Contribution to the development of a methodology coupling natural language processing and machine learning to react to production disturbances.

Juan Pablo Usuga Cadavid

HESAM UNIVERSITÉ

Arts et Métiers Sciences et Technologies

LAMIH UMR CNRS 8201

ÉCOLE DOCTORALE SCIENCES DES MÉTIERS DE L'INGÉNIEUR
[LAMIH UMR CNRS 8201 – Campus de Paris]

THÈSE

présentée par : **Juan Pablo USUGA CADAVID**

soutenue le : **13 octobre 2021**

pour obtenir le grade de : **Docteur d'HESAM Université**

préparée à : **École Nationale Supérieure d'Arts et Métiers**

Spécialité : **Informatique-traitement du signal (AM)**

[FR] Contribution à la définition d'une méthodologie couplant le traitement automatique du langage naturel et l'apprentissage automatique pour réagir aux perturbations de production

[EN] Contribution to the development of a methodology coupling natural language processing and machine learning to react to production disturbances

THÈSE dirigée par :

[Professeur LAMOURI Samir]

et co-dirigée par :

[Professeur GRABOT Bernard]

Jury

Mme Gülgün ALPAN-GAUJAL, Professeure, Grenoble INP	Rapporteure
Mme Mitra FOULADIRAD, Professeure, UTT	Examinatrice
M. Dimitrios KYRITSIS, Professeur, EPFL	Examinateur
M. Kary FRÄMLING, Professeur, Université Aalto	Rapporteur
M. Bernard GRABOT, Professeur, INP-ENIT	Examinateur
M. Samir LAMOURI, Professeur, Arts et Métiers	Examinateur
M. Nazih MECHBAL, Professeur, Arts et Métiers	Examinateur

THÈSE

To my parents.

Essentially all models are wrong, but some are useful – G. Box

Acknowledgements

First and foremost, I wish to express my deepest gratitude to my thesis advisers M. Samir Lamouri and M. Bernard Grabot. Their advice monumentally contributed to the success of my work. In addition, their guidance helped me grow personally. I will undoubtedly owe them for the future successes in my professional and personal lives.

I would like to recognise the thesis jury members: Ms Gülgün ALPAN-GAUJAL, Ms Mitra FOULADIRAD, M. Dimitrios KYRITSIS, M. Kary FRÄMLING, and M. Nazih MECHBAL. It is an honour to receive their feedback and insights on my work. Additionally, I would like to thank M. Robert PELLERIN, who unexpectedly contributed to my work and was always available to support my ideas.

I want to thank the industrial partner in this research, iFAKT France, who supported my work even though it evolved from an unexplored idea. Notably, I wish to thank M. Arnaud FORTIN, who allowed me to acquire professional experience throughout the process of this thesis.

I want to acknowledge my laboratory colleagues: Ms Estefania TOBON VALENCIA, Ms Léa SATTLER, Ms Angie NGUYEN, and Ms Aurélie EDOUARD, and Frédéric ROSIN, who aided me throughout the development of this thesis, encouraging rich discussions that helped me progress in transverse domains.

I cannot express sufficient gratitude to my parents, whose support and guidance have built the foundation of everything that I have accomplished. They have unwaveringly trusted in me since the beginning and helped in achieving all my dreams. Finally, I want to thank Thaís CARBINATTI, whose unconditional support has helped me more than she thinks.

Résumé

Dans l'ère de l'industrie 4.0, exploiter les données stockées dans les systèmes d'information est un axe d'amélioration des systèmes de production. En effet, ces bases de données contiennent des informations pouvant être utilisées par des modèles d'apprentissage automatique (AA) permettant de mieux réagir aux futures perturbations de la production. Dans le cas de la maintenance, les données sont fréquemment récupérées au moyen de rapports établis par les opérateurs. Ces rapports sont souvent rédigés en utilisant des champs de saisie en textes libres avec comme résultats des données non structurées et complexes : elles contiennent des irrégularités comme des acronymes, des jargons, des fautes de frappe, etc. De plus, les données de maintenance présentent souvent des distributions statistiques asymétriques : quelques évènements arrivent plus souvent que d'autres. Ce phénomène est connu sous le nom de « déséquilibre de classes » et peut entraver l'entraînement des modèles d'AA, car ils ont tendance à mieux apprendre les évènements les plus fréquents, en ignorant les plus rares. Enfin, la mise en place de technologies de l'industrie 4.0 doit assurer que l'être humain reste inclus dans la boucle de prise de décision. Si cela n'est pas respecté, les entreprises peuvent être réticentes à adopter ces nouvelles technologies.

Cette thèse se structure autour de l'objectif général d'exploiter des données de maintenance pour mieux réagir aux perturbations de la production. Afin de répondre à cet objectif, nous avons utilisé deux stratégies. D'une part, nous avons mené une revue systématique de la littérature pour identifier des tendances et des perspectives de recherche concernant l'AA appliqué à la planification et au contrôle de la production. Cette étude de la littérature nous a permis de comprendre que la maintenance prédictive peut bénéficier de données non structurées provenant des opérateurs. Leur utilisation peut contribuer à l'inclusion de l'humain dans l'application de nouvelles technologies. D'autre part, nous avons abordé certaines perspectives identifiées au moyen d'études de cas utilisant des données issues de systèmes de productions réels. Ces études de cas ont exploité des données textuelles fournies par les opérateurs qui présentaient des déséquilibres de classes. Nous avons exploré l'utilisation de techniques pour mitiger l'effet des données déséquilibrées et nous avons proposé d'utiliser une architecture récente appelée « *transformer* » pour le traitement automatique du langage naturel.

Mots clés : Apprentissage automatique, Traitement automatique du langage naturel, Industrie 4.0, Apprentissage profond, Maintenance.

Abstract

In the age of Industry 4.0 (I4.0), exploiting data stored in information systems offers an opportunity to improve production systems. Datasets stored in these systems may contain patterns that machine learning (ML) models can recognise to react more effectively to future production disturbances. In the case of industrial maintenance, data are frequently collected through reports provided by operators. However, such reports are often provided using free-form text fields, resulting in complex unstructured data; therefore, they may contain irregularities such as acronyms, jargon, and typos. Furthermore, maintenance data often present asymmetrical distributions, where certain events occur more frequently than others. This phenomenon is known as class imbalance, and it can hinder the training of ML models as they tend to recognise the more frequent events better, ignoring rarer incidents. Finally, when implementing I4.0 technologies, the inclusion of humans in the decision-making process must be ensured. Otherwise, companies may be reluctant to adopt new technologies.

The work presented in this thesis aims to tackle the general objective of harnessing maintenance data to react more effectively to production disturbances. To achieve this, we employed two strategies. First, we performed a systematic literature review to identify the research trends and perspectives regarding the use of ML in production planning and control. This literature analysis allowed us to understand that predictive maintenance may benefit from the unstructured data provided by operators. Additionally, their usage can contribute to the inclusion of humans in the implementation of new technologies. Second, we addressed some of the identified research gaps through case studies that employed data from real production systems. These studies harnessed the free-form text data provided by operators and presented class imbalance. Hence, the proposed case studies explored techniques to mitigate the effect of imbalanced data; moreover, we also suggested the use of a recent architecture for natural language processing called *transformer*.

Keywords: Machine learning, Natural language processing, Industry 4.0, Deep learning, Maintenance.

Table of contents

Table of figures

List of tables

List of abbreviations and acronyms

A	Artificial data
AA	Apprentissage Automatique
Ada	AdaBoost
AGV	Automated Guided Vehicle
ANRT	Association Nationale de la Recherche et de la Technologie
API	Application Programming Interfaces
BDA	Big Data and Analytics
CA	Contextualized Analysis or application
CBM	Condition-Based Maintenance
CE	Cross-Entropy
CMS	Condition Monitoring System
CNN	Convolutional Neural Network
CRM	Customer Relationship Management system
CUA	Commonly Used Activities
DA	Data Acquisition system design and integration
DC	Data Cleaning and formatting
DE	Data Exploration
DL	Deep Learning
DPS	Dynamic Production Schedule
DT	Decision Trees
E	Equipment data
ERP	Enterprise Resource Planning
FE	Feature Extraction
FL	Focal Loss
FS	Feature Selection
FT	Feature Transformation
HT	Hyperparameter Tuning and architecture design
I4.0	Industry 4.0
INRIA	National Institute for Research in Computer Science and Automation
IoT	Internet of Things
LDA	Linear Discriminant Analysis
LIME	Local Interpretable Model-agnostic Explanations
LR	Logistic Regression
LSTM	Long Short-Term Memory neural network
M	Management data
MC	Model Comparison and selection
MCC	Matthews Correlation Coefficient
MEM	Mandatory Elements of a Method
MES	Manufacturing Execution System
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
MT	Model Training, validation, testing, and assessment

MTM	Method Time Measurement
MU	Model Update
MUA	Medium Use Activities
NLP	Natural Language Processing
NN	Neural Network
OC	Overrepresented Class
OEE	Overall Equipment Effectiveness
OUA	Often Used Activities
P	Product data
Pb	Public data
PCA	Principal Component Analysis
PHM	Prognosis & Health Management
PLC	Programmable Logic Controllers
PPC	Production Planning and Control
PS	Production Schedule
QDA	Quadratic Discriminant Analysis
R	Restriction
RF	Random Forest
R-FCN	Region-based Fully Convolutional Network
RGB	Red, Green, and Blue
RL	Reinforcement Learning
RNN	Recurrent Neural Networks
ROS	Random Oversampling
RQ	Research Question
RUL	Remaining Useful Life
RUS	Random Undersampling
S1	Specificity
S2	Sensitivity
SCADA	Supervisory Control And Data Acquisition
SHAP	SHapley Additive exPlanations
SL	Supervised Learning
SQ	Specific Question
SUA	Seldom Use Activities
SVM	Support Vector Machines
TF-IDF	Term Frequency-Inverse Document Frequency
TL	Transfer Learning
U	User data
UC	Underrepresented Class
UL	Unsupervised Learning
WCE	Weighted Cross-Entropy
XGB	XGBoost

Remarks on the industrial '*CIFRE'* partnership of this thesis

This thesis was developed in the framework of a CIFRE (which stands for *Conventions Industrielles de Formation par la Recherche* or Industrial Agreements for Training through Research, in English) partnership with the *Association Nationale de la Recherche et Technologie* (ANRT) under Grant 2018/1266. The research activities were performed at *Arts et Métiers – Sciences et Technologies* (*Ecole Nationale Supérieure d'Arts et Métiers*) in the Laboratory of Automation, Mechanics and Industrial and Human Computer Science (LAMIH UMR CNRS 8201) with the support of the industrial partner, iFAKT France SAS, which develops solutions for the optimisation of industrial processes and monitoring of the shop floor.

The objective of this partnership was to develop research approaches that can harness the data collected from the shop floor for reacting effectively to production disturbances for future integration with the solutions of the company. More specifically, two of the software products of the company were considered, i.e. Integrated Manufacturing Solutions and Integrated Manufacturing Operator (IMS and IMO, respectively). The former performs production scheduling, while the latter serves as a lightweight MES (Manufacturing Execution System), informing operators about the tasks that they must execute and collecting data from the shop floor. Hence, the aim of this research was to exploit the data that can be collected using the IMO to perform predictions and support decision making to allow a more informed production rescheduling with IMS.

My contributions in this research were to:

- Propose a use case for the data collected through IMO to react effectively to production disturbances.
- Support the theoretical development and feasibility tests for the use case, highlighting the challenges and lessons learnt.
- Explore solutions to some of the encountered challenges.

1. Chapter 1: Introduction

1.1. Context

A recent study that used text mining to analyse around 3800 scientific publications related to Industry 4.0 (I4.0) and supply chains identified that predictive maintenance (PdM) is a major research trend (Nguyen et al., 2021). It offers promising improvements by reducing maintenance costs (-25 to 35%), machine downtime (-35 to 45%), breakdown occurrences (-70 to 75%), in addition to increasing production (+25 to 35%) (Sullivan et al., 2010; Montero Jimenez et al., 2020).

PdM aims to organise maintenance activities depending on the health status of the equipment. Thus, maintenance interventions are performed when necessary. This maximises the usage of machines, reduces material costs as parts are replaced when required, and reduces labour costs by mitigating the time spent performing maintenance interventions owing to a better characterisation of breakdowns (Carvalho et al., 2019; Montero Jimenez et al., 2020). Further, by making more informed decisions through predictive analysis, planners can improve the scheduling of future maintenance interventions (Zhang et al., 2019).

Several studies have suggested the following three types of models for enabling PdM (Zhang et al., 2019; Montero Jimenez et al., 2020; Zonta et al., 2020):

1) Knowledge-based models: They are models based on experience that can be represented by sets of rules, cases obtained from previous events, or rules using fuzzy logic. Examples of these are rule-, case-, and fuzzy knowledge-based models (Montero Jimenez et al., 2020).
2) Data-driven models: These models rely on statistics, pattern recognition and artificial intelligence. They can be further classified into statistical models that analyse the behaviour of random variables, stochastic models that study the evolution of random variables over time, and machine learning (ML) models, which are computer programs that can learn from experience (e.g. data) to perform a specific task (Jordan and Mitchell, 2015; Montero Jimenez et al., 2020).
3) Physics-based models: These models employ the laws of physics and mathematical modelling to represent phenomena such as the erosion of tubes, fatigue, or crack propagation. In practice, these models are customarily implemented using simulations, such as finite element methods (Montero Jimenez et al., 2020; Zonta et al., 2020).

Montero Jimenez et al. (2020) suggested a fourth type of model, named multi-model approach, which combines several models. These combinations can group models of different types, such as data-driven with physics-based models or a combination of the same type, such as various knowledge-based models.

This thesis focuses only on data-driven models from these four types of models, specifically, the ML models. Physics-based models often require advanced knowledge in physics and mathematics; however, it is challenging to accurately represent certain physical phenomena. Even if the knowledge-based models can be easily interpreted by practitioners, it may be difficult to formalise experience into reliable rules to perform accurate predictions (Montero Jimenez et al., 2020). For instance, in a study conducted by Ruiz-Sarmiento et al. (2020), the proposed data-driven approach for machine health state estimation achieved 20% better predictive performance than the baseline knowledge-based model. Additionally, it was more reliable in situations with higher uncertainty. Finally, continuous improvements in data acquisition systems have led to a growth in the collected data volumes, which has fostered the interest, application, and success of data-driven models in PdM (Zhang et al., 2019).

Despite the potential advantages, implementing PdM solutions in real-world applications is usually limited by the data acquisition and monitoring system (Acernese et al., 2020). This is because systems relying on sensors and actuators can be expensive to design and maintain. Further, the management of such systems may require rare expertise. Another issue is that data acquisition and monitoring systems are primarily suited to industries with high investment in heavy engineering equipment, where failures can provoke considerable financial losses, such as the semiconductor, oil and gas, and energy industries (Acernese et al., 2020). For other sectors with the rare occurrences of failures or discretised production with relatively low output volumes, it may be difficult to identify machines with dozens of sensors for collecting real-time data. Such configurations could be more expensive than simply relying on classic preventive maintenance (Kusiak, 2019; Acernese et al., 2020). Finally, according to Dalzochio et al. (2020), generating data from sensors poses other challenges; for instance, sensor data often achieves big data scales, where maintaining good performance in terms of latency, scalability, and bandwidth may be challenging.

One way of creating PdM systems that require limited use of sensors and actuators is to harness historical data. Companies have been computerised for a long time, which has allowed them to

store large volumes of data over the years (Grabot, 2020). Using these data sources (e.g. maintenance logs) may allow the design of predictive models with reduced investment in capital expenditures, such as brand new equipment with sensors. However, most of the research on PdM focuses on harnessing time-series data produced by sensors (Montero Jimenez et al., 2020). Historical maintenance data tend to be ignored because of their highly unstructured nature; for example, they often contain text-based data from free-form comments left by operators (Montero Jimenez et al., 2020; Usuga Cadavid et al., 2020b).

Harnessing the free-form text data originating from the shop floor may be challenging as operators tend to use jargon and acronyms, and the data usually contain typos (Usuga Cadavid et al., 2020b). Moreover, two operators can provide different descriptions of the same issue. Thus, the data collection is highly dependent on the judgement, perception, and assumptions of the person who is describing the phenomena; therefore, it can be considered as a subjective data source (Razmi-Farooji et al., 2019). Despite the difficulties in harnessing free-form text data originating from maintenance, ignoring them would be a potential waste, as they encapsulate knowledge from operators that can be meaningful for PdM. Further, operators may feel more comfortable when using PdM systems using their text descriptions as inputs. This is essential to position the human at the core of digital innovations, which is a crucial requirement for encouraging industries to adopt I4.0 technologies (Usuga Cadavid et al., 2020a).

Apart from the highly unstructured nature of text data present in maintenance logs, this data source presents another characteristic hinderance in its exploitation, i.e. intrinsic imbalance. This implies that certain events are naturally over-represented with respect to others (Johnson and Khoshgoftaar, 2019). For instance, noncritical issues that do not block production are common, while critical problems that cripple the system are rarer. Moreover, maintenance interventions requiring only minimal time to be fixed are numerous, while serious events requiring a longer resolution time tend to be rarer. This imbalance adds an extra layer of complexity to the exploitation of maintenance data. In fact, data imbalance severely affects ML models, as these tend to learn the majority classes while ignoring the minorities. Such behaviour may be unacceptable if an ML model cannot identify severe cases leading to a blockage in production. Figure 1.1 illustrates an example of the imbalanced distribution in the machine problems that can stop or continue the work in a production system. In the figure, machine problems leading to a stop in the production system are the minority class, representing approximately 9% of the dataset.

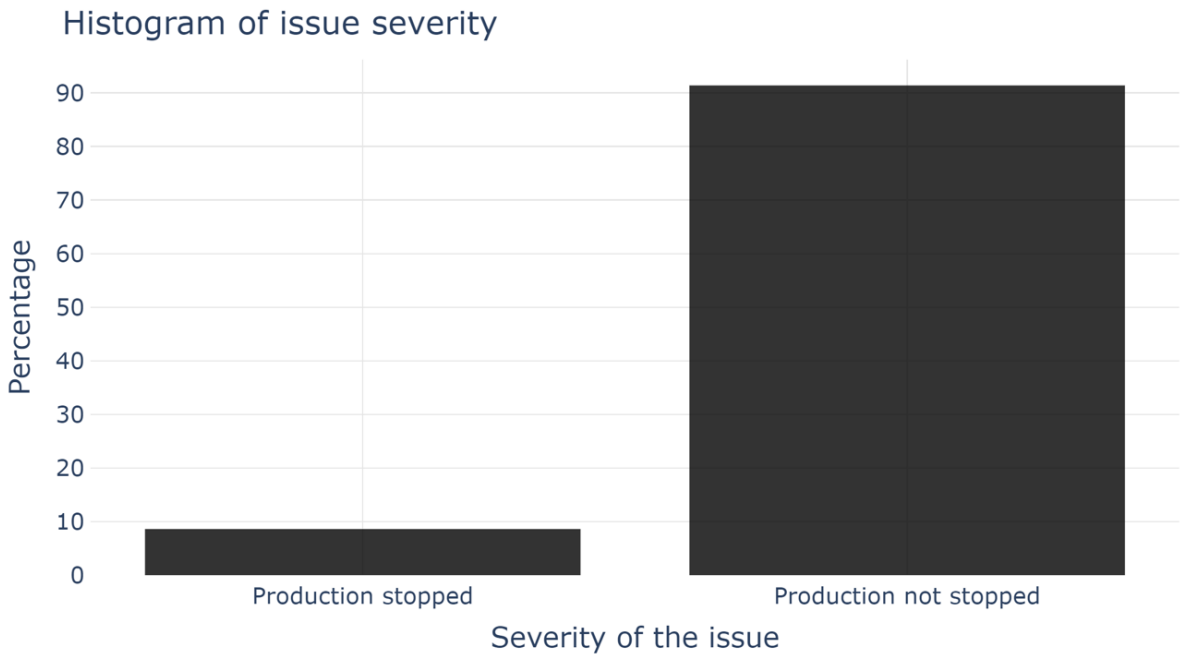


Figure 1.1 Example of imbalanced distribution in maintenance. Figure adapted from (Usuga Cadavid et al., 2020b).

Having presented the advantages of exploiting text data that are available in maintenance logs and some of the challenges, this research focused on using natural language processing (NLP) and ML to value maintenance text data generated on the shop floor to react more effectively to production disturbances. The remainder of this chapter explains the key concepts for this work, presents the proposed approach to address the previously introduced challenges, and describes the structure of the thesis.

1.2. Concepts

1.2.1. Predictive maintenance (PdM)

Estimates suggest that maintenance costs can range between 15 and 60% of operating costs in manufacturing (Zonta et al., 2020). Hence, several strategies have been proposed to manage maintenance actions more effectively. In the scientific literature, authors tend to converge the maintenance approaches into the following four primary approaches (Montero Jimenez et al., 2020; Zonta et al., 2020): corrective, preventive, predictive, and prescriptive maintenance. In corrective maintenance, interventions are performed when signs of degradation or failure have occurred. In preventive maintenance, time intervals are fixed to perform maintenance actions, for instance, by relying on measures such as the number of cycles, kilometres, hours. In PdM,

the precise moment to trigger maintenance interventions is estimated. For instance, by estimating the remaining useful life (RUL) of components or health status of the equipment or by characterising the machine condition, the requirement for maintenance is determined. Finally, prescriptive maintenance allows answering questions based on 'what-if' scenarios, such as 'how to achieve it?' or 'how to encourage the occurrence of a certain event?'

Adopting a corrective maintenance strategy may lead to higher risks of machines being unpredictably unavailable. In contrast, preventive maintenance can provide satisfactory results in terms of machine availability. However, it is known to be wasteful, as components may be replaced despite a high RUL (Montero Jimenez et al., 2020). This thesis focuses on PdM, as it offers opportunities to harness produced data on the shop floor for superior management of maintenance actions. Prescriptive maintenance is presented as the next step of PdM; thus, it is retained as a research perspective. In fact, PdM models may be the basis for supporting effective prescriptive models that are capable of providing the appropriate insights at the appropriate time.

In the scientific literature, two other terms have been used in applications related to PdM, i.e. condition-based maintenance (CBM) and prognosis and health management (PHM). Among these two, PHM was introduced more recently in the early 2000s, while CBM was introduced in the 1940s (Montero Jimenez et al., 2020). Minor differences were observed between these terms. For instance, certain authors have proposed that CBM should primarily focus on monitoring the condition of the equipment by measuring parameters such as temperature, humidity, and vibration with the final objective of detecting symptomatic variations leading to failures (Ghasemi et al., 2007; Acernese et al., 2020). Acernese et al. (2020) proposed that PdM is an evolution of CBM, where the idea is to use variables monitored using CBM to predict the degradation of an item, and thus plan future maintenance actions. Despite the existence of differences, this thesis proposes the use of CBM and PHM under the same 'umbrella term' of PdM. According to a recent systematic review published by Montero Jimenez et al. (2020), there is no consistent difference in the usage of these terms in the relevant scientific literature, and they have been used indistinctly to refer to the same research field. Finally, this thesis focuses on the characterisation of machine health status from the descriptions provided by operators.

1.2.2. Machine learning (ML) and other data-related terms

ML was defined as a computer program capable of learning from experience to perform a specific task (Jordan and Mitchell, 2015). Nevertheless, the term ML when applied to research in the context of I4.0 also suffers from the same issue as PdM, i.e. it is usually used interchangeably with other terms. For example, it is common to observe different terms, such as data mining, statistical learning, data analytics, and artificial intelligence when referring to ML applications.

Certain studies have explored this issue in the context of I4.0. For instance, Usuga Cadavid et al. (2020a) analysed the results related to production planning and control when querying scientific databases with the following keywords: 'Deep Learning AND Machine Learning', 'Data Mining', and 'Statistical Learning'. The findings suggest that ML is associated with more recent research than data mining and statistical learning. The study performed by Schuh et al. (2019) proposed that ML and data mining have a causal relationship, where ML is applied in data mining to generate results. However, this previous study did not analyse the context in which data-related concepts are currently used in research.

Motivated by the lack of consistency in the definitions found in the literature, we performed a collaborative study with A. Nguyen at the Laboratory of Automation, Mechanics and Industrial and Human Computer Science (LAMIH UMR CNRS 8201) to employ text mining to understand the usage of data-related concepts in the recent research on I4.0 and supply chains. Metadata (i.e. titles, abstracts, and keywords) of scientific publications were used to achieve this goal. The paper was presented at the 'SOHOMA'2020: 10th Workshop on Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future' conducted in Paris, France. The complete study can be obtained in (Nguyen et al., 2021). For the analysis, we considered the following seven common data-related concepts: ML, data analytics, big data, data mining, artificial intelligence, data engineering, and data management. This thesis presents the following three primary results for each predefined concept: the temporal evolution of the usage, the most related terms, and a disambiguation matrix proposing definitions, similarities, and differences.

Several scientific databases were queried to conduct the study, resulting in a sample containing 3858 articles. Figure 1.2 shows the described methodology, while Appendix A details the selection criteria for the scientific literature.

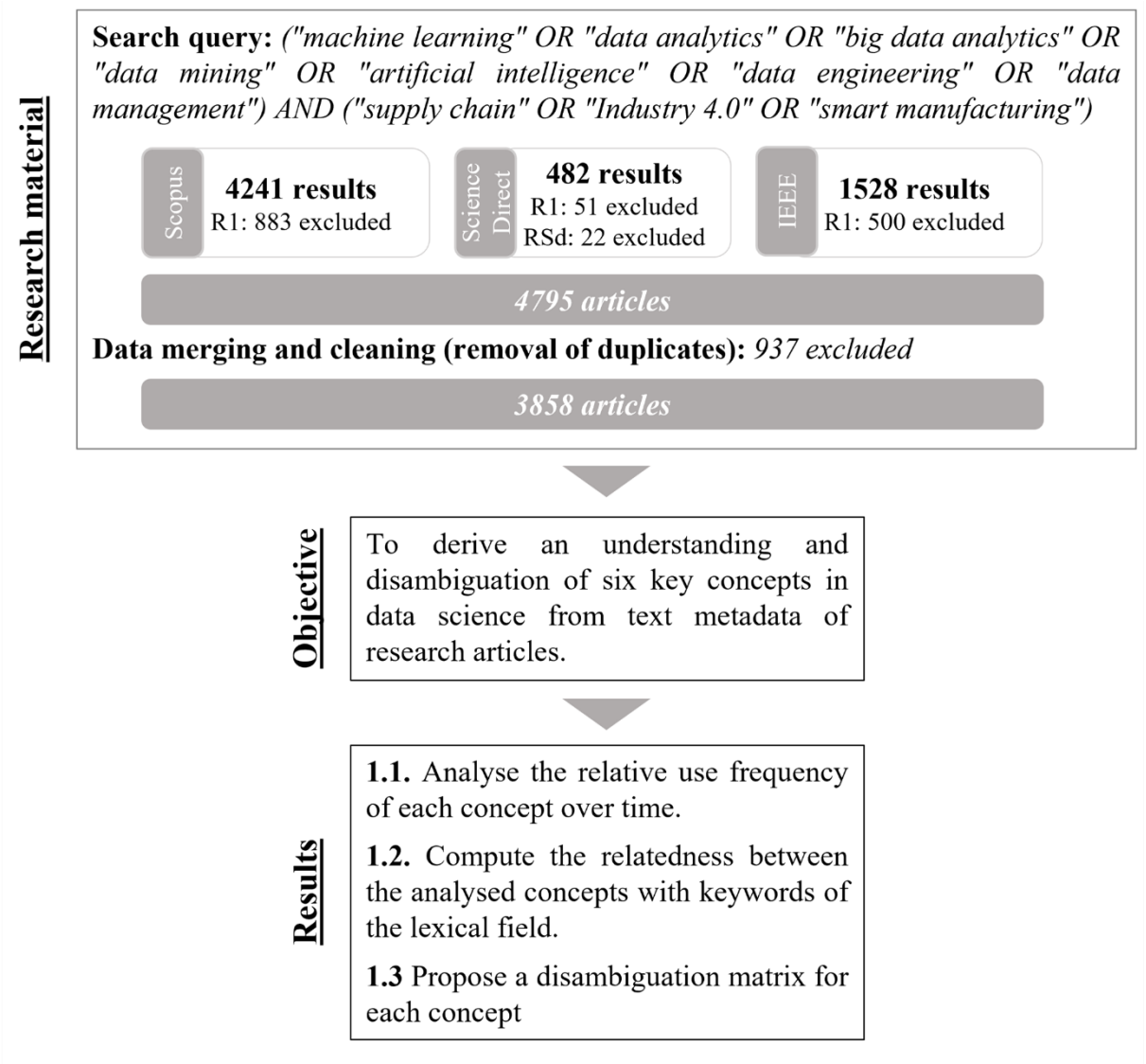


Figure 1.2 Methodology for article selection. Figure adapted from (Nguyen et al., 2021).

1.2.2.1. Temporal evolution for the concept usage

To understand how each concept has been used over time, we calculated the yearly relative frequency between 2011 and 2020 by measuring the percentage of papers using a specific concept in the title, abstract, or keywords. Figure 1.3 shows the results.

Figure 1.3 Relative frequency for each data-related concept across time (Nguyen et al., 2021).

The results confirm the findings of Usuga Cadavid et al. (2020a), where ML is related to more recent research than data mining. The popularity of data mining in research appears to decrease over time, probably replaced by the concepts of ML, data analytics, and big data. Artificial intelligence has increased in popularity over time. This result is expected, as ML is one of the subfields of artificial intelligence. Hence, researchers are more likely to mention both these terms in their studies. Finally, the term 'data engineering' was not found in any title, abstract, or keywords of the sample of papers. This may be because data engineering is more prevalent among industry practitioners than among academicians.

1.2.2.2. Relatedness to other concepts

The Jaccard coefficient was used to measure the relatedness between concepts. It measures the overlap between two sets, and ranges from 0 (no overlap) to 1 (complete overlap). For each proposed concept, the Jaccard coefficient was calculated using the most frequent keywords of the paper sample. Table 1.1 lists the top 10 most related terms to each of the proposed concepts, ranked based on the Jaccard coefficient.

Machine learning		**Data analytics**		**Big data**	
Item	*Rel.*	*Item*	*Rel.*	*Item*	*Rel.*
prediction	0,122	big data	0,372	data analytics	0,372
classification	0,113	internet of things	0,123	internet of things	0,195
neural network	0,109	cloud computing	0,079	cloud computing	0,136
artificial intelligence	0,107	sensors	0,075	artificial intelligence	0,116
big data	0,098	decision making	0,062	machine learning	0,098
internet of things	0,098	machine learning	0,059	security	0,074
sensors	0,078	optimisation	0,059	sensors	0,061
optimisation	0,070	security	0,055	decision making	0,060
deep learning	0,068	automation	0,054	automation	0,058
security	0,066	simulation	0,042	data management	0,055
Data mining		**Artificial intelligence**		**Data management**	
Item	*Rel.*	*Item*	*Rel.*	*Item*	*Rel.*
classification	0,088	internet of things	0,169	RFID	0,076
clustering	0,085	big data	0,116	Security	0,058
prediction	0,077	machine learning	0,107	internet of things	0,057
neural network	0,059	robotics	0,089	big data	0,055
optimisation	0,058	automation	0,088	blockchain	0,054
simulation	0,058	security	0,076	automation	0,043
forecasting	0,055	cloud computing	0,074	sensors	0,041
big data	0,053	blockchain	0,068	decision making	0,039
RFID	0,051	neural network	0,066	cloud computing	0,038
decision making	0,047	optimisation	0,065	clustering	0,027

Table 1.1 Top 10 most related notions to each concept. Table adapted from (Nguyen et al., 2021)

The results show that ML and data mining share common notions such as *prediction*, *classification*, *neural networks*, *big data*, and *optimisation*. However, ML is more related to concepts related to data sources (e.g. internet of things, sensors), while data mining is more linked to decisions (e.g. decision making, forecasting). This may be because ML is employed to harness data from different data sources to support decision making through data mining. This is aligned with the causal relationship between ML and data mining, as proposed by (Schuh et al., 2019).

Big data is the most related term to data analytics, and vice versa. This suggests that these two concepts are strongly intertwined. Moreover, they both share several common concepts, such as most related terms (e.g. internet of things, sensors, cloud computing). This is probably because big data applications typically use data analytics to extract insights from the collected data. It is interesting to note that data analytics is also closely related to *decision making*, similar to the case of data mining. This fact and the observed trend in Figure 1.3 suggest that data analytics is replacing the concept of data mining in recent research.

1.2.2.3. Disambiguation matrix

The results obtained with the first and second objectives of our text mining study were employed to propose a disambiguation matrix. This matrix contains the proposed definition for all six concepts found in the literature. Their similarities and differences are also presented. These definitions are used within the framework of this thesis. Figure 1.4 shows the matrix, where the white cells describe the concept, green cells contain similarities, and red cells outline the differences between the terms.

The disambiguation matrix illustrates that ML, artificial intelligence, data mining, and data analytics have similar meanings. However, the matrix highlights the following differences between these concepts:

1) ML is a subfield of artificial intelligence. Thus, not all applications of artificial intelligence are necessarily related to ML.
2) Data analytics aims to generate insights from data using multidisciplinary techniques, for instance, by using artificial intelligence but not only.
3) Data mining also seeks to create insights from data using statistical models and algorithms, which implies that data mining also draws from artificial intelligence. However, Table 1.1 shows that data analytics is more closely related to new technologies such as the *internet of things*, *cloud computing*, and *sensors*. Data mining tends to be associated with theoretical considerations, such as *classification*, *clustering*, or *optimisation*.

	Machine learning	Data analytics	Big data	Data mining (DM)	Artificial intelligence (AI)	Data management
Machine learning	Computer programs learning from data to improve a performance metric at a certain task [12].	While data analytics targets decision making from data, ML refers to algorithms that can be used to do so.	Big data refers to the data itself while ML involves techniques using them as a resource.	While DM targets decision making from data, ML refers to algorithms that can be used to do so.	AI systems often involve, but are not limited to, using ML	Data management aims to provide data availability, security, and quality, which is not the case for ML.
Data analytics	Data analytics harnesses ML to extract patterns from data, supporting decision making.	Process using multidisciplinary techniques to get insights from data [14].	Big data refers to the data itself while data analytics focuses on deriving insights from them.	Data analytics is more likely to be associated with new technologies and industrial applications, while DM is associated with theoretical considerations.	AI encapsulates other concepts and fields beyond data analytics.	Unlike data management, data analytics aims to support decision making.
Big data	Big data is one of the main resources to train ML models.	Data analytics aims to extract insights from big data.	Data streams characterised by high volume, variety, and velocity [7].	Big data refers to the data itself while DM focuses on deriving insights from them.	AI systems often use big data but are not limited to this type of resource.	Big data refers to the data resources while data management relates to the handling of them.
Data mining (DM)	DM harnesses ML to extract patterns from data, supporting decision making.	DM and data analytics both aim to support decision making using data as a resource.	DM aims to extract insights from big data.	Process using statistical models and algorithms to derive insights from data.	AI encapsulates other concepts and fields beyond DM.	Unlike data management, DM aims to support decision making.
Artificial intelligence (AI)	ML is a subdomain of AI.	AI can employ data analytics to build intelligent systems.	Big data can be used to create intelligent systems through AI.	AI can employ DM to built intelligent systems.	Research field aiming to understand and build intelligent entities [13].	AI encapsulates other concepts and fields beyond the management of information systems.
Data management	Data management stores, prepares, and retrieves data for ML applications and ML algorithms can be used in data management.	Data management stores, prepares, and retrieves data to perform analytics.	Big data is made available through data management.	Data management stores, prepares, and retrieves data for DM applications.	AI includes information availability and security matters, which require data management.	Process consisting of collecting, storing, preparing, and retrieving data to ensure availability, security, and quality [7].

Legend	
Similarities	
Differences	
Definition	

Figure 1.4 Disambiguation matrix for the data-related concepts. Figure adapted from (Nguyen et al., 2021)

1.2.3. Natural language processing

The field of NLP addresses a rather specific goal, which is as old as the idea of computers, i.e. to endow computers with the capability of processing human language to perform valuable tasks (Jurafsky and Martin, 2009). To achieve this, NLP employs techniques derived from computer science to create linguistic systems with multiple purposes, such as learning, understanding, and producing human language content (Hirschberg and Manning, 2015).

What characterises NLP applications is the knowledge of language. For example, a program that counts the number of words in a document belongs to the field of NLP, as it requires

knowledge of what a word is (Jurafsky and Martin, 2009). However, human language is complex. For an NLP system to be proficient, it must grasp several aspects of linguistic knowledge as specified subsequently (Jurafsky and Martin, 2009):

1) Phonetics and phonology: to capture sounds to understand what was said and to communicate with humans.
2) Morphology: to understand how to form words and break them into components; for example, when creating plurals for a particular word or recognising the meaning of contractions such as '*can't*'.
3) Syntax: to accurately represent words and sentences according to predefined structures and relationships dictated by a particular language.
4) Semantics: to consider the meaning of words (lexical semantics) and particular groups of words conveying a specific meaning (compositional semantics), such as the '*European Union*'.
5) Pragmatics: to understand the intentions conveyed by the speaker, which is an essential requirement to differentiate statements from questions or orders.
6) Discourse: to be aware of the actual meaning of linguistic units when they refer to other pieces of information. For example, if a question answering system analyses a text about the invasion of Russia by the French during the Napoleonic Wars and it is asked, '*why did he order the invasion*?', the system must know that the word '*he'* refers to Napoleon.

NLP applications in I4.0 are not necessarily required to tackle all the aforementioned aspects of linguistic knowledge to provide valuable results. However, these aspects shed some light on the complexity and challenges of NLP. In the context of this thesis, we will focus on the application of data-driven models in NLP on text data, specifically, to perform text classification of maintenance reports. These texts are particularly challenging, as operators may use domain-specific acronyms and abbreviations to describe maintenance issues. Moreover, text quality is often low, as workers prefer to be fast rather than spend time on producing an elegant problem description. To illustrate this, Table 1.2 lists certain examples of a real dataset used in this thesis. This dataset contains free-form text descriptions of machine symptoms as an input, and the output is a label indicating whether this resulted in a production stop. For language consistency, both the original version in French and the translation to English are

provided. Further, personal details such as operator names are hidden and represented as '[Hidden – operator name]' or '[*caché – nom de l'opérateur*]' in French.

In Table 1.2, the original text in French exemplifies some of the challenges when harnessing free-form text data in maintenance. For instance, No.1 has a typo in the word '*Continu*', No.2 shows the use of the acronym '*HS*', which stands for 'Hors Service' ('out of order'), and in No.3, the operator wrote the non-existent word '*KC*', which is the phonetic abbreviation for '*cassé*' (broken, in French). In No.4, there are words indicating measures such as '*m*' for metre and '*mm*' for millimetre. An NLP system should recognise that a deviation of 2 mm did not lead to a production stop in this context. No.1 and No.3 show that operators tend to add personal information, such as the name of the person who evaluated the issue in this case. Although this information is useful for a maintenance planner analysing the report, it is probably useless for an algorithm trying to determine the severity of a machine symptom. Finally, No.2 and No.3 exemplify that certain messages can be relatively short, providing limited information on the issue itself.

No.	Original text (French)	English version	Label
1	Continu à se déplacer en mouvement transversal. Coupure du sectionneur et etiquette rouge sur boitier de commande mise par [Caché – nom de l'opérateur] car jugé tres dangereux.	Keeps moving in transverse motion. Circuit breaker was disconnected, and a red label was placed on the control box by [Hidden – operator name] because it was considered too dangerous.	Production stopped
2	IMPORTANT. LAME DE SCIE HS.	IMPORTANT. SAW BLADE OUT OF ORDER.	Production stopped
3	BOUTON DE MISE SOUS TENSION KC [Caché – nom de l'opérateur]	POWER UP BUTTON BROKEN [Hidden – operator name]	Production stopped
4	PROBLEME DIMESIONNELLE APRES COUPE. Lors de la coupe d'une plaque de 2 m, il y a une difference de 2mm d'un bout à l'autre. Difference entre l'affichage et la coupe réelle.	DIMENSIONAL PROBLEM AFTER CUTTING. While cutting a 2 m plate, there is a 2 mm difference from one end to the other. Difference between the display and the real cut.	Production not stopped

Table 1.2 Examples from a real maintenance dataset

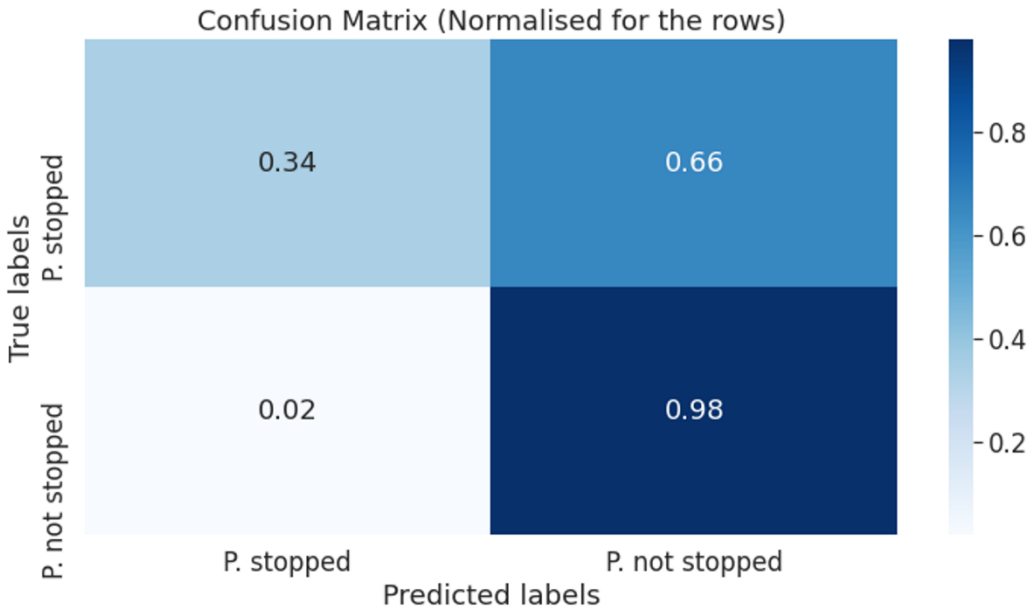
One of the critical steps for handling text data is producing meaningful numerical representations that data-driven models can employ. Hence, this thesis primarily focuses on a recent deep learning (DL) architecture called *transformer* to obtain richer numerical representations. Chapter 5 and Chapter 6 further explain the specificities and advantages of using such architecture.

