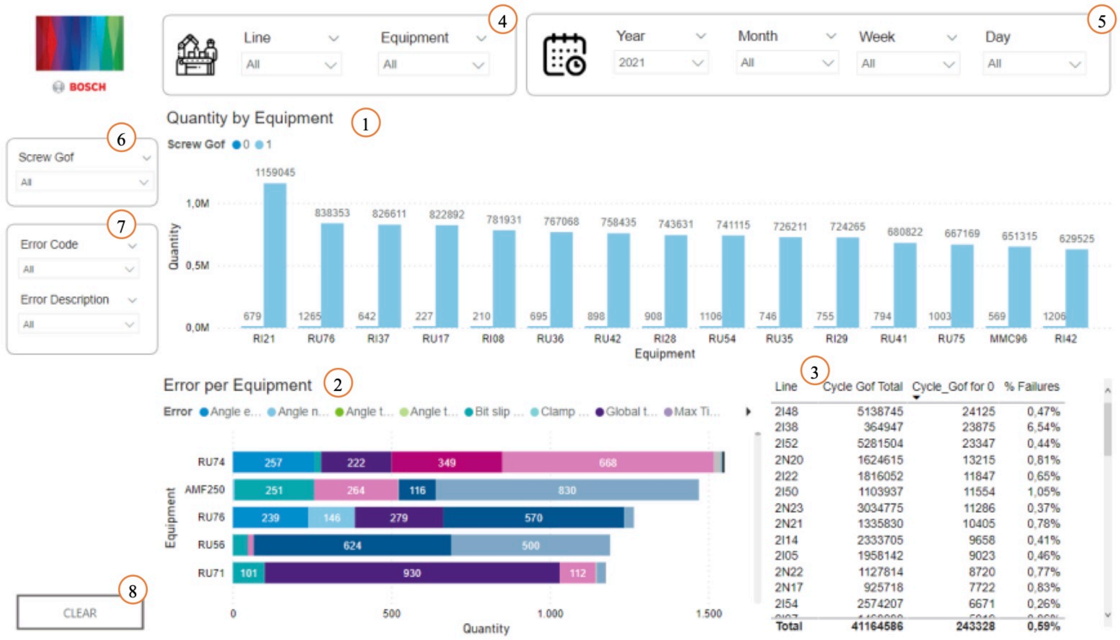


Figure 22: Macro Screw Tightening dashboard, from Galvão et al. (2022).

Figure 23: Detailed Screw Tightening dashboard, from Galvão et al. (2022).

Figure 24: Detailed Screw Tightening dashboard with GoF 0, from Galvão et al. (2022).

station and, for each *cycle_id*, the number of GoF tests OK vs NOK (Not OK) is presented. Fig. 23, part 8, shows in detail what are the results of the GoF test for each screw id based on a previously selected cycle id. Also, the prediction of the GoF test result is presented to the user, since in this phase decision-makers want to see the GoF test results and their prediction to evaluate if they can stop doing GoF tests or if they decide to do it by sampling. In this dashboard, a set of temporal filters can be used, Fig. 23 parts 2 and 3, and it is also possible to filter by GoF test result (part 4). Additionally, the dashboard has two cards to show the values about the percentage of failures (Fig. 23, part 9) and the total of screws (Fig. 23, part 10).

Fig. 24 presents a different serial number of the same *part_number* presented in the dashboard of Fig. 23. For this product, in the selected cycles, 7 screws were tight (OK) and 1 is not ok (NOK), ending the process with 14,3 % of failures. Also, it is possible to see that the tightening fails in the first screw and then the product changes to a different station, starting a new screw cycle that also starts the screwing process. Moreover, it is important to highlight that the prediction was capable to detect the failure in the first screw.

3.7 Summary

In this work, we target an unsupervised approach for screw tightening quality inspection, focusing more on the main task of detecting an anomaly process. Our approach assumes that only normal (thus one-class) angle-torque input pairs are used to train the models. Using recently collected data, from an automotive electronics assembly company. First, we collected around 2.8 million angle-torque pairs, related with 6,162 screw fastening cycles. Four different learning algorithms were compared: unsupervised – LOF, IF and a dense AE (under two learning modes: reset and reuse); and supervised - RF, used for benchmarking purposes. The algorithms were trained with individual angle-torque observations (two input variables), allowing to output an anomaly decision score for each angle-torque pair. The anomaly detection methods were compared by using a realistic rolling window evaluation, which simulated 20 training and test iterations through time. Overall, the best unsupervised learning results were obtained by the IF and AE reuse. Both methods provide an excellent anomaly discrimination capability that is identical or similar to supervised RF (99%) benchmark. In effect, IF produces the best unsupervised anomaly detection result (99%), slightly outperforming the AE reuse (95%) approach. However, it should be noted that the AE reuse method requires much less computational effort, which is relevant in this industrial domain, since it generates high velocity data. Since competitive results were obtained, we have further evaluated the IF and AE methods on a more recent and larger data, regarding three types of assembled products and two months of production, from February 12 to March 2021. In total, 67,337 fastening cycles and roughly 26.9 million angle-torque pair observations were considered in this second evaluation stage. Overall, an excellent anomaly discrimination detection performance was obtained by both methods, with the AE requiring much less computational effort. We have also designed an interactive visualization tool that provides XAI knowlegdge to the human operators, helping them to better identify the angle-torque areas associated with anomalies. The integration demonstration case was presented, providing historical and predictive data made available throughout a set of dashboards fed by the BDW.

4 Conclusions

Chapter Table of Contents:

4.1 Overview and Discussion

The Fourth Industrial Revolution, also known as Industry 4.0 (I4.0), has compelled companies to adapt to a more competitive market, where optimized productive processes are crucial for success. In the assembly industrial sector, reducing costs and increasing efficiency are paramount. Advances in Information Technology (IT) paved the way for improved approaches and tools to store and process sensory data. The introduction of smart sensors and actuators facilitated the use of autonomous or semi-autonomous robots to assist human operators during production processes. This resulted in the collection of large volumes of data (some with the characteristics of Big Data (BD)) at the shop floor across diverse industries. However, some assembly tasks still required human involvement. This PhD thesis was conducted within a large electronics manufacturer, primarily supplying assembled automotive instrument clusters to renowned automobile brands. The focus of this study lay in one of the final phases of the assembly process, specifically the combination of the plastic housing with either the PCBs or other plastic parts. It particularly targets the assembly of individual parts using handheld screwdrivers with torque and angle sensing capabilities, enabling faster and more accurate detection of anomalous processes. By leveraging the power of BD architectures, advanced warehousing techniques, and tailored Machine Learning (ML) algorithms, previously unexplored domains could now be investigated. These advancements held tremendous potential for anomaly detection in industrial environments.

The state of the literature revealed a research gap, with a lack of studies specifically addressing this particular issue (as highlighted in Section 2.8). This study aimed to contribute novel insights and fill this gap, advancing knowledge in the field. The main objective was to develop a ML-based Intelligent Decision Support System (IDSS) for predicting the success or failure of assembly processes. The study leveraged Big Data (BD) extracted from a Big Data Warehouse (BDW) to construct a IDSS.

The production lines in the plant were optimized for efficient assembly cluster throughput. Handheld tools were loaded with the appropriate screw tightening program, playing a critical role in this stage. Each inserted screw triggered a Good or Fail (GoF) status message indicating the success or failure of the fastening process (see Section 3.2). During the course of this work, significant improvements were made to the acquisition and storage of industrial data, ultimately resulting in data being stored in a BDW through an ETL process. As the research literature suggested, datasets in this domain were typically highly unbalanced, with a minimal percentage of faulty processes (see Table 3). Therefore, this study focused on One-Class Classification (OCC), unsupervised approaches for screw tightening quality inspection, with the primary objective of detecting anomalous processes. The models were trained using only normal angle-torque input pairs.

The performance of four learning algorithms, including Local Outlier Factor (LOF), Isolation Forest (IF), and a dense Autoencoder (AE), was initially compared (Section 3.6.1 and Table 4). Additionally, a supervised algorithm, Random Forest (RF), was used for benchmarking. The results showed that the IF and AE reuse methods achieved the best anomaly discrimination capabilities, comparable to the supervised RF benchmark. The AE reuse method demonstrated superior computational efficiency, as demonstrated in our first paper (Ribeiro, Matos, Cortez, et al., 2021), recognized with the best paper award of the conference. The effectiveness of the proposed approach was further validated using a larger dataset from the same automotive electronics assembly company (Section 3.6.2). The evaluation confirmed the excellent anomaly discrimination performance of both the IF and AE reuse methods, with the AE method requiring significantly less computational effort (Fig. 18, Ribeiro, Matos, Moreira, et al. (2022)).

To aid human operators in identifying angle-torque regions associated with anomalies, an interactive visualization tool was designed, providing Explainable Artificial Intelligence (XAI) knowledge (Section 3.6.2). The tool enabled the setting of the final threshold for abnormal screw cases and facilitated reasoning behind ML anomaly detection decisions.

This study also brought together in the form of an integration the scientific and technological contributions from Big Data Analytics (BDA) and ML fields. The proposed architecture created a unified environment between ML processes and the BDW, serving as the main storage component in BD contexts. The architecture enabled monitoring of ML models and the BDW, extending the analytical scope to performance metrics over time (Section 3.6.3, Galvão et al. (2022)).

Conducting experiments in a constrained environment poses various challenges, including issues related to timings, permissions, and accessibility. As a result, certain extended experimentation that was initially planned had to be postponed until there was insufficient time remaining to perform them. Consequently, the performance of the developed algorithms in such an evolving environment remains uncertain. While the initial tests provided insights into the effort required to maintain certain performance levels, several questions still remain unanswered. These include determining whether the same model can be applied across multiple product families and identifying the optimal frequency for retraining the models.

4.2 Future Work

In future research, there are several potential avenues to explore in the expansion of deep learning architectures. One intriguing direction involves the utilization of AE with LSTM layers. This architectural framework exhibits the capability to directly capture sequential data patterns, which holds the promise of bolstering the performance of our system (Cao et al., 2019).

Moreover, it would be worthwhile to extend our data-driven approach by conducting a prolonged comparative analysis with the screw assembly expert system tool. By subjecting both approaches to long-term evaluation, we can gain valuable insights into their respective strengths, weaknesses, and overall capabilities. This endeavor would require continuous monitoring of system performance over an extended period and the acquisition of feedback from screw tightening human operators. This real-world assessment would enable us to gauge the practical effectiveness of our approach and identify any challenges or areas for improvement that may arise.

Bibliography

Vapnik, V. N. (1963). “Pattern recognition using generalized portrait method”. In: *Automation and Remote Control* 24, pp. 774–780 (cit. on p. 29).

Gorry, G. A. and M. S. S. Morton (1971). “A framework for management information systems”. In: *Sloan Management Review* 13.1, pp. 55–70 (cit. on p. 13).

Keen, P. G. W. and M. S. S. Morton (1978). *Decision Support Systems: An organizational perspective*. Addison-Wesley (cit. on p. 13).

Smith, S. (1980). “Use of a microprocessor in the control and monitoring of air tools while tightening threaded fasteners”. In: *Proceedings of the Society of Manufacturing Engineers, Eaton Corporation, Autofact West, Dearborn, MI*, p. 2 (cit. on p. 24).

Peterson, G., B. Niznik, and L. Chan (1988). “Development of an automated screwdriver for use with industrial robots”. In: *IEEE Journal on Robotics and Automation* 4.4, pp. 411–414. doi: 10.1109/56.805 (cit. on p. 24).

Fix, E. and J. L. Hodges (1989). “Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties”. In: *International Statistical Review / Revue Internationale de Statistique* 57.3, pp. 238–247. issn: 03067734, 17515823. doi: 10.2307/1403797. url: http://www.jstor.org/stable/1403797 (cit. on p. 50).

Shimbun, N. K. (1989). *Poka-yoke: improving product quality by preventing defects*. Crc Press (cit. on p. 44).

Nicolson, E. J. and R. S. Fearing (1993). “Compliant Control of Threaded Fastener Insertion”. In: *Proceedings of the 1993 IEEE International Conference on Robotics and Automation, Atlanta, Georgia, USA, May 1993*. IEEE Computer Society Press, pp. 484–490. doi: 10.1109/ROBOT.1993.292026. url: https://doi.org/10.1109/ROBOT.1993.292026 (cit. on p. 23).

MacGregor, J. F. and P. Nomikos (1996). “Monitoring batch processes”. In: *Batch Processing Systems Engineering*. Springer, pp. 242–258 (cit. on p. 47).