1.2.4. Imbalanced classification

Imbalanced distributions are common when events that tend to occur less frequently exist. It can be the case of detecting a rare disease, credit card fraud, or a scarce animal. Imbalance distributions can severely degrade the performance of ML models as learning is usually an optimisation process. Hence, the ML model will predominantly learn the majority classes, as recognising them is sufficient to achieve a good optimisation performance. Consequently, the ML model misclassifies minority classes as majority classes.

Furthermore, the standard performance measures used to assess ML models provide limited results for imbalanced classification. For example, using classification accuracy would lead to spurious conclusions, such as achieving 95% accuracy in a dataset consisting of 95% of the majority class can imply that the algorithm only learnt how to classify the majority class accurately. Moreover, misclassifying minority classes can be harmful in certain contexts; for example, classifying patients as healthy while having certain dangerous heart disease would completely invalidate the use of the ML model.

In the context of maintenance, when classifying maintenance reports into those leading to a production stop and those not leading to it, not detecting that a problem will block the production can be expensive, as this may cripple the production in the future. However, misclassifying a noncritical issue as severe is also problematic, as this would trigger unnecessary alerts, resulting in a loss of productivity. To illustrate the effect of class imbalance, Figure 1.5 shows the confusion matrix for an ML model trained to classify maintenance reports into two classes: production was stopped and production was not stopped. The confusion matrix was normalised with respect to true labels (rows).



*P: Production

Figure 1.5 Resulting normalised confusion matrix in an imbalanced dataset

The confusion matrix shows that only 34% of the observations leading to a production stop were correctly detected, while the remaining 66% were misclassified as not severe issues. Further, it is possible to appreciate the high performance of the ML model when detecting the majority class, with 98% of such observations being correctly classified. However, this ML model would be unsuitable for real-world manufacturing scenarios and will rarely identify critical cases that can cripple production.

As maintenance data are naturally imbalanced (Usuga Cadavid et al., 2020b), class imbalance is of paramount importance in this thesis. The techniques to tackle its effects and their advantages and disadvantages will be further explored in the following chapters.

1.3. Proposed approach

The general objective of this work is to harness the maintenance data to effectively react to production disturbances. It originated from the interest expressed by an industrial partner, i.e. a software development company that produces a tool for operations scheduling, primarily on discrete manufacturing processes, and another tool that serves as a MES. The MES of the company collects several variables, including text descriptions provided by operators when they encounter issues. Therefore, the idea was to exploit the diverse data to assist planners in

rescheduling. The general objective was classified into three specific objectives that will be addressed using two strategies. The specific objectives of this study are as follows:

1) To identify research gaps, opportunities, and trends in the ML domain that can be applied to production planning and control.
2) To evaluate the technical feasibility of the models to address the previously identified research gaps. Specifically, to utilise the free-form text data originating from maintenance logs to achieve quick and accurate reactions to production disturbances.
3) To explore solutions to certain challenges encountered when exploiting real-world maintenance data from production. These challenges, which are identified with the first two objectives, are as follows: including humans in the loop, tackling class imbalance, and knowledge generation and interpretability.

The strategies are as follows:

1) Evaluation of the state-of-the-art methods through a literature review: A systematic literature review was performed at the beginning of the research process to assess the existing research on the utilisation of ML for production planning and control.
2) Evaluation of contributions through case studies: Several case studies were performed to technically evaluate the identified ML models to fill the research gaps identified in the literature review stage and identify other research perspectives that may motivate future work.

This research work is characterised by using real industrial datasets to identify the challenges encountered in real-world scenarios. Additionally, these datasets were obtained from different industries, allowing better generalisation of the conclusions.

1.4. Manuscript structure

The structure of this thesis encompasses nine chapters, which are organised as follows.

Chapter 1 introduces the research work by presenting the context, briefly explaining the key concepts, and providing an overview of the proposed approach.

Chapter 2 presents an update of the systematic literature review focusing on PdM for production planning and control. This update highlights further research gaps and trends in the recent research in this field.

Chapter 3 explains the approach adopted for this thesis and provides a clear view of how the different research contributions relate to addressing the objectives of this study.

Chapter 4 presents an article titled 'Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0', which was published in the *Journal of Intelligent Manufacturing*. This article presents the results of a systematic literature review conducted at the beginning of the research work to identify gaps, opportunities, and trends in the field of ML as applied to production planning and control.

Chapter 5 presents an article titled 'Valuing free-form text data from maintenance logs through transfer learning with CamemBERT', which was published in *Enterprise Information Systems*. This article evaluates the technical feasibility of using transformers to harness free-form text data from maintenance logs to support the decision-making process in production planning and control.

Chapter 6 presents an article titled 'Using Deep Learning to Value Free-Form Text Data for Predictive Maintenance', which was published in the *International Journal of Production Research*. This article extends the research discussed in Chapter 5 by exploring the usage of transformers in datasets from three different companies. Further, this paper explores the notion of interpretability and insight extraction from highly unstructured free-form text data.

Chapter 7 is an article titled 'Artificial Data Generation with Language Models for Imbalanced Classification in Maintenance', which was published in the Springer book series *Studies in Computational Intelligence*. This article explored an alternative data-driven approach to mitigate class imbalance when training ML models. The method focused on generating artificial observations using language models.

Chapter 8 discusses the findings of a study exploring an alternative algorithm-based approach to mitigate class imbalance when training ML models. The method consists of using a loss function called 'focal loss' (FL), which was initially applied in the field of computer vision. The communication was accepted and presented at the *17th IFAC Symposium on Information Control Problems in Manufacturing* (INCOM 2021).

Chapter 9 discusses the main results that were obtained, presents the limitations of the research work, concludes this document, and provides research perspectives.

In the **Appendix**, we explain the details regarding the criteria used for paper selection in the joint study conducted with A. Nguyen (Nguyen et al., 2021).

2. Chapter 2: Literature review

2.1. Motivation

Research related to ML models that are applied to PdM is developing rapidly and gaining interest over time. To illustrate this trend, a query was raised on Scopus on 14 May 2021 using the following string chain in *titles*, *abstracts*, and *keywords*: 'Predictive maintenance' AND ('Machine Learning' OR 'Deep Learning'). Figure 2.1 illustrates the results of this query by showing the number of publications according to the year.

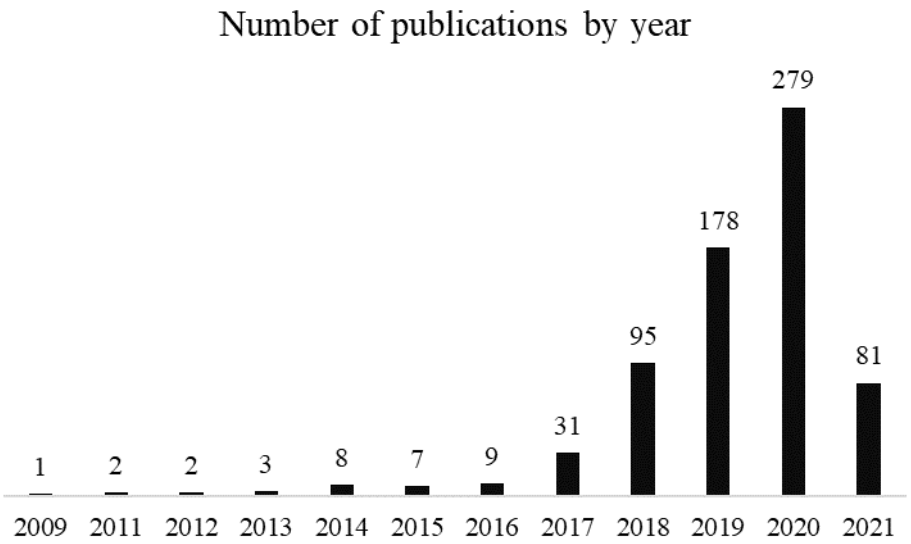


Figure 2.1 Number of publications by year on predictive maintenance using machine learning (ML) or deep learning

The figure shows that ML in PdM research has grown exponentially since 2017. As of May 2021, 81 papers were published on this topic. Because of this rapid evolution in the domain, it is necessary to update a previous systematic literature review (Usuga Cadavid et al., 2020a). Moreover, this previous study broadly focused on ML that is applied to production planning and control. Thus, this chapter updates the review and focuses on ML applied to PdM in the context of production planning and control. This update has two primary objectives: first, to understand the current context and trends of employing ML in PdM, and second, identifying research gaps, obstacles, and their solutions when creating ML models for PdM.

2.2. Methodology

To perform the literature review update, a systematic literature review was performed using a methodology based on the one proposed by Kitchenham et al. (2010). This methodology has

been previously used to derive knowledge from scientific literature (Dalzochio et al., 2020; Montero Jimenez et al., 2020). The protocol for conducting a systematic literature review consists of four phases: definition of research questions, search strategy, study selection, and data synthesis. Each phase is discussed in the following subsections.

2.2.1. Definition of research questions

To identify trends, challenges, and research gaps in recent literature, it is important to understand how researchers currently use ML in PdM and the obstacles they encounter with their respective solutions. Thus, the following two research questions (RQs) are proposed:

- RQ1: How is ML currently applied in PdM?
- RQ2: What are the challenges and their respective solutions when using ML in PdM?

RQ1 primarily serves to identify the current context, trends, and research gaps pertaining to PdM when applying ML models. RQ2 is also used to identify research gaps. However, its primary purpose is to identify challenges and solutions in the recent literature. These RQs allow the definition of specific questions (SQs) that will be answered when analysing the selected papers. The SQs related to RQ1 are:

- SQ1.1: Which industries are applying ML to perform PdM?
- SQ1.2: What are the use cases in PdM that were addressed by recent research?
- SQ1.3: What are the ML techniques used in the recent research on PdM?
- SQ1.4: What is the nature of the data sources used to perform PdM with ML?

The SQs related to the RQ2 are:

- SQ2.1: What are the challenges that are encountered when applying ML to PdM?
- SQ2.2: What are the solutions that are employed to tackle these challenges?

2.2.2. Search strategy

To capture recent research, the scientific database Scopus was queried on 23 April 2021 with the following string chain in *titles*, *abstracts*, and *keywords*: (‘Text mining’ OR ‘Natural Language Processing’ OR ‘Semantic Analysis’ OR ‘Topic Modelling’ OR ‘Machine Learning’ OR ‘Deep learning’) AND (‘Industry’ OR ‘Manufacturing’) AND (‘Maintenance’). The

motivation behind this string chain was to retrieve recent research using ML (or its subfield DL) when applied to industrial maintenance as well as applications related to NLP.

2.2.3. Study selection

After running the string chain, we applied the following restrictions (R) to focus on the recent research: First, only papers published in 2020 and 2021 were considered (R1), as the sample papers in our previous study stopped in 2019. Second, to capture the state of research in real-world applications, only case studies were considered (R2). Therefore, all reviews and surveys were discarded. Third, *titles* and *abstracts* were analysed to select papers related to manufacturing or production and predictive maintenance (R3). Finally, the shortlisted articles were thoroughly investigated. After a full-text analysis, each paper was graded by answering five questions:

1) Is the industry of the study mentioned?
2) Is the purpose of the research clear throughout the paper?
3) Are the ML techniques clearly presented?
4) Are the used data sources introduced?
5) Are there elements to identify the challenges encountered and their respective solutions?

Each question was answered with one of three possible answers: Yes = 1, Partially = 0.5, and No = 0. Only papers with a grade of at least four were considered in the final sample (R4). After the study selection phase, 19 articles constituted the final paper sample for the systematic literature review. Figure 2.2 summarises the search strategy and study selection phases.

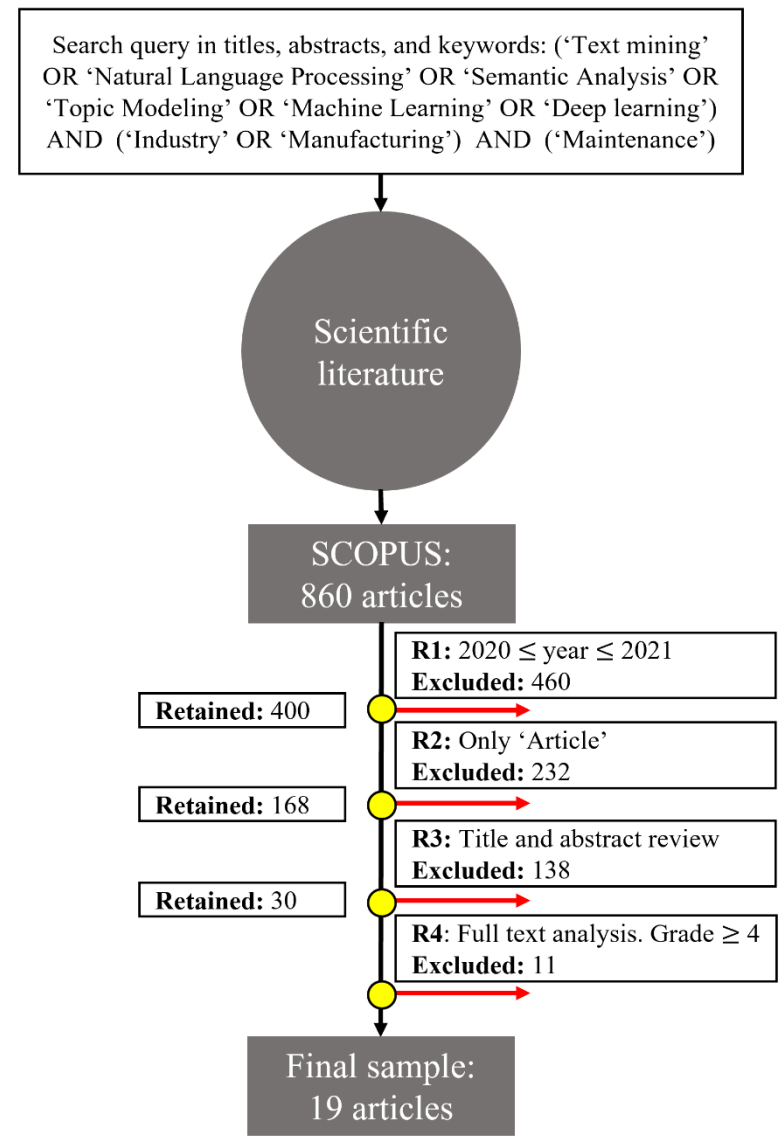


Figure 2.2 Summary of the search strategy and study selection phases

2.2.4. Data synthesis: the analytical framework

This subsection details the analytical framework used to answer the SQs by explaining how we responded to each question when performing the full-text analysis. For certain SQs, there appears to be no clear consensus in recent PdM literature reviews to provide a predefined framework (Razmi-Farooji et al., 2019; Zhang et al., 2019; Dalzochio et al., 2020; Montero Jimenez et al., 2020; Zonta et al., 2020). Therefore, for the industries (SQ1.1), use cases (SQ1.2), and solutions to the encountered challenges (SQ2.2), we propose our *ad hoc* classifications. The axes of the analytical framework used to address SQs are presented in the following subsections.

2.2.4.1. First axis of the analytical framework: industries and use cases

To analyse industries and use cases where PdM applications tend to focus, we measured these elements for each article. As there appears to be no clear consensus in the literature on a predefined set of use cases in PdM, we noted the use cases for each article and then grouped them to propose categories from our experience in the domain.

2.2.4.2. Second axis of the analytical framework: ML techniques

To measure ML techniques frequently used in PdM research, we collected the best-performing ML model from each article. In cases where the paper used several ML models at different stages of the proposed approach, all of these models were considered. Further, the learning type was recorded. ML models usually belong to one of the three following learning paradigms: supervised learning (SL), unsupervised learning (UL), and reinforcement learning (RL). According to Usuga Cadavid et al. (2020a), SL focuses on estimating a function $Y = f(X)$ by learning how to map inputs X to outputs Y. UL aims to uncover hidden patterns in data X. In this learning paradigm, there are no predefined target outputs Y. Finally, RL seeks to train an agent to learn an optimised policy of actions in a specific environment. For further information regarding the learning paradigms, please refer to (Jordan and Mitchell, 2015). Moreover, a detailed explanation of self-supervised learning and semi-supervised learning is beyond the scope of this thesis.

2.2.4.3. Third axis of the analytical framework: data sources

One of the critical enablers of ML models is the availability of data. Therefore, it is crucial to understand the sources of data that are generally employed by PdM researchers. Hence, we use the classification proposed by Razmi-Farooji et al. (2019), which consists of the following four data source categories:

1) Maintenance event data: These data are collected from the production planning or maintenance system. A typical example is maintenance logs (also referred to as maintenance information control) obtained in computerised maintenance management systems. Other examples of this data source are the technical data of equipment, technical drawings of machines, maintenance manuals, spare parts lists, equipment life plan showing the inspection dates for preventive maintenance, maintenance jobs catalogues, and operations and safety data.

2) Condition monitoring data: These are related to the data collected from sensors installed in the manufacturing system. They also encompass the data collected manually from smartphones or tablets. Examples of this data source are value-type data collected at a point in time, waveform-type data obtained from sensors measuring time series, and multi-dimensional data measuring several variables collected at a point in time.
3) Product data: These correspond to the data obtained from products, such as product specifications, bill of materials, failure modes.
4) Business data: These encompass data that are related to procurement, customer and supplier relationship management, and human resources. Typically, business data are used to perform maintenance planning by considering technician skills, availability, and absenteeism.

For each article in the sample, we use the categories above to identify and classify the data sources employed to perform PdM using ML. Finally, Razmi-Farooji et al. (2019) also suggest that maintenance data sources can be either objective or subjective. Objective data sources are invariant to human judgement and assumptions, while subjective data sources are subject to the knowledge of the operators or their comprehension of a particular situation. Examples of objective data sources are sensor systems, whereas subjective data sources are operator-written reports. The notion of objective or subjective sources was measured for each paper in the sample.

2.2.4.4. Fourth axis of the analytical framework: challenges and solutions

To understand the common challenges when performing PdM with ML, we decided to use the framework proposed by Razmi-Farooji et al. (2019). Six common data management challenges for maintenance were identified in their study through a literature review, interviews, and questionnaires completed by an industrial organisation. The proposed challenges are as follows.

1) Data acquisition: Choosing an adequate data acquisition method is vital when designing systems for PdM. It is an important decision that determines the design process of the system, how it works, and the incurred costs to develop and maintain it. This category encompasses challenges such as where to store the data, how to connect to the sensors, and how to identify the required data sources.

2) Rapid information flow: Developing effective systems in PdM may result in unfruitful efforts if stakeholders cannot access the correct information at the appropriate time. Additionally, the process of informing members may become complex when diverse groups of people are involved. This category addresses the challenge of keeping the stakeholders informed.
3) Data quality: Data obtained from real-world manufacturing systems are rarely ready to be used 'as provided'. They tend to contain different scales, errors, missing values, and outliers. For instance, subjective data sources tend to present several data quality issues when they are collected manually. Objective data sources, such as sensor data, also suffer from data quality, as faulty sensors or external noise can affect data acquisition (Razmi-Farooji et al., 2019). Thus, this category comprises challenges such as handling missing values, performing feature engineering to extract useful features, and mitigating the influence of class imbalance.
4) Voluminous and heterogeneous data: Data acquisition systems in maintenance can easily reach big data levels. They can present a high velocity of data generation, a variety of measured parameters and formats, and high volumes with large datasets which cannot be saved in standard personal computers owing to the lack of space. Therefore, this category includes challenges such as handling high-dimensional data and high data generation rates, enabling fast querying to retrieve data.
5) Data exchange and interoperability: Enabling data sharing among stakeholders and their utilisation across different information systems presents inter- and intra-organisational challenges. However, it is necessary to ensure the success of PdM systems. Hence, this category deals with challenges related to data accessibility for all stakeholders, data exchange between sensors and information systems, deployment of applications in production, etc.
6) Data conversion: Data obtained from different sources may be difficult to exploit owing to disparities in the data formats or software versions. Consequently, this category focuses on handling data from diverse formats in PdM.

For each paper, the challenges corresponding to the categories presented above were noted. Additionally, the solutions applied to tackle these obstacles were also recorded. Thus, this axis of the analytical framework shows challenge-solution pairs aiming to provide the common obstacles with their usual solution in recent PdM research.

2.3. Results

The following subsections explain the results obtained when answering the two RQs.

2.3.1. First research question (RQ): how is ML currently applied in PdM?

2.3.1.1. First axis of the analytical framework: industries and use cases

Figure 2.3 shows the results for industries (a) and use cases (b) identified through a systematic literature review.

While considering the industries in Figure 2.3 (a), it can be observed that the automotive sector has attracted the most attention from PdM applications in recent research. Industries related to electronics, such as semiconductors and the circuit board industry, have also been addressed in PdM. Apart from these trends, PdM applications appear to be applied in a diverse spectrum of industries, ranging from the pharmaceutical industry to the consumer goods industry. Such versatility is shown in the study performed by Chiu et al. (2020), where the authors propose a framework that can be applied to both the semiconductor and solar cell industries. Additionally, the results suggest that PdM can be used in manufacturing environments characterised by continuous production, such as in the oil and gas industry, or discretised production, such as in the automotive industry. Although various studies have reported the concerned industries, 31% of the studies (6 out of 19) did not mention the concerned sector.

While considering the use cases in Figure 2.3 (b), the most common usage of PdM in recent research was to perform fault detection. This use case refers to models used to predict the occurrence of a failure in the future or to characterise the possible failures that may occur given the characteristics of the system. Examples of this use case are the prediction of the severity of a fault in a particular machine (Kiangala and Wang, 2020; Usuga Cadavid et al., 2020b), determining whether a fault arises from a specific system, such as the air pressure system (Fathy et al., 2021), or triggering an alarm before a fault occurs, allowing shop-floor engineers to take appropriate measures in advance (Chiu et al., 2020).

The second most addressed use case was health state estimation, which predicts the state of equipment from measured data. This is the case of determining the degradation state in coilers for the stainless steel hot-rolling process (Ruiz-Sarmiento et al., 2020) or estimating the condition of specific components in an ultrasonic welding machine (Nazir and Shao, 2021).

RUL estimation was another common use case in the paper sample. This use case has been classically used to support PdM practices for replacing components according to their RUL and not based on a predefined schedule, as is usually performed in preventive maintenance. Thus, waste is minimised by replacing the parts when required. An example of RUL estimation was the study conducted by Ayvaz and Alpay (2021), where the authors employed data collected from an assembly line producing baby diapers (consumer goods industry) to estimate the remaining time before failure.

Other meaningful yet less popular use cases included the following: time-to-clean estimation, where data from heat exchangers were employed to raise the alarm when the tubes required cleaning (Soualhi et al., 2021), support for maintenance operations, where Ortega et al. (2021) helped in the diagnosis and repair of electronic boards through augmented reality and infrared cameras, and bottleneck maintenance diagnostic, where Subramaniyan et al. (2020) used maintenance logs to better understand the maintenance issues in bottleneck stations.

A final remark pertaining to the use cases is that they are not mutually exclusive. Certain studies addressed more than one use case to create more complete applications, as the data collected for a particular use case may be meaningful for application to another. For instance, Aliev and Antonelli (2021) employed data from collaborative robots to estimate the temperature at certain joints of the robot (prediction of machine working conditions) and predict whether safety stops are required (fault detection). Further, Usuga Cadavid et al. (2020b) harnessed machine symptom descriptions written by operators to predict whether the issue will lead to a stop in production (fault detection) and helped to determine the required workload to solve the problem (estimation of maintenance work duration).

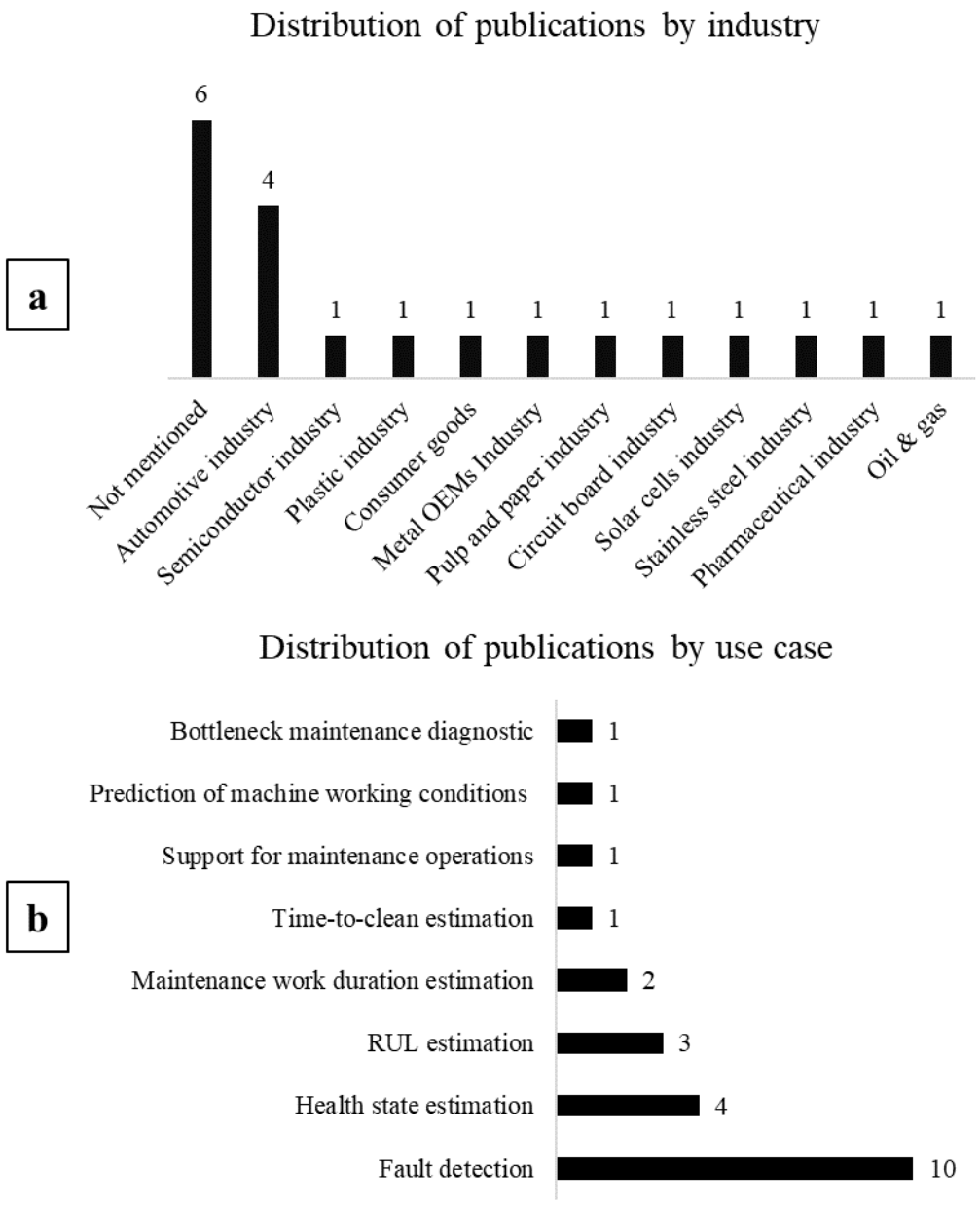


Figure 2.3 Results for the first axis of the analytical framework: industries (a) and use cases (b)

2.3.1.2. Second axis of the analytical framework: ML techniques

Figure 2.4 shows the results for this axis. It illustrates the most common ML techniques (a), grouping based on the category (b), and classification based on learning type (c).

While considering the ML techniques and their categories, Figure 2.4 (b) shows that the classic ML models are still extensively used in PdM, despite the hype around neural networks. This

category encompasses models such as support vector machines (SVM), linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), k-means, or principal component analysis (PCA). Although models in this category were common in recent PdM applications, Figure 2.4 (a) shows that PCA and k-means were the most commonly used classic ML techniques. These two models belong to the UL paradigm. They can detect patterns in the data to identify the underlying clusters or perform feature compression for dimensionality reduction. As several studies in the paper sample harnessed high-dimensional data from sensors, it was usual to observe synergies between SL and UL (SL-UL), which explains the high utilisation of the classic ML models. For example, PCA allows data compression in a low-dimensional space. Then, the compressed features are employed to train an ML model. k-means was also used in synergy with SL techniques, as in (Usuga Cadavid et al., 2020b), where the k-means pre-processes the output variable to discretise it and reduce the data imbalance.

The second most popular technique category was neural networks. In this category, convolutional neural networks (CNNs) and long short-term memory neural networks (LSTMs) were the most frequently applied techniques. Owing to their capacity in keeping track of past observations, LSTMs are models that are applicable when dealing with time series. This is suitable for PdM applications as they often tend to harness sensors to measure the evolution of variables over time. For instance, Soualhi et al. (2021) trained an LSTM on measuring variables from sensor data, such as temperature and pressure, to predict long-term fouling in heat exchanger tubes.

CNNs are commonly known to be helpful in applications that use image data. For instance, Ortega et al. (2021) used a pretrained CNN named YOLOv4 to perform object recognition and pose estimation to support the maintenance of electronic boards. However, CNNs can also handle time series, such as in (Lehmann et al., 2020), where CNNs employed time-series data collected by sensors to determine faulty machines. Another approach that uses CNNs for time series is to transform the data into images; for example, in (Kiangala and Wang, 2020), time series were transformed into images using Gramian angular fields. After this mathematical pre-processing, the generated images were used to train a CNN to perform fault detection.

Two papers used transfer learning for their applications, i.e., (Ortega et al., 2021) and (Usuga Cadavid et al., 2020b). The former used a pretrained CNN named YOLOv4 (Bochkovskiy et al., 2020), while the latter employed a pretrained transformer named CamemBERT for NLP

(Martin et al., 2019). Transfer learning is an interesting approach in which models are pretrained on massive amounts of data. By completing this training, the models learn rich representations of the data. Then, they can be fine-tuned to new tasks, achieving excellent performance in less time, with fewer data. Transfer learning may be an enabler for cases where data are scarce, which is a common situation in certain manufacturing contexts. For further information on transfer learning, please refer to (Pan and Yang, 2010).

Ensemble learning models were the least used when compared to the other two types. However, the random forest model, an ensemble learning model, was the most employed among all the other techniques. This may be because of its excellent performance when fitting complex non-linear relationships.

Figure 2.4 (c) shows that the most-used learning type was SL (58% of publications), although it is increasingly employed in synergy with UL to create more capable models (32% of publications). This result aligns with our previous systematic literature review, where SL and SL-UL were extensively applied in production planning and control (Usuga Cadavid et al., 2020a). Only two studies (~11%) employed UL with no other synergy, which were (Subramaniyan et al., 2020), where the authors harnessed k-means to understand maintenance data from bottleneck stations, and (Zhai et al., 2021), where conditional variational autoencoders were trained to learn probability distributions and perform health state estimation. Finally, it was surprising that no RL applications were found in PdM. Our previous study also showed a low number of studies using this learning paradigm in PdM, despite its extensive usage in other production planning and control domains (Usuga Cadavid et al., 2020a). This may be because the current focus of PdM is the prediction of well-defined outcomes, for which SL is well suited. Particularly, RL is primarily used in applications where optimised policies of actions must be learnt, and such use cases may not be adapted to the current needs or maturity of PdM.

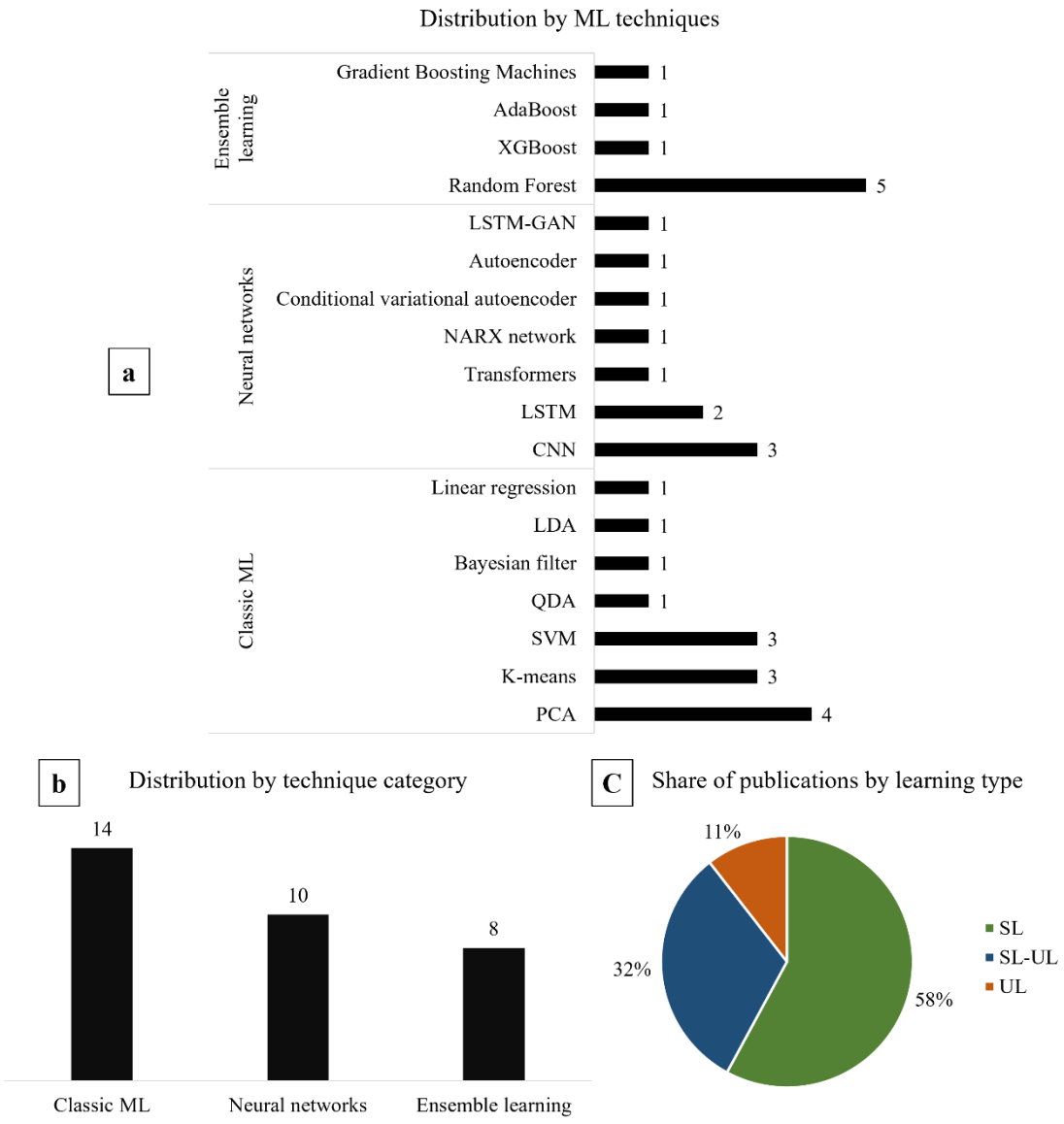


Figure 2.4 Results for the second axis of the analytical framework: ML techniques (a), technique categories (b), and learning types (c)

2.3.1.3. Third axis of the analytical framework: data sources

Figure 2.5 displays the frequency of usage of each data source (a) and detailed distribution of the data sources (b).

Figure 2.5 (a) shows that the most employed data source category in PdM with ML is condition monitoring data. Additionally, Figure 2.5 (b) indicates that, in this category, researchers primarily utilised waveform-type data, representing sensor data that measures time series. This result aligns with a previous systematic literature review by Montero Jimenez et al. (2020), who stated that most studies related to the diagnosis and prognosis of PdM employ time-series data.

Maintenance event data were the second most-used data source, and it was often employed along with condition monitoring data. Although creating systems that combine condition monitoring data with other sources such as maintenance event data may benefit manufacturing systems, not all companies can design data acquisition systems to collect sensor data. Thus, it is essential to study the potential of solely using data that has already been collected by the company, such as maintenance event data. This was the case for three studies in the sample papers (Subramaniyan et al., 2020; Usuga Cadavid et al., 2020b; Khalid et al., 2021), which validated that PdM applications can be created without relying on data collected through acquisition systems requiring sensors.

While considering the two remaining data sources, only one study used product data (Ruiz-Sarmiento et al., 2020), where the properties of steel plates were employed. No study in the list harnessed business data. Although this may imply that product and business data have limited applicability for the current scope of PdM, these data sources should not be ignored as they may provide potential benefits. For example, better scheduling of maintenance activities can be achieved by considering technician skills (business data) or more effective PdM models can be created by being aware of the characteristics of the highly customised products that are to be manufactured (product data).

While considering the objective and subjective data sources, only the following two studies employed subjective data: in (Usuga Cadavid et al., 2020b), authors used free-form text descriptions provided by operators to describe machine symptoms, and in (Ortega et al., 2021), authors utilised, among other data sources, images captured with cameras connected to a portable computer. In both cases, the difficulty of harnessing subjective data sources arises from the influence of humans on the inputs when performing data acquisition; for example, what can be judged as informative by a particular operator may not be significant for an algorithm trying to understand the underlying pattern. To tackle this challenge, these studies relied on dedicated techniques that can handle such subjectivity. Usuga Cadavid et al. (2020b) employed transformers for NLP, which are more robust to unknown words owing to pretrained tokenisation strategies and their capability to generate contextualised word embeddings. Ortega et al. (2021) designed a dedicated augmented reality system relying on the pretrained CNN YOLOv4 and other state-of-the-art methods such as LINEMOD for 3D object detection.

Future research must focus on effectively including subjective data sources, as they are generally easy to collect in production environments that rely on human operations. Additionally, creating systems that exploit subjective data sources may help increase the acceptance of new systems by operators, as these systems can better integrate into the manner in which humans work. For instance, instead of imposing predefined forms for reporting machine symptoms that can be cumbersome to fill, a system can simply use the free-form text provided by technicians.

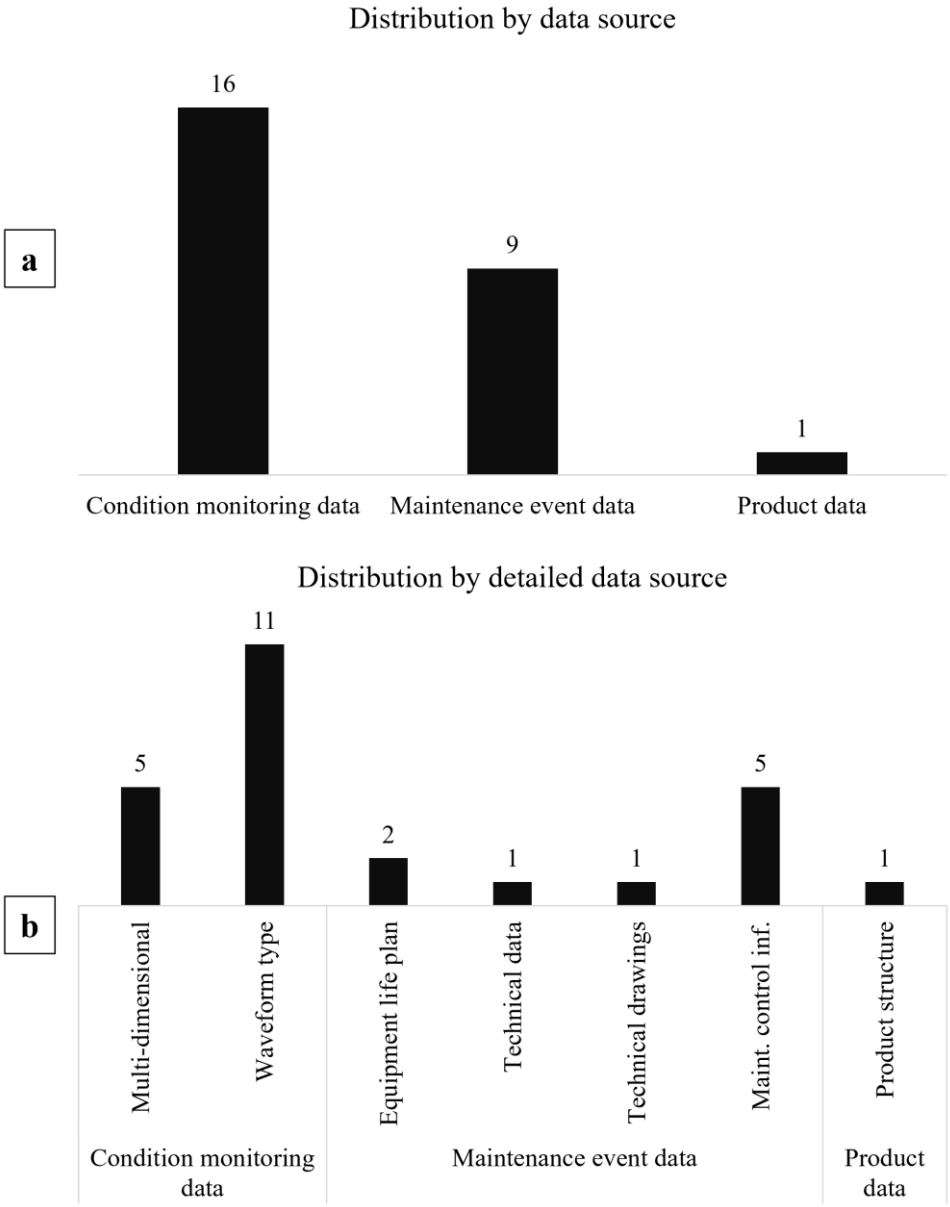


Figure 2.5 Results for the third axis of the analytical framework: distribution of publications by data sources (a) and by detailed data sources (b)

To summarise the findings of each axis of the analytical framework related to the first RQ, Table 2.1 lists the primary results for each article in the paper sample. The two studies that used subjective data sources are highlighted in red.

Reference	Industry	Use Case	Best Technique	Learning Type	Data Source
(Ayvaz and Alpay, 2021)	Consumer goods	RUL estimation	*Random forest *PCA	SL-UL	Condition monitoring data
(Aliev and Antonelli, 2021)	Not mentioned	*Prediction of machine working conditions *Fault detection	*Linear regression *Gradient Boosting Machines	SL	Condition monitoring data
(Nazir and Shao, 2021)	Not mentioned	Health state estimation	QDA, LDA, and SVM provided similar results	SL	Condition monitoring data
(Soualhi et al., 2021)	Pulp and paper industry	Time-to-clean estimation	*LSTM *Nonlinear auto-regressive exogenous model *Autoencoder	SL-UL	Condition monitoring data
(Zhai et al., 2021)	Automotive industry	Health state estimation	*Conditional variational autoencoder *K-Means	UL	*Condition monitoring data *Maintenance event data
(Fathy et al., 2021)	Automotive industry	Fault detection	*XGBoost *PCA	SL-UL	Condition monitoring data
(Liu et al., 2021)	Not mentioned	*Health state estimation *Fault detection	LSTM-GAN	SL	Condition monitoring data
(Ortega et al., 2021)	Circuit board industry	Support for maintenance operations	CNN (YOLOv4)	SL	*Condition monitoring data *Maintenance event data
(Lehmann et al., 2020)	Metal processing OEMs industry	Fault detection	CNN	SL	Condition monitoring data
(Subramaniyan et al., 2020)	Automotive industry	Bottleneck maintenance diagnostic	K-means	UL	Maintenance event data
(Borith et al., 2020)	Automotive industry	Fault detection	SVM	SL	Condition monitoring data

Reference	Industry	Use Case	Best Technique	Learning Type	Data Source
(Quatrini et al., 2020)	Pharmaceutical industry	Fault detection	Random forest	SL	Condition monitoring data
(Panicucci et al., 2020)	Not mentioned	RUL estimation	Random forest	SL	Condition monitoring data
(Usuga Cadavid et al., 2020b)	Not mentioned	*Fault detection *Maintenance work duration estimation	*Transformers (CamemBERT) *K-means	SL-UL	Maintenance event data
(Kiangala and Wang, 2020)	Not mentioned	Fault detection	*CNN *PCA	SL-UL	Condition monitoring data
(Ruiz-Sarmiento et al., 2020)	Stainless steel industry	Health state estimation	Bayesian filter	SL	*Condition monitoring data *Product data *Maintenance event data
(Chiu et al., 2020)	*Semiconductor industry *Solar cells industry	*Fault detection *RUL estimation	*Random forest *LSTM	SL	Condition monitoring data
(Khalid et al., 2021)	Oil & gas	Maintenance work duration estimation	*AdaBoost *Random forest *PCA	SL-UL	Maintenance event data
(Acernese et al., 2020)	Plastic industry	Fault detection	SVM	SL	Condition monitoring data

Table 2.1 Summary for the first research question

2.3.2. Second RQ: What are the challenges and their respective solutions when using ML in PdM?

Table 2.2 lists the 'main challenges' addressed by each reference. These main challenges were drawn from the study conducted by Razmi-Farooji et al. (2019). The table also indicates the total number of papers addressing each main challenge. To better explain the common obstacles encountered in each main challenge, the following subsections describe certain recurrent interrogations that researchers must deal with when tackling the main challenges. We identified, formalised, and proposed these common interrogations through the full-text analysis of the papers, and they will be referred to as 'detailed challenges'. Although they do not represent an exhaustive list of the detailed challenges encountered in real-world applications, this provides

more detail on the state of each main challenge in recent research. Finally, for the sake of analysis, the three papers that do not use condition monitoring data are highlighted in red in the following subsections. These studies may provide interesting insights, as they are the only studies in the sample that do not rely on data collected through sensors.

Table 2.2 shows that most of the studies in the sample encountered obstacles related to data quality as well as voluminous and heterogeneous data. This is an anticipated finding as these two challenges are inherent to real-world applications of ML models. Data in manufacturing tend to be unsuitable for exploitation in its raw form. Moreover, datasets contain several variables obtained from disparate sources. However, not all the variables may be relevant to the model and including them all would increase the complexity of the model.

Data acquisition, rapid information flow, and data exchange and interoperability were also frequently addressed. In the case of data acquisition as well as data exchange and interoperability, this may be owing to the high number of papers using condition monitoring data obtained from sensors. Moreover, it is interesting to highlight that the three articles that did not use data from sensors did not tackle these two challenges (Subramaniyan et al., 2020; Usuga Cadavid et al., 2020b; Khalid et al., 2021). This suggests that using previously collected data from information systems, such as maintenance event data, may reduce the burden of designing data acquisition systems and integrating the results into the existing infrastructure of the company. While considering the rapid information flow, several studies proposed methods to keep the stakeholders informed, suggesting that the maturity of PdM in I4.0 is increasing, as it is now generating exploitable applications to cover business requirements.

Only three papers studied challenges related to data conversion resulting from disparities in data formats or software versions, leading to potential errors when creating PdM systems. Although it is an important topic to consider when designing systems, it has been frequently ignored in recent research. This may be because data conversion issues tend to be expected in larger industrial environments, where a high number of computers and sensors lead to data conversion obstacles. In research scenarios with a smaller scale, this issue may be diminished. Nonetheless, we recognise that data conversion issues should be further explored in PdM research through case studies dealing with large-scale systems, as real-world manufacturing scenarios will probably encounter these obstacles.

Reference	Data acquisition	Rapid information flow	Data quality	Voluminous heterogeneous data	Data exchange and interoperability	Data conversion
(Ayvaz and Alpay, 2021)	X	X	X	X	X	X
(Aliev and Antonelli, 2021)	X	X		X	X	
(Nazir and Shao, 2021)	X		X	X		
(Soualhi et al., 2021)			X	X		
(Zhai et al., 2021)			X	X		
(Fathy et al., 2021)			X	X		
(Liu et al., 2021)		X	X		X	
(Ortega et al., 2021)	X	X	X	X		
(Lehmann et al., 2020)	X			X	X	X
(Subramaniyan et al., 2020)		X	X	X		
(Borith et al., 2020)	X		X			
(Quatrini et al., 2020)			X	X		

Reference	Data acquisition	Rapid information flow	Data quality	Voluminous heterogeneous data	Data exchange and interoperability	Data conversion
(Panicucci et al., 2020)	X	X	X	X	X	X
(Usuga Cadavid et al., 2020b)			X			
(Kiangala and Wang, 2020)			X	X		
(Ruiz-Sarmiento et al., 2020)	X	X		X	X	
(Chiu et al., 2020)	X	X		X	X	
(Khalid et al., 2021)			X	X		
(Acernese et al., 2020)	X		X	X	X	
Total	**10**	**8**	**15**	**16**	**8**	**3**

Table 2.2 Main challenges addressed by each paper in the sample

2.3.2.1. Fourth axis of the analytical framework: challenges and solutions in data acquisition

Table 2.3 summarises all the proposed detailed challenges in data acquisition with their respective solutions for each paper. Detailed challenges are presented as questions, and the solutions to tackle them as applied in each study are provided. Additionally, Table 2.4 provides an overview of the particulars of the detailed challenge addressed by each publication, thereby allowing a better comprehension of the types of obstacles that are commonly addressed in the literature.

Reference	Challenges and solutions for data acquisition
(Ayvaz and Alpay, 2021)	*Where can the data be stored? Cloud database
(Aliev and Antonelli, 2021)	*Where can the data be stored? Database in a server *How can we connect to the sensors? WiFi
(Nazir and Shao, 2021)	*How can the data sources be identified? Domain knowledge
(Ortega et al., 2021)	*How can the data acquisition system be prepared? Sensor calibration
(Lehmann et al., 2020)	*How can we connect to the sensors? Gateway *Where can the data be stored? Cloud data lake
(Borith et al., 2020)	*Where can the data be stored? Local database
(Panicucci et al., 2020)	*How can data acquisition be managed? Edge gateway
(Ruiz-Sarmiento et al., 2020)	*Where can the data be stored? Centralised server
(Chiu et al., 2020)	*Where can the data be stored? Cloud database *How can data acquisition be managed? Programmable Logic Controller (PLC) transceiver *How can we connect to the sensors? WiFi
(Acernese et al., 2020)	*How can data acquisition be managed? PLC controller *Where can the data be stored? Personal computer *How can the data acquisition system be validated? In situ tests on machines

Table 2.3 Different papers and the detailed challenges and solutions related to data acquisition that are addressed in each paper

Reference	Where can the data be stored?	How can we connect to the sensors?	How can the data sources be identified?	How can the data acquisition system be prepared?	How can data acquisition be managed?	How can the data acquisition system be validated?
(Ayvaz and Alpay, 2021)	X					
(Aliev and Antonelli, 2021)	X	X				
(Nazir and Shao, 2021)			X			
(Ortega et al., 2021)				X		
(Lehmann et al., 2020)	X	X				
(Borith et al., 2020)	X					
(Panicucci et al., 2020)					X	
(Ruiz-Sarmiento et al., 2020)	X					
(Chiu et al., 2020)	X	X			X	
(Acernese et al., 2020)	X				X	X
Total	**7**	**3**	**1**	**1**	**3**	**1**

Table 2.4 Summary of detailed challenges addressed by each paper for data acquisition

From Table 2.4, we propose the following six detailed challenges regarding data acquisition:

1) Where can the data be stored? Deciding where to store the data in PdM systems is vital to ensure the success of the proposed solution. Data should be stored in a place capable of handling the large volumes of generated data and can achieve secure access when the data are required.
2) How can we connect to the sensors? Connecting the sensors to the data acquisition system may be a non-trivial issue, as this connection should be reliable for continuously monitoring the variables of interest.
3) How can the data sources be identified? Manufacturing systems may present many parameters and measuring them may result in prohibitive costs for the data acquisition system. Therefore, it is vital to target suitable data sources in advance.
4) How can the data acquisition system be prepared? Data acquisition systems should be calibrated to ensure that the variables of interest are accurately measured. This calibration often requires specialised knowledge that may be rare.
5) How can data acquisition be managed? Integrating the systems that measure several parameters at different frequencies requires specialised methods or devices to manage data acquisition in an organised manner.
6) How can the data acquisition system be validated? The data acquisition system should be tested in real environments before deploying them in production to avoid issues arising from unforeseen scenarios.

Interestingly, all the studies presented in this subsection employed condition monitoring data from the sensors. In other words, the three papers that did not use sensor data did not encounter challenges related to data acquisition, probably because of the exploitation of historical data, which had already been collected through time.

The most frequently encountered detailed challenge was data storage, with seven papers exploring solutions for this obstacle. It was typically solved through cloud databases or cloud data lakes, which shows that cloud technologies can offer valuable advantages for PdM. However, certain studies have employed more straightforward solutions, such as local databases in personal computers. This may indicate that PdM applications do not require cloud technologies to store the data for successfully delivering results. However, large-scale applications may require cloud services owing to their flexibility.

Connecting to sensors and managing data acquisition were challenges that were each addressed in three studies. The former was tackled through WiFi and gateways, while the latter was performed using PLC. Studies that applied these solutions employed condition monitoring data from sensors, which exemplifies why it may be challenging to harness this data source, which is that it often requires multidisciplinary knowledge in networks, control, and telecommunications to ensure proper system implementation and functioning.

Finally, results regarding data acquisition suggest that using sensor data raises other questions, such as identifying the useful data sources to define the monitoring system, preparing it once installed, and validating it to ensure its proper functioning. These challenges were solved using expert knowledge, specialised calibration methods, and in situ tests on machines, respectively, which may be difficult to obtain owing to the limited time from experts, scarce skills in human resources, or the impossibility of stopping a machine in production, respectively.

2.3.2.2. Fourth axis of the analytical framework: challenges and solutions in rapid information flow

Table 2.5 summarises the proposed detailed challenges in rapid information flow with their respective solutions for each paper. Detailed challenges are presented as questions, and the solutions to tackle them as applied in each study are provided. Additionally, Table 2.6 provides an overview of the particulars of the detailed challenges tackled by each publication, thereby allowing a better comprehension of the different obstacles that are commonly addressed in the literature.

Reference	**Challenges and solutions for rapid information flow**
(Ayvaz and Alpay, 2021)	*How can the stakeholders be kept informed? Dedicated application *How can the solution be created? Flask Application Programming Interfaces (API) to develop a web service
(Aliev and Antonelli, 2021)	*How can the stakeholders be kept informed? Graphical user interface *How can the solution be created? Node-RED
(Liu et al., 2021)	*How can the stakeholders be kept informed? Monitoring interface, integration with a digital twin, integration with production scheduling
(Ortega et al., 2021)	*How can the stakeholders be kept informed? Graphical user interface, dedicated application

Reference	Challenges and solutions for rapid information flow
(Subramaniyan et al., 2020)	*How can the stakeholders be kept informed? Data visualisation
(Panicucci et al., 2020)	*How can the stakeholders be kept informed? Integration with production scheduling, dedicated application, data visualisation, *How can the solution be created? Unity platform
(Ruiz-Sarmiento et al., 2020)	*How can the stakeholders be kept informed? Data visualisation
(Chiu et al., 2020)	*How can the stakeholders be kept informed? Integration with automatic control of machines

Table 2.5 Different papers and the detailed challenges and solutions related to rapid information flow that are addressed in each paper

Reference	How can the stakeholders be kept informed?	How can the solution be created?
(Ayvaz and Alpay, 2021)	X	X
(Aliev and Antonelli, 2021)	X	X
(Liu et al., 2021)	X	
(Ortega et al., 2021)	X	
(Subramaniyan et al., 2020)	X	
(Panicucci et al., 2020)	X	X
(Ruiz-Sarmiento et al., 2020)	X	
(Chiu et al., 2020)	X	
Total	**8**	**3**

Table 2.6 Summary of detailed challenges addressed by each paper for rapid information flow

Based on Table 2.6, we propose two detailed challenges regarding rapid information flow:

1) How can the stakeholders be kept informed? PdM applications create a rapid flow of information between the concerned stakeholders. However, applications in I4.0 should avoid creating isolated islands of information, as this may hinder decision making (Razmi-Farooji et al., 2019). Hence, special efforts must be expended to keep the stakeholders informed.

2) How can the solution be created? Choosing the right methods or tools to develop a solution that will keep stakeholders informed is a non-trivial task. It must be adapted both technically and organisationally to the context of the company where it is applied. Hence, it remains a common obstacle when creating PdM systems.

From Table 2.6, it can be noted that all the papers exploring obstacles related to rapid information flow focused on the detailed challenge of keeping the stakeholders informed. It is a vital part of PdM applications with ML models as it provides actionable insights for manufacturing. Recent research has often developed dedicated applications, graphical user interfaces, monitoring interfaces, and data visualisation dashboards to keep the stakeholders informed. Additionally, three studies integrated the proposed PdM application with other functions on the shop floor, such as production scheduling (Panicucci et al., 2020; Liu et al., 2021) and automatic machine control (Chiu et al., 2020).

Integrating PdM solutions with other functions in a company is vital to align the stakeholders and achieve a faster decision-making process. For example, by employing predictions from the PdM system, which can indicate when a particular machine will require maintenance, production rescheduling can be performed to avoid losses in production. Moreover, let us assume a case where the PdM model detects that the operating parameters of specific equipment will contribute to faster degradation. In such a case, it can send instructions to the control system to adapt the functioning of the machine and extend its useful life. Finally, Liu et al. (2021) proposed the integration with existing digital twins, which may provide diverse advantages as digital twins keep track of the state of the shop floor, allowing more informed decision making. In future research, integration with other functions in the company should be explored further.

While considering the creation of the proposed solution, only three papers reported the utilisation of a tool to develop solutions that can keep the stakeholders informed. These tools primarily consisted of platforms or APIs such as Unity and Node-RED or the Flask API to develop web applications. However, this challenge seems less relevant for future research, although we believe it is important from an industrial perspective.

2.3.2.3. Fourth axis of the analytical framework: challenges and solutions in data quality

Table 2.7 summarises the proposed detailed challenges in data quality with their respective solutions for each paper. Detailed challenges are presented as questions, and the solutions to

tackle them as applied in each study are provided. Additionally, Table 2.8 provides an overview of the particulars of the detailed challenges that were tackled by each publication, allowing a better comprehension of the different obstacles that are commonly addressed in the literature.

From Table 2.8, we propose the following seven detailed challenges concerning data quality:

1) How can the missing values be handled? Missing values are a common issue in real-world datasets. For instance, in manufacturing, they can arise from causes such as failures in sensors, operators not filling fields in electronic forms, parameters that are only measured in specific situations. Handling missing values is a challenge, as leaving them in the data used to train ML models can hurt the learning process.
2) How can different variable scales be handled? Data from multiple sources measuring many variables will probably have different scales. For example, a particular sensor that measures the temperature in Celsius and another sensor that gauges the revolutions per minute of a shaft in a high-speed engine will undoubtedly provide data in different scales. Using these data without pre-processing in certain ML models such as k-means, SVM, or PCA would lead to misleading results, as the dimensions with a lower scale may be given less importance.
3) How can the class imbalance be handled? Class imbalance is a typical issue in several domains, where certain events tend to be over-represented. Therefore, ML models are prone to recognising these frequent classes while ignoring the less regular ones. This can be harmful in maintenance, as rare situations may lead to severe problems.
4) How can useful features be extracted? Modern manufacturing systems measure several parameters simultaneously, resulting in datasets with a multitude of variables. Using all of them would lead to statistical problems, such as the curse of dimensionality and overfitting or too complex models that hinder the interpretation of results. Therefore, using methods to perform feature engineering or feature extraction to identify a more relevant set of variables may be essential when designing PdM systems.
5) How can noise be handled in the data? External factors when measuring variables or defects in the measuring system can lead to noise in the data. Such noise should be treated to ensure that the learning process of ML models accurately identifies the underlying patterns.

6) How can the lack of labelled data be tackled? Labelled data that indicates the expected outputs for a particular set of inputs are scarce in manufacturing when compared to unlabelled data. Additionally, efforts to produce labelled datasets can be expensive, time consuming, and unreliable if not performed with the support of experts. Even if manufacturing data are usually unlabelled, they still contain valuable patterns that can be exploited to improve the production system (Usuga Cadavid et al., 2020a). Therefore, identifying ways to harness unlabelled data is critical for extracting the value of the high data volumes produced by modern companies.
7) How can the outliers be handled? Outliers are values that significantly differ from the other data points in the distribution. Even if they may represent correctly measured observations, their existence can hinder the proper learning of ML algorithms. Thus, identifying these values, processing them, or using robust methods for outliers is essential to create reliable ML models.

From Table 2.8, it can be noted that the most frequent challenge encountered in recent research regarding data quality was extracting useful features from the data, with more than half of the studies proposing solutions. This challenge was identified in papers exploiting both sensor and non-sensor data, suggesting that it exists in all contexts. To address this obstacle, a wide range of solutions was applied. From these, the following four groups of solutions were identified:

1) ML techniques: For example, autoencoders were used to create compressed and continuous latent spaces from high-dimensional inputs or k-means was implemented to identify groups that can serve as input categories or output labels.
2) Expert knowledge: Numerous studies followed the advice of previously published research or of industry specialists to create new meaningful variables, while other studies employed theoretical considerations to calculate them. For example, Soualhi et al. (2021) used the heat transfer rate to estimate a health indicator for heat exchangers.
3) Statistical treatment: Descriptive statistics were used to summarise the population by estimating statistics such as the mean and standard deviation. Further, simple pre-processing techniques, such as one-hot encoding to handle categorical variables, are observed in this category.

4) Mathematical treatment and analysis: Mathematical functions were used to derive richer representations of the data. For instance, Kiangala and Wang (2020) employed the Gramian angular field to encode time series as images. Then, these images were used to train a CNN.

Although we propose four groups of solutions, Table 2.7 shows that one single group of solutions may not be sufficient for extracting meaningful features independently. In fact, they are often combined to generate better variables. As obtaining features that can improve the learning of ML models is crucial for enhancing the harnessing of existing data, we expect future research to focus on this topic as it is a challenge where solutions can be obtained from multidisciplinary fields. Hence, there are numerous possibilities.

Other frequently addressed challenges were handling missing values and managing different data scales. For missing values, solutions commonly employed imputation, which consists of filling the missing values with predefined rules, such as replacing the missing value by the mean, mode or estimating by utilising other variables in the dataset. However, several studies decided to discard observations containing missing data. Although this is a straightforward solution, it may reduce the dataset size, depriving the model of valuable information. However, removing missing values may be justified in certain scenarios, such as in the study by Subramaniyan et al. (2020), where the authors conducted discussions with maintenance engineers (expert knowledge) to determine whether missing values could be safely removed. While considering the management of different data scales, recent literature appears to converge towards common scaling techniques such as normalisation and standardisation.

While considering class imbalance, only five studies aimed to address this issue. Some of the solutions to this obstacle were removing the data belonging to rare types of failures, generating artificial data, class weighting and cost-sensitive learning, and data pre-processing with k-means to equilibrate the outputs. It is interesting to note that there are only a few papers discussing this topic. Indeed, as maintenance datasets tend to be inherently imbalanced (Usuga Cadavid et al., 2020b), it was expected that richer literature regarding class imbalance in maintenance would be available. However, most studies focused exclusively on the most frequent failure types, thus removing the class imbalance by ignoring the minority classes. Nevertheless, this topic should be further explored in future research, as the prediction of rare failures is essential for increasing the maturity of PdM applications in I4.0.

Unlabelled data in manufacturing are typical. However, only four papers explored solutions for this issue. Such low interest on this topic may be because most recent research is still focused on exploiting data from labelled datasets. However, as PdM grows in maturity, research aiming at tackling unlabelled data is expected to attract more interest in the following years. There were diverse solutions to tackle this obstacle, including generating artificial images with BlenderProc (Ortega et al., 2021), using expert knowledge to define rules for labelling (Acernese et al., 2020; Quatrini et al., 2020), and implementing unsupervised ML models with expert knowledge and statistical analysis to learn probability distributions (Zhai et al., 2021). Future research should focus on new ways to generate labelled data with less reliance on experts, as they often have limited time to dedicate to ML projects. Additionally, PdM applications employing condition monitoring data from sensors may encounter high volumes of unlabelled data. Hence, further work should focus on automating the recognition of labels in the sensor data.

Finally, the following two challenges were rarely encountered in recent literature: handling noise and outliers in the data. To control noise, studies commonly employed statistical analysis, mathematical pre-processing of data, and ML models, such as autoencoders. To handle outliers, techniques were less sophisticated and primarily focused on removing outliers that were previously identified by experts or using statistical methods. Hence, future research should focus on strategies that are robust to outliers to avoid losing information by performing data removal.

Reference	Challenges and solutions for data quality
(Ayvaz and Alpay, 2021)	*How can the missing values be handled? Imputation by the median *How can different variable scales be handled? Common scaling techniques (standardisation) *How can the class imbalance be handled? Data removal
(Nazir and Shao, 2021)	*How can useful features be extracted? Statistical treatment, expert knowledge
(Soualhi et al., 2021)	*How can the missing values be handled? Imputation through non-linear interpolation *How can noise be handled in the data? ML (autoencoder) *How can useful features be extracted? Expert knowledge
(Zhai et al., 2021)	*How can the lack of labelled data be tackled? Expert knowledge, data visualisation and statistical analysis, ML (conditional variational autoencoder) *How can different variable scales be handled? Common scaling techniques (standardisation) *How can useful features be extracted? ML models (k-means and conditional variational autoencoder) *How can the missing values be handled? Imputation by forward-fill imputation or by interpolation
(Fathy et al., 2021)	*How can the missing values be handled? Imputation by the mean *How can the class imbalance be handled? Artificial data generation, cost-sensitive learning
(Liu et al., 2021)	*How can noise be handled in the data? Mathematical treatment and analysis
(Ortega et al., 2021)	*How can the lack of labelled data be tackled? Artificial data generation
(Subramaniyan et al., 2020)	*How can the missing values be handled? Expert knowledge, data removal *How can the outliers be handled? Expert knowledge, data removal *How can useful features be extracted? Expert knowledge, Statistical treatment (one-hot encoding) *How can different variable scales be handled? Common scaling techniques (standardisation)
(Borith et al., 2020)	*How can useful features be extracted? Statistical treatment, expert knowledge *How can different variable scales be handled? Common scaling techniques (Min-Max normalisation)

Reference	Challenges and solutions for data quality
(Quatrini et al., 2020)	*How can the class imbalance be handled? Add a new feature to the dataset *How can useful features be extracted? ML *How can the lack of labelled data be tackled? Expert knowledge
(Panicucci et al., 2020)	*How can useful features be extracted? Statistical treatment (computation of sample statistics)
(Usuga Cadavid et al., 2020b)	*How can the missing values be handled? Data removal *How can useful features be extracted? Expert knowledge *How can the class imbalance be handled? Random over-sampling, pre-processing with ML (k-means and silhouette diagrams)
(Kiangala and Wang, 2020)	*How can useful features be extracted? Mathematical treatment and analysis *How can different variable scales be handled? Common scaling techniques (Min-Max normalisation)
(Khalid et al., 2021)	*How can useful features be extracted? Statistical treatment: one-hot encoding *How can the missing values be handled? Data removal *How can the outliers be handled? Data removal *How can different variable scales be handled? Common scaling techniques (normalisation)
(Acernese et al., 2020)	*How can the lack of labelled data be tackled? Expert knowledge and manual labelling *How can the outliers be handled? Data removal with statistical methods *How can the missing values be handled? Data removal *How can noise be handled in the data? Statistical analysis and processing *How can different variable scales be handled? Common scaling techniques (standardisation) *How can useful features be extracted? Mathematical treatment and analysis, statistical treatment *How can the class imbalance be handled? Class weighting

Table 2.7 Different papers and the detailed challenges and solutions related to data quality addressed in each paper

Reference	How can the missing values be handled?	How can different variable scales be handled?	How can the class imbalance be handled?	How can useful features be extracted?	How can noise be handled in the data?	How can the lack of labelled data be tackled?	How can the outliers be handled?
(Ayvaz and Alpay, 2021)	X	X	X				
(Nazir and Shao, 2021)				X			
(Soualhi et al., 2021)	X			X	X		
(Zhai et al., 2021)	X	X		X		X	
(Fathy et al., 2021)	X		X				
(Liu et al., 2021)					X		
(Ortega et al., 2021)						X	
(Subramaniyan et al., 2020)	X	X		X			X
(Borith et al., 2020)		X		X			
(Quatrini et al., 2020)			X	X		X	
(Panicucci et al., 2020)				X			
(Usuga Cadavid et al., 2020b)	X		X	X			
(Kiangala and Wang, 2020)		X		X			
(Khalid et al., 2021)	X	X		X			X
(Acernese et al., 2020)	X	X	X	X	X	X	X
Total	**8**	**7**	**5**	**11**	**3**	**4**	**3**

Table 2.8 Summary of detailed challenges addressed by each paper for data quality

2.3.2.4. Fourth axis of the analytical framework: Challenges and solutions for voluminous and heterogeneous data

Table 2.9 summarises the proposed detailed challenges for voluminous and heterogeneous data with their respective solutions for each paper. Detailed challenges are presented as questions, and the solutions to tackle them as applied in each study are provided. Additionally, Table 2.10 provides an overview of the particulars related to the detailed challenges addressed by each publication, allowing a better comprehension of the different obstacles that are commonly addressed in the literature.

Based on Table 2.10, we propose the following four detailed challenges related to voluminous and heterogeneous data:

1) How can high-dimensional data be handled? Data collected from multiple sources may present high levels of heterogeneity, which can affect the performance of ML models. Moreover, certain variables may be redundant in the model. Hence, the use of methods to determine the most relevant variables to be used or compressing them into lower-dimensional spaces may improve the performance of PdM models.
2) How can high data generation rates be handled? One of the characteristics of big data is the velocity at which the data are produced. Such high generation rates may overwhelm the applications of PdM, such as when the prediction time of the system is significantly higher than the data generation rate. In this scenario, such an issue hinders the deployment of the system for real-time predictions.
3) How can rapid data visualisation be enabled? When dashboards are required to display large amounts of data, it may be challenging to ensure that graphs are visible and updated in a reasonable time. Therefore, for cases using big data, enabling rapid data visualisation is essential when reporting the results to users.
4) How can fast data querying be enabled? When handling voluminous data, strategies must be deployed to obtain rapid results when querying databases. Otherwise, solutions depending on the results of such queries may be unfeasible in real-world environments.

To date, the most addressed detailed challenge was the handling of high-dimensional data in both types of studies, i.e. those using sensor and non-sensor data. It was often tackled through a mix of expert knowledge, statistical analysis, and ML models. For expert knowledge, using

previous studies or advice from specialists was a commonly used strategy. The typical statistical analysis techniques were correlation analysis, data visualisation, and rules using descriptive statistics, such as standard deviation, to identify the most relevant variables. Finally, the common ML models used were random forest or AdaBoost to determine variable importance, PCA or autoencoders to compress high-dimensional inputs into a low-dimensional space, and linear regression to perform bivariate analysis.

One study employed specialised techniques for data fusion from sensors to merge high-dimensional data (Ortega et al., 2021). In this study, the authors used sensor registration to perform a one-to-one pixel correspondence between images captured using thermal and RGB cameras. This highlights that employing the condition monitoring data obtained from sensors not only poses challenges related to data acquisition when designing and maintaining the monitoring system, but also related to the data preparation, which is required subsequently.

There appears to be no clear consensus about a single solution that provides the best results to tackle the challenge of high-dimensional data. Instead, mixing several strategies appears to be the most frequently employed option. This challenge may be explored in future research, as the broad scope of solutions that can be applied promises further developments in the coming years.

Handling high generation rates, enabling rapid data visualisation, and allowing fast data querying were addressed by only one paper (Lehmann et al., 2020). This finding may suggest that current PdM research pertaining to ML is less focused on big data and more centred on the study of ML models and their implications in the studied context. The fact that only one study in the sample thoroughly explored the use of ML when applied to big data in manufacturing is a surprising result, as researchers and companies often discuss its benefits for I4.0. Hence, future research in PdM should focus on exploring the applications, implications, and limitations of ML models in the context of big data.

Reference	Challenges and solutions for voluminous and heterogeneous data
(Ayvaz and Alpay, 2021)	*How can high-dimensional data be handled? Statistical analysis and processing (correlation analysis), manual removal of duplicate variables, ML (PCA and feature selection with random forest), expert knowledge
(Aliev and Antonelli, 2021)	*How can high-dimensional data be handled? Statistical analysis and processing (correlation analysis)

Reference	Challenges and solutions for voluminous and heterogeneous data
(Nazir and Shao, 2021)	*How can high-dimensional data be handled? Variable removal by hand-crafted rules, ML (SelectKbest, SelectFromModel, FeatureImportance, and recursive feature elimination with cross-validation)
(Soualhi et al., 2021)	*How can high-dimensional data be handled? ML (autoencoder)
(Zhai et al., 2021)	*How can high-dimensional data be handled? Statistical analysis and processing
(Fathy et al., 2021)	*How can high-dimensional data be handled? ML (PCA)
(Ortega et al., 2021)	*How can high-dimensional data be handled? Data fusion techniques (sensor registration)
(Lehmann et al., 2020)	*How can high data generation rates be handled? Field Programmable Gate Arrays *How can rapid data visualisation be enabled? Delta Tables *How can fast data querying be enabled? Z-ordering
(Subramaniyan et al., 2020)	*How can high-dimensional data be handled? Expert knowledge
(Quatrini et al., 2020)	*How can high-dimensional data be handled? Expert knowledge
(Panicucci et al., 2020)	*How can high-dimensional data be handled? Statistical analysis and processing (correlation analysis)
(Kiangala and Wang, 2020)	*How can high-dimensional data be handled? ML (PCA)
(Ruiz-Sarmiento et al., 2020)	*How can high-dimensional data be handled? Expert knowledge, statistical analysis and processing, data visualisation, ML (bivariate analysis with linear regression)
(Chiu et al., 2020)	*How can high-dimensional data be handled? ML (variable importance with random forest)
(Khalid et al., 2021)	*How can high-dimensional data be handled? ML (variable importance with AdaBoost and PCA)
(Acernese et al., 2020)	*How can high-dimensional data be handled? Statistical analysis and processing

Table 2.9 Different papers and the detailed challenges and solutions related to voluminous and heterogeneous data addressed in these papers

Reference	How can high-dimensional data be handled?	How can high data generation rates be handled?	How can rapid data visualisation be enabled?	How can fast data querying be enabled?
(Ayvaz and Alpay, 2021)	X			
(Aliev and Antonelli, 2021)	X			
(Nazir and Shao, 2021)	X			
(Soualhi et al., 2021)	X			
(Zhai et al., 2021)	X			
(Fathy et al., 2021)	X			
(Ortega et al., 2021)	X			
(Lehmann et al., 2020)		X	X	X
(Subramaniyan et al., 2020)	X			
(Quatrini et al., 2020)	X			
(Panicucci et al., 2020)	X			
(Kiangala and Wang, 2020)	X			
(Ruiz-Sarmiento et al., 2020)	X			
(Chiu et al., 2020)	X			
(Khalid et al., 2021)	X			
(Acernese et al., 2020)	X			
Total	**15**	**1**	**1**	**1**

Table 2.10 Summary of detailed challenges addressed by each paper for voluminous and heterogeneous data

2.3.2.5. Fourth axis of the analytical framework: challenges and solutions in data exchange and interoperability.

Table 2.11 summarises the proposed detailed challenges in data exchange and interoperability with their respective solutions for each paper. The detailed challenges are presented as questions, and the solutions to tackle them as applied in each study are provided. Additionally, Table 2.12 presents an overview of the particulars of the detailed challenges addressed by each publication, allowing a better comprehension of the different obstacles that are commonly addressed in the literature.

From Table 2.12, we propose the following seven detailed challenges concerning data exchange and interoperability:

1) How can the data be accessed? Choosing the right strategy to store the data to allow data exchange between systems and stakeholders is essential to ensure the success of PdM tools.
2) How can the solution be hosted? Deciding where to host the solution will define the usability and scope of the PdM solution. For example, it may be straightforward to deploy an application on a personal computer. However, this would limit its use in a company. Cloud servers can be employed to host the solution and extend its reach, but the costs may be higher.
3) How can the data exchange be ensured? Defining how to communicate the data between the functional blocks of the system is a non-trivial task. For instance, this may be performed through simple solutions, such as cables connecting sensors to the data acquisition system, to more sophisticated strategies such as WiFi networks.
4) How can data exchange be managed? Managing the data stream, instructions, and actions to be performed when exchanging and collecting the data is an important choice when defining how data will be shared among the different functional blocks of the PdM system.
5) How can the errors in data streaming be tackled? Data exchange systems can encounter complications, such as a computer in a cluster experiencing an issue and having its data compromised. Therefore, it is essential to manage such problems as upstream as possible to avoid information loss.

6) How can the application be deployed and managed? Deploying applications in production that many people use is challenging. This relates to questions related to managing the life cycle of the application, such as ensuring scalability, handling different software versions and dependencies, and rapidly deploying the application for new users.
7) How can adapted response times be ensured? PdM systems with unsuitable response times are not adapted for deployment in production. Therefore, engineers designing such systems must perform appropriate design choices, both from a modelling and infrastructure perspective, to maintaining response times compliant with the requirements of the manufacturing context.

The most frequently addressed detailed challenge in data exchange and interoperability was ensuring access to the data. Solutions such as databases or data lakes in the cloud were typically employed to address this challenge. This highlights the importance of cloud technologies in I4.0, as they provide flexible scalability and almost ubiquitous ways to access data. Additionally, to tackle the challenge of hosting the solution, cloud computing was utilised. Specifically, cloud servers or edge computing solutions were utilised. Edge computing also contributed to ensuring the adapted response times. In fact, edge computing brings the storage and computations closer to where results are needed, thus improving the response times and helping to achieve real-time predictions (Mahdavinejad et al., 2018; Panicucci et al., 2020).

Generally, evidence suggests that cloud computing (including edge computing) appears to help in addressing various challenges, such as storing the data, hosting the solution, and reducing the response times. Therefore, the future of PdM with ML is likely to focus on the development and best practices for creating, deploying, and managing ML models for PdM using cloud computing. For example, to tackle the challenge of deploying and managing an application, Panicucci et al. (2020) employed Docker, a relatively recent platform released in 2013 that allows the delivery of software in containers. Each container is an isolated environment with the required software libraries for the proper functioning of an application. Thus, Docker containers facilitate faster software deployment and updates to new users. Further, when supported by cloud computing, an application running on several devices can be managed and updated at scale.

The remaining detailed challenges in this category were rarely addressed, probably because they involve already existing technical solutions for which there appear to be existing mature solutions, attracting less interest from researchers in the field of ML. For instance, to manage data exchange, recent studies have used protocols or message broker services. To ensure data exchange, WiFi or Ethernet networks were employed. Finally, to tackle errors in data streaming, checkpointing and a dedicated program to deal with cluster failures were used. The former method focused on recovering the data from the last successfully processed timestamp, while the latter aimed to recover the data between the last saved timestamp and the restart of the cluster (Lehmann et al., 2020).

Interestingly, none of the three studies that used non-sensor data tackled the challenges in this category. This does not imply that using historical maintenance log data avoids data exchange and interoperability obstacles. Instead, this suggests that the maturity of recent PdM research using maintenance event data may not be up to the point of encountering questions regarding the data exchange and interoperability between information systems.

Reference	Challenges and solutions for data exchange and interoperability
(Ayvaz and Alpay, 2021)	*How can the data be accessed? Cloud database *How can the solution be hosted? Cloud servers
(Aliev and Antonelli, 2021)	*How can data exchange be managed? Use of protocols (Real-Time Data Exchange, MODBUS, and Message Queuing Telemetry Transport (MQTT))
(Liu et al., 2021)	*How can the data exchange be ensured? Ethernet
(Lehmann et al., 2020)	*How can the data be accessed? Cloud data lake *How can the errors in data streaming be tackled? Checkpointing and dedicated program to check for cluster failures
(Panicucci et al., 2020)	*How can the application be deployed and managed? Docker *How can the solution be hosted? Cloud and gateways (edge computing) *How can data exchange be managed? Broker service *How can adapted response times be ensured? Edge computing
(Ruiz-Sarmiento et al., 2020)	*How can the data be accessed? NoSQL database
(Chiu et al., 2020)	*How can the data be accessed? Cloud database *How can the data exchange be ensured? WiFi
(Acernese et al., 2020)	*How can data exchange be managed? Use of protocols (file transfer protocol)

Table 2.11 Different papers and the detailed challenges and solutions related to data exchange and interoperability addressed in these papers

Reference	How can the data be accessed?	How can the solution be hosted?	How can the data exchange be ensured?	How can data exchange be managed?	How can the errors in data streaming be tackled?	How can the application be deployed and managed?	How can adapted response times be ensured?
(Ayvaz and Alpay, 2021)	X	X					
(Aliev and Antonelli, 2021)				X			
(Liu et al., 2021)			X				
(Lehmann et al., 2020)	X				X		
(Panicucci et al., 2020)		X		X		X	X
(Ruiz-Sarmiento et al., 2020)	X						
(Chiu et al., 2020)	X		X				
(Acernese et al., 2020)				X			
Total	**4**	**2**	**2**	**3**	**1**	**1**	**1**

Table 2.12 Summary of detailed challenges addressed by each paper for data exchange and interoperability

2.3.2.6. Fourth axis of the analytical framework: challenges and solutions in data conversion

Only one 'detailed challenge' was identified for data conversion: handling diverse data formats. Table 2.13 summarises the solution proposed to tackle this obstacle for each paper.

Reference	Challenges and solutions for data conversion
(Ayvaz and Alpay, 2021)	How to handle diverse data formats? Use of protocols (MQTT)
(Lehmann et al., 2020)	How to handle diverse data formats? A dedicated program to check for data consistency
(Panicucci et al., 2020)	How to handle diverse data formats? Use of a predefined data model

Table 2.13 Different papers and the detailed challenges and solutions related to data conversion addressed in these papers

From Table 2.13, it can be noted that only one detailed challenge was identified, i.e. handling diverse data formats. Processing various data formats simultaneously is common in real-world datasets, which tend to mix multiple data sources. For example, it is possible to have maintenance reports describing a machine failure consisting of free-form text descriptions and images provided by operators and sensor readings at the moment of breakdown. To effectively apply ML models in PdM, systems must automatically handle diverse data formats.

The results suggest that there is no standard solution to overcome the challenge of handling diverse data formats. The encountered solutions were as follows: using protocols to convert the collected sensor data into a single data type, employing dedicated computer programs to check for data consistency, thus avoiding discrepancies caused by software updates, and creating predefined data models for the PdM system.

The fact that a few papers encountered challenges related to data conversion does not imply that this problem has been resolved. This issue is probably more interesting from an industrial perspective than from an academic perspective. Further, the current maturity of PdM has not yet reached a stage where further challenges can be identified. For instance, Lehmann et al. (2020) highlighted a relevant source of data inconsistencies, i.e. software updates. This issue can be commonly found in large-scale systems running several devices, where each device may not have the same software and library versions. This problem is likely to be observed in PdM

applications using cloud computing, as cloud computing allows software deployment in several devices. However, cloud computing does not appear to be sufficiently mature in PdM with ML.

2.4. Conclusion

Research on PdM with ML has undergone an exponential growth in recent years. Hence, this chapter aimed to provide an updated glimpse of the current state of scientific literature by answering the following two RQs: how is ML currently applied in PdM and what are the challenges and their respective solutions when using ML in PdM?

These RQs allowed the definition of SQs to provide a more detailed analysis. We explored the industries, use cases, ML techniques, and data sources that are commonly addressed for the first RQ. For the second RQ, we analysed the challenges and solutions commonly encountered when developing PdM solutions with ML.

The results from the first RQ suggested that PdM research is primarily used in the automotive and electronics (semiconductors, circuit boards) industries. However, applications were observed in a wide range of sectors, showing the versatility of PdM with ML. The most frequently addressed use cases were fault detection and health state estimation.

Recent research has often used classic ML techniques, predominantly with the UL paradigm. They have often been employed in synergy with SL techniques. The most employed technique was random forest, which is suitable for PdM applications because of its capacity to learn complex non-linear relationships. Neural networks, such as LSTM and CNN, have also been applied. While SL was the most frequently encountered learning paradigm, RL was not found in the sample papers. Finally, transfer learning was used in NLP and computer vision applied to PdM to exploit previously trained models.