Bickford, J. (1998). *Handbook of Bolts and Bolted Joints*. Taylor & Francis (cit. on p. 37).

Lara, B., K. Althoefer, and L. Seneviratne (1998). "Automated robot-based screw insertion system". In: *IECON '98. Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.98CH36200)*. Vol. 4, 2440–2445 vol.4. doi: 10.1109/IECON.1998.724109 (cit. on p. 20).

Lara, B., K. Althoefer, and L. D. Seneviratne (1999). "Use of artificial neural networks for the monitoring of screw insertions". In: *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients, October 17-21,1999, Hyundai Hotel, Kyongju, Korea*. IEEE, pp. 579–584. doi: 10.1109/IROS.1999.813066. url: https://doi.org/10.1109/IROS.1999.813066 (cit. on p. 38).

Breunig, M. M. et al. (2000). "LOF: Identifying Density-Based Local Outliers". In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA*. Ed. by W. Chen, J. F. Naughton, and P. A. Bernstein. ACM, pp. 93–104. doi: 10.1145/342009.335388. url: https://doi.org/10.1145/342009.335388 (cit. on pp. 29, 37, 50).

Chapman, P. et al. (2000). "CRISP-DM 1.0: Step-by-step data mining guide". In: (cit. on pp. xii, 25, 27, 28).

Domingos, P. (2000-01). "A Unified Bias-Variance Decomposition and its Applications." In: *Proceedings of the 17th International Conference on Machine Learning (ICML '00)*, pp. 231–238 (cit. on p. 35).

Tashman, L. J. (2000). "Out-of-sample tests of forecasting accuracy: an analysis and review". In: *International Journal of Forecasting* 16.4. The M3- Competition, pp. 437–450. issn: 0169-2070. doi: https://doi.org/10.1016/S0169-2070(00)00065-0. url: http://www.sciencedirect.com/science/article/pii/S0169207000000650 (cit. on p. 52).

Breiman, L. (2001). "Random Forests". In: *Mach. Learn.* 45.1, pp. 5–32. doi: 10.1023/A:1010933404324. url: https://doi.org/10.1023/A:1010933404324 (cit. on pp. 30, 52).

Schölkopf, B. et al. (2001). "Estimating the Support of a High-Dimensional Distribution". In: *Neural Comput.* 13.7, pp. 1443–1471. doi: 10.1162/089976601750264965. url: https://doi.org/10.1162/089976601750264965 (cit. on p. 30).

Coulouris, G., J. Dollimore, and T. Kindberg (2002). *Distributed systems - concepts and designs (3. ed.)* International computer science series. Addison-Wesley-Longman. isbn: 978-0-201-61918-8 (cit. on p. 36).

Japkowicz, N. and S. Stephen (2002). "The class imbalance problem: A systematic study". In: *Intell. Data Anal.* 6.5, pp. 429–449. url: http://content.iospress.com/articles/intelligent-data-analysis/ida00103 (cit. on p. 34).

Kiang, M. Y. (2003). "A comparative assessment of classification methods". In: *Decis. Support Syst.* 35.4, pp. 441–454. doi: 10.1016/S0167-9236(02)00110-0. url: https://doi.org/10.1016/S0167-9236(02)00110-0 (cit. on p. 35).

Hevner, A. R. et al. (2004). "Design Science in Information Systems Research". In: *MIS Q.* 28.1, pp. 75–105. url: http://misq.org/design-science-in-information-systems-research.html (cit. on pp. 7, 10).

Klingajay, M. and N. I. Giannoccaro (2004-12). “The Automated Threaded Fastening Based on On-line Identification”. In: *International Journal of Advanced Robotic Systems* 1. doi: 10.5772/5810 (cit. on p. 38).

Althoefer, K., B. Lara, and L. Seneviratne (2005-02). “Monitoring of Self-Tapping Screw Fastenings Using Artificial Neural Networks”. In: *Journal of Manufacturing Science and Engineering-transactions of The Asme - J MANUF SCI ENG* 127. doi: 10.1115/1.1831286 (cit. on p. 38).

II, D. K. W. (2005). “Discovering Knowledge in Data, an Introduction to Data Mining”. In: *Inf. Process. Manag.* 41.5, pp. 1307–1309. doi: 10.1016/j.ipm.2005.01.006. url: https://doi.org/10.1016/j.ipm.2005.01.006 (cit. on p. 32).

Bengio, Y. et al. (2006). “Greedy Layer-Wise Training of Deep Networks”. In: *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*. Ed. by B. Schölkopf, J. C. Platt, and T. Hofmann. MIT Press, pp. 153–160. url: https://proceedings.neurips.cc/paper/2006/hash/5da713a690c067105aeb2fae32403405-Abstract.html (cit. on p. 31).

Fawcett, T. (2006a). “An introduction to ROC analysis”. In: *Pattern Recognit. Lett.* 27.8, pp. 861–874. doi: 10.1016/j.patrec.2005.10.010. url: https://doi.org/10.1016/j.patrec.2005.10.010 (cit. on p. 53).

Althoefer, K., B. Lara, Y. H. Zweiri, et al. (2008). “Automated failure classification for assembly with self-tapping threaded fastenings using artificial neural networks”. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 222.6, pp. 1081–1095. doi: 10.1243/09544062JMES546. eprint: https://doi.org/10.1243/09544062JMES546. url: https://doi.org/10.1243/09544062JMES546 (cit. on pp. 24, 38).

Dean, J. and S. Ghemawat (2008). “MapReduce: simplified data processing on large clusters”. In: *Commun. ACM* 51.1, pp. 107–113. doi: 10.1145/1327452.1327492. url: https://doi.org/10.1145/1327452.1327492 (cit. on p. 37).

Liu, F. T., K. M. Ting, and Z. Zhou (2008). “Isolation Forest”. In: *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*. IEEE Computer Society, pp. 413–422. doi: 10.1109/ICDM.2008.17. url: https://doi.org/10.1109/ICDM.2008.17 (cit. on pp. 29, 50).

Peffers, K. et al. (2008). “A Design Science Research Methodology for Information Systems Research”. In: *J. Manag. Inf. Syst.* 24.3, pp. 45–77. doi: 10.2753/mis0742-1222240302. url: https://doi.org/10.2753/mis0742-1222240302 (cit. on pp. xii, 7–10).