While considering the data sources, recent research has extensively focused on using data generated by sensors. Although this proved to deliver applications in PdM effectively, another option is to harness historical data collected across the years. This second option may be cheaper and more straightforward to implement, as the data acquisition system need not be designed, and rare skills in networks, control, and telecommunications may be required to a lesser extent. Finally, it is rare to find studies handling subjective data sources, such as written reports from operators or images. These data sources tend to be ignored because of their highly unstructured nature. However, discarding them may waste precious information that is useful for PdM.

The second RQ showed that the efforts of researchers are primarily focused on handling voluminous and heterogeneous data, designing data acquisition systems, and dealing with data quality problems. The analysis of the detailed challenges yielded the following three main conclusions:

1) Cloud computing technologies provide solutions that may overcome the obstacles identified in data acquisition, data exchange, and interoperability.
2) Although maintenance data are intrinsically imbalanced, it was rare to find studies exploring techniques to mitigate the effect of class imbalance.
3) Even if companies and researchers constantly communicate about the advent of big data in manufacturing, most of the papers related to PdM with ML did not fully explore all the challenges imposed by big data. The vast majority of them were limited to the handling of high-dimensional data, while other questions such as dealing with velocity in data generation as well as ensuring fast data querying and visualisation were ignored.

As a final word regarding the state of the recent research, our previous systematic literature review suggested that one of the essential research perspectives was handling the concept drift issue (Usuga Cadavid et al., 2020a). The concept drift issue occurs when there are changes in the environment that produced the data that was initially employed to train the ML model. Such changes make ML models obsolete, as the learnt statistical distributions and properties are no longer valid. Nevertheless, manufacturing environments are constantly subjected to variations in processes, machines, raw materials, products, etc. (Ruiz-Sarmiento et al., 2020). Although this issue was recognised by two studies in the paper sample (Panicucci et al., 2020; Ruiz-Sarmiento et al., 2020), none provided applied solutions. Therefore, the concept drift issue remains one of the most relevant avenues for future research on PdM with ML.

3. Chapter 3: Proposed approach and structure of the thesis

3.1. Objectives

The general objective of this thesis is to harness maintenance data to react more effectively to production disturbances. Three specific objectives were derived from this general objective, which are explained in the following subsections. Figure 3.1 shows a summarised view of the proposed approach with the published articles and respective chapters that have contributed to this research.

3.1.1. First specific objective: identification of research gaps, opportunities, and trends

This objective aimed to identify topics requiring further research related to the application of ML to production planning and control in the context of I4.0. The motivation behind this choice is that production planning and control is a vast domain encompassing several functions such as logistics, maintenance, quality control, and scheduling. Therefore, the aim is to understand how ML is applied to improve production systems in real-world applications, allowing us to have a broad view of what elements may guide our future research and how to orient them towards maintenance.

We performed a systematic literature review focusing on ML in production planning and control to address this specific objective. The article is completely presented in Chapter 4. Furthermore, Figure 3.1 shows that this contribution was motivated by the general objective of this thesis, thus generating the required inputs to guide subsequent studies.

3.1.2. Second specific objective: evaluation of the technical feasibility of models to address the previously identified research gaps

After identifying the research gaps resulting from the first objective, the next step is to assess the original contributions to tackle them. Thus, this objective is aimed at performing technical feasibility tests to validate our proposed approach and identify further research gaps, as certain research opportunities are easier to recognise when implementing real-world applications. Specifically, this study explored the use of free-form text data from maintenance logs to support fault detection and estimation of maintenance work duration. This article is completely presented in Chapter 5. The results allowed us to validate the technical feasibility and to set the basis for further studies addressing the identified future work perspectives.

3.1.3. Third specific objective: explore solutions to certain challenges encountered when exploiting real-world maintenance data from production

The last step was to extend our technical feasibility test by harnessing the results of the second specific objective along with the unaddressed research perspectives identified with the first specific objective. Figure 3.1 shows that these results motivated two research paths named A and B.

Research path A focused on extending Chapter 5 by performing external validation to generalise the conclusions. Hence, the performance of the proposed approach was evaluated using other real-world datasets and with more ML techniques. Additionally, it explored the capability of DL models in PdM to generate knowledge from data, which is a research gap identified with the systematic literature review and technical feasibility test. The entire article is presented in Chapter 6.

Research path B was entirely devoted to alternative techniques to tackle class imbalance. First, we used DL language models to generate artificial data and mitigate class imbalance. Chapter 7 presents this article. Second, we assessed the usage of a loss function called 'FL' to reduce the effect of class imbalance in NLP. Lin et al. (2017) initially proposed this loss function in the domain of computer vision for object detection, providing promising results. Chapter 8 describes the proposed approach, findings, and conclusions of this study.

3.2. Strategies

After defining the specific objectives that guided this research, we defined two strategies that allowed us to generate the results to accomplish the established goals. These strategies are explained in the following two subsections, summarising the key future research perspectives for each article and how these motivated the other studies.

To provide a clearer view of how these two strategies enabled us to connect the multiple studies performed in this thesis, Figure 3.2 shows the relationships between these articles. For articles belonging to the first two specific objectives, arrows indicate how some of the proposed research gaps or future work perspectives motivated the other papers.

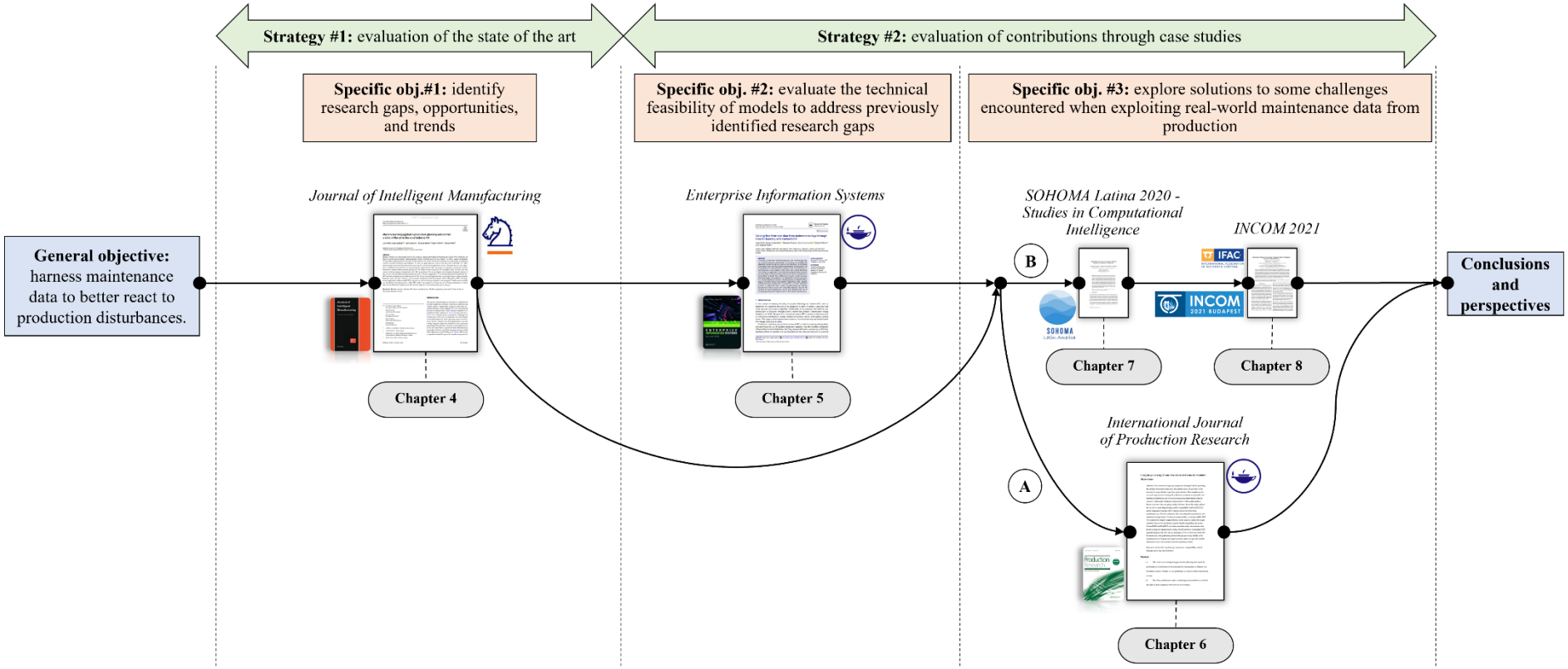


Figure 3.1 Summary of the proposed approach

3.2.1. First strategy: evaluation of the state-of-the-art methods using a literature review study

The first strategy exclusively focused on addressing the first specific objective, whose aim is to identify research gaps, opportunities, and trends in recent literature. Hence, the first strategy was to use a systematic literature review, an effective tool for deriving insights from the scientific literature.

A systematic literature review was observed in (Usuga Cadavid et al., 2020a), where we conducted a rigorous selection of recent case studies using ML in production planning and control. This process led to the selection of 93 papers that were thoroughly analysed. After performing the analysis, several trends, research gaps, and future research avenues were highlighted. The ones that primarily motivated further studies in this thesis are:

1) Including humans in the loop of ML applications in I4.0: Considering human factors in the developed ML solutions is essential to ensure acceptance from operators and the top management. For example, by ensuring minimal changes to the way they currently work or avoiding imposing new constraints on their actions, humans can feel included in the loop of ML models. However, our study showed that this inclusion has rarely been addressed in recent research.
2) Valuing historical data from information systems: Data obtained from information systems were identified to be one of the most employed data sources in our study, suggesting that companies are willing to value historical data collected through the years. Additionally, this finding indicates that research on ML for production planning and control need not be performed exclusively with sensor data.
3) Exploring class imbalance mitigation and transfer learning: Our study highlighted that researchers frequently encountered data availability issues to test their models, forcing them to employ artificially generated data instead. In fact, data are often scarce, justifying the use of transfer learning models, as these models require less data to achieve superior performance. Moreover, data containing a balanced number of instances of each class to train classification algorithms are rare in scenarios with uncommon events, such as severe machine failures in maintenance. Thus, class imbalance mitigation techniques must be used.

4) Enabling knowledge generation from data: Our study suggested that recent research frequently focused on knowledge generation from data, as it is vital for increasing the benefits and value of ML applications.

3.2.2. Case studies

We utilised case studies as the second strategy to tackle the two remaining objectives and address the identified research gaps and trends obtained with the first strategy. Case studies are a helpful tool for testing new approaches and identifying research gaps and limitations through an empirical approach. Therefore, the approach for this strategy was to empirically test our contributions and then generalise them with further research. Four case studies were conducted. They are explained in the following subsections.

3.2.2.1. Article 2: Technical feasibility test

The technical feasibility test was motivated by the following identified research gaps and trends from the literature review: including humans in the loop, valuing historical data from information systems, and exploring class imbalance mitigation techniques and transfer learning.

To include humans in the loop and value historical data from information systems, we decided to employ free-form text comments from historical maintenance logs to train ML models for fault detection and maintenance work duration estimation. Using the free-form text comments reduces the changes in the way operators work, as we consider their inputs *as provided*. In fact, certain companies have replaced their free-form text fields in maintenance reports with predefined options from drop-down menus. Although this solution homogenises the collected data, it can be cumbersome to enter data, and predefined options may not fit all the situations. Moreover, it can generate dissatisfaction among operators who prefer free-form text comments because of their flexibility and ease of creation.

As free-form text comments from maintenance logs are highly unstructured and datasets may not be sufficiently large, we decided to test the use of recent DL models called transformers, thereby allowing transfer learning to be performed for NLP. Moreover, other common ML models were tested for comparison.

Finally, as we exploited maintenance data, it presented a class imbalance. To tackle this, we explored a technique called random over-sampling (ROS) for fault detection and pre-processing the outputs with k-means for the estimation of maintenance work duration.

This study sheds light on several research gaps and limitations, from which the following are helpful for the other papers presented in this thesis:

1) Exploring other techniques to handle the effect of class imbalance: This article highlights that ROS is an effective data-level method for mitigating the impact of class imbalance. Nevertheless, it can cause overfitting to the oversampled classes (Wang et al., 2016). Moreover, training times can dramatically increase as the dataset size is artificially increased (Johnson and Khoshgoftaar, 2019). Therefore, other techniques that reduce overfitting to minority classes or avoid increased training times should be explored.
2) External validation of the results and conclusions: Other datasets should be employed to generalise the findings obtained with the technical feasibility test. For example, the findings of this study suggest that fine-tuned transformer models outperform other more common ML models, such as random forests. However, this should be validated on different datasets to generalise the conclusions from more robust empirical evidence.
3) Explore knowledge generation using text data from maintenance logs: This research avenue was also proposed in the future research perspectives that were identified in the systematic literature review. The technical feasibility test suggested that future work should explore generating knowledge and insights from highly unstructured inputs, such as free-form text data coming from maintenance.

3.2.2.2. Article 3: External validation and knowledge generation

This study was motivated by the need to externally validate the results obtained using the technical feasibility test, generate knowledge from data, and explore other techniques to address the class imbalance.

For the external validation, this study employed three datasets from different companies to apply the approach proposed in the technical feasibility test. Moreover, it explored other common ML models with better optimisation of hyperparameters to compare and verify the

superiority of transformers. To generate knowledge, we used a technique called Local Interpretable Model-agnostic Explanations (LIME), proposed by Ribeiro et al. (2016), to support a method to extract insights from highly unstructured text data. Finally, other classic techniques to tackle class imbalance (i.e. random under-sampling (RUS) and class weighting) were compared with ROS.

3.2.2.3. Article 4: Alternative mitigation of class imbalance with language models

The motivation of this study arises from the need to explore other techniques to tackle class imbalance, which is identified with the technical feasibility test. To reduce the possibility of overfitting to the minority classes when employing ROS, we proposed creating artificial observations with language models.

Language models learn the probability distributions of word sequences. Thus, given a particular input of words, a language model will generate the most likely words that may follow. We harnessed these models to learn to mimic maintenance descriptions belonging to the minority class and used the trained model to generate new instances to reduce the class imbalance. The idea behind this was to artificially create observations that are *sufficiently similar* to the original text, but with marginal variations to lessen the degree of overfitting.

3.2.2.4. Chapter 8: Alternative mitigation of class imbalance with language models

This study was motivated by the need to explore other techniques to tackle class imbalance, which are identified using the technical feasibility test. To reduce the increase in the training time when employing ROS, techniques that modify the way ML models learn can be used. These are called algorithm-level techniques. A common algorithm-level approach is to use a loss function called *weighted cross-entropy (WCE)*, which assigns weights to the classes so that minority classes have larger weights, provoking a more significant penalty if the model misclassifies them. However, this strategy does not help the ML model focus on learning to classify *hard* instances. Therefore, in this study, we explored the use of the *FL*, which modifies the cross-entropy (CE) loss to help the model learn both the minority class and the difficult to classify observations.

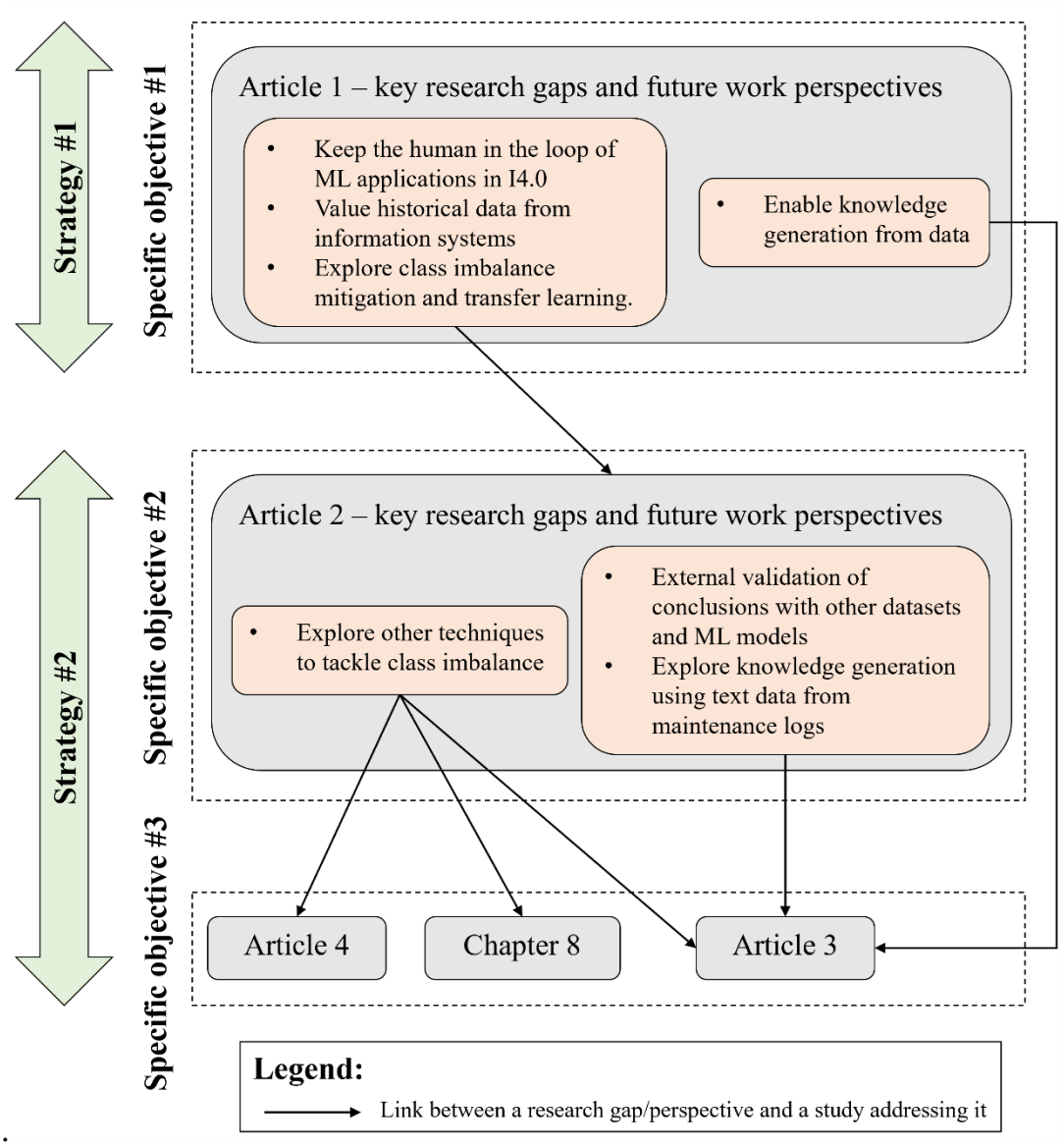


Figure 3.2 Relationships between different studies in this thesis

3.3. Industrial contribution

The industrial contribution is to provide a theoretical background, methods, and strategies to harness the maintenance data to react to production disturbances and improve production systems. Specifically, three industrial contributions were derived from each of the specific objectives.

1) The first specific objective identifies the possible applications, challenges, and benefits of using ML models for production planning and control. This may be useful for managers who are starting digitalisation projects, who are willing to know

the existing techniques while making initial assumptions about foreseen improvements and risks.

2) The second objective shows an example of how to include free-form text data from manufacturing in predictive systems to keep the human in the loop of future applications. Moreover, it highlights the recent advances in NLP applied to maintenance data, focusing on transformers. This is useful for guiding technological choices for companies willing to exploit highly unstructured text data containing jargon and typos. Additionally, this objective provides information about the existing risks when there are imbalanced data.
3) The third objective is helpful for enterprises dealing with imbalanced datasets and is willing to mitigate this issue in their applications. It provides a method for extracting insights from highly unstructured text data.

Although the second and third contributions employed data from maintenance logs, the conclusions of this work may be meaningful in other contexts where descriptions remain relatively short, and there is a class imbalance, such as product reviews, customer feedback, or social network comments. Long texts, such as books, are beyond the scope of this research, as they violate certain technical limitations of the transformer models employed in this thesis.

4. Chapter 4: Article 1 - Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0

Name of the journal: Journal of Intelligent Manufacturing

Received: 15 July 2019

Accepted: 30 December 2019

Authors: Juan Pablo Usuga Cadavid, Samir Lamouri, Bernard Grabot, Robert Pellerin, Arnaud Fortin.

Corresponding author: Juan Pablo Usuga Cadavid

DOI: https://doi.org/10.1007/s10845-019-01531-7

Abstract: Because of their cross-functional nature in the company, enhancing Production Planning and Control (PPC) functions can lead to a global improvement of manufacturing systems. With the advent of the Industry 4.0 (I4.0), copious availability of data, high-computing power and large storage capacity have made of Machine Learning (ML) approaches an appealing solution to tackle manufacturing challenges. As such, this paper presents a state-of-the-art of ML-aided PPC (ML-PPC) done through a systematic literature review analyzing 93 recent research application articles. This study has two main objectives: contribute to the definition of a methodology to implement ML-PPC and propose a mapping to classify the scientific literature to identify further research perspectives. To achieve the first objective, ML techniques, tools, activities, and data sources which are required to implement a ML-PPC are reviewed. The second objective is developed through the analysis of the use cases and the addressed characteristics of the I4.0. Results suggest that 75% of the possible research domains in ML-PPC are barely explored or not addressed at all. This lack of research originates from two possible causes: firstly, scientific literature rarely considers customer, environmental, and human-in-the-loop aspects when linking ML to PPC. Secondly, recent applications seldom couple PPC to logistics as well as to design of products and processes. Finally, two key pitfalls are identified in the implementation of ML-PPC models: the complexity of using Internet of Things technologies to collect data and the difficulty of updating the ML model to adapt it to the manufacturing system changes.

Keywords Machine Learning, Industry 4.0, Smart Manufacturing, Production Planning and Control, State-of-the-art, Systematic literature review.

4.1. Introduction

The current manufacturing environment is characterized by high complexity, dynamic production conditions and volatile markets. Additionally, companies must offer customized products while engaging low costs and reducing the time-to-market if they want to remain competitive in a globalized world (Schuh et al., 2017a; Carvajal Soto et al., 2019). This situation poses tremendous challenges for manufacturers who seek to implement new technologies to meet their objectives while expecting a return on investment. Several countries have developed projects that aim to help companies adapt their industries to new production technologies. For instance, Germany created Industry 4.0 (I4.0), the United States proposed the Smart Manufacturing Leadership Coalition, and China introduced the plan called China Manufacturing 2025 (Wang et al., 2018b). This has led to significant financial support for manufacturing research; for example in the European Union around €7 billion will be invested by 2020 in Factories of the Future (Kusiak, 2017).

Among the Industry 4.0 groups of technologies (Ruessmann et al., 2015), Big Data and Analytics (BDA) allows the constantly growing mass of produced data to be harnessed to generate added value. In fact, data generation in modern manufacturing has undergone explosive growth, reaching around 1000 Exabytes per year (Tao et al., 2018). However, the potential of this data has been found to be insufficiently exploited by companies (Manns et al., 2015; Moeuf et al., 2018). As BDA enables the exploitation of data, the scope of this review will focus on this technology, and more specifically ML applied in Production Planning and Control.

In the context of I4.0, Production Planning and Control (PPC) can be defined as the function determining the global quantities to be produced (production plan) to satisfy the commercial plan and to meet the profitability, productivity and delivery time objectives. It also encompasses the control of production process, allowing real-time synchronization of resources as well as product customization (Tony Arnold et al., 2012; Moeuf et al., 2018). In this review, I4.0 is considered a synonym of Smart Manufacturing, as they both refer to technological advances that value data to draw improvements in production. For example, Ruessmann et al. (2015) proposed nine technologies for I4.0 while Kusiak (2019) suggested six, but for Smart Manufacturing. Both proposals tend to refer to similar technologies and variations depend on the authors' focus. Hence, as the PPC is a core function of manufacturing, this paper regards its

improvement through I4.0 technologies, namely ML, which concerns BDA. Regarding ML, the definition that will be retained is the one of a computer program capable of learning from experience to improve a performance measure at a given task (Mitchell, 1997).

Classical approaches to performing PPC include analytical methods and precise simulations, providing solutions that may rapidly become unfeasible in the execution phase due to the stochastic nature of the production system and uncertainties such as machine breakdowns, scrap rate, delayed deliveries, etc. Moreover, Enterprise Resource Planning (ERP) systems perform poorly at the operative level (Gyulai et al., 2015). To tackle this issue, ML can endow the PPC with the capacity of learning from historical or real-time data to react to predictable and unpredictable events. Even though this may suggest that organizations must invest in data warehousing to handle the mass amount of collected data, studies have reported that enterprises successfully implementing data-driven solutions have experienced a payback of 10-70 times their investment in data warehousing (Rainer, 2013).

Having introduced the synergism between ML and PPC, this study aims to provide an analysis of its state-of-the-art through a systematic literature review. This will contribute to the definition of a methodology to implement a ML-PPC and to the proposal of a map to classify scientific literature. This paper analyzes research produced in the context of the I4.0 and is guided by five research questions:

1) Which are the activities employed to perform a ML-PPC?
2) Which are the techniques and tools used to implement a ML-PPC?
3) Which are the currently harnessed data sources to implement a ML-PPC?
4) Which are the addressed use cases by the recent scientific literature in ML-PPC?
5) Which are the characteristics of the I4.0 targeted by the recent scientific literature in ML-PPC?

The first three questions are related to the first objective of this research. They will contribute to the definition of a methodology to implement a ML-PPC. The last two questions address the second objective, as they will provide the basis to create a classification map.

The remainder of this paper is organized as follows: section “Research methodology and contribution” will explain the systematic literature review methodology employed to search and choose the sample of scientific articles. Additionally, the contribution of this paper with respect

to similar studies will be briefly highlighted and a short bibliometric analysis is presented to assess the keywords used as string chains. The "Analytical framework" section will explain the four axes encompassed by the analytical framework. Afterwards, the "Results" section will focus on the results of the systematic literature review and an analysis of it. Finally, the "Conclusion and further research perspectives" section will conclude this study and provide further research perspectives.

4.2. Research methodology and contribution

To meet the two objectives of this study, a systematic literature review was carried out following the method proposed by Tranfield et al. (2003) who extended research methods from the medical sector to the management sciences. This method has been successfully employed by other authors to draw insights from the scientific literature (Garengo et al., 2005; Moeuf et al., 2018). This literature review focuses exclusively on applications of ML in PPC in the context of I4.0.

In another domain, Zhong et al. (2016), proposed a bibliometric analysis of big data applications on different sectors such as healthcare, supply chain, finance, etc. but its focus on manufacturing was limited. (Kusiak, 2017; Tao et al., 2018) and (Wang et al., 2018b) provided a literature analysis of data-driven smart manufacturing, citing representative references. However, these references were not chosen through a systematic literature review. Finally, (Sharp et al., 2018) could be considered as a study close to this paper as the authors used a pre-defined methodology to select the articles to analyze. Nevertheless, they employed Natural Language Processing (NLP) to analyze around 4000 unique articles and provide insights about the scientific literature of ML applied in I4.0. The use of NLP can be useful to identify important trends, but it does not allow the authors to analyze the detail of the reviewed papers, where it is likely to find interesting research gaps and insights. On the other hand, a systematic review allows the authors to both follow a rigorous methodology and perform a detailed study of each chosen article.

Even though the PPC is closely related to the domain of supply chain, the latter is not included in the scope of this review as its vastness would increase the risk of straying from the focus on PPC. Therefore, to learn about recent trends on this topic, the authors invite readers to refer to Hosseini et al. (2019), who performed a comprehensive review of quantitative methods, technologies, definitions, and key drivers of supply chain resilience. In fact, supply chain

resilience is a growing research area that examines the ability of a supply chain to respond to disruptive events (Hosseini et al., 2019). Applications of this topic have been done by Hosseini and Barker (2016), who applied Bayesian networks to perform supplier selection based on primary, green, and resilience criteria; and Hosseini and Ivanov (2019), who proposed a method using Bayesian networks to assess the resilience of suppliers in identifying critical links in a supply network.

The queries were performed between 10/10/2018 and 24/03/2019 in two scientific databases: ScienceDirect and SCOPUS. The following keywords conducted the research:

- (“Deep Learning” OR “Machine Learning”) AND (“Production scheduling”)
- (“Deep Learning” OR “Machine Learning”) AND (“Production scheduling”)
- (“Deep Learning” OR “Machine Learning”) AND (“Production scheduling”)
- (“Deep Learning” OR “Machine Learning”) AND (“Production scheduling”)

To consider the context of I4.0, only papers published since 2011 were considered because this year corresponds to the formal introduction of I4.0 at the Hannover Fair. Additionally, only communications labeled as “Research Articles” in ScienceDirect and “Conference paper” OR “Article” in SCOPUS were included to solely capture articles presenting application models. Subsequently, a review of titles and abstracts allowed for the exclusion of articles not related to ML-PPC. After the removal of duplicates, a full text analysis allowed a final selection that excluded papers that did not fit with research questions. The sample size obtained encompasses 93 scientific papers. The article selection methodology with its Restrictions (R) is described in Figure. 4.1.

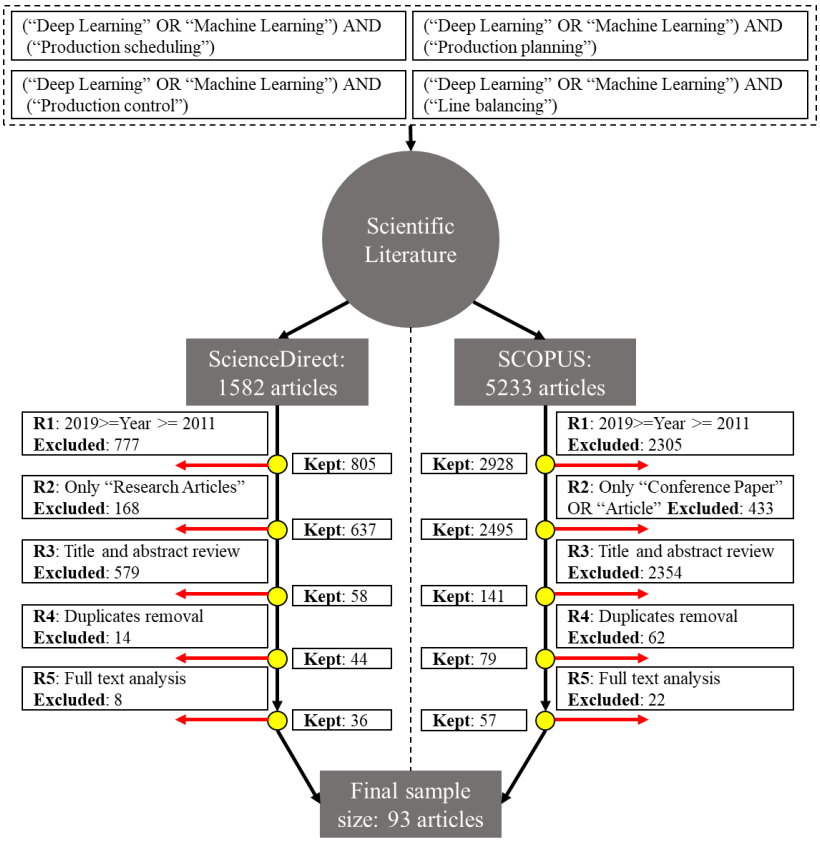


Figure. 4.1 Search strategy used to capture the scientific literature.

4.2.1. A brief focus on the query keywords

The used string chains represent a core strategic choice for review. Therefore, this sub section aims to provide an analysis of the employed keywords.

Concerning the keywords used in the first parenthesis of the string chains, the use of "Deep Learning" and "Machine Learning" was done for two reasons: firstly, they are relatively new terms, which eases the identification of recent trends in the literature; and secondly, they are directly related to one of the two core subjects in this study, which is ML. Other terms such as "Data Mining" or "Statistical Learning" could have been sensible choices too, as they are often used interchangeably with "Machine Learning" and "Deep Learning". Nevertheless, using these two terms might have deviated this study from its core topic. In fact, a recent study suggests that the differences between ML and Data Mining are not consistently defined in the literature. Thus, Data Mining is mostly considered to be the process of generating useful knowledge from data (Schuh et al., 2019). To do so, it draws from other fields such as Artificial Intelligence, Statistics, ML, and Data Analytics. Therefore, Data Mining can be a vast topic,

and does not exclusively concern ML, which could potentially affect the focus of this study. As there seems to be no clear boundary between these terms, a short bibliometric analysis was performed to assess the chosen keywords. The analysis was done using VOSviewer, software developed by the University of Leiden to draw insights from scientific literature. Furthermore, using keywords related to specific ML techniques such as “Random Forest” or “k-means” did not seem appropriate due to the risk of introducing a bias when answering the second research question. In fact, this could have artificially boosted the results of the queried techniques.

The bibliometric analysis followed a similar methodology to that used to choose the final article sample (cf. Figure. 4.1). The objective was to briefly assess the influence of different keywords on the queries’ results. For the analysis, three different string chains were considered: “Deep Learning” OR “Machine Learning”, “Data Mining”, and “Statistical Learning”. The queries were performed on 06/10/2019 and the detail of the search strategy can be found in Appendix I. Finally, as the aim is to analyze the available literature when querying with a certain string chain, no Title and abstract review was done as this could introduce a bias into the results due to the authors’ influence.

The bibliometric analysis focused on the keywords defined by the authors for all of the papers of each of the three samples. To represent the results, the network visualization from VOSviewer was employed. In such a network, the nodes represent the keywords or items, their sizes represent the keyword importance determined by the number of occurrences, and the links between the nodes represent their co-occurrence. Furthermore, the relatedness between two terms is represented through their spatial distance in the network: two keywords closely related will be spatially closer. For this review, the obtained networks were displayed under the “overlay visualization,” which shows the average publication year for each of the keywords through a color scale. For clarity reasons, a filter was applied on the minimum number of occurrences to display; at most, 50 items per graph. Also, the queried keywords were highlighted with a red frame to assist in their identification. The networks are presented on Figure 4.2, Figure 4.3, and Figure 4.4.

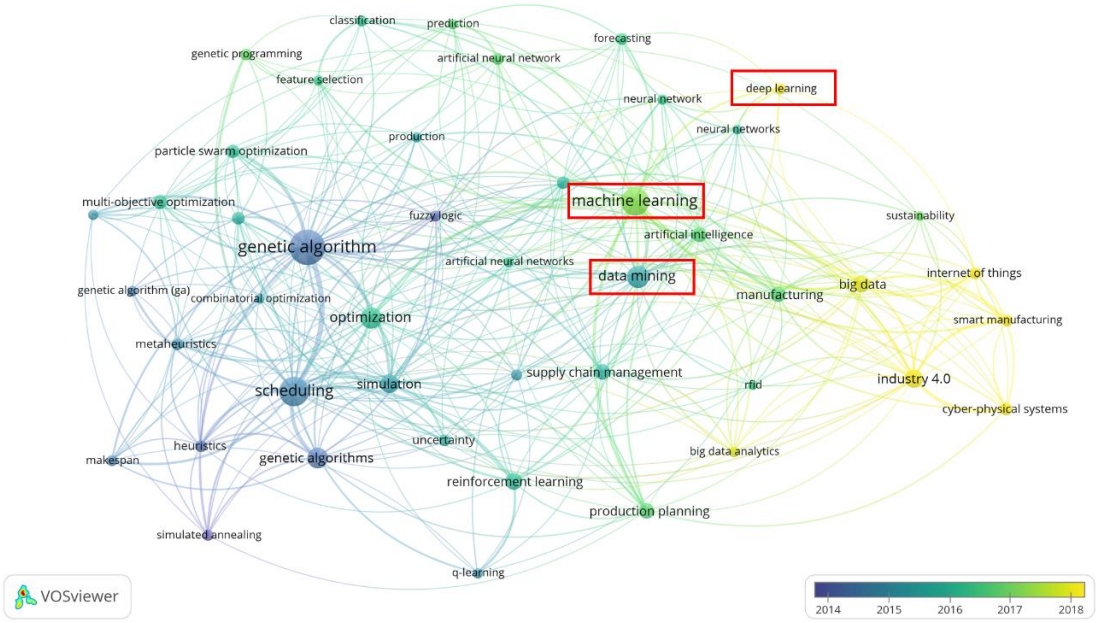


Figure 4.2 Network visualization with the average publication year for “Deep Learning” OR “Machine Learning.”

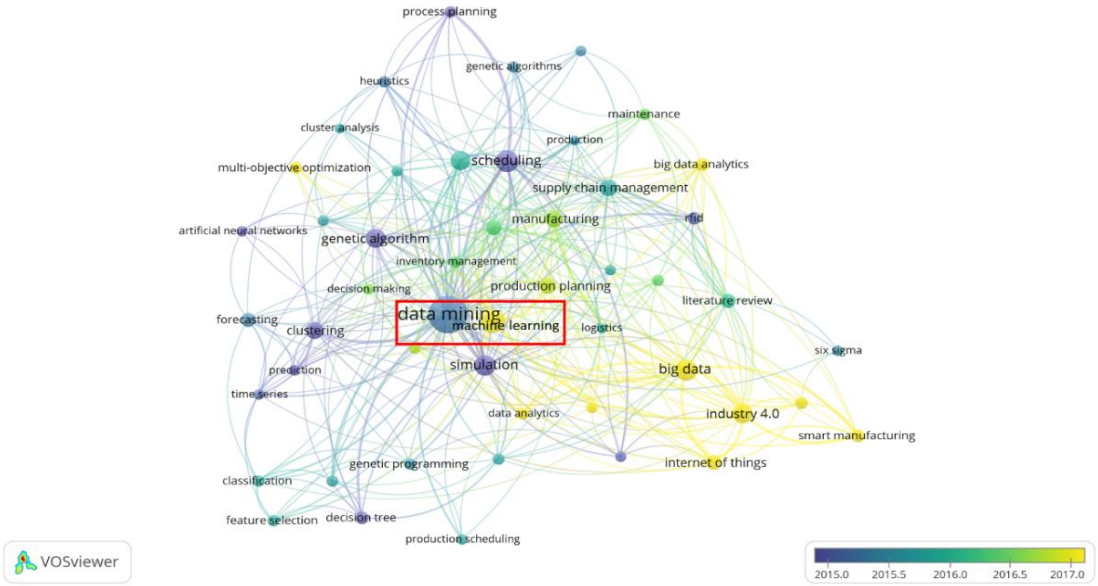


Figure 4.3 Network visualization with the average publication year for “Data Mining.”

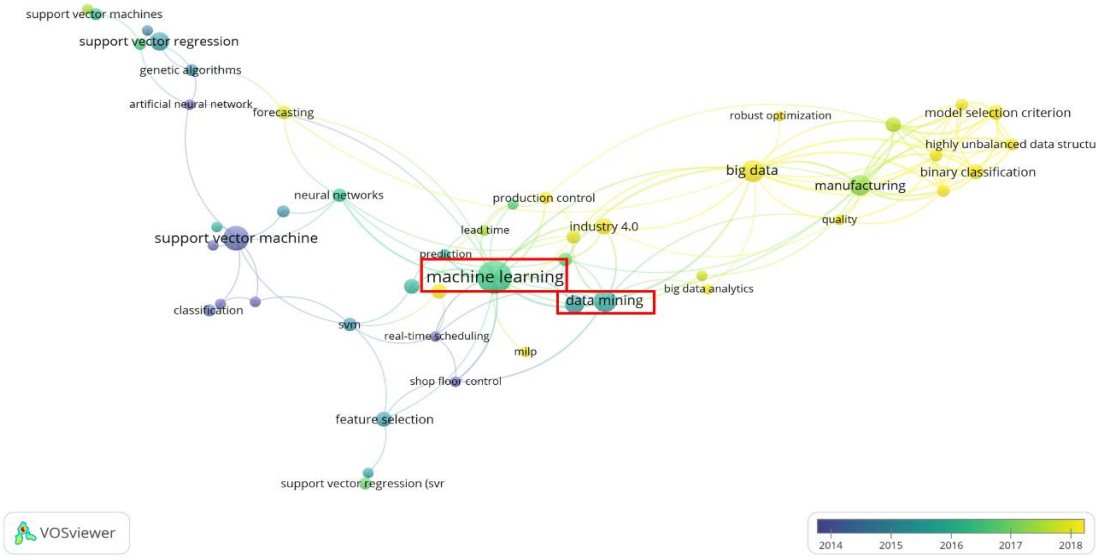


Figure 4.4 Network visualization with the average publication year for “Statistical Learning.”

Results from the bibliometric study suggest that “Statistical Learning” may not be a common keyword to find in ML-PPC research because the size of the obtained article sample (241 articles) is far below the results obtained with the other two queries. In fact, “Deep Learning” OR “Machine Learning” and “Data Mining” provided 2862 and 2166 articles, respectively (cf. Appendix I). This is also stated in all of the networks, in which the item “Statistical Learning” does not appear, probably due to the filter excluding keywords with a low number of occurrences.

Analyzing the relatedness between “Data Mining” and “Machine Learning” by their spatial distance on the networks provides an idea of how these concepts are associated: they are spatially closer on the “Data Mining” Network (Figure 4.3) than on the “Deep Learning” OR “Machine Learning” network (Figure 4.2). This suggests that Data Mining tends to relate more often to ML, rather than ML to Data Mining. Such a relation may support what is said in (Schuh et al., 2019), in which Data Mining is considered a field drawing from ML, Artificial Intelligence, Statistics, etc. to produce useful insights.

Findings from the network visualizations show that the item “Machine Learning” is always associated with a more recent average publication year than “Data Mining”. This supports the idea that “Machine Learning” is a relatively new term, which can lead to the identification of

recent trends in literature. Furthermore, querying with “Deep Learning” OR “Machine Learning” provides a more recent average publication year (2017.06) for the item “Machine Learning” than with the other two queries: 2016.95 when querying with “Data Mining” and 2016.41 when querying with “Statistical Learning”. Finally, “Deep Learning” OR “Machine Learning” was the only query enabling the inclusion of the item “Deep Learning” with enough occurrences (25) to pass the filter, which is a recent research topic with an average publication year of 2018.28.