Amancio-Filho, S. and J. dos Santos (2009-08). “Joining of Polymers and Polymer-Metal Hybrid Structures: Recent Developments and Trends”. In: *Polymer Engineering & Science* 49, pp. 1461–1476. doi: 10.1002/pen.21424 (cit. on pp. 16–18).

Bengio, Y. (2009). “Learning Deep Architectures for AI”. In: *Found. Trends Mach. Learn.* 2.1, pp. 1–127. doi: 10.1561/2200000006. url: https://doi.org/10.1561/2200000006 (cit. on p. 30).

Chandola, V., A. Banerjee, and V. Kumar (2009). “Anomaly detection: A survey”. In: *ACM Comput. Surv.* 41.3, 15:1–15:58. doi: 10.1145/1541880.1541882. url: https://doi.org/10.1145/1541880.1541882 (cit. on p. 37).

Geng, X. and K. Smith-Miles (2009). “Incremental Learning”. In: *Encyclopedia of Biometrics*. Ed. by S. Z. Li and A. Jain. Boston, MA: Springer US, pp. 731–735. isbn: 978-0-387-73003-5. doi: 10.1007/978-0-387-73003-5_304. url: https://doi.org/10.1007/978-0-387-73003-5_304 (cit. on p. 35).

Sun, Y., A. K. C. Wong, and M. S. Kamel (2009). “Classification of Imbalanced Data: a Review”. In: *Int. J. Pattern Recognit. Artif. Intell.* 23.4, pp. 687–719. doi: 10.1142/S0218001409007326. url: https://doi.org/10.1142/S0218001409007326 (cit. on p. 32).

Genuer, R., J. Poggi, and C. Tuleau-Malot (2010). “Variable selection using random forests”. In: *Pattern Recognit. Lett.* 31.14, pp. 2225–2236. doi: 10.1016/j.patrec.2010.03.014. url: https://doi.org/10.1016/j.patrec.2010.03.014 (cit. on p. 30).

Vincent, P. et al. (2010). “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion”. In: *J. Mach. Learn. Res.* 11, pp. 3371–3408. doi: 10.5555/1756006.1953039. url: https://dl.acm.org/doi/10.5555/1756006.1953039 (cit. on p. 31).

Zaharia, M., M. Chowdhury, M. J. Franklin, et al. (2010). “Spark: Cluster Computing with Working Sets”. In: *2nd USENIX Workshop on Hot Topics in Cloud Computing, HotCloud’10, Boston, MA, USA, June 22, 2010*. Ed. by E. M. Nahum and D. Xu. USENIX Association. url: https://www.usenix.org/conference/hotcloud-10/spark-cluster-computing-working-sets (cit. on p. 37).

“Chapter 10 - CUDA in a Cloud and Cluster Environments” (2011). In: *CUDA Application Design and Development*. Ed. by R. Farber. Boston: Morgan Kaufmann, pp. 241–264. isbn: 978-0-12-388426-8. doi: https://doi.org/10.1016/B978-0-12-388426-8.00010-0. url: https://www.sciencedirect.com/science/article/pii/B9780123884268000100 (cit. on p. 37).

Ghoting, A. et al. (2011). “NIMBLE: a toolkit for the implementation of parallel data mining and machine learning algorithms on mapreduce”. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*. Ed. by C. Apté, J. Ghosh, and P. Smyth. ACM, pp. 334–342. doi: 10.1145/2020408.2020464. url: https://doi.org/10.1145/2020408.2020464 (cit. on p. 36).

Pedregosa, F. et al. (2011). “Scikit-learn: Machine Learning in Python”. In: *J. Mach. Learn. Res.* 12, pp. 2825–2830. url: http://dl.acm.org/citation.cfm?id=2078195 (cit. on p. 49).

Turban, E., R. Sharda, and D. Delen (2011). *Decision Support and Business Intelligence Systems (9th Edition)*. 9th. Pearson Hall, p. 720. isbn: 9780132453233. url: https://books.google.pt/books?id=T7WHQgAACAAJ (cit. on p. 14).

Witten, I. H., E. Frank, and M. A. Hall (2011). *Data mining: practical machine learning tools and techniques, 3rd Edition*. Morgan Kaufmann, Elsevier. isbn: 9780123748560. url: https://www.worldcat.org/oclc/262433473 (cit. on pp. 29–31).

Beyer, M. and D. Laney (2012). “The Importance of ‘Big Data’: a Definition”. In: *Gartner Research Report*. url: https://www.gartner.com/en/documents/2057415?_its=JTdCJTIydmlkJTIyJTNBJTIyNjc5OTBiZTUtMTJjNS00MWE4LTgyNjctOTI4NDhjMzQ3NzE5JTIyJTJDJTIyc3RhdGUlMjIlM0ElMjJybHR%5C%2BMTY4NDU5ODA0Nn5sYW5kfjJfMTY0NjVfc2VvXzlhY2IwMjk3ZDJmODkwNTZhOGEyMTc3ODg3MmZkOGM0JTIyJTJDJTIyc2l0ZUlkJTIyJTNBNDAxMzElN0Q%5C%3D (cit. on pp. 33, 34).

Chen, H., R. H. L. Chiang, and V. C. Storey (2012). “Business Intelligence and Analytics: From Big Data to Big Impact”. In: *MIS Q.* 36.4, pp. 1165–1188. url: http://www.misq.org/skin/frontend/default/misq/pdf/V36I4/SI%5C_ChenIntroduction.pdf (cit. on p. 14).

Domingos, P. M. (2012). “A few useful things to know about machine learning”. In: *Commun. ACM* 55.10, p. 78. doi: 10.1145/2347736.2347755. url: https://doi.org/10.1145/2347736.2347755 (cit. on p. 35).

Fan, W. and A. Bifet (2012). “Mining big data: current status, and forecast to the future”. In: *SIGKDD Explor.* 14.2, pp. 1–5. doi: 10.1145/2481244.2481246. url: https://doi.org/10.1145/2481244.2481246 (cit. on p. 33).

Galar, M. et al. (2012). “A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches”. In: *IEEE Trans. Syst. Man Cybern. Part C* 42.4, pp. 463–484. doi: 10.1109/TSMCC.2011.2161285. url: https://doi.org/10.1109/TSMCC.2011.2161285 (cit. on p. 31).