From the bibliometric analysis, it could be concluded that using “Deep Learning” OR “Machine Learning” as part of the query keywords is appropriate enough, as this allows for the identification of a big sample of recent papers, enabling the identification of new trends. It seems that “Statistical Learning” does not provide enough recent results to be considered. Finally, even if “Data Mining” is closely related to “Machine Learning,” it covers a vast domain that can deviate from the focus of this review.

Regarding the keywords employed in the second parenthesis of the string chains, the objective was to represent the main functions of the PPC under the definition provided in the introduction section. Consequently, a determination of the global production quantities was represented by “Production Planning” and the aspect of the main objectives (i.e. profitability, productivity, and delivery time) was depicted by “Production Control”. Finally, the real time synchronization of resources as well as product customizations were represented by both “Production Scheduling” and “Line Balancing,” given the fact that companies should be able to perform balanced scheduling even when facing customized client orders.

As the PPC is a transverse topic tangled with other functions such as maintenance, quality control, logistics, etc., the challenge was to decide whether or not these related subjects should be included as explicit keywords for the queries. The final choice was to not include them through keywords, as this would broaden the perimeter of the research too much, losing a focus on PPC. Nevertheless, it was decided to include, in the final article sample, the studies dealing with other functions only if they were related to the PPC.

4.3. Analytical framework

This section presents the four axes that build the analytical framework that will be employed to harness knowledge and insight from the final sample of 93 scientific articles.

4.3.1. First axis of the analytical framework: the elements of a method

This axis concerns the first and second research questions: the activities, techniques, and tools to implement a ML-PPC model. To link these three elements, the concept of "Mandatory Elements of a Method" (MEM) proposed by Zellner (2011) is used. In fact, this concept has been successfully employed by other authors to propose methodologies in research domains such as product development (Lemieux et al., 2015) and lean in hospitals (Curatolo et al., 2014). Moreover, (Talhi et al., 2017) suggested its use to develop a methodology in the context of cloud manufacturing applied to product lifecycle management. Thus, the MEM suits the first objective of this study, which concerns the definition of a methodology to implement a ML-PPC. There are five elements in the MEM:

1) A procedure: order of activities to be followed when the method is employed.
2) Techniques: the means to generate the results. Activities from the procedure are supported by techniques, while the latter is supported by tools.
3) Results: they correspond to the output of an activity.
4) Role: the point of view adopted by the person who performs the activity and is responsible for it.
5) Information model: this refers to the relation between the first four mandatory elements.

In the scope of this study, the first two elements are the concern. Firstly, to evaluate the procedure, the activities used to perform a ML-PPC implementation will be recognized and their use will be measured. By activities, this research refers to tasks such as "model comparison and selection" or "data cleaning". Secondly, to address the techniques, ML models and tools will be identified, and their use will be measured. ML models point to elements such as Support Vector Machines or Neural Networks, while tools relate to programming languages or software used to implement these ML models.

To provide further insight concerning the ML techniques, the learning types will also be measured. This will be used to summarize the information regarding the techniques as well as to ease the identification of trends and research perspectives. Additionally, the learning types will serve as a bridge between the first and second objectives of this study, as they will be used in the mapping to classify scientific literature. Based on the work of Jordan and Mitchell (2015), three main learning types can be identified:

1) Supervised Learning (SL), which concerns ML techniques approximating a function $f(X) = Y$ by learning the relationship between the inputs X and the outputs Y. For instance, learning the mapping between the Red, Green, and Blue (RGB) codes (input X) in an image and the objects in it (output Y) to determine if a certain picture contains a misplaced product in a stock rack.
2) Unsupervised Learning (UL), which encompasses techniques allowing data exploration to find patterns and hidden structures in a given dataset X. For instance, finding categories in maintenance reports by using the description of the problem and the duration of the maintenance intervention.
3) Reinforcement Learning (RL), which are techniques allowing the learning of actions to be performed by an agent interacting with a certain environment to maximize a reward. For example, teaching an Automated Guided Vehicle (AGV) in a warehouse how to avoid obstacles to maximize the number of delivered packages.

4.3.2. Second axis of the analytical framework: employed data sources

This axis addresses the third research question: the harnessed data sources. Identifying which are the data sources used to perform a ML-PPC is capital. In fact, data could be considered as the raw material allowing ML models to develop autonomous computer knowledge gain (Sharp et al., 2018). Moreover, the quality of the final model will depend to a great extent on the quality and appropriateness of the used data. Therefore, the choice of the data source is an important decision when training a ML model. To address this axis of the analytical framework, the data source types proposed by Tao et al. (2018) will be used. They mention that there are five main data sources used in the data-driven smart manufacturing:

1) Management data (M): historical data coming from company's information systems such as the ERP, Manufacturing Execution System (MES), Customer Relationship Management system (CRM), etc. M data will concern production planning, maintenance, logistics, customer information, etc.
2) Equipment data (E): data coming from Internet of Things (IoT) technologies implemented in the factory. It refers to sensors installed in physical resources such as machines, places such as workstations or human resources such as workers. In the case of workers, data is collected passively, such as by RFID sensors installed on helmets.

3) User data (U): consumer information collected from e-commerce platforms, social media, etc. It also encompasses feedback given by workers or experts that will be used to train the ML-PPC model. User data coming from workers is collected actively, for example through interviews or questionnaires.
4) Product data (P): data originating from products or services either during the production process or from the final consumer.
5) Public data (Pb): data available in public databases from universities, governments or from other researchers.

The analysis of the 93 shortlisted articles suggested that some of them did not fit into the five data sources proposed by Tao et al. (2018): these communications used artificially generated data through computer simulations. Therefore, a sixth data source is proposed, which corresponds to the first contribution of this paper to the scientific literature:

6) Artificial data (A): data generated by computers (e.g. simulations) to assess ML-PPC implementations.

4.3.3. Third axis of the analytical framework: the use cases of the ML-PPC in the I4.0

This axis concerns the fourth question: it aims to show which applications can be achieved when applying a ML-PPC. Moreover, identifying the use cases and quantifying their use frequency is important to detect trends as well as further research gaps. By use cases, this study refers to the different possible applications in a certain domain, such as maintenance, quality control, distribution, etc. In fact, as the PPC is entwined with several manufacturing subjects, is difficult to perform a complete review on PPC if these topics are ignored. For example, if there were a predictive maintenance study meant to enable a more robust production scheduling, such application would be directly related to the PPC through maintenance. To start this analysis, the use cases of I4.0 initially proposed by Tao et al. (2018) were considered. They identified six of them:

1) Smart Maintenance: harnessing data to perform preventive and predictive maintenance. For instance, monitoring machine components to estimate the best date to perform a maintenance intervention.

2) Quality Control: applying BDA to supervise the manufacturing process or products, seeking for possible quality problems and/or allowing the identification of root causes.
3) Process Control and Monitoring: constantly analyzing data coming from the shop floor to perform a smart adjustment of the functioning parameters of physical resources (machines, AGVs, etc.). The objective is to automatically control these physical resources and/or optimize their parameters with respect to the working conditions.
4) Inventory and Distribution Control: stock management, parts and tools tracking, and distribution control with the use real-time and/or historical data.
5) Smart Planning and Scheduling: considering production uncertainties to perform a production planning and scheduling closer to the current state of the production system. For instance, considering unexpected maintenance problems to reschedule a production order and minimize the delay.
6) Smart Design of Products and Processes: using BDA to support new products and processes development. For instance, using NLP to analyze the technical requirements of a new product and then to propose the potentially suitable manufacturing process.

The analysis of the 93 scientific articles suggests that these six use cases are not enough to fully characterize the recent publications. Additionally, papers not fitting in the initially proposed use cases shared the same application: time estimation (cycle time, operation time, etc.). Consequently, a seventh use case is proposed:

7) Time estimation: adaptation of different manufacturing related times to current working conditions. For instance, estimating the operation times to the actual work rate of each employee instead of using the data from the Method Time Measurement (MTM) approach.

4.3.4. Fourth axis of the analytical framework: the characteristics of I4.0

The I4.0 aims to transform the collected data during the product's lifecycle into "intelligence" to enhance the manufacturing process (Tao et al., 2018). With this transformation, the objective is to reduce costs while improving the quality, productivity, sustainability of the production system (Wang et al., 2018b). However, what specific benefits could be expected when

embracing the I4.0? To answer this question, the characteristics of I4.0 need to be identified. Tao et al. (2018) argue that I4.0 enables the following paradigms:

1) Customer-Centric Product Development: production systems in the I4.0 should be able to adjust their parameters by considering variables coming from customers such as their behavior, their needs, the way they use the products, *inter alia*. It is the case of manufacturing personalized products, designing processes from the customer requirements or proposing a target manufacturing cost for each consumer profile.
2) Self-Organization of Resources: I4.0 should endow production systems with the capacity of considering data coming from the manufacturing process to better engage the available resources. Additionally, this data should also be used to plan capital and operational expenditures. For example, updating the scheduling of machines the shop floor after new urgent order is released.
3) Self-Execution of Resources and Processes: in the I4.0, resources should become "smart" by providing them a real-time awareness and interaction capacity with the manufacturing environment (Huang et al., 2019). Therefore, the self-execution of resources concerns their faculty of making decisions depending on the received information or measured data. It is the case of machines automatically adapting their functioning parameters to work optimally or trolleys automatically replenishing workstations when these reach a certain level of security stock.
4) Self-Regulation of the Production Process: unexpected events should be effectively handled in the I4.0. Thus, this characteristic concerns the capability to perform the required adjustments to respond to unpredicted problems. For example, relaunching the scheduling process for a certain production line when one of the machines experienced a breakdown.
5) Self-Learning of the Production Process: this characteristic follows a similar logic as the self-regulation of processes in terms of adjustability. However, it relates to the capacity of the production system to adapt to predicted events. It is the case of predictive maintenance, which uses BDA to estimate the remaining useful life of machine's components. Afterwards, the manufacturing system can adapt to the results of this prediction.

After concluding the analysis of the 93 articles, three characteristics seem to be overlooked: the environmental dimension, the knowledge generation, and the inclusion of the human being. To

consider these dimensions that seem to not be explicitly raised in the work of Tao et al. (2018), three new characteristics are proposed:

6) Environment-Centric Processes: estimations suggest that the electronics and home appliances industry scrapped around 100 million goods in China in 2012 (Tian et al., 2013). As exemplified, the environmental impact of industry is far from being negligible, which is the reason why industrialized countries have started to tighten regulations and engage environmentally friendly practices in manufacturing (Tuncel et al., 2014). Research done in the context of I4.0 must not overlook this aspect. Therefore, this characteristic concerns the use of new technologies to create environment-centric processes. For example, optimizing the disassembly scheduling process to maximize the number of components that can be recycled.
7) Knowledge Discovery and Generation: most of the companies have been computerized for a long time, which has eased the collection of data. Despite the access to a plethora of information systems, generating knowledge from raw data still supposes a major industrial and academic challenge. Besides, the generation of knowledge is a mandatory step to improve the adoption of BDA by companies (Grabot, 2020). In fact, knowledge could be considered as one of the most valuable assets in manufacturing (Harding et al., 2006), the reason why generating it represents an important gain behind the adoption of BDA. Therefore, as I4.0 is characterized by allowing knowledge creation, research efforts must include it to generate value. One example of this is harnessing data from maintenance reports to provide the production of responsible real-time information about the root causes of machine breakdowns.
8) Smart Human Interaction: even with the advent of multiple I4.0 technologies, its adoption would be significantly hindered by not keeping humans in the loop or not considering their interaction with the proposed solutions. For instance, Thomas et al. (2018a) experienced the case of a company that was not willing to introduce an improved version of a quality control system because it somehow excluded the person from the process. Therefore, this characteristic concerns the consideration and/or inclusion of a human being when implementing new technologies. Examples of this would be a worker behavior recognition system based on computer vision or software interacting with operators through NLP.

Figure 4.5 summarizes this section. It also presents the relationship between the Research Questions (RQ), the analytical framework axes, the research objectives, and the expected outputs of this study.

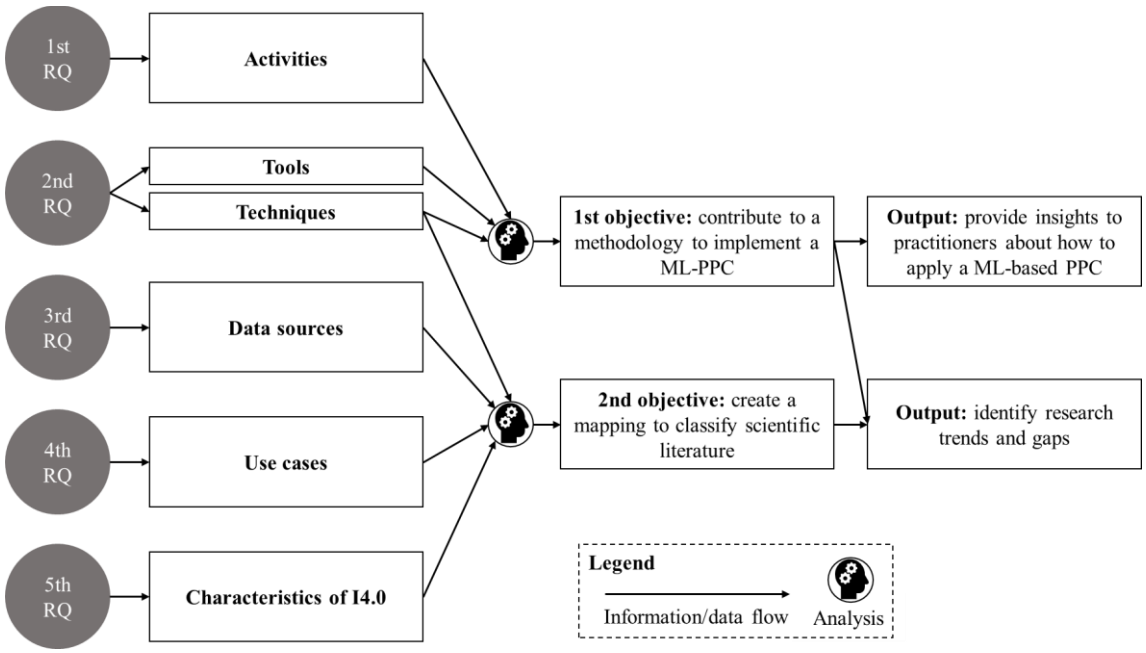


Figure 4.5 Relationship between the building blocks, research objectives and expected outputs of this study.

4.4. Results

4.4.1. First research question: activities employed in ML-PPC

To identify the activities, the tasks used to implement a ML-PPC in each of the 93 communications were identified. Afterwards, these tasks were grouped into categories to ease the information analysis. These groups of activities were analyzed by two experts to keep the most meaningful ones. Results suggest eleven standard and recurrent activities:

1) Data Acquisition system design and integration (DA): design and implementation of IoT systems to collect data. This activity also encompasses the data storage and communication protocols.
2) Data Exploration (DE): use of data visualization techniques, inferential statistics, and others to derive initial insights and conclusions about the dataset.

3) Data Cleaning and formatting (DC): data preparation from the raw data to make it exploitable by the ML-PPC model. It concerns tasks such as outlier removal or missing values handling.
4) Feature Selection (FS): choice of the most suitable inputs to the ML-PPC model. It can be done through statistical techniques, e.g. stepwise regression or by means of expert insight.
5) Feature Extraction (FE): use of variables from the initial dataset to calculate more meaningful features.
6) Feature Transformation (FT): representation of the initial features into different spaces or scales using techniques such as normalization, standardization or kernel transformations.
7) Hyperparameter Tuning and architecture design (HT): definition of the ML model architecture and adjustment of its hyperparameters to improve the performance. For instance, optimizing the learning rate and defining the activation function in a neural network.
8) Model Training, validation, testing, and assessment (MT): using the data to perform the training, validation and testing process. It can be done through techniques such as k-fold cross-validation. It also encompasses the choice of the training/validation/testing set split and the model's performance assessment.
9) Model Comparison and selection (MC): several ML techniques can be used to achieve a certain task. This activity concerns the comparison of multiple ML models to choose the one that better suits the needs.
10) Contextualized Analysis or application (CA): going further than just assessing the model's performance. It concerns the actual implementation of the ML-PPC model or the analysis of its results in the context of the problem that is addressed by the study.
11) Model Update (MU): data used to train ML models represents the context of the studied environment at a given moment. However, this context is dynamic, hence the ML-PPC model must be adapted. Therefore, this task concerns the model update through new data.

To address this research question, the percentage of papers using each activity was measured. These results are summarized in Figure 4.6. Findings suggests that four groups of activities can be proposed following their usage:

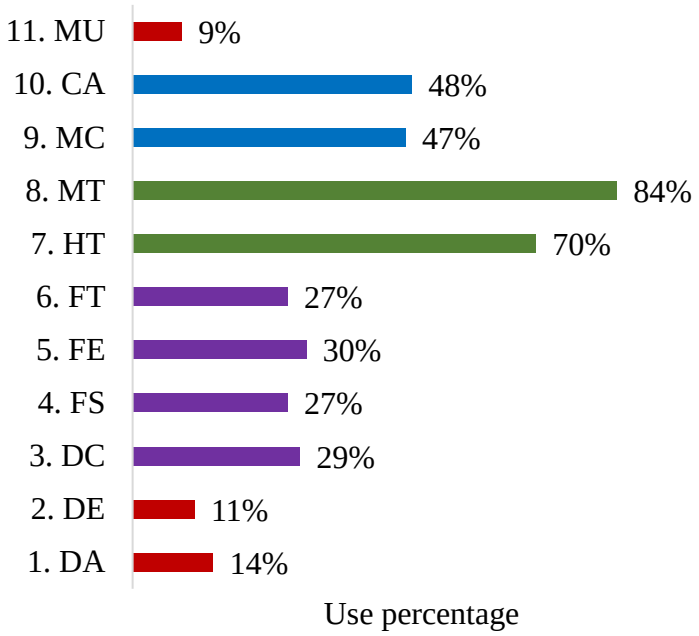


Figure 4.6 Use percentage by activity. CUAs in green, OUAs in blue, MUAs in purple, and SUAs in red.

These groups show that a considerable amount of research papers only focus on the architecture design, training, and assessment of ML-PPC models (CUAs cluster), while not employing or documenting the use of other activities. Considering OUAs, it is surprising to find that only half of the communications used the CA, which corresponds to an actual implementation of the proposed model in the context of the study. This suggests that half of the studies go no further than just training and evaluating the performance of the model.

MUAs group encompasses data pre-processing tasks, which are capital to any ML implementation. Even if these activities are frequently employed in practice, their low usage is probably because researchers do not mention them, implying a lack of documentation. Moreover, as one of the characteristics of big data is the variety (in type, nature, format, etc.) (Zhou et al., 2017), it is crucial to employ data pre-processing activities to ensure the quality of the final models. Consequently, this lack of documentation can represent a pitfall to practitioners willing to apply ML-PPC based on research papers.

Finally, SUAs cluster highlights the most important research gaps in scientific literature. Three key findings can be inferred from activities in this group: firstly, the low usage of DA highlights the challenge of coupling IoT technologies with ML-PPC. This is a major pitfall to deploy ML-PPC in companies, as they normally need real-time data or statuses from their manufacturing systems. Secondly, the lack of DE utilization could mean that ML-PPC applications tend to jump directly to activities in the CUAs cluster while overlooking descriptive and basic inferential statistics techniques. This represents an obstacle to generating knowledge from data, as DE can draw conclusions easily interpretable by non-ML specialists. Finally, the rare use of MU implies that adapting the ML-PPC model to a dynamic manufacturing context is seldom addressed. This unpredictable change of the statistical properties and relationships between variables over time is known as *concept drift* (Hammami et al., 2017). Not addressing this issue can be harmful for the model reliability in the long term.

4.4.2. Second research question: techniques and tools used in ML-PPC

Concerning the techniques, results present the number of times a given ML model is used. In the case of communications comparing several techniques, only the one chosen by the authors because of its better performance was considered. If this best-performing model employs several techniques, each of them is counted as used once.

There are numerous ML techniques in scientific literature. Therefore, to ease the analysis of results, a grouping of techniques in families is proposed is Table 4.1. These families were determined with the help of a ML expert. It is important to mention that the column "Concerned techniques" in Table 4.1 is not an exhaustive list, it is limited to techniques found in the systematic literature review.

Family	Concerned techniques
Association rule	Association rule
Bayesian models	Bayesian networks, naïve bayes
Canonical Variable Analysis	Canonical Variable Analysis
Clustering	c-Means, density peak clustering, hierarchical clustering, k-Means
Ensemble learning	Bagging, gradient boosting, machine learner fusion-regression, random forests, stacking

Family	Concerned techniques
FURIA	FURIA
k-NN	K-Nearest Neighbors (k-NN), Neighborhood Component Feature Selection
Neural Network (NN)	Artificial neural agent, autoencoders, convolutional neural network (CNN), deep belief networks, extreme learning machine, long short-term memory (LSTM), multi-layer perceptron, self-organizing maps, stacked denoising autoencoders
Principal component analysis (Princip. Comp. analysis)	Principal component analysis
Q-Learning	Q-Learning
Regression	Gaussian process regression, linear regression, logistic regression, polynomial regression, radial basis function approximation
R-learning	R-learning
Sarsa	Sarsa
Supervised Locally Linear Embedding (Sup. Locally Linear Embed.)	Supervised Locally Linear Embedding
Support vector machines	Support vector machines (SVM)
Decision Trees (DT)	Decision trees

Table 4.1 Technique families with their respective ML models.

Results are presented in Figure 4.7. They suggest that NN, Q-Learning, and DT are the most used techniques in ML-PPC. The extensive use of NN is probably due to their ability to learn complex non-linear relationships between variables, often delivering good performance when compared to other techniques. Even if Q-Learning remains, by far, the most used RL technique, other RL models such as Sarsa or R-Learning are used, which points an interest in agent-based modeling in ML-PPC. Finally, the attention drawn by DT techniques is probably linked to their excellent trade-off between accuracy and interpretability, allowing knowledge generation.

The high use of Clustering techniques could be explained by the fact that data in manufacturing systems is normally unlabeled and can contain meaningful unknown patterns. Therefore, clustering can be employed to discover groups as well as hidden structures in datasets.

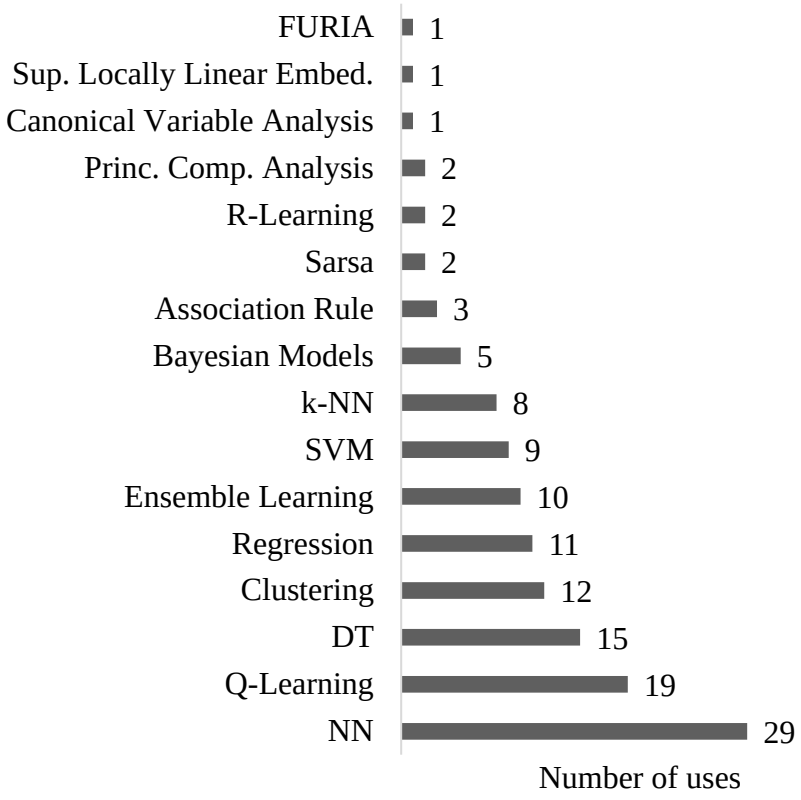


Figure 4.7 Number of uses by technique family.

The usage evolution of the six most used technique families was also measured. Figures representing this can be found in Appendix II. Due to an imbalance in the amount of articles over the different years, results are presented as relative frequencies. For example, if the NN achieved a usage of 27% in 2018, it means that 27% of all the techniques used in that year corresponded to such a model. Results suggest that there is a strong growth in the use of NN since 2015, this is possibly due to the growing computing power, recent findings in terms of architectures such as CNNs or LSTMs, and the development of specialized frameworks like PyTorch, TensorFlow, Keras, etc. which ease the task of implementing such models. Moreover, results show a growing interest on Ensemble learning techniques which evolved from not being used between 2011-2013 to accounting for 14% of applications in 2018. This can possibly explain the loss of interest on DT since 2017, as Random forests (a type of Ensemble learning) can achieve better performance by using committees of decision trees.

As NN and Ensemble learning families seem to be recently attracting the research community, a detailed view of their encompassed techniques is presented in Appendix III. Concerning NN, the most used technique is the Multi-layer perceptron, which is the classic architecture of a NN. However, more specialized architectures belonging to deep learning are starting to appear in PPC research. Such is the case of the CNNs, LSTMs, and Deep Belief Networks. These techniques have presented good performance when dealing with specific problems, such as image recognition for CNNs, time series analysis for LSTMs or feature extraction for Deep Belief Networks. In the case of Ensemble learning, the most used technique is, by far, the Random forests. They seem to provide excellent results while enabling knowledge generation. In fact, they allow the most meaningful variables to be easily identified in the SL task, which is the reason why researchers tend to use them to both attain accuracy and model interpretability.

To measure the utilization of the learning types, each paper was analyzed, and the learning types used were identified and counted. As a given model can use several ML techniques, it can refer to several learning types at the same time. Hence, the different synergies between these were also considered. Results are presented in Figure 4.8.

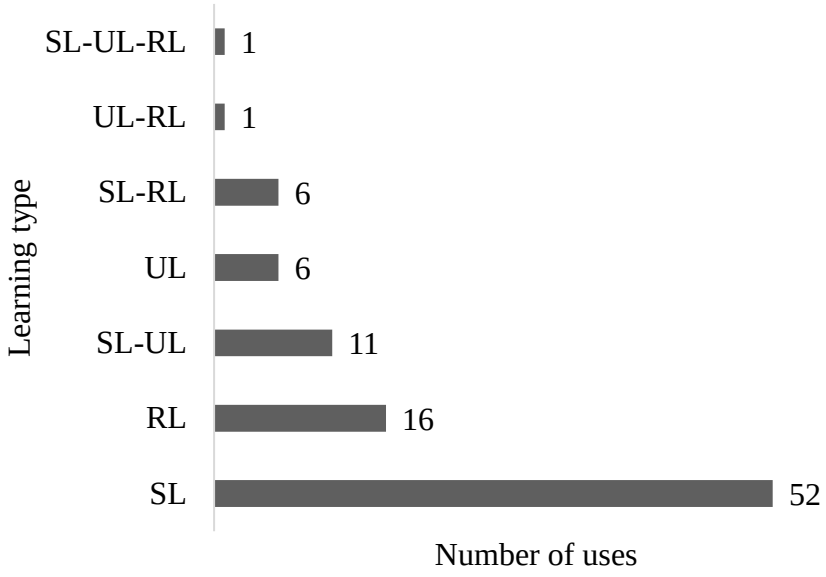


Figure 4.8 Number of uses by learning type.

Findings show that the most used learning type is SL. This is probably because SL addresses two recurrent needs in applied research: classification and regression. In fact, SL can be used

to learn the relationship between an input X and an output Y that can be either discrete in the case of classification or continuous for regression. Furthermore, it was found that RL techniques are extensively used, which confirms the interest behind agent-based models.

Concerning UL, it seems to be especially used with SL (SL-UL), which suggests a strong synergy between these two learning types. The reason behind this could be that UL techniques are normally used to perform data pre-processing, as with Principal Component Analysis, or discovery of hidden patterns in datasets, e.g. with Clustering. There are 6 papers using just UL, however, this learning type seems to unlock all of its potential when used in synergies, allowing for the design of more complex models.

Even if there are some SL-RL synergies, they are not very common. This is probably because SL is normally coupled with RL when there is a need of performing rapid estimations of functions to save computing time. However, it seems that most of the applications do not reach a scale that needs this kind of configuration. Finally, it was found that using UL-RL and SL-UL-RL is rare in the scientific literature. This does not mean that their synergy does not provide advantages, it is just that there may not be a current need for it. Also, it could be that coupling these learning types over-complexifies the model design, which prevents its use.

Concerning the tools, only programming languages or software used to implement the ML model were considered. Therefore, other tools such as discrete event simulation software are out of the scope of this research. Results are presented in Figure 4.9.

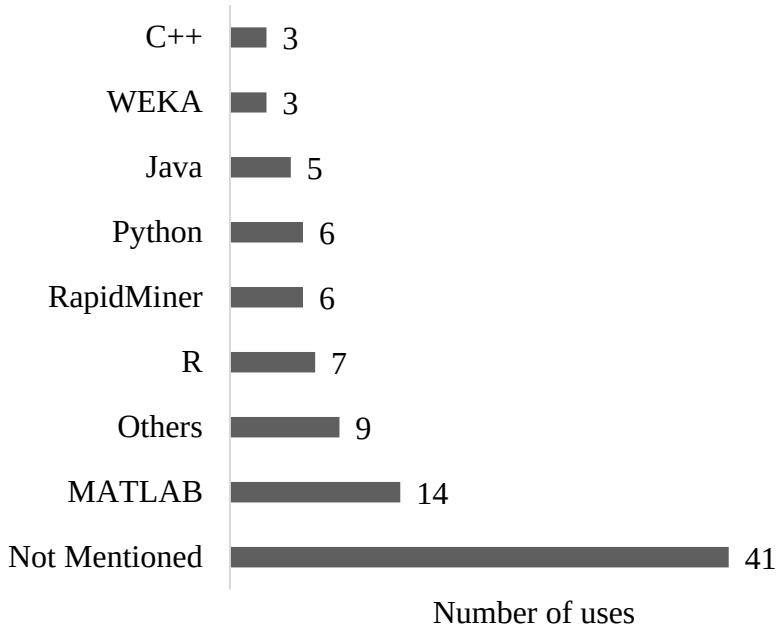


Figure 4.9 Number of uses by tool.

For clarity sake, tools being used only once were grouped in the category denominated as "Others". These tools were: ACE Datamining System, C#, Clementine, GeNIe Modeler, Hugin 8.1, NetLogo, Neural-SIM, Visual C++, and Xelopes Library. Additionally, it is important to mention that most of the researchers do not mention the tool the use to implement the models.

It could be said that MATLAB is, by far, the most used tool to perform ML-PPC in research. Besides its robust calculation capacity, the reason behind this could be that universities often invest in licenses for this software; therefore, they expect their researchers to use this tool. R is the second most used tool, which may be because it is a free software targeting statistical applications, including ML. Finally, the third most used tools are both RapidMiner and Python. The former eases the implementation of ML models thanks to its visual programming logic, while the latter is a multipurpose programming language recently characterized by its ML libraries and frameworks such as Scikit-learn, PyToch, Keras, etc.

4.4.3. Third research question: used data sources to implement a ML-PPC

To answer this question, the data sources used by each of the analyzed papers were identified. These results are summarized in Table 4.2. The column "Identification" (ID) will assign a number to each communication. This will be used later to establish a mapping of the scientific literature.

ID	Reference	M	E	U	P	Pb	A	ID	Reference	M	E	U	P	Pb	A
1	(Aissani et al., 2012)						X	35	(Lai and Liu, 2012)	X					
2	(Altaf et al., 2018)				X			36	(Lai et al., 2018)	X					
3	(Bergmann et al., 2016)						X	37	(Leng et al., 2018)	X			X		
4	(Cai et al., 2016)		X	X				38	(Li et al., 2012a)				X		
5	(Cao et al., 2019)		X					39	(Li et al., 2012b)						X
6	(Carvajal Soto et al., 2019)					X		40	(Li et al., 2013)						X
7	(Chen et al., 2015)						X	41	(Li et al., 2018)		X				
8	(Diaz-Rozo et al., 2017)		X					42	(Liao, 2018)						X
9	(Ding and Jiang, 2018)	X	X		X			43	(Lieber et al., 2013)		X		X		
10	(Dinis et al., 2019)	X		X				44	(Lingitz et al., 2018)	X					
11	(Dolgui et al., 2018)	X						45	(Lubosch et al., 2018)					X	X
12	(Doltsinis et al., 2014)		X					46	(Lv et al., 2018a)	X					
13	(Fotuhi et al., 2013)					X	X	47	(Lv et al., 2018b)	X				X	X
14	(Gao et al., 2014)						X	48	(Ma et al., 2017)	X					X
15	(Gyulai et al., 2014)	X					X	49	(Maghrebi et al., 2016)	X					
16	(Gyulai et al., 2015)	X	X				X	50	(Manns et al., 2015)	X			X		
17	(Gyulai et al., 2018b)	X					X	51	(Manupati et al., 2013)						X
18	(Gyulai et al., 2018a)	X			X			52	(Mori and Mahalec, 2015)	X					
19	(Habib Zahmani and Atmani, 2018)						X	53	(Ou et al., 2018)	X					X
20	(Hammami et al., 2016)						X	54	(Ou et al., 2019)						X
21	(Heger et al., 2016)						X	55	(Palombarini and Martínez, 2012)					X	
22	(Huang et al., 2019)		X		X		X	56	(Priore et al., 2018)					X	X
23	(Ji and Wang, 2017)						X	57	(Qu et al., 2015)						X
24	(Jiang et al., 2016)	X					X	58	(Qu et al., 2016b)						X
25	(Jurkovic et al., 2018)		X					59	(Qu et al., 2016a)						X
26	(Kartal et al., 2016)	X						60	(Reboiro-Jato et al., 2011)	X					
27	(Khader and Yoon, 2018)		X					61	(Reuter et al., 2016)	X					
28	(Kho et al., 2018)		X					62	(Rostami et al., 2018)		X				
29	(Kim and Lim, 2018)						X	63	(Sahebjamnia et al., 2016)	X					X
30	(Kim and Nembhard, 2013)					X		64	(Schuh et al., 2017b)	X					
31	(Kosmopoulos et al., 2012)		X					65	(Shahzad and Mebarki, 2012)						X
32	(Kretschmer et al., 2017)	X			X			66	(Shiue et al., 2012)					X	
33	(Kruger et al., 2011)		X					67	(Shiue et al., 2018)					X	X
34	(Kumar et al., 2018)					X		68	(Solti et al., 2018)				X		X

ID	Reference	M	E	U	P	Pb	A
69	(Stein et al., 2018)				X		
70	(Stricker et al., 2018)						X
71	(Thomas et al., 2018a)	X					
72	(Thomas et al., 2018b)						X
73	(Tian et al., 2013)					X	
74	(Tong et al., 2016)	X					
75	(Tuncel et al., 2014)					X	
76	(Wang and Jiang, 2018)		X				
77	(Wang and Jiang, 2019)		X		X		
78	(Wang and Yan, 2016)						X
79	(Wang et al., 2015)						X
80	(Wang et al., 2017)	X					
81	(Wang et al., 2018a)	X				X	
82	(Wang et al., 2018c)					X	
83	(Waschneck et al., 2018)						X
84	(Wauters et al., 2012)	X					
85	(Wu et al., 2015)	X					X
86	(Xanthopoulos et al., 2017)						X
87	(Yang et al., 2016)		X				
88	(Yeh et al., 2011)	X					
89	(Yuan et al., 2014)						X
90	(Zhang et al., 2011)	X					X
91	(Zhang et al., 2012)						X
92	(Zhong et al., 2014)		X				
93	(Zhou et al., 2018)		X				X
Totals		**M**	**E**	**U**	**P**	**Pb**	**A**
		33	20	2	12	14	43

Table 4.2 Data sources used by each of the analyzed scientific articles.

Results show that “Artificial data” is the most used data source in recent scientific literature. This probably highlights the difficulty of accessing data coming from companies. Additionally, it is important to remember the extensive use of RL techniques. These models normally require constant access to data concerning the real-time status of the production system, which can be difficult to find in real factories. Therefore, researchers normally use Artificial or Public data to test their models. This issue could be addressed by creating digital twins, but this still represents a research challenge.

The extensive use of artificial data suggests that there are data availability issues. This poses two main challenges: firstly, dealing with highly unbalanced datasets when training, for instance, SL algorithms for classification, and secondly, accessing enough data to enable good generalization capacity, especially in deep learning models.

The first challenge is common when training ML models to identify disruptions. In fact, disruptive events in PPC such as machine breakdowns or quality problems tend to be scarce when compared to the total size of the dataset. Thus, ML techniques struggle to learn these events. To tackle this issue, some authors have proposed solutions such as data augmentation, a common practice in computer vision that consists of artificially creating new training examples by modifying existent observations (Perez and Wang, 2017; Mikołajczyk and Grochowski, 2018). Another approach is to use crafted algorithms adapted to class-imbalance. Bi and Zhang (2018) performed a comprehensive comparison of state-of-the-art ML techniques adapted to this issue. The second challenge normally concerns the training of deep learning models as they need voluminous data to learn meaningful representations. This issue is normally tackled by transfer learning, which is the use of models already trained on a source task to perform another related task (Wang et al., 2018b), for instance, using a CNN trained to recognize pedestrians in the street to recognize operators on the shop floor. A comprehensive survey of transfer learning can be found in (Pan and Yang, 2010).

Management is the second most used data source. Hence, there seems to be a strong interest in valuing enterprise data stored in information systems by making it available for researchers and practitioners. Furthermore, the use of Equipment and Product data suggests that recent applications are starting to employ data coming from IoT technologies installed in machines or semi-finished products. However, there are still tremendous research gaps when harnessing user data to implement ML-PPC models. Two studies used this data source, but only under the form

of expert feedback to train the ML model. No study included consumer feedback from e-commerce platforms or social media to influence the PPC.

4.4.4. Fourth research question: addressed use cases by recent scientific literature

To answer this question, each analyzed article was allocated to one of the seven proposed types of use cases. This allows to measure their importance in the scientific literature (Figure 4.10).

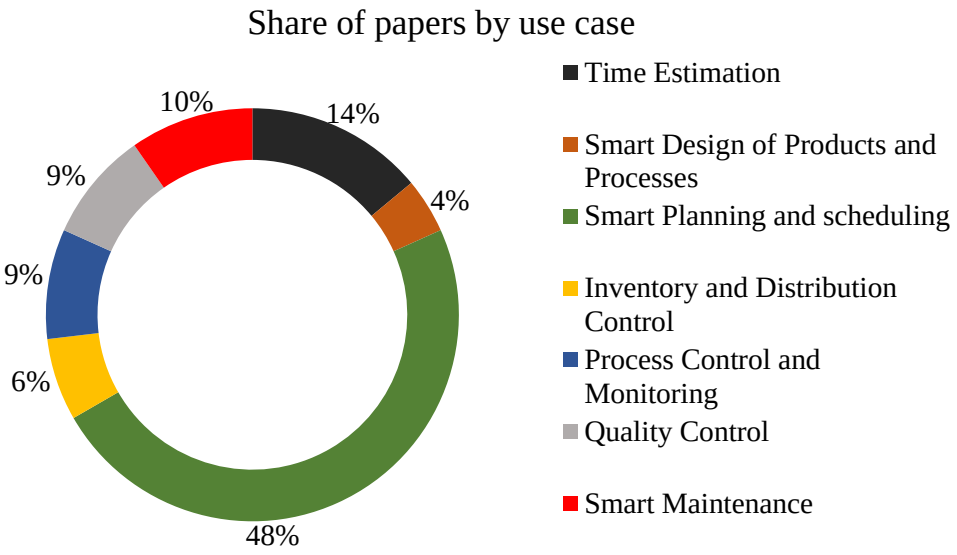


Figure 4.10 Share of the analyzed sample by proposed use case.

Results point out that Smart Planning and Scheduling is the most addressed use case in recent scientific literature, with nearly half of the communications discussing it. This result may come from two main reasons: firstly, the string chains used in the methodology are closely related to this use case; secondly, it normally uses structured data relatively easy to get from information systems, which eases the task of implementing a data-driven approach. The strong use of Time Estimation in ML-PPC (14% of the papers) suggests that classical time measurement methods are not compliant with the growing complexity of the manufacturing systems, which may represent a pitfall to perform a reliable planning. Therefore, ML models considering more diverse variables as inputs are being adopted. Moreover, some researchers have addressed the coupling of Smart Maintenance, Process Control and Monitoring, and Quality Control with the PPC. However, there is still effort to be made, as the share of these use cases was no higher than 10%.

Finally, two use cases are targeted as critical: The Inventory and Distribution Control (6%) and the Smart Design of Products and Processes (4%). These findings suggest two things: first, a lack of integration of the logistic functions into the ML-PPC, and secondly, a difficulty for harnessing insights from data to serve product and process design. This difficulty is probably because data employed in design is highly unstructured (text data, image data, etc.) and greatly depends on people's experience.

4.4.5. Fifth research question: the characteristics of I4.0

To quantify their usage, the addressed characteristics in each of the 93 analyzed papers were identified and counted. Results are summarized in Figure 4.11. In this figure, the sum of all the totals is higher than 93 as one ML-PPC model can satisfy several characteristics.