Hwang, J.-Y. et al. (2012). “Low-cost automatic screw machine using a commercial electric screwdriver”. In: *2012 12th International Conference on Control, Automation and Systems*, pp. 1055–1060 (cit. on p. 21).

Parker, C. (2012). “Unexpected challenges in large scale machine learning”. In: *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, BigMine 2012, Beijing, China, August 12, 2012*. Ed. by W. Fan et al. ACM, pp. 1–6. doi: 10.1145/2351316.2351317. url: https://doi.org/10.1145/2351316.2351317 (cit. on pp. 33–35).

Zaharia, M., M. Chowdhury, T. Das, et al. (2012). “Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing”. In: *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2012, San Jose, CA, USA, April 25-27, 2012*. Ed. by S. D. Gribble and D. Katabi. USENIX Association, pp. 15–28. url: https://www.usenix.org/conference/nsdi12/technical-sessions/presentation/zaharia (cit. on p. 37).

Delen, D. and H. Demirkan (2013). “Data, information and analytics as services”. In: *Decis. Support Syst.* 55.1, pp. 359–363. doi: 10.1016/j.dss.2012.05.044. url: https://doi.org/10.1016/j.dss.2012.05.044 (cit. on p. 14).

Demchenko, Y. et al. (2013). “Addressing big data issues in Scientific Data Infrastructure”. In: *2013 International Conference on Collaboration Technologies and Systems, CTS 2013, San Diego, CA, USA, May 20-24, 2013*. Ed. by G. C. Fox and W. W. Smari. IEEE, pp. 48–55. doi: 10.1109/CTS.2013.6567203. url: https://doi.org/10.1109/CTS.2013.6567203 (cit. on p. 33).

Ding, Z. and M. Fei (2013). “An Anomaly Detection Approach Based on Isolation Forest Algorithm for Streaming Data Using Sliding Window”. In: *3rd IFAC International Conference on Intelligent Control and Automation Science, ICONS 2013, Sichuan, Chengdu, China, September 2-4, 2013*. Ed. by P. M. Ferreira. International Federation of Automatic Control, pp. 12–17. doi: 10.3182/20130902-3-CN-3020.00044. url: https://doi.org/10.3182/20130902-3-CN-3020.00044 (cit. on p. 29).

Hollander, M., D. A. Wolfe, and E. Chicken (2013). *Nonparametric statistical methods*. John Wiley & Sons (cit. on pp. 27, 54).

Järvenpää, E., M. Lanz, and R. Tuokko (2013-09). “Capability-Based Adaptation of Production Systems - Practical Case Study in TUT-Microfactory Environment”. In: *Key Engineering Materials* 572, pp. 245–248. doi: 10.4028/www.scientific.net/KEM.572.245 (cit. on p. 22).

Matsuno, T., J. Huang, and T. Fukuda (2013). “Fault detection algorithm for external thread fastening by robotic manipulator using linear support vector machine classifier”. In: *2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, May 6-10, 2013*. IEEE, pp. 3443–3450. doi: 10.1109/ICRA.2013.6631058. url: https://doi.org/10.1109/ICRA.2013.6631058 (cit. on pp. 20, 38).

Swan, M. (2013). “The Quantified Self: Fundamental Disruption in Big Data Science and Biological Discovery”. In: *Big Data* 1.2, pp. 85–99. doi: 10.1089/big.2012.0002. url: https://doi.org/10.1089/big.2012.0002 (cit. on p. 35).

Arnott, D. and G. Pervan (2014). “A critical analysis of decision support systems research revisited: the rise of design science”. In: *J. Inf. Technol.* 29.4, pp. 269–293. doi: 10.1057/jit.2014.16. url: https://doi.org/10.1057/jit.2014.16 (cit. on pp. xii, 13–15).

Dongre, P. B. and L. G. Malik (2014). “A review on real time data stream classification and adapting to various concept drift scenarios”. In: *2014 IEEE International Advance Computing Conference (IACC)*, pp. 533–537. doi: 10.1109/IAdCC.2014.6779381 (cit. on p. 35).

Fan, J., F. Han, and H. Liu (2014-02). “Challenges of Big Data analysis”. In: *National Science Review* 1.2, pp. 293–314. issn: 2095-5138. doi: 10.1093/nsr/nwt032. eprint: https://academic.oup.com/nsr/article-pdf/1/2/293/31565398/nwt032.pdf. url: https://doi.org/10.1093/nsr/nwt032 (cit. on p. 35).

Ghanavati, M. et al. (2014). “An Effective Integrated Method for Learning Big Imbalanced Data”. In: *2014 IEEE International Congress on Big Data, Anchorage, AK, USA, June 27 - July 2, 2014*. IEEE Computer Society, pp. 691–698. doi: 10.1109/BigData.Congress.2014.102. url: https://doi.org/10.1109/BigData.Congress.2014.102 (cit. on p. 34).

Grolinger, K. et al. (2014). “Challenges for MapReduce in Big Data”. In: *2014 IEEE World Congress on Services, SERVICES 2014, Anchorage, AK, USA, June 27 - July 2, 2014*. IEEE Computer Society, pp. 182–189. doi: 10.1109/SERVICES.2014.41. url: https://doi.org/10.1109/SERVICES.2014.41 (cit. on p. 34).

Jagadish, H. V. et al. (2014). “Big data and its technical challenges”. In: *Commun. ACM* 57.7, pp. 86–94. doi: 10.1145/2611567. url: https://doi.org/10.1145/2611567 (cit. on pp. 34, 36).

Khan, M. A.-d., M. F. Uddin, and N. Gupta (2014). “Seven V’s of Big Data understanding Big Data to extract value”. In: *Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education*, pp. 1–5. doi: 10.1109/ASEEZone1.2014.6820689 (cit. on pp. 33, 34).

Kingma, D. P. and M. Welling (2014). “Auto-Encoding Variational Bayes”. In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. url: http://arxiv.org/abs/1312.6114 (cit. on p. 31).

Lasi, H. et al. (2014). “Industry 4.0”. In: *Bus. Inf. Syst. Eng.* 6.4, pp. 239–242. doi: 10.1007/s12599-014-0334-4. url: https://doi.org/10.1007/s12599-014-0334-4 (cit. on pp. 10, 11).

Lee, J., H.-A. Kao, and S. Yang (2014). “Service innovation and smart analytics for Industry 4.0 and big data environment”. In: vol. 16. Cited by: 1211; All Open Access, Bronze Open Access, pp. 3–8. doi: 10.1016/j.procir.2014.02.001. url: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84905408361&doi=10.1016%5C%2fj.procir.2014.02.001&partnerID=40&md5=0843fd1f4a451db2c0975802ac8bc258 (cit. on p. 33).

Leskovec, J., A. Rajaraman, and J. D. Ullman (2014). *Mining of Massive Datasets, 2nd Ed.* Cambridge University Press. isbn: 978-1107077232. url: http://www.mmds.org/ (cit. on p. 35).

Narasimhan, R. and T. Bhuvaneshwari (2014). “Big Data - A Brief Study”. In: ed. by I. J. of Scientific & Engineering Research. Vol. 5. 9, pp. 350–353 (cit. on p. 33).

Srivastava, N. et al. (2014). “Dropout: a simple way to prevent neural networks from overfitting”. In: *J. Mach. Learn. Res.* 15.1, pp. 1929–1958. doi: 10.5555/2627435.2670313. url: https://dl.acm.org/doi/10.5555/2627435.2670313 (cit. on p. 30).

Al-Fuqaha, A. I. et al. (2015). “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications”. In: *IEEE Commun. Surv. Tutorials* 17.4, pp. 2347–2376. doi: 10.1109/COMST.2015.2444095. url: https://doi.org/10.1109/COMST.2015.2444095 (cit. on pp. 11, 12).

Biau, G. and E. Scornet (2015-11). “A Random Forest Guided Tour”. In: *TEST* 25. doi: 10.1007/s11749-016-0481-7 (cit. on p. 30).

Cai, L. and Y. Zhu (2015). “The Challenges of Data Quality and Data Quality Assessment in the Big Data Era”. In: *Data Sci. J.* 14, p. 2. doi: 10.5334/dsj-2015-002. url: https://doi.org/10.5334/dsj-2015-002 (cit. on p. 59).

Gandomi, A. and M. Haider (2015). “Beyond the hype: Big data concepts, methods, and analytics”. In: *Int. J. Inf. Manag.* 35.2, pp. 137–144. doi: 10.1016/j.ijinfomgt.2014.10.007. url: https://doi.org/10.1016/j.ijinfomgt.2014.10.007 (cit. on p. 35).

Ioffe, S. and C. Szegedy (2015). “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *CoRR* abs/1502.03167. arXiv: 1502.03167. url: http://arxiv.org/abs/1502.03167 (cit. on p. 51).

Kaklauskas, A. (2015). *Biometric and Intelligent Decision Making Support*. Vol. 81. Intelligent Systems Reference Library. Springer, p. 220. isbn: 978-3-319-13658-5. doi: 10.1007/978-3-319-13659-2. url: https://doi.org/10.1007/978-3-319-13659-2 (cit. on p. 13).

Najafabadi, M. M. et al. (2015). “Deep learning applications and challenges in big data analytics”. In: *J. Big Data* 2, p. 1. doi: 10.1186/s40537-014-0007-7. url: https://doi.org/10.1186/s40537-014-0007-7 (cit. on p. 35).

Park, K. and G. E. Okudan Kremer (2015). “Assessment of static complexity in design and manufacturing of a product family and its impact on manufacturing performance”. In: *International Journal of Production Economics* 169, pp. 215–232. issn: 0925-5273. doi: https://doi.org/10.1016/j.ijpe.2015.07.036. url: https://www.sciencedirect.com/science/article/pii/S0925527315002856 (cit. on p. 16).

Sharda, R. et al. (2015). *Business Intelligence and Analytics: Systems for Decision Support*. 10th. Pearson, p. 688. doi: 10.4324/9781315206455-12 (cit. on p. 14).

Speck, J. A. (2015). *Mechanical fastening, joining, and assembly; 2nd ed.* Hoboken, NJ: Taylor and Francis. url: https://cds.cern.ch/record/2000060 (cit. on p. 23).

Wang, J. et al. (2015). “Big data provenance: Challenges, state of the art and opportunities”. In: *2015 IEEE International Conference on Big Data (IEEE BigData 2015), Santa Clara, CA, USA, October 29 - November 1, 2015*. IEEE Computer Society, pp. 2509–2516. doi: 10.1109/BigData.2015.7364047. url: https://doi.org/10.1109/BigData.2015.7364047 (cit. on p. 35).

Zheng, Y. (2015). “Methodologies for Cross-Domain Data Fusion: An Overview”. In: *IEEE Trans. Big Data* 1.1, pp. 16–34. doi: 10.1109/TBDATA.2015.2465959. url: https://doi.org/10.1109/TBDATA.2015.2465959 (cit. on p. 35).

Krawczyk, B. (2016). “Learning from imbalanced data: open challenges and future directions”. In: *Prog. Artif. Intell.* 5.4, pp. 221–232. doi: 10.1007/s13748-016-0094-0. url: https://doi.org/10.1007/s13748-016-0094-0 (cit. on p. 35).

Kune, R. et al. (2016). “The anatomy of big data computing”. In: *Softw. Pract. Exp.* 46.1, pp. 79–105. doi: 10.1002/spe.2374. url: https://doi.org/10.1002/spe.2374 (cit. on p. 33).

Lovelace, R. et al. (2016). “From Big Noise to Big Data: Toward the Verification of Large Data sets for Understanding Regional Retail Flows”. In: *Geographical Analysis* 48.1, pp. 59–81. doi: https://doi.org/10.1111/gean.12081. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/gean.12081. url: https://onlinelibrary.wiley.com/doi/abs/10.1111/gean.12081 (cit. on p. 36).

Nieoczym, A. and R. Longwic (2016). “CONTROL OF THE PROCESS OF SCREWING IN THE INDUSTRIAL SCREWDRIVERS”. In: *Advances in Science and Technology Research Journal* 10.30, pp. 202–206.

issn: 2080-4075. doi: 10.12913/22998624/62808. url: https://doi.org/10.12913/22998624/62808 (cit. on p. 24).

Nvidia (2016). *GPU Applications: Transforming Computational Research and Engineering*. url: http://www.nvidia.ca/object/machine-learning.html (cit. on p. 36).