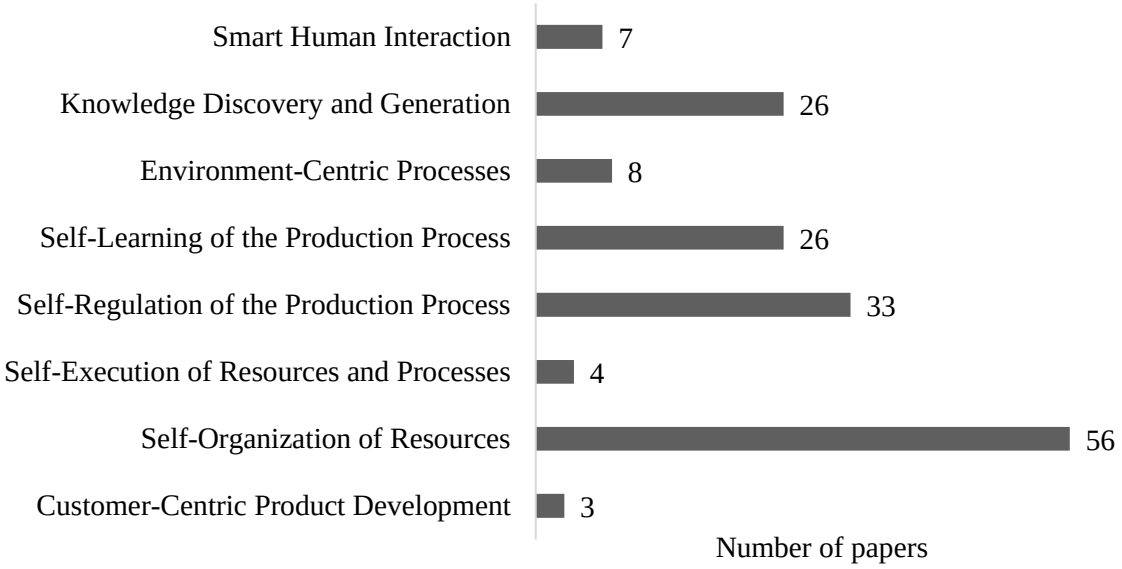


Figure 4.11 Number of papers by I4.0 characteristic.

Findings show that the Self-Organization of Resources is, by far, the most addressed characteristic (56 uses) in ML-PPC applications. This result was expected, as this characteristic can be achieved through production planning and scheduling, two functions directly related to the PPC and found to be extensively employed in the use cases. Therefore, it can be concluded that the ML-PPC based models effectively enable this characteristic.

The Self-Regulation of the Production Process (33 papers), the Self-Learning of the Production Process (26 papers), as well as the Knowledge Discovery and Generation (26 papers) appear to be moderately boarded. This leads to two main conclusions: first, ML-PPC models effectively

endow manufacturing systems with the capacity of adapting to unexpected events and predicting production problems. This is suitable to handle the stochastic nature of production environments. Secondly, ML is suitable to generate knowledge from PPC data, which is crucial in I4.0, where data is abundant, and it can provide useful guidelines to improve the company's know-how.

Four characteristics were rarely satisfied: The Customer-Centric Product Development (3 papers), the Self-Execution of Resources and Processes (4 papers), the Smart Human Interaction (7 papers), and the Environment-Centric Processes (8 papers), which points to strong research perspectives of ML-PPC applications enabling these features. Concerning the Customer-Centric Product Development, it was rare to find papers including customer-related variables into their PPC. This can be due to the difficulty to access data from customers or end users. For instance, as observed in the data sources section, user data was seldom employed.

The low number of papers dealing with Self-Execution of Resources and Processes suggests that it is unusual to couple the PPC with autonomous physical resources. This can be due to the complexity of such systems as they require important capital investments as well as multi-disciplinary knowledge in production systems, mechatronics, and control theory.

It was very surprising to find that the Smart Human Interaction (7 papers) and the Environment-Centric Processes (8 papers) are rarely addressed. Indeed, manufacturing systems can be human based in several steps such as during the execution in the shop floor or during the tactical planning definition. Not considering the interaction of the proposed ML-PPC models with humans can be harmful for the deployment of the proposed system, as it may worsen the working conditions. Therefore, thinking about this human-ML interaction is the cornerstone for a successful adoption. Concerning the Environment-Centric Processes, scarce applications tried to minimize the environmental impact of production processes through ML-PPC. In a world where natural resources are becoming rare, this is a non-negligible aspect that must be considered, not only because of the tightening of environmental laws by governments but also because of the ethical responsibility of companies.

4.4.6. Cross-axes analysis: mapping the scientific literature through use cases, I4.0 characteristics, and learning types

To address the second objective of this study, a mapping of the scientific literature in ML-PPC is proposed. This is achieved through a cross-analysis employing the use cases, characteristics of I4.0, and learning types. Results are represented via a cross-matrix having the use cases in the vertical axis and the characteristics of I4.0 in the horizontal axis. This matrix also allows the maturity of a given use case to be assessed. For instance, a mature use case in the scientific literature will tend to satisfy more I4.0 characteristics. From this point of view, the crossing between a characteristic of I4.0 and a use case will be referred as a *domain*.

The ID numbers defined on Table 4.2 are employed to place the analyzed articles in the matrix. Additionally, the learning types employed by each communication are represented using a color code. Figure 4.12 provides a summarized view of this matrix, allowing for a high-level analysis that will help to identify research gaps and trends in ML-PPC. Figure 4.13 is a detailed view of the matrix indicating the scientific articles with their respective learning types found in each domain.

Figure 4.12 shows that among the 56 possible domains, 18 (32%) were not addressed at all. Furthermore, 24 (43%) domains lie in the range of 1 to 3 papers. This means that nearly half of the domains are in an exploration phase. These two remarks lead to conclude that ML-PPC in the I4.0 is still an active research topic with strong perspectives.

From Figure 4.13, it can be said that there is a strong trend of using multiple synergies between learning types across all of the different use cases. However, there are no applications of RL in Time Estimation and in Smart Design of Products and Processes. The reason for this may be that these use cases have strong strategic impacts. Therefore, current ML implementations in such applications aim to support decisions rather than automating them such as with agent-based systems driven by RL.

There are two use cases achieving a high maturity: Smart Planning and Scheduling and Process Control and Monitoring. They both cover all but one of the characteristics of I4.0. In the case of Smart Planning and Scheduling, it fails to address the Self-Execution of Resources and Processes, which suggests that there are research perspectives in coupling the production planning and scheduling with autonomous physical resources. For the Process Control and

Monitoring, there is a lack of applications satisfying the Customer-Centric Product Development, which would be an automatic optimization of physical resources from the analysis of customer-related variables.

Knowledge Discovery and Generation is the only characteristic addressed by all the use cases, which denotes an intense interest in knowledge creation from data. Furthermore, there is a strong presence of SL, UL, and SL-UL in this characteristic. This implies an important affinity between these learning types and the generation of useful information from raw data. Following a similar trend, there seems to be a generalized interest in Environment-Centric Processes, a characteristic that is addressed by almost all of the use cases. However, its low number of papers implies that there are strong research avenues to be explored.

Communications addressing the Self-Execution of Resources and Processes focused exclusively on Process Control and Monitoring applications. This shows that the dynamic optimization of working parameters of the machines allows data-driven intelligent resources to be created. However, this characteristic has further potential to be explored in PPC research with other use cases, such as in Inventory and Distribution Control with autonomous AGVs to serve logistic needs or in quality, by automating processes.

Figure 4.12 Summarized view of the cross-matrix: number of papers by domain.

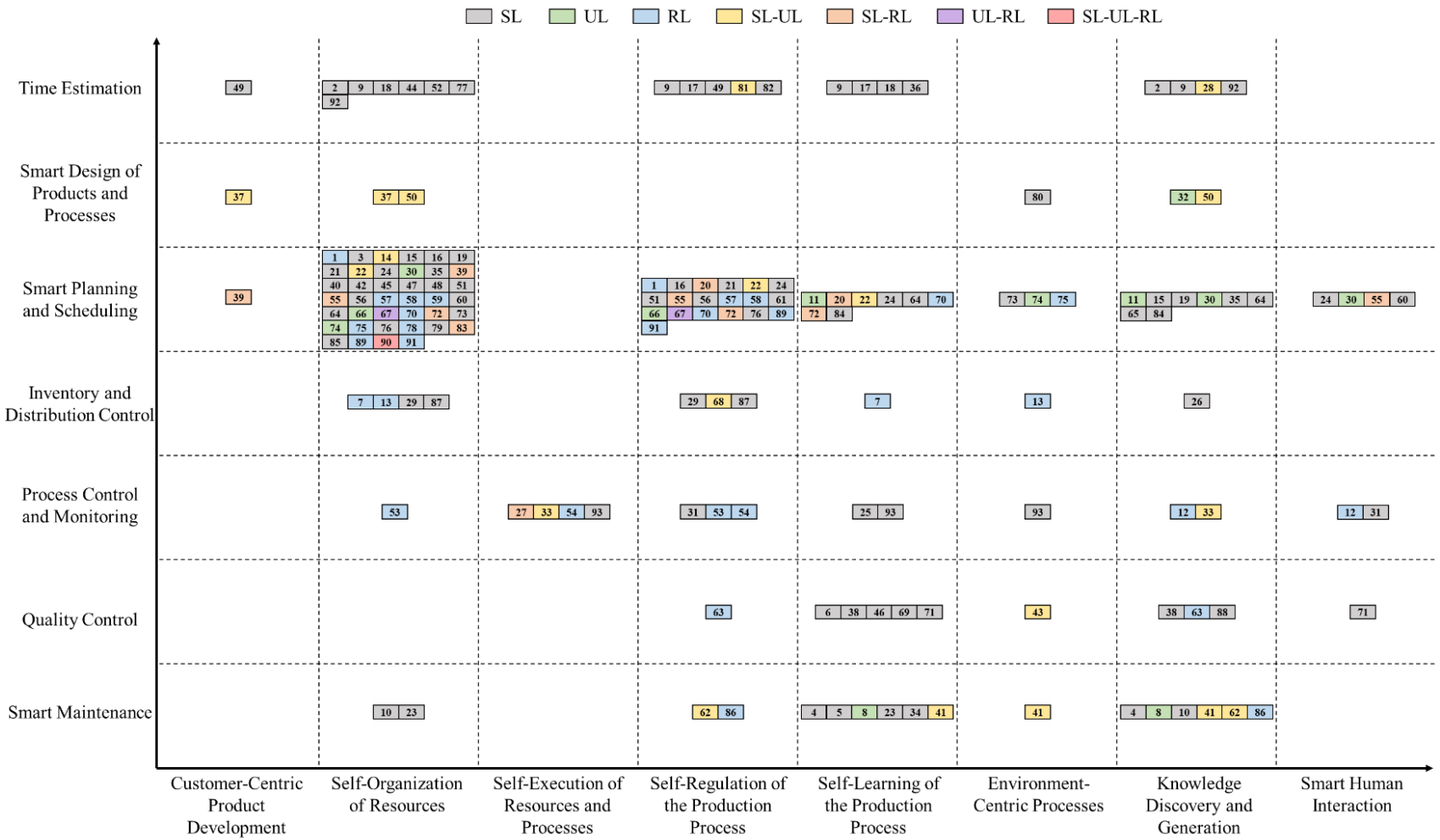


Figure 4.13 Detailed view of the cross-matrix for use cases, characteristics of I4.0, and learning types.

4.5. Conclusion and further research perspectives

This state-of-the-art analysis studied 93 research articles chosen through the logic of a systematic literature review. These papers were analyzed by means of an analytical framework composed of four axes. First, the elements of a method were reviewed, which enabled an analysis of activities, techniques, and tools to perform a ML-PPC. Secondly, the data sources employed to implement a ML-PPC model were recognized and assessed. Thirdly, an analysis of the use cases enabled the recognition of the applications of data-driven models in the 4.0. Fourthly, the characteristics of I4.0 were identified and assessed through their usage. Additionally, a mapping of the scientific literature was proposed by means of the use cases, characteristics of I4.0 and ML learning types.

Results concerning the activities allowed the recognition of eleven recurrent tasks that are employed to create a ML-PPC model. They were grouped in four clusters following their use percentage: CUAs (Commonly Used Activities), OUAs (Often Used Activities), MUAs (Medium Use Activities), and SUAs (Seldom Use Activities). From these clusters, it can be concluded that activities belonging to the CUAs and OUAs clusters are well documented in the scientific literature. MUAs activities mainly contain data pre-processing tasks, which are necessary but not commonly documented by researchers. Finally, the SUAs cluster suggests that there are three activities rarely addressed in literature: the design and implementation of data acquisition methods from the manufacturing system, the exploration of data to get insights, and the constant adaptation of the proposed ML-PPC model to the environment dynamics.

An extensive review of the techniques identified the most used families in scientific literature. These were found to be the NN, Q-Learning, DT, Clustering, Regression, and Ensemble learning. From these results, a temporal evolution analysis of the top 6 most used families was performed. Findings suggested a growing interest in NN and Ensemble learning, which motivated a focused study on the detailed techniques encompassed by these families. Concerning the NN, the Multi-layer perceptron was the most used technique. Nevertheless, more specialized deep learning techniques such as CNNs, LSTMs, and Deep Belief Networks are starting to be employed. With respect to Ensemble learning, the most used technique was Random forests.

The ML learning types were also reviewed. Findings showed that scientific literature mainly focused on the individual use of SL and RL. However, synergies between learning types are also employed. For instance, the most used synergy was SL-UL, which allows to explore and pre-process the data through UL to improve the SL training. The UL-RL and SL-UL-RL synergies had only one use each, which could be considered as a research gap, advising improvements in its integration. In fact, each learning type has its advantages and limitations. Hence, it is important to explore more synergy possibilities, as they may help overcome individual limits.

Other than increasing data availability, one option to encourage the utilization of UL-RL and SL-UL-RL is to boost the development of specialized libraries to build complex models coupling several learning types. Examples of this are deep learning frameworks such as TensorFlow, Keras, PyTorch, etc. which have eased the implementation of deep learning applications. This has allowed researchers to spend more time on the addressed problem than on the coding stage.

Results concerning the tools showed that MATLAB, R, Python, and RapidMiner are the most used tools in developing ML-PPC models in research. However, most authors did not mention the tool used, which is a limit of this study. Furthermore, it is important to mention that these results come from a sample of scientific articles, meaning that results are mainly valid in an academic context. If there are practitioners willing to implement ML-PPC models in companies, other aspects need to be analyzed such as the cost of the software, its scalability, skill availability in the labor market, compatibility with existing information systems, etc.

The current horizon of data sources used is dominated by Artificial and Management data. The former points to a difficulty in collecting all of the data required to implement ML-PPC models, while the latter suggest that companies are interested in valuing their data stored in information systems. Data coming from IoT sources such as Equipment and Product data was moderately used, nevertheless showing an interest in these technologies to collect data. Finally, ML-PPC models failed to integrate User data, probably because it is complex to collect and it engages an important responsibility concerning data privacy.

The most addressed use cases were Smart Planning and Scheduling and Time Estimation, probably because they are directly concerned by the PPC, which may lead to its high utilization. The fact that there are research articles in all of the use cases suggests that the PPC is a

transversal function that benefits from several applications. Therefore, when designing a ML-PPC system for a company, the impact on all of the use cases must be assessed. Finally, it was found that Inventory and Distribution Control, as well as Smart Design of Products and Processes, are seldom addressed. This suggests that there is still a lot of progress to be made when coupling the PPC to logistics as well as product and process design through ML.

Concerning the characteristics of I4.0, results suggest that scientific literature in ML-PPC is extremely focused on satisfying the Self-Organization of Resources, which was expected, as one of the main goals of the PPC is resource management to satisfy the commercial plan. At a second level, the Self-Regulation of the Production Process, the Self-Learning of the Production Process, and the Knowledge Discovery and Generation seem to be more frequently addressed. However, Figure 4.13 showed that they are mainly employed for Smart Planning and Scheduling, implying a lack of research in the other applications. Finally, there are three characteristics that are partially overlooked by researchers: Environment-Centric Processes, Smart Human Interaction, and Customer-Centric Product Development. The first two are essential characteristics of building more responsible production systems as they aim to include human beings and reduce the environmental impact of manufacturing processes. The latter relates to the alignment of the PPC to the customer's needs. Hence, it appears that recent ML-PPC research ignores the influence of the customer in the manufacturing process.

As illustrated in the proposed cross-matrix, 75% of the possible research domains are barely addressed or were not explored at all. This means that the ML-PPC is still a key topic for the enablement of I4.0, which presents strong research avenues. The main future research perspectives could be summarized in the following three key items:

1) Reinforce the role of IoT in ML-PPC: this would allow an improvement to the data acquisition system's design and would provide a means to perform a model update to tackle the concept drift issue. To do so, the ML mindset and workflow should be shifted from a linear to a circular process, considering the need to constantly retrain through new data. This way of thinking would enable the identification, from an early development stage, of the retraining policy and the necessary variables that could be measured again at a sensitive cost. By defining these two aspects, the data acquisition system design will be less complex to conceive, as the needs will be clearer. This would avoid investment in sensors and resources and architecture that

would not be exploited. Concerning the retraining policy, a review in the context of PPC reporting common practices, advantages and pitfalls seems to be missing in the scientific literature.

2) Improve the integration between the PPC, logistics, and design: it was stated that the PPC benefits from different use cases. However, recent literature seems to overlook logistics as well as product and process design applications coupled with the PPC. To tackle this challenge, it is necessary enable data availability, continuity and sharing over the design, logistics, and production departments. This could be achieved through interoperability as well as communication of intra-organizational systems such as the PLM, ERP, and MES. Even if projects that are meant to couple such systems are costly, they are necessary to ensure data availability and quality. One way to achieve this is the use of data lakes, which have been recognized as suitable to handle big data repositories of a structured and unstructured nature (Llave, 2018; Lo Giudice et al., 2019). For instance, Llave (2018) concluded, through expert interviews, that one of the key purposes of data lakes is to serve as experimentation platforms for data scientists.
3) Set human interaction and environmental aspect as priorities to ensure the development of ethical manufacturing in I4.0: exploring the interaction of humans with the proposed ML-PPC models is paramount to building inclusive technologies at the service of society. To achieve this, the short- and long-term impact of ML-PPC systems on employees' working conditions must be assessed. If the system degrades them, it must be redesigned. Concerning the second aspect, seeking a reduction in the environmental impact of manufacturing through ML could provide important developments. This can be addressed from a purely PPC approach by optimizing, for instance, the scheduling of disassembly processes or by improving the prediction of production times to avoid energy waste. Other approaches could be the optimization of the supply chain. Even though the supply chain was not covered in this review, it is an appropriate domain for researchers to implement ML applications. For instance, by considering environmental criteria when choosing suppliers, as in (Hosseini and Barker, 2016).

Some of the research gaps indicated in this review could motivate future work. Future work will be focused on the following aspects:

1) The proposed activities will be reviewed to determine an order between them, creating a procedure: this would help shift from a linear to a circular workflow when implementing ML-PPC models.
2) The most suitable techniques and tools will be linked to each of the activities with sectorial information: linking techniques, tools, and activities is the key to creating good practices that could be helpful to new practitioners, both in research and industry. Furthermore, according to Kusiak (2017, 2019), there are profound differences in the volume of data generation and usage across different industries. Therefore, future work will aim to identify trends categorized by sectorial information.
3) The current state of data availability solutions and workarounds will be explored: as data availability was found to be a main issue, a review of techniques to tackle the class-imbalance problem and the use of transfer learning in the context of PPC will be performed. Additionally, the utilization of data lakes for ML-PPC will also be explored.
4) Future research avenues will be proposed through an NLP analysis: NLP may enable the discovery of non-trivial trends present in the corpus of the 93 sampled articles. This will complement the results of the systematic literature review.

Acknowledgements

This work was financially supported by a partnership between the company iFAKT France SAS and the ANRT (Association Nationale de la Recherche et de la Technologie) under the grant 2018/1266. Furthermore, the authors thank the Editor-in-chief and three anonymous referees who helped improve the quality of this paper through their comments and suggestions.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix

Appendix I: Detail on the search strategy for the bibliometric analysis

See Figure 4.14,Figure 4.15, and Figure 4.16.

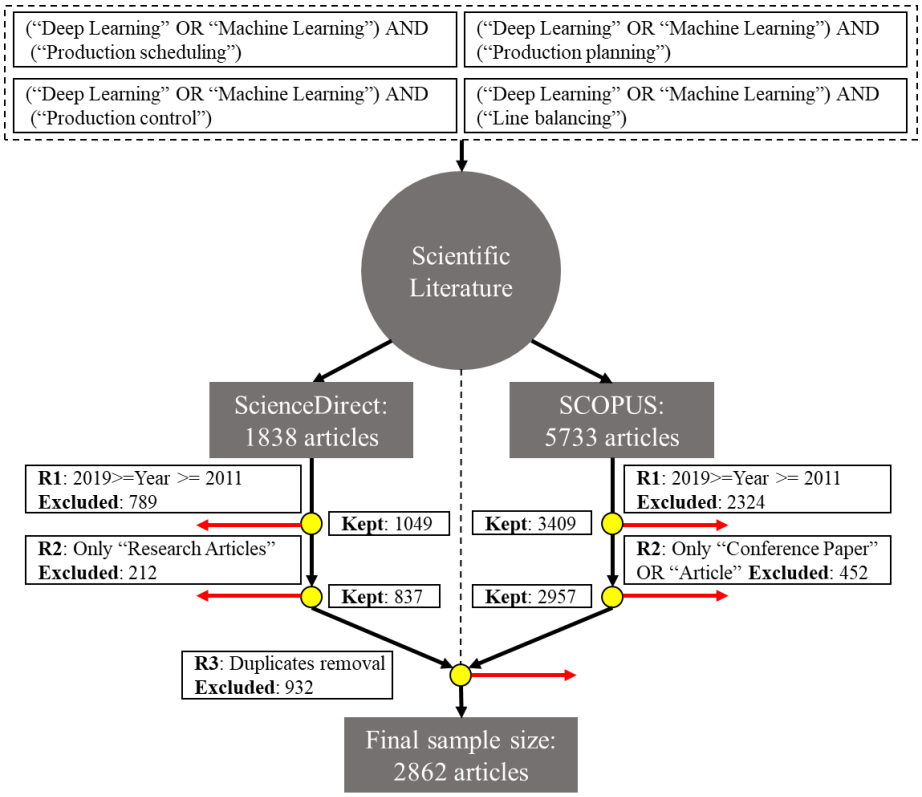


Figure 4.14 Search strategy detail for "Deep Learning" OR "Machine Learning."

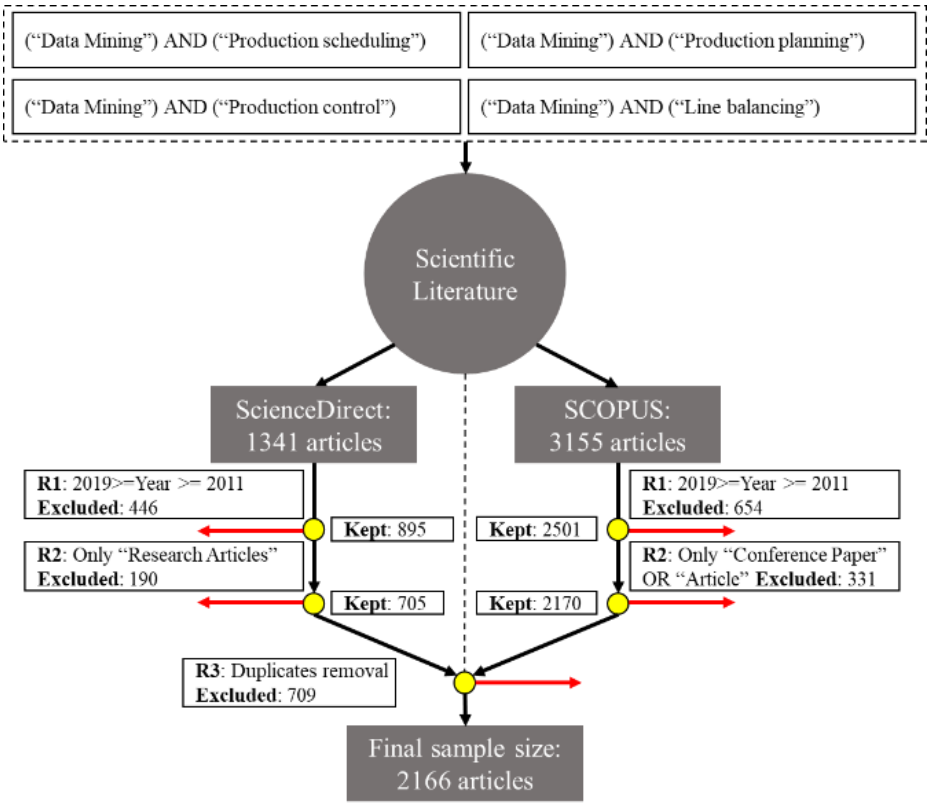


Figure 4.15 Search strategy detail for "Data Mining."

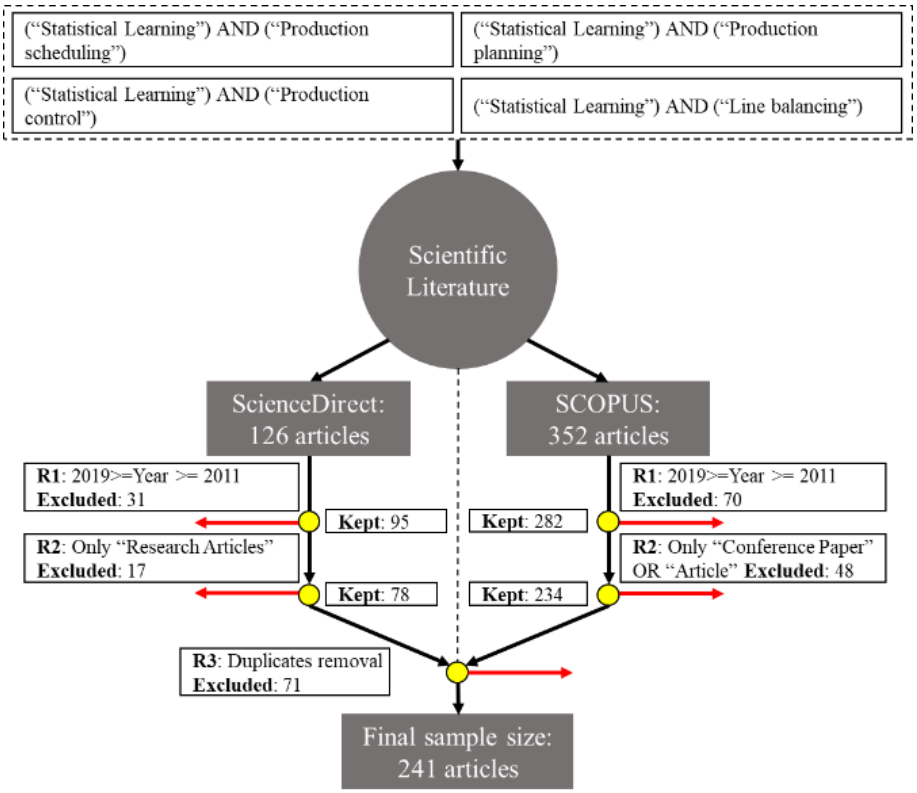


Figure 4.16 Search strategy detail for "Statistical Learning."

Appendix II. usage evolution of the top 6 most used techniques

See Figure 4.17, Figure 4.18, Figure 4.19, Figure 4.20, Figure 4.21, and Figure 4.22.

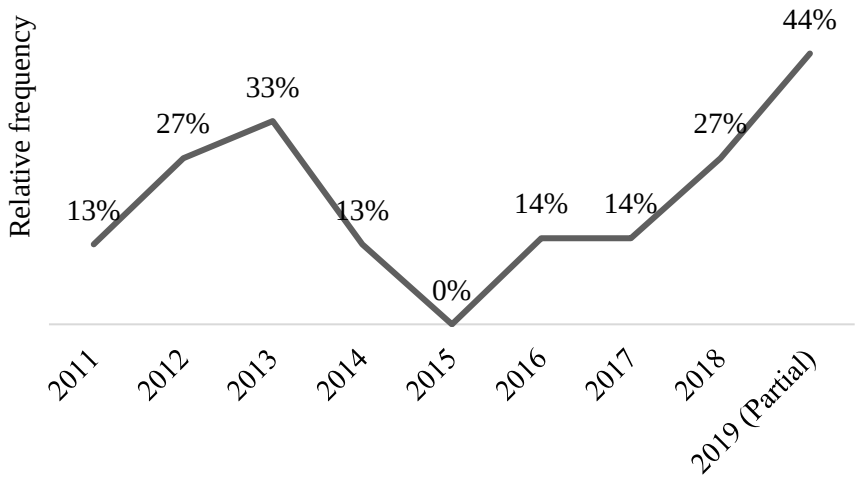


Figure 4.17 Usage evolution for Neural Networks.

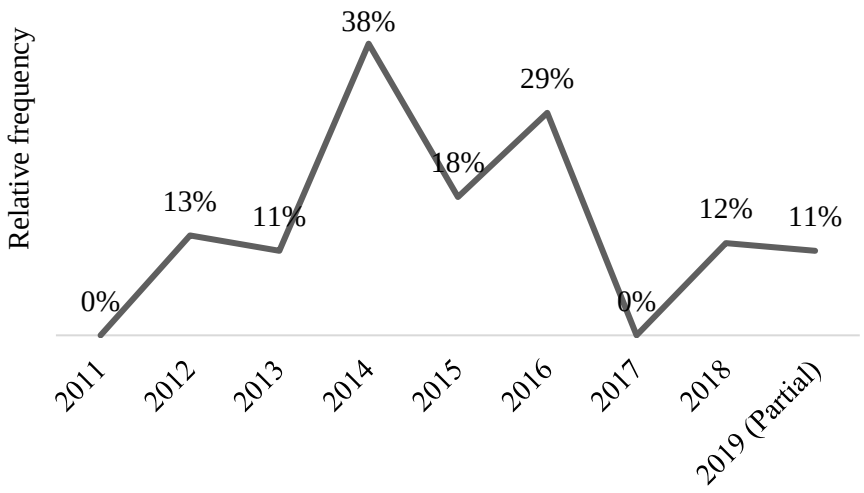


Figure 4.18 Usage evolution for Q-Learning.

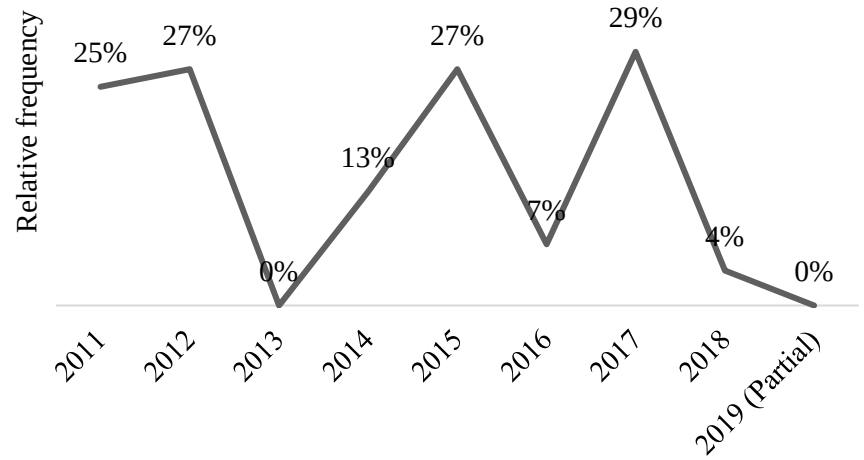


Figure 4.19 Usage evolution for Decision Trees.

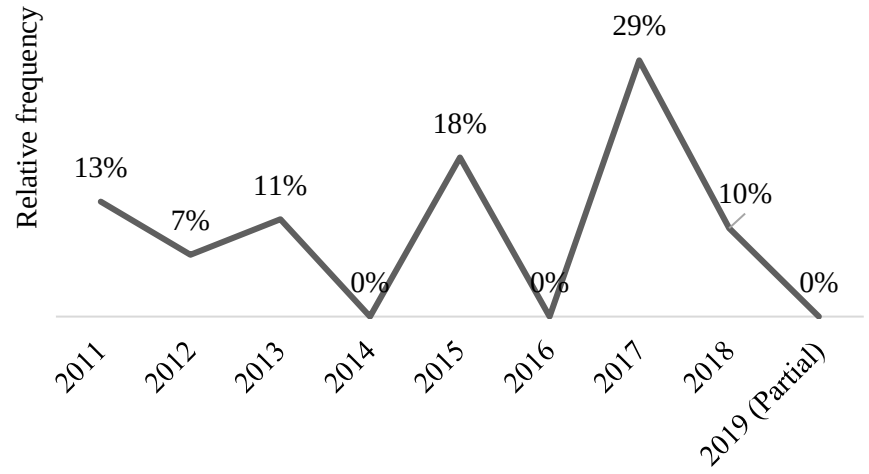


Figure 4.20 Usage evolution for Clustering.

Figure 4.21 Usage evolution for Regression.

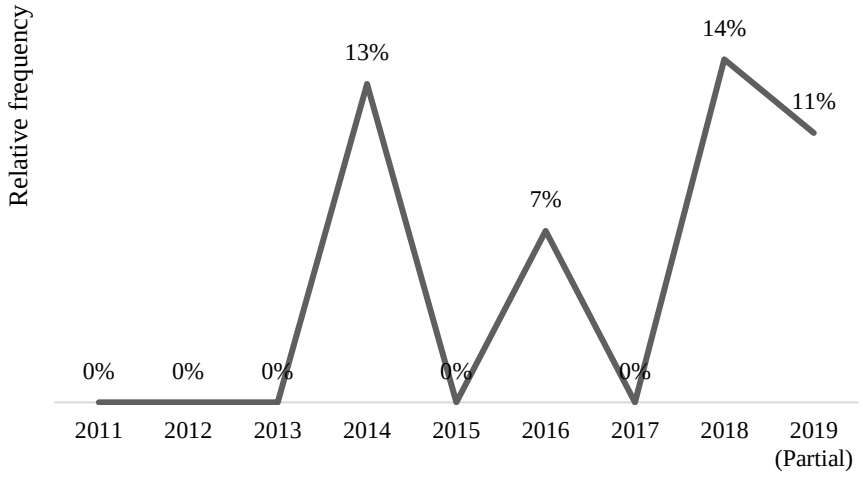


Figure 4.22 Usage evolution for Ensemble Learning.

Appendix III: detail on NN and Ensemble learning techniques

See Figure 4.23.

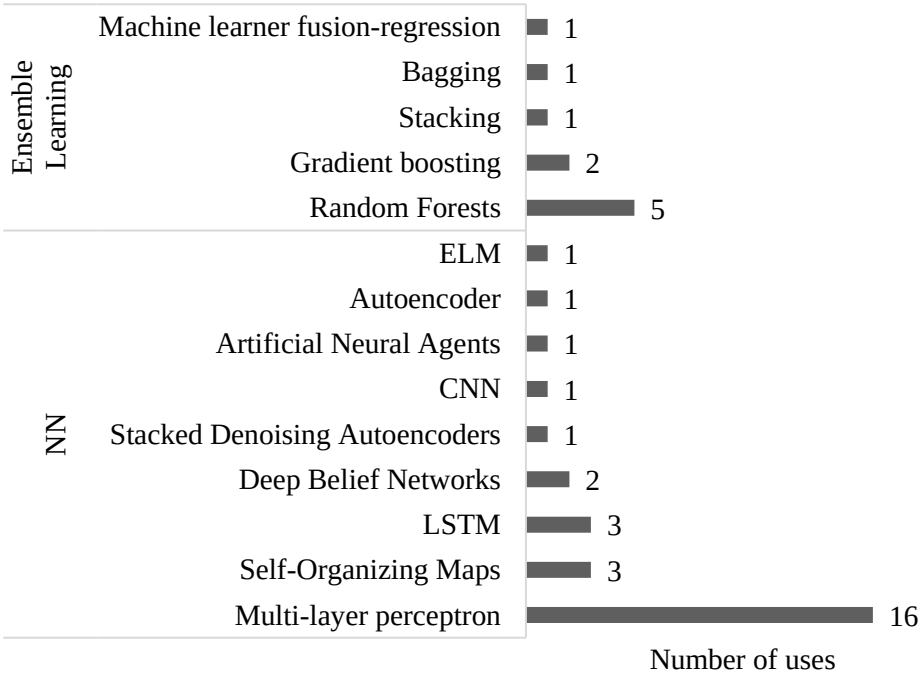


Figure 4.23 Detail on the techniques of the NN and Ensemble learning families.

5. Chapter 5: Article 2 - Valuing free-form text data from maintenance logs through transfer learning with CamemBERT

Name of the journal: Enterprise Information Systems

Received: 27 February 2020

Accepted: 28 June 2020

Authors: Juan Pablo Usuga Cadavid, Bernard Grabot, Samir Lamouri, Robert Pellerin

and Arnaud Fortin

Corresponding author: Juan Pablo Usuga Cadavid

DOI: https://doi.org/10.1080/17517575.2020.1790043

Abstract: Coupling a production scheduling process with maintenance logs can provide important advantages. For instance, this enables the adaptation of planning to the reality of the shop floor. Nevertheless, maintenance logs are often highly unstructured, as they mainly rely on free-form text comments from operators, and are imbalanced, as commonplace issues happen more often than critical problems. This hinders the application of machine learning methods to exploit this data. Thus, this study explores the use of a recent model named CamemBERT to tackle these difficulties through transfer learning. More specifically, the purpose is to predict the criticality and duration of a maintenance issue from the description provided. Findings suggest that fine-tuning CamemBERT outperforms other classical and feature-based approaches. Furthermore, the class imbalance problem is addressed from a data pre-processing and training perspective: firstly, k-means with silhouette diagrams allowed the creation of more homogenous classes, and secondly, the use of resampling enabled an improvement in the model's performance.

Keywords: transfer learning; deep learning; maintenance; industry 4.0; natural language processing; class imbalance

5.1. Introduction

In the context of Industry 4.0 (I4.0), Production Planning and Control (PPC) aims to determine the quantities that are to be produced in order to satisfy a sales plan and meet required performance objectives. Additionally, it encompasses the real-time synchronization of resources through process control and product customization (Usuga Cadavid et al., 2020a). Because of its transversal nature, PPC is related to functions such as scheduling, maintenance,

quality, product and process design, and logistics, among others. This suggests that improvements made on these functions may also lead to better PPC (Usuga Cadavid et al., 2020a).

Production scheduling, one of the functions of PPC, is ideal for meeting delivery dates and optimizing the use of available production capacity. From this function, companies influence four clusters of decisions: batching, resource allocation, sequencing, and timing. Batching defines the number and size of production lots, resource allocation is related to the association of tasks with production resources, sequencing determines the production order of the lots, and timing provides start and finish times for each operation (Muñoz and Capón-García, 2019). Nowadays, solvers can propose a feasible Production Schedule (PS) that respects constraints such as resource availability and operation sequencing. However, once the PS is in a shop floor, it is subject to production disturbances such as stochastic operation times, delivery delays, machine breakdowns, etc. Such problems may lead to an infeasible PS in the execution phase (Gyulai et al., 2015).

Poor adherence to the PS may affect the fulfilment of engaged delivery dates, harming a company's competitive advantage. In fact, meeting delivery dates is a key factor in gaining an advantage over competitors, as it helps to reduce production costs by lowering stocks and making better use of production capacity (Reuter et al., 2016; Schuh et al., 2017a). Nevertheless, even if companies in high-wage countries consider fulfilling delivery dates to be their main logistical target, they struggle to achieve this as a result of production disturbances (Reuter et al., 2016).

A sensible solution that could improve adherence to a PS is the creation of a Dynamic Production Schedule (DPS) that can adapt to disturbances through rescheduling. To achieve such a DPS, companies may benefit from I4.0 technologies, as they have proven to increase both efficiency and customer satisfaction (Wang et al., 2018b). Ruessmann et al. (2015) identified nine groups of technologies enabling the I4.0, among which Big Data Analytics (BDA) proposes tools and techniques to exploit large amounts of data. Additionally, as computerisation has enabled the storage of industrial datasets through the years (Grabot, 2020), BDA may provide meaningful advantages to improve the performance of production systems and achieve the development of a DPS.

Machine Learning (ML) is one of the research fields encompassed by BDA. It has been widely applied in PPC (Usuga Cadavid et al., 2020a) and it refers to computer programs that are able to learn from data to improve their performance in a given task (Mitchell, 1997). Nevertheless, training ML algorithms from scratch to solve complex tasks may require large amounts of data as well as high computing power and time, which can be prohibitive for some companies. In fact, the maturity levels across different industries regarding the data generation is rather heterogeneous: while some industries, such as the semiconductor industry, are able to generate tremendous amounts of data in relatively small lapses of time, other industries struggle to achieve the same volume and quality (Kusiak, 2019). To tackle this issue, Transfer Learning (TL) can be employed. This is inspired by the fact that human beings can apply knowledge that was learnt from previous experiences to effectively solve new problems (Pan and Yang, 2010). Hence, the idea behind TL is to adapt ML models that have been pre-trained on external datasets to a new task or domain, which reduces the need to have access to a large in-house training set.

Having introduced the potential benefits of using TL when developing a DPS with limited access to data, it is important to mention that this research aims to employ language-specific state-of-the-art models from the Natural Language Processing (NLP) domain to exploit maintenance logs. More specifically, the purpose is to predict the criticality and duration of a maintenance problem from the free-form text description provided by an operator. By free-form text, this paper refers to unstructured text entries in which users can provide any kind of annotation. Hence, this study has two main research objectives:

1) Compare the performance of a classical NLP approach with a recent deep learning model when trained and adapted to a highly imbalanced maintenance dataset;
2) Assess the pertinence of using recent NLP models to support a DPS.

The first research objective seeks to evaluate the use of TL through a recent French-specific NLP model named CamemBERT (Martin et al., 2019) when compared to a classical approach (i.e. Term Frequency-Inverse Document Frequency). Furthermore, both approaches are trained on a highly imbalanced dataset. Hence, the purpose is to modify the initial models through certain techniques that aim to tackle the class imbalance issue and assess their impact. The second research question will assess the capability of such a TL model to meet the needs of designing a DPS when working with purely free-form text data. To the best of the authors' knowledge, this is the first use and adaptation of CamemBERT to an industrial dataset

presenting class imbalance. Furthermore, it is worth noting that this research is the extension of a previous study in which authors validated the technical feasibility of NLP applied to maintenance logs (Usuga Cadavid et al., 2019).

The remainder of this paper is organized as follows: the section "Context and state of the art" will deepen the problem statement and it will provide a brief literature review of NLP models to generate word representations, as well as their applications. The "Materials and methods" section will describe the dataset used, the architectures tested, and training policy. The "Results" section will focus on the results of the tested architectures and an analysis of those results. The "Discussion" section will present the limitations of the study and directions for future work. Finally, the conclusion section provides a synthesis of the key points this study.

5.2. Context and state of the art

5.2.1. Problem statement: exploiting free-form text data from maintenance to develop a DPS

A DPS should consider predictable and unpredictable events. According to Tao et al. (2018), the capacity of adapting to unexpected events is called the self-regulation of processes, while the capacity of adapting to predicted events is self-adaptation. This research focuses on the self-regulation characteristic. Furthermore, authors such as Wang and Jiang (2018) have suggested that there are two kinds of unexpected disturbances: dominant and recessive disturbances. The former immediately cripples the production process (e.g. severe machine breakdowns, broken tools), while the latter introduces noise to the production system, preventing it from working to its nominal capacity (e.g. machine adjustments, failure and subsequent replacement of a secondary component). The concept of recessive disturbances used in this study slightly differs from the one used by Wang and Jiang (2018). They suggest that recessive disturbances always appear as a cumulative delay. However, the definition used in this research is wider: recessive disturbances are those that do not directly block the production process.

To successfully implement a DPS, two main questions arise when a disturbance occurs: whether it is dominant or recessive and how much workload will be needed for it to be solved. The first question refers to the criticality of the issue, while the second is related to the duration of the required maintenance intervention. Both questions must be answered to determine whether rescheduling is necessary. Figure 5.1 shows a simplified example of the function of a DPS: it

is a Gantt diagram showing the timing and resource allocation of seven operations ("Op.") on three machines. Arrows between operations indicate precedence constraints and the expected delivery date is shown with a dashed blue line at the end of the last operation ("Op. 7"). Thus, Figure 5.1a shows an initial PS where operation 6 is subject to an unexpected disturbance. Figure 5.1b and Figure 5.1c describe the case of a dominant disturbance: they show how the estimated delay is considered in the impacted operation to determine whether rescheduling is not necessary (Figure 5.1b), meaning that the delivery date remains unchanged; or, if rescheduling is needed (Figure 5.1c), the expected delivery date is changed. Finally, Figure 5.1d shows the case of a recessive disturbance. In such a case, only an ulterior maintenance intervention is needed.

To train ML models capable of determining the criticality and duration of a maintenance intervention, shop floor data is needed. However, most of the advances in I4.0 are limited by data in terms of variety, velocity, veracity, and volume (Zhou et al., 2017; Kusiak, 2019). For instance, not all companies can afford the infrastructure required by sensor-oriented data sources. Instead, management data coming from enterprise information systems has proven to be one of the most employed data sources in ML applied on PPC (Usuga Cadavid et al., 2020a), probably because it is easier to access, as data has already been collected through the years. Hence, to train the models proposed in this study, maintenance logs of a company were used.

Written descriptions of maintenance interventions may be a useful input to develop ML models. However, these are often in a free-text form, which means that they are highly unstructured. Thus, two operators reporting the same issue may provide different descriptions. Therefore, it is important to create models able to consider this variability. The next subsection reviews the use of word representations in NLP, which are the means employed to vectorize raw text into acceptable inputs by ML models.

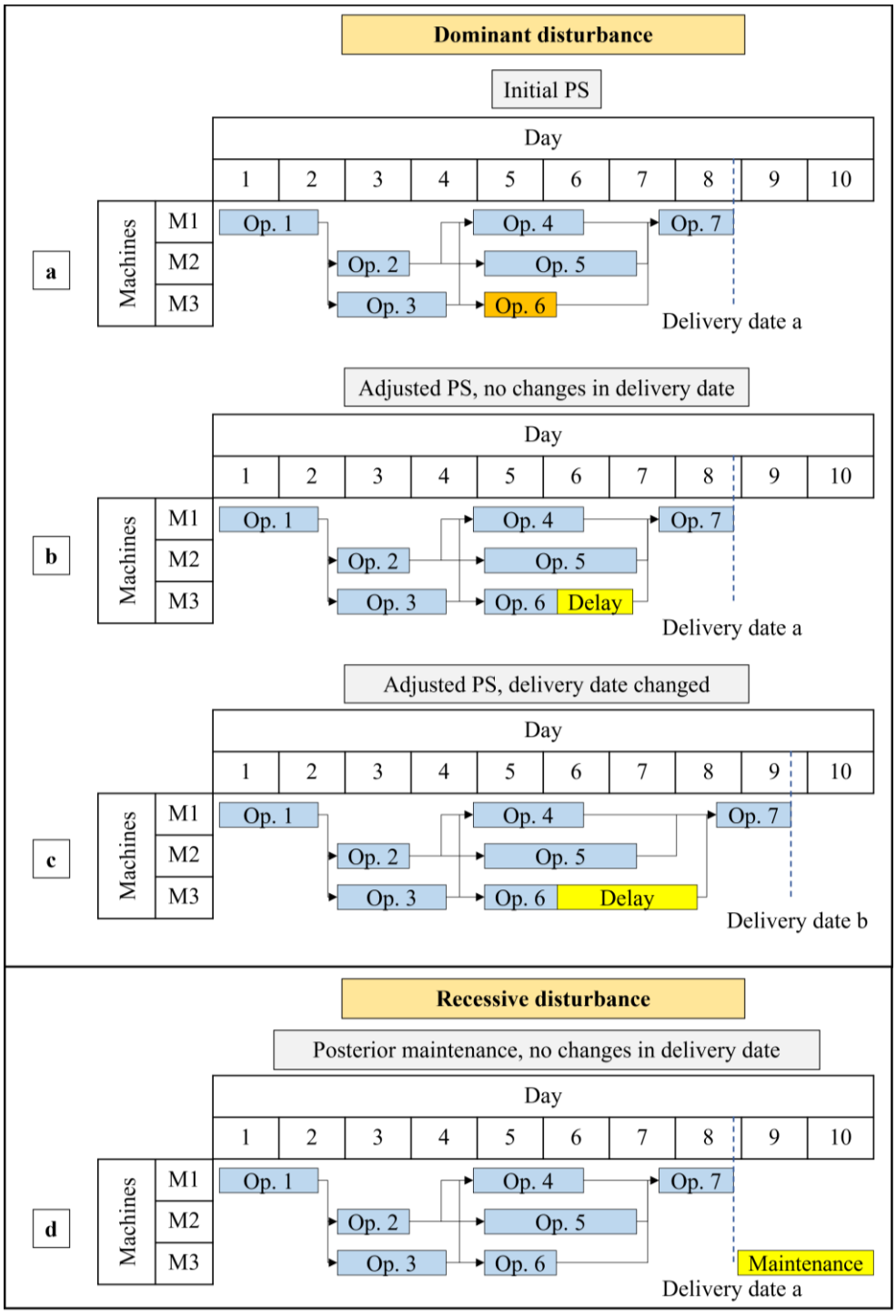


Figure 5.1 The functioning of a DPS. (a) Initial PS. (b) Dominant disturbance not affecting the delivery date. (c) Dominant disturbance affecting the delivery date. (d) Recessive disturbance not affecting the delivery date.

5.2.2. Word representations in NLP

To translate raw text inputs into representations exploitable by ML models, a vectorization must be performed. One of the classical NLP approaches to perform text vectorization is using Term Frequency-Inverse Document Frequency (TF-IDF), which consists of multiplying the frequency with which a term occurs in a phrase by a weight, in this case, determined through a measure of its rarity in the vocabulary (Karen, 1972). In such a way, trivial terms are valued less than rare words that may be semantically more important.

TF-IDF has been widely employed in research. For instance, in the field of recommender systems, Beel et al. (2016) analysed more than 200 research articles and found that TF-IDF was the most popular approach employed in Content-based filtering, accounting for 70% of the communications. Nevertheless, using TF-IDF has significant shortcomings. For example, it builds a fixed dictionary of words that represents the task-specific vocabulary. Thus, if the model receives a word that was misspelled or abbreviated, it will not recognize it. To consider new terms, the model needs to be constantly retrained on new data. Such a limitation hinders the successful application of TF-IDF on free-form text maintenance logs, as operators tend to write with several abbreviations and heterogeneous jargon, which affects the quality of the maintenance logs (Sexton et al., 2017). Furthermore, approaches such as TF-IDF are often accompanied by hand-crafted text pre-processing steps such as tokenization, stop-word removal, stemming, and n-gram conversion. This adds a non-negligible burden when creating a model, as the parameters ruling most of these steps must be determined through trial and error. For a detailed example of the use of TF-IDF with its pre-processing pipeline, please refer to a previous work, Usuga Cadavid et al. (2019).

In 2013, non-contextualized word embeddings, which are vector representations for words, became a research trend. Examples of non-contextualized word embedding generators are Google's Word2Vec (Mikolov et al., 2013), Stanford's GloVe (Pennington et al., 2014), and Facebook's FastText (Bojanowski et al., 2016). Using embeddings diminishes the burden of hand-crafted text pre-processing steps. For instance, FastText automatically generates subwords, which are segments of a full word. This can be used to segment misspelled words or abbreviations into known entities, reducing the number of unknown terms, especially when working with free-form text. Furthermore, generated word vectors keep their semantic meaning, suggesting that embeddings preserve a certain degree of language understanding. For example,

summing the vectors for the terms *Germany* and *Capital* results in a vector close to the word *Berlin* (Mikolov et al., 2013).

The main drawback of non-contextualized word embeddings is that they do not consider that the context of the phrase determines the vector representation for the word. Thus, the polysemy of words is ignored. This means that a term such as *bar* will have the same vector representation, regardless of whether the phrase is "*to replace the damaged steel bar"* or "*meeting my colleagues at the bar"*. Hence, research effort has been performed to create contextualized word embeddings using recurrent neural networks (McCann et al., 2017; Peters et al., 2017) and attention mechanisms (Vaswani et al., 2017; Devlin et al., 2018). These efforts have led to the creation of models such as ELMo (Peters et al., 2018), OpenAI GPT (Radford and Salimans, 2018), and BERT (Devlin et al., 2018), which are able to generate contextualized word embeddings. Notably, BERT significantly improved the state-of-the-art results in NLP. In fact, Devlin et al. (2018) used a novel training strategy and an architecture based on transformers, which rely on attention mechanisms (Vaswani et al., 2017). When compared to pure recurrent neural networks, attention mechanisms are more effective in learning long-term dependencies, a fruitful property when modelling languages. Additionally, the attention-based approach allows for the parallelization of computations, accelerating the learning process (Vaswani et al., 2017).

Since BERT was released, similar new models have been created, improving its initial performance. To get detailed insight on the latest developments, please refer to the *related work* section in Martin et al. (2019) and Le et al. (2019).

Despite the existence of a multilingual version of BERT, efforts to create models adapted to other languages have rarely reached the same level as their English counterparts (Martin et al., 2019). To illustrate this, the English model named RoBERTa (an improved version of BERT), was trained on a corpora summing around 160GB of uncompressed text (Liu et al., 2019), while the multilingual version of BERT for French only used 57GB (Martin et al., 2019). Therefore, a portion of recent research has focused on the creation of language-specific models.

As this study focuses on maintenance logs written in French, the model named CamemBERT will be employed. This model was inspired by RoBERTa's architecture and achieved state-of-the-art results through GPU training during 17h on 138GB of uncompressed text in French (Martin et al., 2019). Despite the existence of a more recent model for French, called FlauBERT

(published one month later), its performance seems to be just as competitive as CamemBERT and not necessarily superior (Le et al., 2019). Therefore, the choice was to address the research objectives of this study with CamemBERT and to leave a comparison of these two models applied in maintenance logs for future research.

According to Devlin et al. (2018), there are two approaches to perform TL through contextualized word embeddings: feature-based and fine-tuning. The feature-based approach seeks to use the pre-trained model as a feature extractor employed to vectorize the input text, to subsequently couple a task-specific architecture. This is the case of ELMo. The fine-tuning approach aims to reduce the burden of creating task-specific architectures, hence the model is adapted by training on the task of interest through the adjustment of all the pre-trained parameters. In such a way, the final model benefits from the previous knowledge to achieve better performance in a few epochs. OpenAI GPT and BERT-based architectures are examples of the fine-tuning approach.

Even though Devlin et al. (2018) recommend using BERT-based architectures in a fine-tuning mode, they have also performed tests using the feature-based approach. Results have suggested that, for some tasks, the feature-based mode may be competitive with the fine-tuning approach. Hence, this research will test both on CamemBERT to determine which performs better, as performance seems to be task specific. As far as the authors know, this is the first research that applies CamemBERT to an industrial application related to maintenance and PPC.

5.2.3. Recent applications of word representations in Industry 4.0

To assess the recent applications of word representations in the literature, a brief systematic literature analysis was performed following the method proposed by Tranfield et al. (2003). It has already been successfully applied by other authors to draw conclusions from the literature (Garengo et al., 2005; Moeuf et al., 2018; Usuga Cadavid et al., 2020a).

The queries were conducted on the 15th of January 2020 in the scientific database SCOPUS using the following string chain: ("*Deep contextualized embeddings*" OR "*word embedding*" OR "*Natural Language processing*") AND ("*Smart Manufacturing*" OR "*Industry 4.0*"). As this research is mainly focused on word embeddings, only papers published as of 2013 were considered, as this year corresponds to the introduction of Word2Vec by Mikolov et al. (2013). Finally, to primarily obtain case studies, literature reviews and surveys were removed by only

considering *Articles* OR *Conference papers*. Then, a title and abstract analysis, as well as a full-text study, allowed for communications that were far from the topic of interest to be excluded. The proposed search strategy retained 10 papers. Figure 5.2 summarizes the search strategy.

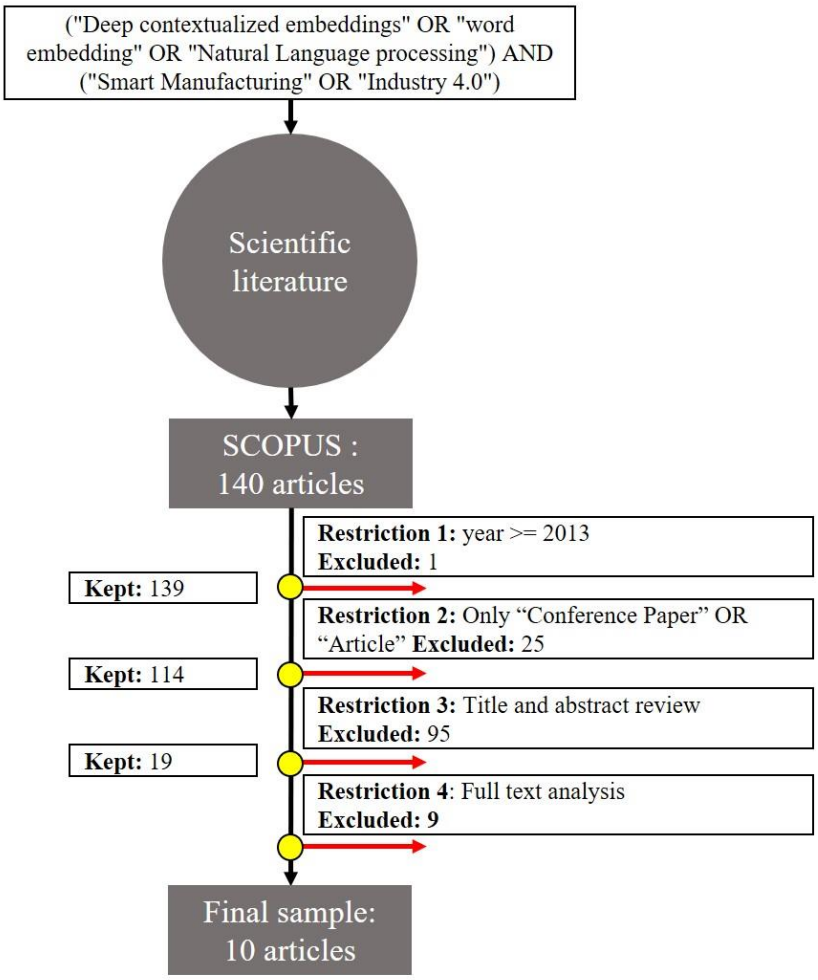


Figure 5.2 Search strategy to assess recent scientific literature.

Results from the literature analysis suggest that no study has applied contextualized word embeddings under the frame of Industry 4.0. It seems that most of the research uses either non-contextualized word embeddings or classical approaches, such as TF-IDF, word occurrence counting, and one-hot-encoding. For example, in the social domain, Lynn et al. (2019) trained a type of recurrent neural network called a bidirectional-GRU (Chung et al., 2014) to recognize misogynistic articles on a webpage called the *Urban Dictionary*. To find word representations of texts, they employed one-hot-encoding. Regarding online social support websites, Chen (2019) used non-contextualized embeddings, more specifically Doc2Vec (Le and Mikolov,

2014), to develop a deep neural network capable of classifying messages from users into informational or emotional support.

Both Trappey et al. (2019) and Keyvanpour and Serpush (2019) worked on the improvement of information retrieval in research through clustering: while the former used Doc2Vec with k-means on several thousands of patents and papers to discover new trends regarding solar power technologies, the latter employed a modified version of TF-IDF with k-means to assist the retrieval of biomedical studies on scientific databases.

Regarding the field of product design, Ireland and Liu (2018) developed a comprehensive framework concerning the exploitation of product reviews from Amazon to assist in the improvement of products. Through the use of several techniques such as TF-IDF, the apriori algorithm, support vector machines, naïve bayes, and others, they derived insights from product reviews to ease the identification of the critical features contributing to negative as well as positive customer experiences.

In manufacturing, more specifically in maintenance, Sharp et al. (2017) and Sexton et al. (2017) focused on the extraction of information from maintenance logs to support the resolution of problems through a description provided about the issue. More precisely, Sharp et al. (2017) vectorized the maintenance logs through word occurrence counting to train a support vector machine able to classify the problem into one of the different diagnostic tags provided by an expert. In a similar way, Sexton et al. (2017) used the TF-IDF approach with a linear support vector machine to classify the words in maintenance logs into three main categories: *item*, *problem*, and *solution*. This tagging process of words enriches already-existing maintenance datasets.

Still in the manufacturing sector, the comprehensive work of Madhusudanan et al. (2017, 2019) focused on the extraction of assembly knowledge from PLM data. Although they did not use ML or word embeddings, they provide interesting insight regarding the data volume and quality in manufacturing: they considered the lack of exploitable training data as one of the main obstacles to employing ML in industrial applications. For that reason, they preferred to use a manually constructed lexicon to meet their research objectives. This conclusion suggests that novel methods such as CamemBERT may help overcome this limitation. Through TL, it is possible to achieve a good performance on new tasks even when training data is scarce.

Finally, in the domain of human resources, Bondielli and Marcelloni (2019) employed Doc2Vec and hierarchical clustering to extract more pertinent profiles from candidate resumes in order to support the recruiting process.

The analysis of the literature suggests that using TL through contextualized word embeddings is rare, especially in the context of I4.0. In fact, its usage may not only improve the results of ML algorithms, but also provide a solution to the data scarcity issue. Furthermore, as contextualized word embeddings represent a recent topic in NLP, comparing their performance to other approaches may provide interesting insights.

5.3. Materials and methods

This section presents the dataset employed, as well as the pre-processing steps involved. Additionally, the ML architectures used are described. Finally, the policy used to train these architectures, compare them, and select the best performing model will be detailed.

5.3.1. Employed dataset

The dataset was provided by a manufacturing company, whose name and industry are not mentioned for reasons of confidentiality. It contained about 20000 maintenance logs. However, due to the presence of missing data on the variables of interest, the exploitable dataset used in this research only contains around 7000 observations. The details of the available variables are described in Table 5.1.

Variable	Data type	Detail
Equipment description	String (free-form text)	Provided name of the equipment, filled by the operator
Symptoms	String (free-form text)	Description of the symptoms leading to the problem
Equipment importance	String (categorical: ordinal)	Three categories of equipment importance, from most to least relevant: 1) Essential 2) Important 3) Secondary
Type of disturbance	Binary	Either if the disturbance was dominant (0) or recessive (1)
Workload	Positive integer	Number of hours required to solve the issue

Table 5.1 Detail on the employed maintenance logs dataset.

5.3.1.1. Pre-processing the inputs

One of the advantages when using embeddings (both contextualized and non-contextualized) is that the pre-processing steps are minimized. In other words, less time can be spent on the development of complex hand-crafted input features. Thus, to illustrate such an advantage, it was decided to concatenate the variables "Equipment description," "Symptoms," and "Equipment importance" into a single piece of text for each observation. This single text will be called the "Input sequence," and it will be the sole input of the proposed ML models. The objective of this is to show that there is no need to heavily pre-process the different variables to create useful representations when using models such as CamemBERT. Figure 5.3 illustrates this pre-processing step with an example.

Original dataset

Obs.	Equipment description	Symptoms	Equipment importance	Type of disturbance	Workload (h)
1	Electrical pump of hangar 3	The pump was squeaking and then stopped	Important	Dominant	7
…	…	…	…	…	…
N	Manual milling machine	Saddle difficult to move in the Y direction	Secondary	Recessive	3

Concatenation into a single input

Pre-processed dataset

Obs.	Input sequence	Type of disturbance	Workload (h)
1	Electrical pump of hangar 3. The pump was squeaking and then stopped. Important	Dominant	7
…	…	…	…
N	Manual milling machine . Saddle difficult to move in the Y direction. Secondary	Recessive	3

Figure 5.3 Example of the pre-processing that was performed.

5.3.1.2. Brief analysis of the distribution of the outputs

Maintenance logs are typically imbalanced datasets. This means that some of the classes have a much higher frequency. In the employed data, issues not blocking the production (recessive disturbances) are more abundant than cases in which the production was blocked (dominant disturbances). Similarly, issues requiring low workloads to be solved tend to be more frequent than the extreme cases, in which the problem will take several hours. This can be observed in Figure 5.4 and Figure 5.5, representing the histograms for the type of disturbance and the workload, respectively.

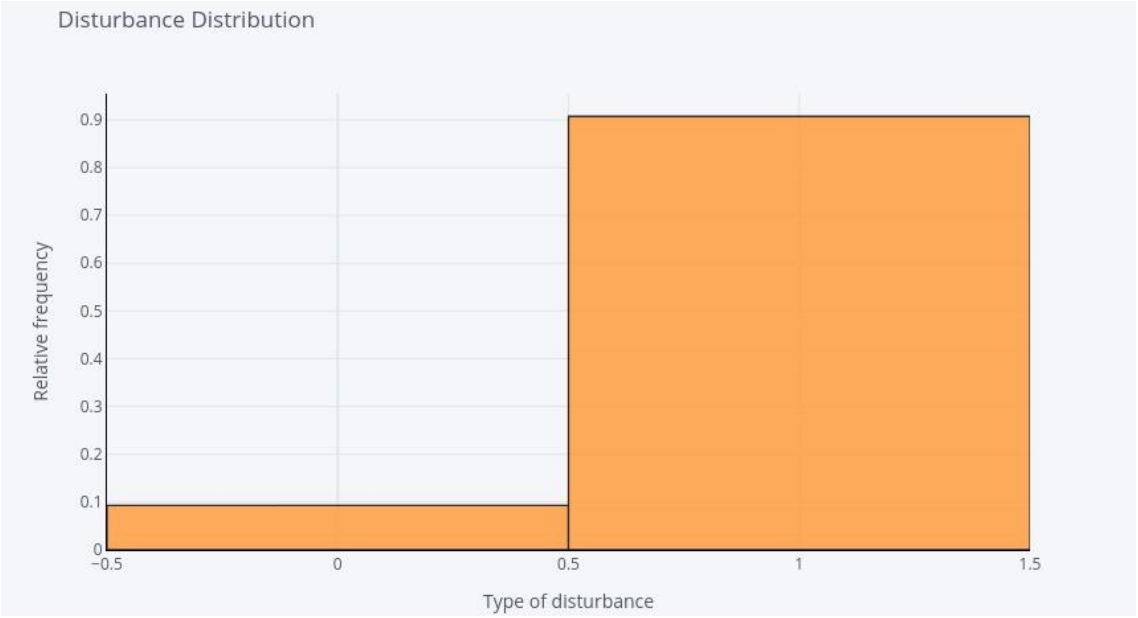


Figure 5.4 Histogram for the disturbance type: dominant (0) and recessive (1) disturbances.

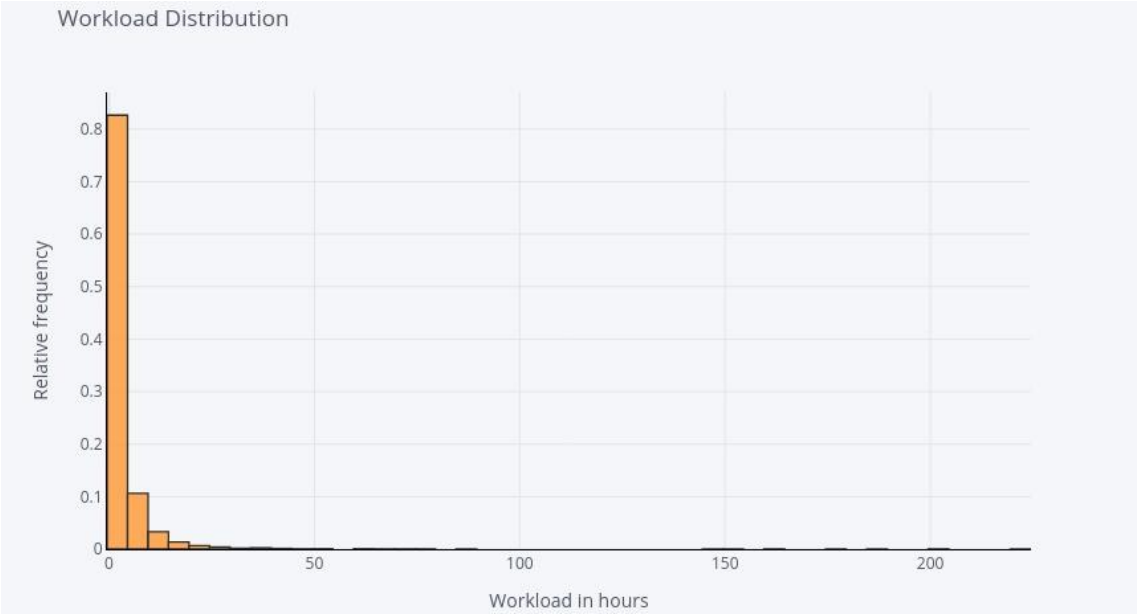


Figure 5.5 Histogram for the workload.

As observed in the distributions, only around 10% of the maintenance logs represent a dominant disturbance. Also, disturbances requiring fewer than 4 hours of workload represent 80% of the dataset. Class imbalance can be harmful for ML algorithms, as they tend to learn the overrepresented class better. For more information on this topic, please refer to Bi and Zhang (2018), who empirically compared recent algorithms that were able to tackle class imbalance, and Johnson and Khoshgoftaar (2019), who surveyed the application of deep learning

algorithms on imbalanced datasets. In this research, some solutions to tackle class imbalances will be employed. Nevertheless, future work will exclusively focus on this aspect.

As the workload distribution seems to be the most imbalanced variable, it will be pre-processed to ease the learning. Thus, a clustering algorithm (i.e. k-means) is used to find groups of maintenance logs with respect to their workload. Then, these clusters will be the labels that the learning algorithms will learn to predict.

5.3.1.3. Pre-processing the outputs to handle class imbalance

To reduce the class imbalance in the workload variable, the k-means algorithm with Euclidean distance was employed to find categories of maintenance logs by their workload. However, as k-means only considers the distance between the observations and not the number of instances per cluster, this may lead to strongly imbalanced clusters. Hence, to further reduce the class imbalance, the atypical workload durations were identified and excluded when training the clustering model. These outliers will represent a cluster called "Expert", meaning that their high duration may need an assessment by a human expert. The outliers were defined as the values that are higher than the third quartile, plus 1.5 times the interquartile range. It is important to note that the quartiles were calculated using the linear method, which is method number 5 in (Hyndman and Fan, 1996). Figure 5.6 shows a zoomed view of the outlier detection results in a boxplot. The datapoints are also displayed at the left of the boxplot.

Figure 5.6 Boxplot and outlier detection for the workload.

Having excluded the outliers, the next step is to find clusters among the remaining observations using k-means. Before performing this step, the initial dataset was split into a training and a test

set following a 75/25% distribution, respectively. This partition was performed at this stage to avoid training any algorithm on data available on the test set, which could induce overfitting. To choose the number of clusters, silhouette diagrams were employed (Rousseeuw, 1987). The advantage of this approach over the commonly used elbow method is that the silhouette diagrams allow the size of the cluster to be seen, as well as provide a measure of its quality. The cluster size is represented by the height of the bar, meaning that clusters with more instances will be bigger. The cluster quality is measured through the silhouette coefficient, which ranges from -1 to 1. A value of 1 means that the instances inside the cluster are well inside it and far from other clusters; a value of 0 implies that the instance is close to a boundary with other clusters and a value of -1 suggests that the instance should be in another cluster (Géron, 2019). Figure 5.7 shows the silhouette diagrams for 3, 4, 5, and 6 clusters with their respective average silhouette coefficient marked as a red dotted line.

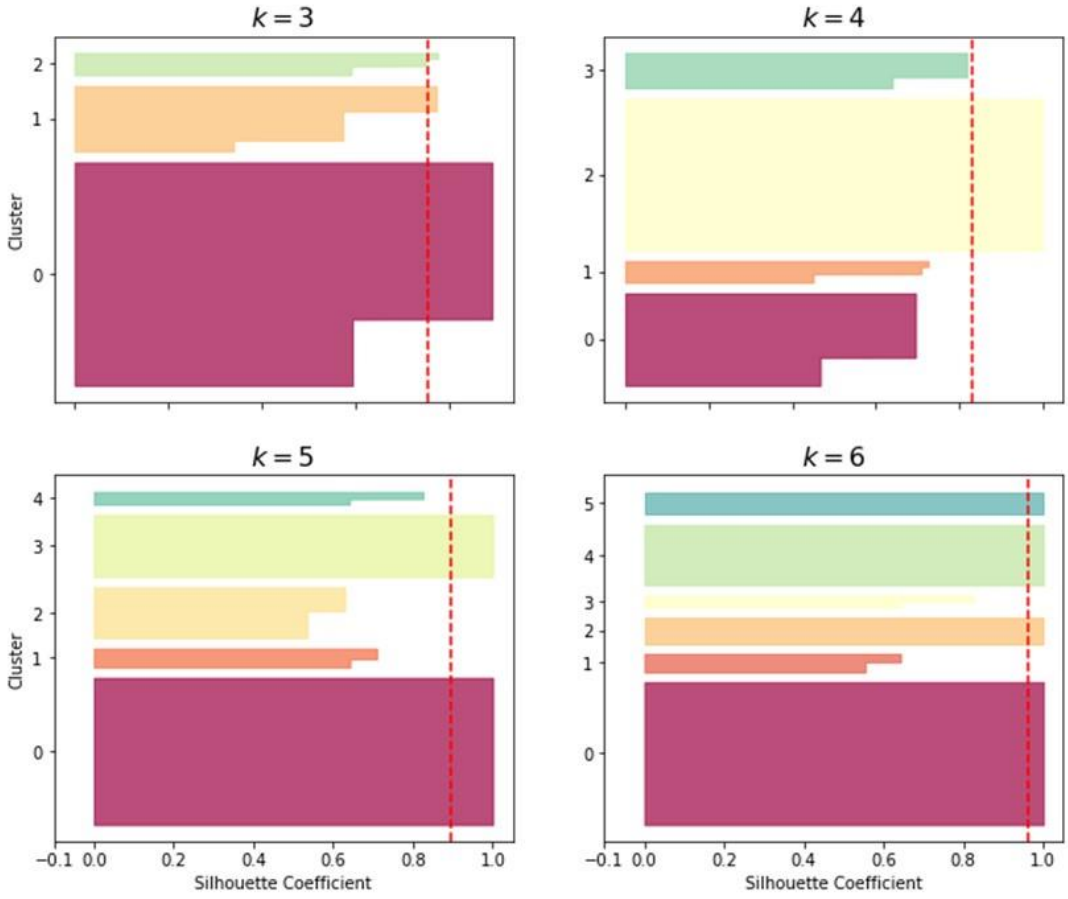


Figure 5.7 Silhouette diagrams with a different number of clusters.

Silhouette diagrams suggest that four clusters may be the best choice in order to have both good clusters as well as a reduced class imbalance. In fact, using four clusters offers a relatively good

silhouette score of 0.83 while keeping a less pronounced class imbalance distribution with respect to the other options. The silhouette diagrams have the advantage of visualising the size of several cluster choices at the same time. As the final steps, the clusters are added to the test set using the already trained model, and the "Expert" cluster containing the outliers is included. Table 5.2 summarizes the final cluster results with the concerned workload values on the training set.

Cluster ID	**Percentage of observations**	**Encompassed workload values (h)**	**Description**
0	28.35%	2-3	Low workload. Not very severe maintenance issues
1	6.60%	6-8	High workload. Severe maintenance issues
2	46.62%	1	Very low workload. Common and simple maintenance issues
3	10.81%	4-5	Medium workload. Relatively severe maintenance issues
4	7.63%	≥9	"Expert" cluster: specific cases normally leading to a high workload. An expert is needed to estimate the actual severity.

Table 5.2 Cluster details for the workload.

After these two pre-processing steps, the objective will be to predict if the provided input sequence leads to a dominant or recessive disturbance (type of disturbance) and to predict the most likely cluster for the expected workload. The next subsection will focus on the ML architectures that will be used.

5.3.2. ML architectures to be compared

A total of four groups of architectures will be trained: a baseline model, a TF-IDF approach, a feature-based approach, and a fine-tuning approach. They will be detailed in the following subsections.

5.3.2.1. Baseline model

This type of architecture will serve exclusively to provide a reference for the performance of the other models. This baseline model will adopt a "dummy" approach, always predicting the most frequent class. For instance, in the case of the type of disturbance prediction, the baseline model will always predict the "recessive" class.

5.3.2.2. TF-IDF approach

To show the advantages of using contextualized word embeddings, it is necessary to compare them with a classical and extensively used approach. As remarked by Beel et al. (2016), TF-IDF is one of the most employed methods to vectorize text inputs in NLP. Therefore, a Random Forest (RF) and a Logistic Regression (LR) will be trained on the maintenance logs that are vectorized through TF-IDF.

Using TF-IDF often requires the following pre-processing steps: tokenization, stopword removal, stemming, n-grams conversion, and filtering. Tokenization consists of obtaining the list of separated words (called tokens) present in a phrase. Stopword removal is the action of deleting commonplace tokens that may not add semantic value. Stemming consists of transforming each token to its root form, which reduces the size of the vocabulary. N-gram conversion seeks to find groups of tokens that may add more value when grouped into a single token. Finally, filtering removes tokens that are rarely used. A more detailed explanation of each step can be found in (Usuga Cadavid et al., 2019).

To train the models, a grid search with 5-fold cross-validation was employed, as it allows several combinations of hyperparameters to be tested. The TfidfVectorizer from scikit-learn (Pedregosa et al., 2011) was used. Furthermore, inspired by the fact that CamemBERT outputs 768-dimensional embeddings for the phrases, the output of the vectorizer was also set to 768 tokens. To perform the stopword removal and stemming, the NLTK library in Python was employed (Bird and Loper, 2004). Table 5.3 summarizes the combination of hyperparameters that were left to be optimized by the grid search strategy.

TF-IDF vectorizer	Cluster ID	Percentage of observations
• N-gram range: 1, 2. • Removing stopwords: True, False. • Tokenization strategy: plain tokenization, tokenization plus stemming.	• Default values from scikit-learn	• Norm: L1, L2 • C penalty for regularization: 1, 10, 100

Table 5.3 Hyperparameters to be tested by the grid search.

Finally, to consider the class imbalance, two approaches were used: class weighting and resampling. The former consists of assigning a higher weight to observations of the underrepresented classes. Thus, the ML learning model will tend to learn such classes better in order to maintain good performance by avoiding higher penalties. The second strategy consists of artificially increasing the number of observations of underrepresented classes. This is done through sampling with replacement from the training set. The employed resampling strategy ensures that the underrepresented classes have as many observations as the most frequent class.

5.3.2.3. Feature-based approach with CamemBERT

The feature-based approach employed the PyTorch version of the base CamemBERT model that is available on the Transformers library (Huggingface, 2019). When used as a feature extractor for document classification, CamemBERT outputs a 768-dimensional vector corresponding to the contextualized word embeddings of the phrase. After finding the vector representation for all of the input sequences on the maintenance logs, these vectors can be used to train ML models.

The ML models that will be coupled to CamemBERT's output are detailed in Table 5.4. The densely connected neural networks were created with Keras 2.2.4 using Tensorflow 1.15.0 backend. As the maximum length of the input sequences in the training set was 54, the maximum length for the CamemBERT inputs was fixed to 64 to be compliant with the common practice of using powers of two. Nevertheless, the impact of the maximum length in the performance is to be explored in future research. The detail of CamemBERT in the feature-based mode is shown in Figure 5.8.

ML model	Parameters
Densely connected neural network (Dense Net)	• 3 hidden layers with 300, 300, and 200 units each. • ReLU in hidden layers. Softmax in the output layer. • Loss function: sparse categorical cross-entropy. • Optimizer: Adam (Kingma and Ba, 2014) with default Keras parameters. • Epochs: 30 and 60 in disturbance classification and workload cluster classification, respectively. • Batch size: 32. • Early stopping with patience equal to 3 and 5 in disturbance classification and workload cluster classification, respectively. The best model is kept.
Dense Net with weighted loss (D. Net Weighted)	• Same parameters as for the Dense Net. • Weighted loss to consider the class imbalance.
D. Net Weighted with dropout (D. Net Dropout)	• Same parameters as for the D. Net Weighted. • Dropout of 0.2 on each hidden layer to avoid overfitting (Srivastava et al., 2014).
D. Net Weighted with weight decay (D. Net Decay)	• Same parameters as for the D. Net Weighted. • L2 norm of 0.001 on the last two hidden layers for disturbance classification and on all the hidden layers for workload cluster classification.
Random Forest (RF-Feature based)	• Default parameters from scikit-learn. • Class weighting.
Logistic Regression (LR-Feature based)	Class weighting and grid search with a 5-fold cross-validation with the following parameters: • C penalty for regularization: 1, 10, 100.

Table 5.4 Hyperparameters for the models trained with the feature-based approach.

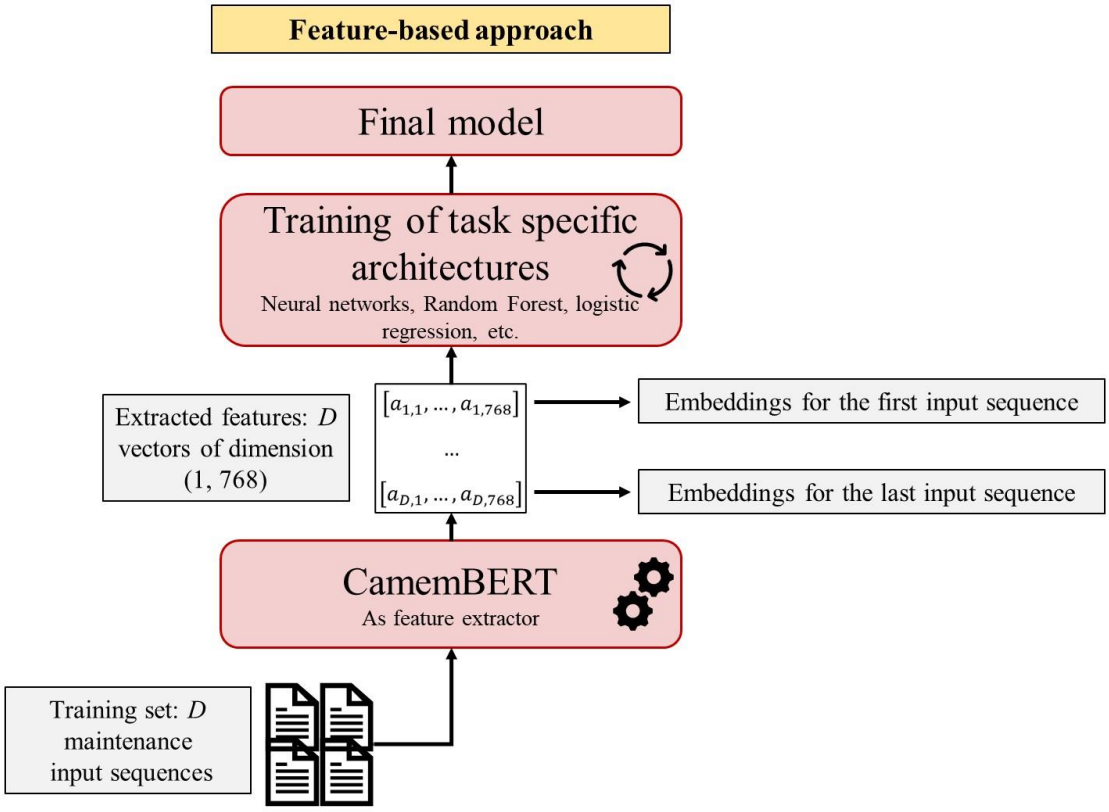


Figure 5.8 Functioning of the feature-based approach with CamemBERT.

To generate contextualized word embeddings, CamemBERT transforms the raw text from the input sequence into a list of tokens through its own tokenizer. In this process, CamemBERT appends two special tokens: the [CLS] and [SEP] token. Then, it maps each token into a predefined key and runs the CamemBERT transformer architecture, which has previously learnt word representations. The last hidden layer in CamemBERT will output a 768-dimensional vector for each of the tokens, corresponding to their respective contextualized embedding. As this research focuses on classification, only the vector for the [CLS] token is relevant. In fact, according to Devlin et al. (2018), it corresponds to the aggregated representation of the input sequence for classification tasks.

5.3.2.4. Fine-tuning approach with CamemBERT

The objective of the fine-tuning approach is to jointly train CamemBERT with a linear layer added on top of the pooled output. Thus, the contextualized embeddings will adapt to the target task. The main advantage of this approach is that it requires minimal architecture modifications, easing the model creation process.

To create the model, the class called CamembertForSequenceClassification available in the Transformers library was employed (Huggingface, 2019). This class automatically adds a linear layer on top of the base CamemBERT. Two versions of this model were trained: one with the training dataset without modifications and a second with resampling to tackle the class imbalance. The resampling strategy was the same as in the TF-IDF approach. The maximum length for the CamemBERT inputs was fixed to 64. Finally, the code was inspired and adapted from a publicly available implementation for BERT done by McCormick and Ryan (2019).