Ponpitakchai, S. (2016). "Monitoring Screw Fastening Process: an Application of SVM Classification". In: *Naresuan University Engineering Journal (NUEJ)*. doi: https://doi.org/10.14456/nuej.2016.11 (cit. on p. 38).

Steen, M. van and A. S. Tanenbaum (2016). "A brief introduction to distributed systems". In: *Computing* 98.10, pp. 967–1009. doi: 10.1007/s00607-016-0508-7. url: https://doi.org/10.1007/s00607-016-0508-7 (cit. on p. 36).

Stojanovic, L. et al. (2016). "Big-data-driven anomaly detection in industry (4.0): An approach and a case study". In: *2016 IEEE International Conference on Big Data (IEEE BigData 2016), Washington DC, USA, December 5-8, 2016*. Ed. by J. Joshi et al. IEEE Computer Society, pp. 1647–1652. doi: 10.1109/BigData.2016.7840777. url: https://doi.org/10.1109/BigData.2016.7840777 (cit. on p. 26).

Calude, C. and G. Longo (2017-09). "The Deluge of Spurious Correlations in Big Data". In: *Foundations of Science* 22. doi: 10.1007/s10699-016-9489-4 (cit. on p. 35).

Colombo, A. W. et al. (2017). "Industrial Cyberphysical Systems: A Backbone of the Fourth Industrial Revolution". In: *IEEE Industrial Electronics Magazine* 11.1, pp. 6–16. doi: 10.1109/MIE.2017.2648857 (cit. on p. 13).

Diez-Olivan, A. et al. (2017). "Kernel Density-Based Pattern Classification in Blind Fasteners Installation". In: *Hybrid Artificial Intelligent Systems*. Ed. by F. J. Martínez de Pisón et al. Cham: Springer International Publishing, pp. 195–206. isbn: 978-3-319-59650-1 (cit. on p. 38).

Gulli, A. and S. Pal (2017). *Deep learning with Keras*. Packt Publishing Ltd (cit. on p. 49).

ISO 5393:2017 Rotary tools for threaded fasteners — Performance test method (2017-11). url: https://www.iso.org/standard/63133.html (cit. on p. 37).

L'Heureux, A. et al. (2017). "Machine Learning With Big Data: Challenges and Approaches". In: *IEEE Access* 5, pp. 7776–7797. doi: 10.1109/ACCESS.2017.2696365. url: https://doi.org/10.1109/ACCESS.2017.2696365 (cit. on pp. xii, 34, 35).

Liao, Y. et al. (2017). "Past, present and future of Industry 4.0 - a systematic literature review and research agenda proposal". In: *Int. J. Prod. Res.* 55.12, pp. 3609–3629. doi: 10.1080/00207543.2017.1308576. url: https://doi.org/10.1080/00207543.2017.1308576 (cit. on pp. 10, 11).

Lu, Y. (2017). "Industry 4.0: A survey on technologies, applications and open research issues". In: *J. Ind. Inf. Integr.* 6, pp. 1–10. doi: 10.1016/j.jii.2017.04.005. url: https://doi.org/10.1016/j.jii.2017.04.005 (cit. on p. 10).

Vaswani, A. et al. (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by I. Guyon et al., pp. 5998–6008. url: https://proceedings.

neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (cit. on p. 30).

Zhao, Y. et al. (2017). "Spatio-Temporal AutoEncoder for Video Anomaly Detection". In: *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017*. Ed. by Q. Liu et al. ACM, pp. 1933–1941. doi: 10.1145/3123266.3123451. url: https://doi.org/10.1145/3123266.3123451 (cit. on p. 31).

Zhou, C. and R. C. Paffenroth (2017). "Anomaly detection with robust deep autoencoders". In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 665–674 (cit. on pp. 31, 38).

Alguliyev, R. M., Y. N. Imamverdiyev, and L. V. Sukhostat (2018). "Cyber-physical systems and their security issues". In: *Comput. Ind.* 100, pp. 212–223. doi: 10.1016/j.compind.2018.04.017. url: https://doi.org/10.1016/j.compind.2018.04.017 (cit. on pp. 12, 13).

Boyes, H. et al. (2018). "The industrial internet of things (IIoT): An analysis framework". In: *Comput. Ind.* 101, pp. 1–12. doi: 10.1016/j.compind.2018.04.015. url: https://doi.org/10.1016/j.compind.2018.04.015 (cit. on pp. 10, 11, 13).

Chen, Z. et al. (2018). "Autoencoder-based network anomaly detection". In: *2018 Wireless Telecommunications Symposium (WTS)*, pp. 1–5. doi: 10.1109/WTS.2018.8363930 (cit. on p. 31).

Dalenogare, L. S. et al. (2018). "The expected contribution of Industry 4.0 technologies for industrial performance". In: *International Journal of Production Economics* 204, pp. 383–394. issn: 0925-5273. doi: https://doi.org/10.1016/j.ijpe.2018.08.019. url: https://www.sciencedirect.com/science/article/pii/S0925527318303372 (cit. on pp. 10, 11).

Speck, J. (2018-10). *Mechanical Fastening, Joining, and Assembly*. isbn: 9781351228640. doi: 10.1201/9781351228640 (cit. on pp. 15–19).

Sterling, T., M. Anderson, and M. Brodowicz (2018). "Chapter 19 - MapReduce". In: *High Performance Computing*. Ed. by T. Sterling, M. Anderson, and M. Brodowicz. Boston: Morgan Kaufmann, pp. 579–589. isbn: 978-0-12-420158-3. doi: https://doi.org/10.1016/B978-0-12-420158-3.00019-8. url: https://www.sciencedirect.com/science/article/pii/B9780124201583000198 (cit. on p. 37).

Xu, H. et al. (2018). "A Survey on Industrial Internet of Things: A Cyber-Physical Systems Perspective". In: *IEEE Access* 6, pp. 78238–78259. doi: 10.1109/ACCESS.2018.2884906. url: https://doi.org/10.1109/ACCESS.2018.2884906 (cit. on pp. xii, 10–13).

Zong, B. et al. (2018). "Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. url: https://openreview.net/forum?id=BJJLHbb0- (cit. on p. 31).

Alla, S. and S. K. Adari (2019). *Beginning Anomaly Detection Using Python-Based Deep Learning*. Berkeley, CA: Apress. isbn: 978-1-4842-5176-8. doi: https://doi.org/10.1007/978-1-4842-5177-5. url: http://link.springer.com/10.1007/978-1-4842-5177-5 (cit. on pp. 37, 51).

Cao, X. et al. (2019). "Anomaly Detection for Screw Tightening Timing Data with LSTM Recurrent Neural Network". In: *2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, pp. 348–352. doi: `10.1109/MSN48538.2019.00072` (cit. on pp. 38, 69).

Cheng, Z., C. Zou, and J. Dong (2019). "Outlier detection using isolation forest and local outlier factor". In: *Proceedings of the Conference on Research in Adaptive and Convergent Systems, RACS 2019, Chongqing, China, September 24-27, 2019*. Ed. by C. Hung et al. ACM, pp. 161–168. doi: `10.1145/3338840.3355641`. url: `https://doi.org/10.1145/3338840.3355641` (cit. on p. 29).

Gong, D. et al. (2019-10). "Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (cit. on p. 31).

Jia, Z. et al. (2019). "A Survey of Automated Threaded Fastening". In: *IEEE Trans Autom. Sci. Eng.* 16.1, pp. 298–310. doi: `10.1109/TASE.2018.2835382`. url: `https://doi.org/10.1109/TASE.2018.2835382` (cit. on pp. xii, 22, 23).

Matos, L. M. et al. (2019). "Using Deep Learning for Mobile Marketing User Conversion Prediction". In: *International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019*. IEEE, pp. 1–8. doi: `10.1109/IJCNN.2019.8851888`. url: `https://doi.org/10.1109/IJCNN.2019.8851888` (cit. on pp. 51, 52).

Santos, M. Y. and C. Costa (2020). "Big Data: Concepts, Warehousing, and Analytics". In: *Big Data: Concepts, Warehousing, and Analytics*, pp. i–xxx (cit. on p. 60).

Alghushairy, O. et al. (2021). "A Review of Local Outlier Factor Algorithms for Outlier Detection in Big Data Streams". In: *Big Data Cogn. Comput.* 5.1, p. 1. doi: `10.3390/bdcc5010001`. url: `https://doi.org/10.3390/bdcc5010001` (cit. on p. 29).

Ferhat, M. et al. (2021). "Incremental discovery of new defects: application to screwing process monitoring". In: *CIRP Annals* 70.1, pp. 369–372. issn: 0007-8506. doi: `https://doi.org/10.1016/j.cirp.2021.04.026`. url: `https://www.sciencedirect.com/science/article/pii/S0007850621000500` (cit. on p. 38).

Lourenço, J. M. (2021). *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. url: `https://github.com/joaomlourenco/novathesis/raw/main/template.pdf` (cit. on pp. ii, iv).

Pereira, P. J., P. Cortez, and R. Mendes (2021). "Multi-objective Grammatical Evolution of Decision Trees for Mobile Marketing user conversion prediction". In: *Expert Systems with Applications* 168, p. 114287 (cit. on p. 53).

Regaya, Y., F. Fadli, and A. Amira (2021). "Point-Denoise: Unsupervised outlier detection for 3D point clouds enhancement". In: *Multim. Tools Appl.* 80.18, pp. 28161–28177. doi: `10.1007/s11042-021-10924-x`. url: `https://doi.org/10.1007/s11042-021-10924-x` (cit. on pp. xii, 51).

Reich, C. L. and S. Vijaykumar (2021). "A Possibility in Algorithmic Fairness: Can Calibration and Equal Error Rates Be Reconciled?" In: *2nd Symposium on Foundations of Responsible Computing, FORC 2021, June 9-11, 2021, Virtual Conference*. Ed. by K. Ligett and S. Gupta. Vol. 192. LIPIcs. Schloss

Dagstuhl - Leibniz-Zentrum für Informatik, 4:1–4:21. doi: 10.4230/LIPIcs.FORC.2021.4. url: https://doi.org/10.4230/LIPIcs.FORC.2021.4 (cit. on p. 53).

Ribeiro, D., L. M. Matos, P. Cortez, et al. (2021). "A Comparison of Anomaly Detection Methods for Industrial Screw Tightening". In: *Computational Science and Its Applications - ICCSA 2021 - 21st International Conference, Cagliari, Italy, September 13-16, 2021, Proceedings, Part II*. Ed. by O. Gervasi et al. Vol. 12950. Lecture Notes in Computer Science. Springer, pp. 485–500. doi: 10.1007/978-3-030-86960-1_34. url: https://doi.org/10.1007/978-3-030-86960-1%5C_34 (cit. on pp. 2, 37, 43, 44, 49, 69).

Solace (2021-05). *Advanced Event Broker. An event mesh for connected enterprises*. url: https://solace.com/ (cit. on p. 44).

Galvão, J. et al. (2022). "Bosch's Industry 4.0 Advanced Data Analytics: Historical and Predictive Data Integration for Decision Support". In: *Research Challenges in Information Science - 16th International Conference, RCIS 2022, Barcelona, Spain, May 17-20, 2022, Proceedings*. Ed. by R. S. S. Guizzardi, J. Ralyté, and X. Franch. Vol. 446. Lecture Notes in Business Information Processing. Springer, pp. 580–596. doi: 10.1007/978-3-031-05760-1_34. url: https://doi.org/10.1007/978-3-031-05760-1%5C_34 (cit. on pp. xiii, 3, 61, 62, 64, 65, 69).

Lipovetsky, S. (2022). "Statistical and Machine-Learning Data Mining: Techniques for Better Predictive Modeling and Analysis of Big Data: 3rd Edition, by Bruce Ratner. Chapman and Hall/CRC, Taylor & Francis Group, Boca Raton, FL, 2020; ISBN 978-0-367-57360-7; xxxiii + 655 pp." In: *Technometrics* 64.1, pp. 145–148. doi: 10.1080/00401706.2021.2020521. url: https://doi.org/10.1080/00401706.2021.2020521 (cit. on p. 35).

Ribeiro, D., L. M. Matos, G. Moreira, et al. (2022). "Isolation Forests and Deep Autoencoders for Industrial Screw Tightening Anomaly Detection". In: *Comput.* 11.4, p. 54. doi: 10.3390/computers11040054. url: https://doi.org/10.3390/computers11040054 (cit. on pp. 3, 69).

Fawcett, T. (2006b). "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27, pp. 861–874 (cit. on pp. 32, 48).