When training the model, the choice was not to modify its architecture. In this way, the obtained results will reflect the true capabilities of the model in a fine-tuning mode. Nevertheless, future research will focus on modifying the loss function of the fine-tuning approach to consider class weighting. The model was developed with PyTorch following the hyperparameters in Table 5.5. Finally, the details of the fine-tuning approach are represented in Figure 5.9.

ML model	**Parameters**
CamemBERT - fine-tuning: • Fine tuning-Imbalanced: no resampling. • Fine tuning-Balanced: with resampling.	• Optimizer: AdamW (Loshchilov and Hutter, 2017) with a learning rate of $2 * 10^{-5}$and an epsilon of $1 * 10^{-8}$ to prevent division by zero. • Epochs: 4. • Batch size: 32. • Learning rate warm-up with zero warm-up steps

Table 5.5 Hyperparameters for the models in the fine-tuning approach.

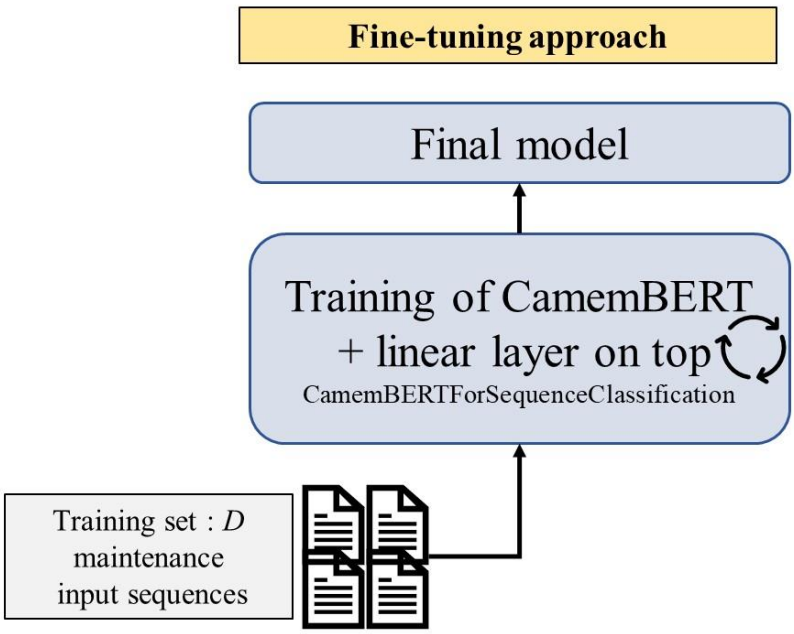


Figure 5.9 Functioning of the fine-tuning approach with CamemBERT.

Having explained the ML architectures that will be used, the next subsection will describe the model training, comparison, and selection policy.

5.3.3. Policy for training, comparing, and selecting the best model

The policy consists of creating four groups of models:

1) Baseline group: this only contains the baseline model;
2) TF-IDF group: models trained through TF-IDF vectorization. More specifically, it encompasses the RF and LR in their class weighting and resampling versions;
3) Feature-based group: models presented in Table 5.4 using the contextualized embeddings produced by CamemBERT in the feature-based mode; and
4) Fine-tuning group: CamemBERT model trained on the base and resampled dataset following the fine-tuning mode.

Each of these groups will receive a copy of the training set, from which 10% of data will be kept for validation. The reduced training set will be used to train the models, while the validation set will be employed to compare their performance. The best performing model on the validation set for each group will be selected and retrained on the full training set. Then, the test set will be utilized to choose the final model among the best models of each group. To

measure the performance, the Matthews Correlation Coefficient (MCC) is employed. Figure 5.10 summarizes the policy.

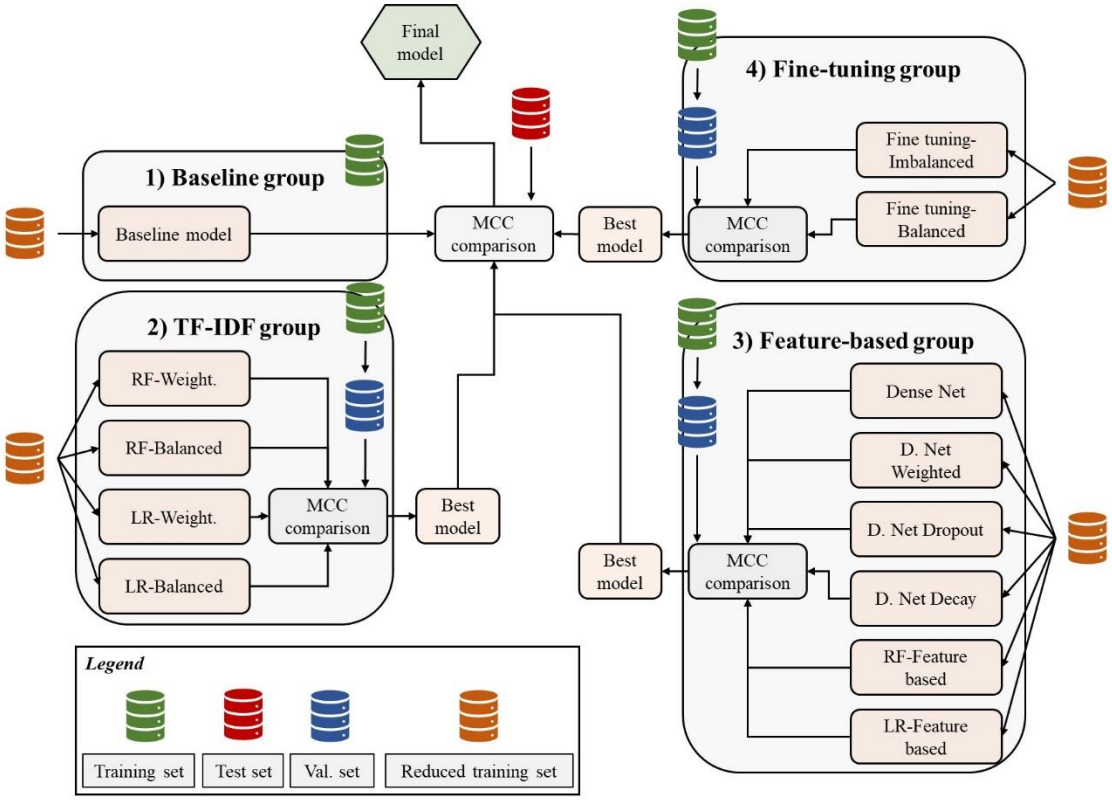


Figure 5.10 Proposed policy.

There are numerous ways of measuring the performance in classification. The classical approaches measure it through accuracy, sensitivity, specificity, and F1-score. More specifically, the F1-score globally measures the quality of the classifier. Nevertheless, this indicator has been found to be misleading, especially when working with imbalanced datasets (Hand and Christen, 2018). To tackle this issue, other measures have been proposed for classification, such as Cohen's kappa score (Cohen, 1960). However, a recent research work recommends avoiding such a score in classification problems, as it presents anomalous behavior in certain specific cases (Delgado and Tibau, 2019). Instead, the MCC is recommended. The MCC can be seen as the Pearson Correlation Coefficient applied to a discrete case (Powers, 2011). Hence, its interpretation is alike: a value of 1 suggests a perfect classifier, 0 is an average random prediction, and -1 an inverse prediction.

The MCC has been validated when used in imbalanced datasets by several recent studies when compared to the F1-score (Chicco and Jurman, 2020) , Cohen's kappa score (Delgado and

Tibau, 2019), and Confusion Entropy (Jurman et al., 2012). Thus, the MCC will be employed as a unique comparison measure among the proposed ML models. However, the accuracy will also be provided, as researchers are familiar with this measure. Furthermore, in the case of disturbance classification, the sensitivity with respect to the dominant disturbances will be presented as well. This will provide further insight about how well the class imbalance was handled by the model with regards to the underrepresented class.

5.4. Results

As the neural networks in the feature-based, as well as the fine-tuning approach, were trained on a GPU, some randomness is introduced. Thus, the training and validation loops were run ten times, on each occasion measuring the performance in order to obtain a more reliable estimate. The same procedure was followed for performance on the test set.

5.4.1. Results for the disturbance type classification

Average validation accuracy, sensitivity and MCC are presented in Table 5.6. The best model within each group is highlighted in bold. For the feature-based and fine-tune approaches, the boxplots are presented in Figure 5.11 and Figure 5.12, respectively. In the boxplots, the mean is represented by the dashed line.

Group number/ Model name	Accuracy	Sensitivity	MCC
2/ LR-Balanced	0.845	0.500	0.303
2/ LR-Weight.	0.833	0.609	0.343
2/ RF-Balanced	0.913	0.348	0.386
2/ RF-Weight.	**0.923**	**0.239**	**0.390**
3/ D. Net Decay	0.747	0.559	0.214
3/ D. Net Dropout	0.791	0.411	0.173
3/ D. Net Weighted	0.794	0.459	0.207
3/ Dense Net	0.913	0.072	0.204
3/ LR-Feature based	**0.790**	**0.587**	**0.273**
3/ RF-Feature based	0.915	0.065	0.244
4/ Fine tuning-Balanced	**0.884**	**0.502**	**0.383**
4/ Fine tuning-Imbalanced	0.918	0.202	0.335

Table 5.6 Validation results for groups 2, 3, and 4 in disturbance classification.

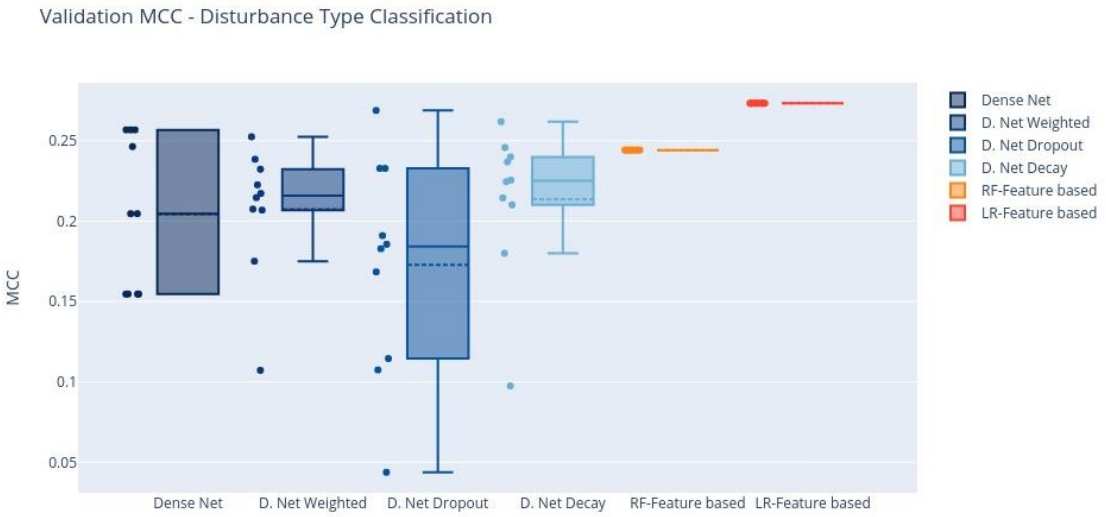


Figure 5.11 Box plots for the validation of MCC in feature-based mode for disturbance classification.

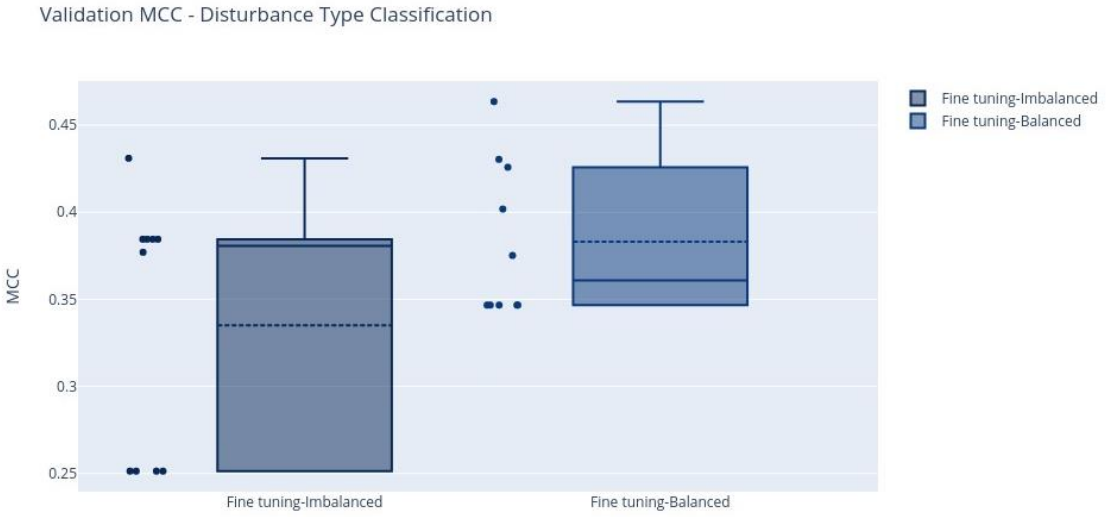


Figure 5.12 Box plots for the validation MCC in fine-tuning mode for disturbance classification.

In the second group, the RF, using class weighting, provided the best validation performance. In the third group, the logistic regression is the model that better exploited the contextualized word embeddings extracted from the input sequences. Furthermore, the LR achieves a sensitivity of almost 0.587, which means that around 60% of the production problems blocking the production are successfully detected. This suggests a strong capacity to handle imbalanced datasets when working with binary classification problems. In the fourth group, the fine-tuning approach with resampling achieved the best MCC, implying that resampling may be a useful approach to tackle class imbalance issues with deep learning techniques. Finally, the results

suggest that resampling reduces the variability of the results with respect to the imbalanced fine-tuning.

The best model for each group was tested in the test set. Table 5.7 presents the results with the best model in bold and Figure 5.13 presents the box plots for the MCC.

Group number/ Model name	Accuracy	Sensitivity	MCC
1/ Baseline Model	0.914	0.000	0.000
2/ RF-Weight.	0.927	0.236	0.386
3/ LR-Feature based	0.783	0.632	0.282
4/ Fine tuning-Balanced	**0.899**	**0.553**	**0.433**

Table 5.7 Test results in a disturbance classification.

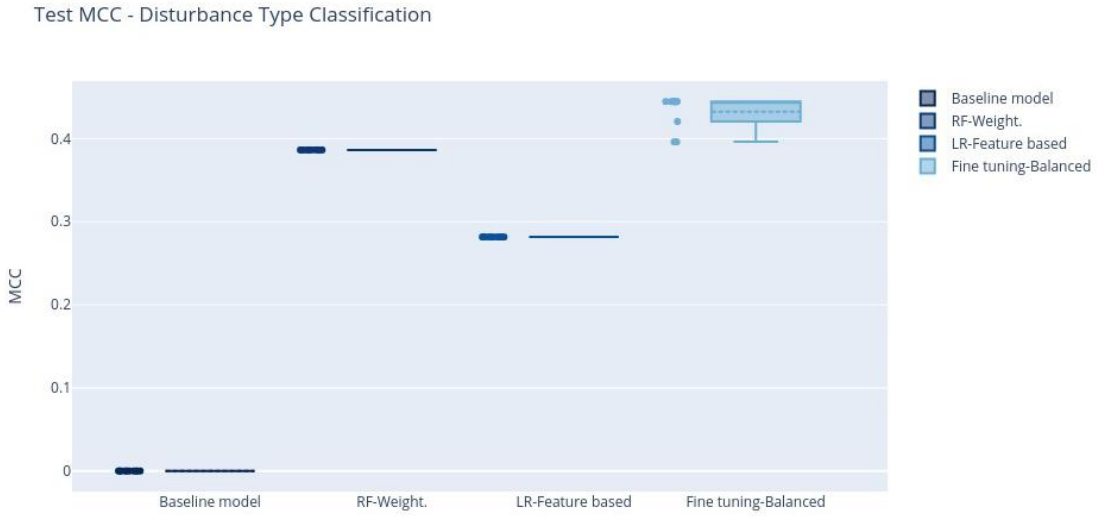


Figure 5.13 Box plots for the test MCC in a disturbance classification.

Fine-tuning with resampling is, by far, the best model in terms of MCC. This suggests that using TL in NLP is a fruitful effort. It is important to recall that the fine-tuning approach does not modify the architecture of the model. Instead, the only modification was to add a resampling strategy to the data. In other words, CamemBERT can achieve excellent performance with minimal task-specific work. This result confirms the recommendation of Devlin et al. (2018) regarding the use of BERT-based architectures in a fine-tuning mode and hints that it may be a useful technique when working with maintenance logs. Furthermore, even if CamemBERT achieves slightly worse accuracy than the baseline model, it is worth nothing that the latter will always predict that disturbances are recessive. This prediction may be harmful when deployed in the shop floor, as critical problems will be ignored. In contrast, CamemBERT achieves a

sensitivity of 0.55, meaning than more than a half of the critical problems blocking the production process will be successfully detected.

Finally, even though it is surprising to see that the TF-IDF approach (RF-Weight.) achieves a good MCC, recall that this follows a heavy process of hand-crafted pre-processing, when compared with the almost *plug and play* process of CamemBERT.

5.4.2. Results for the workload cluster classification

Average validation accuracy and MCC are presented in Table 5.8. The best model in each group is highlighted in bold. For the feature-based and fine-tune approaches, the boxplots are presented in Figure 5.14 and Figure 5.15, respectively.

Group number/ Model name	Accuracy	MCC
2/ LR-Balanced	0.381	0.140
2/ LR-Weight.	0.361	0.156
2/ RF-Balanced	0.456	0.159
2/ RF-Weight.	**0.486**	**0.196**
3/ D. Net Decay	0.450	0.166
3/ D. Net Dropout	0.418	0.176
3/ D. Net Weighted	0.415	0.174
3/ Dense Net	0.479	0.143
3/ LR-Feature based	**0.373**	**0.183**
3/ RF-Feature based	0.492	0.138
4/ Fine tuning-Balanced	**0.410**	**0.199**
4/ Fine tuning-Imbalanced	0.473	0.154

Table 5.8 Validation results for groups 2, 3, and 4 in the workload cluster classification.

Figure 5.14 Box plots for the validation MCC in a feature-based mode for workload cluster classification.

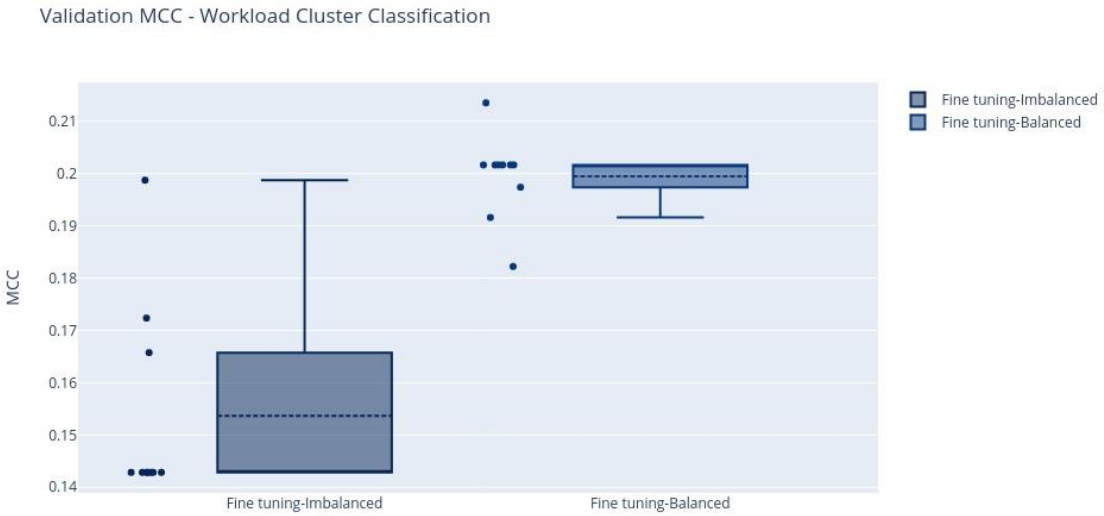


Figure 5.15 Box plots for the validation of MCC in fine-tuning mode for a workload cluster classification.

Results for the workload cluster classification are comparable to those obtained in the disturbance type classification in terms of the best performing models. This suggests that when analyzing maintenance logs, some models tend to have better results. More specifically, when using TF-IDF, it seems that RF with class weighting is the best-performing model. In the cases where feature-based approaches are used, LR with class weighting is the technique that best exploits the embeddings produced. Finally, when using the fine-tuning mode, resampling seems to largely improve performance.

It seems that the class imbalance problem further affects the performance of ML models when the number of classes increases. This can be articulated from the fact that the MCC is lower for the workload cluster classification than for the disturbance type classification.

The performances of the best models in each group were measured on the test set. Results are shown in Table 5.9 and in Figure 5.16.

Group number/ Model name	Accuracy	MCC
1/ Baseline Model	0.468	0.000
2/ RF-Weight.	0.493	0.189
3/ LR-Feature based	0.352	0.139
4/ Fine tuning-Balanced	**0.424**	**0.210**

Table 5.9 Test results in workload cluster classification.

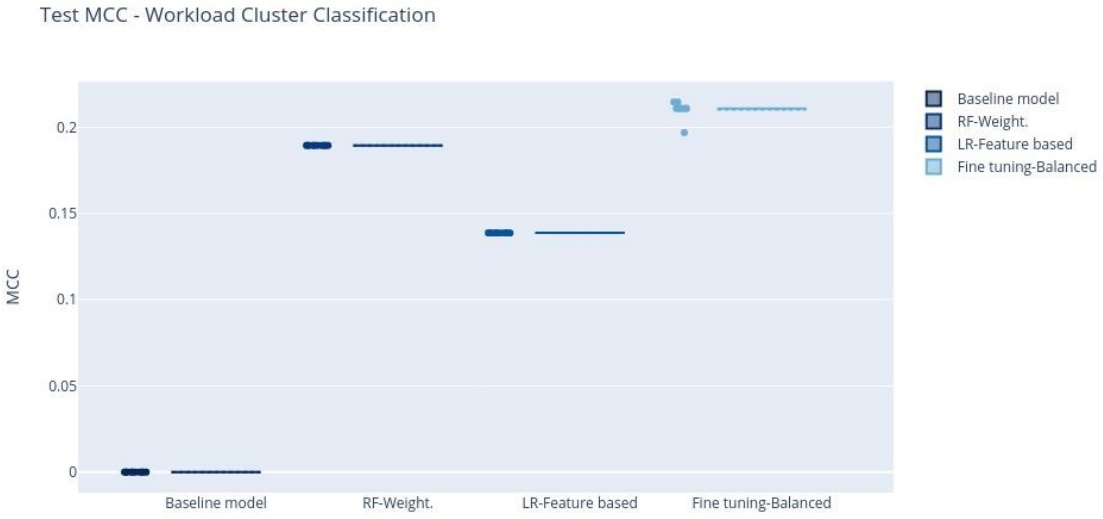


Figure 5.16 Box plots for the test MCC.

The performance on the test set suggests that a fine-tuning approach is the best model in terms of MCC. This supports the findings of the disturbance type classification, where CamemBERT in fine-tuning with resampling also outperformed the other architectures. MCC values show that the classification performance is less than in a disturbance classification. This may suggest two things: firstly, the class imbalance is more severe in a workload cluster classification, which damages the performance of ML algorithms. And secondly, it may be that the data available in the input sequence does not provide enough information to achieve fully effective learning. For instance, logistics data concerning the availability of spare parts to perform maintenance interventions was not included. Despite the less successful results when classifying the

workload, this study suggests that the fine-tuning approach may be the most suitable way to exploit free-form text data from maintenance logs, setting the groundwork for future research.

5.5. Discussion

5.5.1. Limitations of the study

This research has three main limitations: first, the hyperparameters for the fine-tuning approach (e.g. maximum phrase length, learning rate, number of epochs, etc.) were not fine-tuned, mainly to demonstrate the performance of the model without much human intervention. Nevertheless, this needs to be explored further to improve the results. Secondly, only one industrial dataset was employed to conduct the experiments. As CamemBERT is a recent model, it is important to test the proposed approach with a wider variety of datasets. More specifically, it should be tested on those containing bilingual maintenance logs, as this is often found in companies where Anglicisms are common. Finally, the input sequence was created through the concatenation of the "Equipment description," "Symptoms," and "Equipment importance". The objective was to show that CamemBERT could achieve good performance without requiring heavy feature engineering. Nevertheless, this choice should be further explored in future research, as it may not be the most optimal way to pre-process the inputs.

5.5.2. Directions for further research

Future work will focus on the following three axes:

1) *Compare the performance of CamemBERT with other language-specific architectures in imbalanced maintenance datasets:* Despite CamemBERT being a new model devoted to the French language, more recent research efforts in this domain have led to the development of another model named FlauBERT. Thus, future work will focus on the comparison of these two architectures along with a more extensive use of techniques to tackle class imbalance in deep learning. Regarding class imbalance, this paper only used data-level (i.e. resampling) techniques to tackle this issue. This may not be suitable for other applications in I4.0 where resampling massive industrial datasets could lead to unacceptable training times or memory issues. Hence, other procedures belonging to algorithm-level and

hybrid approaches, as proposed by Johnson and Khoshgoftaar (2019), must be tested.

2) *Evaluate the performance on other datasets:* To validate the generalization capacity of the proposed approach, the best performing model between CamemBERT and FlauBERT, along with techniques to mitigate the impact of class imbalance, will be applied to several maintenance log datasets. The objective will be to determine the key architectures that lead to globally successful results. Furthermore, special attention will be devoted to testing the performance of these models when working with maintenance logs containing bilingual free-form text descriptions.
3) *Explore knowledge generation from maintenance logs:* In the context of ML applied to PPC in I4.0, knowledge generation has received great attention by the research community (Usuga Cadavid et al., 2020a). In fact, knowledge is considered one of the most valuable assets in manufacturing (Harding et al., 2006). Researchers working with transformer-based models have also focused on this aspect, deriving from a recent research field called BERTology, which studies the inner workings of such models (Rogers et al., 2020). Thus, further research will explore how to generate and communicate knowledge from maintenance logs extracted through BERT-based architectures.

5.6. Conclusion

This study has explored the basis on which to create a DPS that can adapt to maintenance issues. The purpose was to estimate the criticality and duration of a maintenance problem from the description provided by an operator. To do so, free-form text data coming from maintenance logs was exploited through TL. The TL was performed through a recent NLP model named CamemBERT, both in a feature-based and fine-tuning modes. As the dataset presented a highly imbalanced structure, pre-processing steps, including outlier detection and k-means, were used to reduce the disparities among the classes. Additionally, strategies such as class weighting and resampling were also employed.

Results showed that employing CamemBERT in the fine-tuning mode with resampling outperformed the other techniques that rely on feature-based and TF-IDF approaches. This implies that fine-tuning not only helps obtain better results in terms of performance, but it also

reduces the burden of creating task-specific architectures that heavily rely on hand-crafted pre-processing steps.

Findings of this study suggested that estimating the criticality of a maintenance log through its description yielded better results than the duration estimation. This may be due to the lack of other features such as logistics data describing the availability of spare parts to perform maintenance interventions.

Highlights

- A BERT-based model adapted to French is compared to classical approaches in natural language processing
- Since maintenance datasets tend to be highly imbalanced, some solutions to tackle this issue are used that show significant improvements to the model's performance.
- Two approaches using transfer learning are compared: the fine-tuning and the feature-based approach.
- A real industrial dataset is employed to validate the proposed models.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was financially supported by a partnership between the company iFAKT France SAS and the ANRT (Association Nationale de la Recherche et de la Technologie) under the Grant 2018/1266.

6. Chapter 6: Article 3 - Using Deep Learning to Value Free-Form Text Data for Predictive Maintenance

Name of the journal: International Journal of Production Research

Received: 13 October 2020

Accepted: 28 June 2021

Authors: Juan Pablo Usuga-Cadavid, Samir Lamouri, Bernard Grabot, Arnaud Fortin

Corresponding author: Juan Pablo Usuga-Cadavid

DOI: https://doi.org/10.1080/00207543.2021.1951868

Abstract: Past maintenance logs may encapsulate meaningful data for predicting the duration of machine breakdowns, the potential causes of a problem, or the necessity to stop production to perform repair activities. These insights may be accessed using machine learning (ML). However, maintenance logs tend to have imbalanced distributions and rely on noisy unstructured text data provided by operators. Additionally, the limited interpretability of ML models results in human reluctance when accepting model predictions. Hence, this study explored the use of two recent deep learning models (CamemBERT and FlauBERT) for natural language processing (NLP) to harness unstructured data from maintenance logs. The class imbalance effect was mitigated using data-level and algorithm-level approaches. To improve interpretability, a technique called LIME was employed to interpret single predictions and to propose a method for insight extraction from several maintenance reports. Results suggest three key points: CamemBERT and FlauBERT can achieve excellent results with minimum text pre-processing and hyperparameter tuning. Second, random oversampling (ROS) generally mitigates the effect of class imbalance. However, ROS was observed to be unnecessary when performing pertinent data pre-processing. Finally, at the maintenance level, the proposed insight extraction method can provide valuable information from a set of poorly structured maintenance reports.

Keywords: industry 4.0; deep learning; maintenance; interpretability; natural language processing; class imbalance

Highlights

- This study aimed at improving production planning and control by predicting useful information for assessing the consequences of failures (e.g. breakdown causes, whether to stop production, or duration of the maintenance action).

- Therefore, maintenance reports containing unstructured texts stored in the ERP of three companies were used as test scenarios.
- Two state-of-the-art deep learning models for NLP were explored: CamemBERT and FlauBERT. These models are based on transformers, a recent architecture that significantly improves the results of NLP tasks. Also, their performance is compared with classic ML models in several datasets.
- To tackle the effect of class imbalance, this research explored both data-level and algorithm-level approaches. Furthermore, a novel method to discretise numerical variables and reduce class imbalance using k-means, silhouette coefficients, and silhouette diagrams is proposed.
- LIME was used to interpret single predictions and to support a method that aims to extract insights useful for managers from several machine breakdown descriptions.

6.1. Introduction

Production planning and control (PPC) aims to determine the quantities to be produced to satisfy a sales plan while satisfying performance objectives frequently linked to on-time delivery and efficient use of resources (Usuga Cadavid et al., 2020a). PPC can be considered a bridge between scheduling, maintenance, quality, logistics, etc. The quality of production scheduling, which is one of the main functions of PPC, heavily depends on the accurate knowledge of the actual capacity of the resources, and thus on their short-term availability. Consequently, the maintenance function involves repairing machines after failures or breakdowns as well as capitalising and availing knowledge on the short-term state of the resources. Therefore, maintenance managers have significant expectations for the tools and technologies embedded in Industry 4.0 (I4.0), particularly the Internet of Things (IoT), enabling the generation of real-time data flow on the machine state and data mining/machine learning (ML), providing tools for analysing past experiences.

Narrowing the scope to PPC, a systematic literature review performed by Usuga Cadavid et al. (2020a) indicated that the coupling of PPC and ML in the context of I4.0 has attracted notable interest from researchers in recent years. In this context, ‘smart maintenance’ harnesses manufacturing system data to react better to production disturbances, for instance, by predicting the severity of an unexpected machine breakdown and the duration of the maintenance action,

to the benefit of production scheduling (Usuga Cadavid et al., 2019, 2020b). Poor maintenance policies can considerably affect production. For instance, Gupta and Vardhan's (2016) study on a tractor manufacturer in India observed that breakdown losses accounted for the second highest source of financial loss in the company, and energy consumption was the first. From the perspective of total productive maintenance, better reaction to machine breakdowns can increase the overall equipment effectiveness (OEE), which is calculated from the equipment availability, performance, and quality rates (Gupta and Vardhan, 2016). Here, improving the reaction to unexpected machine breakdowns should result in an increase in the availability rate.

To achieve a better reactivity, the use of manufacturing execution systems (MESs) is important: they function as the interface between the shopfloor and enterprise resource planning (ERP) systems, in which company data are stored. Thus, MESs enable data collection to support smart maintenance and perform proper adjustments to the production scheduling (Saenz De Ugarte et al., 2009; Usuga Cadavid et al., 2020b). Although this concept may seem simple, integrating MES with other information systems such as high-level planners and schedulers remains a challenge (Saenz De Ugarte et al., 2009).

Maintenance logs collected from various sources, such as MESs, are frequently stored in information systems. These records may provide useful insights for improving the production system; thus, companies are frequently interested in valuing them. In Grabot (2020), one of the authors presented experiments aimed at extracting knowledge from records of maintenance events in three different companies, all interested in assessing the potential of data mining techniques. For the three companies, the aim when formatting their databases (all extracted from SAP ECC but customised to specific requirements) was first to 'easily' capitalise interpretable data, uniformized using taxonomies and entered in the ERP using drop-down menus. Nevertheless, the precision of the recorded information depends on the predefined taxonomy, which is difficult to evolve without losing past records. Therefore, the companies decided to also store more precise (and to be fully evolutive, non-formatted) data using free texts, even if the companies could not efficiently use this information at that time.

More precisely, the records in the databases gathered data considered to be relevant to describing a maintenance activity with precise timestamps, symptoms, causes, actions, actors, parts changed, estimated duration, actual duration, etc. The records were to be filled by the people successively involved in the managing of the failure: workshop actors, maintenance

actors, maintenance manager for closure, etc. The data models of the three databases are available in (Grabot, 2020).

Despite the richness of the provided databases, the first tests focused on structured data, such as maintenance durations and taxonomies of symptoms, causes, and maintenance actions. Encouraged by these results, the companies also requested that the additional free texts entered by the maintenance actors, suspected to contain a very rich (even if hidden) reusable knowledge, be considered.

Using these records, the companies were primarily interested in predicting the data listed in Table 6.1 when a failure occurred.

	Severity prediction	**Breakdown-duration prediction**	**Cause prediction**
Company A	X	X	
Company B		X	X
Company C		X	
Type of task	Binary classification	Multi-class classification	Multi-class classification

Table 6.1 Data to be predicted using each company's dataset.

As the datasets contained different pieces of information (the three companies were using the maintenance module of SAP ECC, but the records had been intensively customised), not all the proposed tasks could be performed for all the companies. As shown in Table 6.1, Company A aimed to predict, from a machine breakdown description, whether this problem would stop the production process and how much time would be required to fix the problem. Company B aimed to estimate the approximate resolution time and propose potential causes of the problem. Finally, Company C desired to know how much time would be required to solve the problem. Defining agility as the ability to operate efficiently in an environment subjected to uncertainty and changes (Borangiu et al., 2015), knowing these elements (severity, breakdown duration, and causes) when working in maintenance should result in a more agile PPC by enabling a more informed decision-making process and better control of resource availability. Precisions on the companies and on the datasets they provided are described in Section 6.3.1.

Addressing these requirements is not an easy task. Free-form text data can be difficult to analyse because of their highly unstructured nature, as operators tend to use heterogeneous languages containing typos, acronyms, abbreviations, jargon, etc. In addition to being highly unstructured,

maintenance logs are intrinsically imbalanced (Usuga Cadavid et al., 2020b): some categories of data are over-represented with respect to others. For instance, severe machine breakdowns that cripple a production process are less frequent than common problems. This creates an additional challenge to exploit maintenance logs using ML, as supervised learning algorithms tend to perform poorly when detecting under-represented classes.

Providing ML-based tools to support decision-making may result in challenges regarding the acceptance of predictions by users, particularly in industries where human interaction is at the core of the production process. ML models, particularly those using deep learning, quickly become too complex to be interpreted by humans, undermining the trust people place in their predictions. This occurs in applications like medical diagnosis, where the interpretation of the model is a key element of trust, and in the identification of possible spurious correlations (Ribeiro et al., 2016). Furthermore, the lack of digital skills and digital culture in some companies has been identified as a critical problem in adopting I4.0 technologies (Ivanov et al., 2020). Hence, a simplified method of interpreting ML models that do not require advanced expertise is essential.

Having introduced the potential benefits of harnessing text data from maintenance logs as well as its challenges regarding class imbalance and interpretability, the objectives of this study were as follows:

Concerning the expected results:

(R1) Determine whether a maintenance problem will stop the production process (severity prediction).

(R2) Determine the approximate time required to fix the problem (breakdown-duration prediction).

(R3) Determine the potential cause of the problem (cause prediction).

Concerning the tools used:

(T1) In the context of maintenance event records, compare the performance of algorithm-level and data-level techniques to mitigate the effect of class imbalance on two recent deep learning models (i.e. CamemBERT and FlauBERT) used for natural language processing (NLP). Also, the effect of a novel method employed to discretise numerical variables and reduce class

imbalance is assessed. The proposed method uses k-means, silhouette coefficients, and silhouette diagrams.

(T2) Explore the use of a model-agnostic interpretation technique (i.e. local interpretable model-agnostic explanations (LIME)) to enable the interpretability of deep learning predictions in NLP applications and extract insights supporting decision making.

Objective T1 would aid in determining which approach mitigates the most class imbalance effect and enable the identification of the most suitable deep learning model for a particular scenario. Furthermore, both binary and multiclass classification scenarios were explored. Objective T2 would enable the interpretability of these models and the generation of insights to maintain humans in the loop of ML applications.

The remainder of this paper is organised as follows: Section 2 provides a brief background on attention mechanisms and imbalanced classification techniques. Research related to predictive maintenance is reviewed. Section 3 describes the employed datasets and the characteristics of the companies providing them, the data pre-processing steps, the models to be employed, and the training policy. Section 4 compares the deep learning models and interpretability results. Finally, section 5 presents the implications, limitations, future research avenues, and conclusions for this study.

6.2. Related studies

6.2.1. Brief background

This subsection presents a brief background on attention mechanisms, transformers, and approaches to tackle the class imbalance to facilitate understanding the research gaps identified in related work and subsequent sections of this paper.

When operating with NLP data, the context of usage is an important dimension to consider, as it affects the meaning of words. Recently, researchers have used recurrent neural networks (RNNs) to capture context in NLP, as these networks can maintain records of past and future inputs. However, RNNs and their variants consider inputs sequentially, precluding parallelisation. Additionally, very long-term dependencies are not captured in text (Vaswani et al., 2017). Recently, attention mechanisms for encoder–decoder architectures using feed-forward neural networks instead of RNNs have been introduced. This facilitated more effective

learning of long-term dependencies in the text and parallelisation of computations, accelerating the learning process (Vaswani et al., 2017). Furthermore, attention mechanisms were used to develop architectures called transformers, which achieved state-of-the-art results in NLP tasks.

Briefly, the original transformer proposed by Vaswani et al. (2017) is a neural network in an encoder–decoder architecture, where both the encoder and decoder employ multi-headed attention. Thus, the encoder maps an input sequence (e.g. token embeddings of the original text) into a continuous multidimensional representation. The decoder receives this continuous representation and generates an output sequence. Multi-headed attention corresponds to several parallel attention layers (Vaswani et al., 2017). Attention mechanisms endow neural networks with the capacity to focus 'more' on specific parts of the inputs. For instance, when performing a machine translation for the phrase 'the dog ate the food, but it was already cold' to French ('*le chien a mangé la nourriture, mais elle était déjà froide*'), it is important to determine what the word 'it' refers to: is it 'the dog' or 'the food'? As the word 'food' seems to be the correct answer, a properly trained attention mechanism will assign a higher weight to this word when generating the translation '*elle*' for the word 'it'. By creating multi-headed attention, transformers can focus on different sections of the input more richly.

Examples of transformers are BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and GPT-2 (Radford et al., 2018). Transformers facilitate the learning of contextualised embeddings, which are highly dimensional and continuous representations for words that retain the semantic meaning, as already achieved by classic embedding techniques such as Word2Vec (Mikolov et al., 2013), and maintain the context of words. For instance, using transformers will provide different representations for the word *bank* depending on whether it refers to a fish bank or a financial establishment. For a detailed review of the numerical representations of text data, please refer to the *related work* section in Martin et al. (2019).

The training of a transformer typically follows two phases: pre-training and fine-tuning. Pre-training is typically performed on large corpora, and it focuses on tasks that may not have direct applicability to industrial problems but that enable the network to learn representations of language. For instance, BERT was pre-trained on masked language modelling and next-sentence prediction. In other words, it was pre-trained to 'fill the blanks' of hidden words in sentences and to determine whether one sentence follows another. Although these tasks may have limited applicability in other domains, they enable BERT to encapsulate useful language

representations. The fine-tuning stage consists of adapting the pre-trained weights of the transformer to a specific task, such as sentiment analysis. Thus, transformers harness the already learned language representations to rapidly learn a new task. This is generally referred to as transfer learning (Pan and Yang, 2010). Figure 6.1 shows a simplified example of the functioning of a transformer. Here, the example shows BERT trained to classify whether the description of a machine problem will stop production (severity prediction). This figure was inspired by the illustrations provided on the original BERT paper (Devlin et al., 2018), the post by Alammar, (2018), and the eBook by McCormick (2020). Five key steps are delimited in the figure to enable better comprehension.

In Figure 6.1, step 1 shows the scenario of an operator reporting a problem of an oil leakage on a particular machine. This can be achieved, for instance, through an MES deployed on the shopfloor. Step 2 shows how the raw text is pre-processed. This pre-processing step encompasses three substeps:

1) Appending two special tokens: the '[CLS]' token at the beginning of the sentence and the '[SEP]' token at the end. These extra tokens are specific to the inner operations of the transformer. Here, the [CLS] token encapsulates document embeddings, which are used for classification. The [SEP] token can be ignored as it is used for applications requiring two separate documents.
2) Tokenizing the text using a predefined strategy. Here, BERT uses the WordPiece tokenization proposed by Wu et al. (2016). This special tokenization generates words and sub-words, mitigating the risk of finding 'out-of-vocabulary' words. For example, the word 'leakage' was split into two tokens: 'leak' (word) and '##age' (sub-word). All possible words and sub-words are fixed in a predefined vocabulary, which for BERT accounts for approximately 30,000 tokens.
3) Mapping each token to its respective token embeddings, which is a 768-dimensional representation.

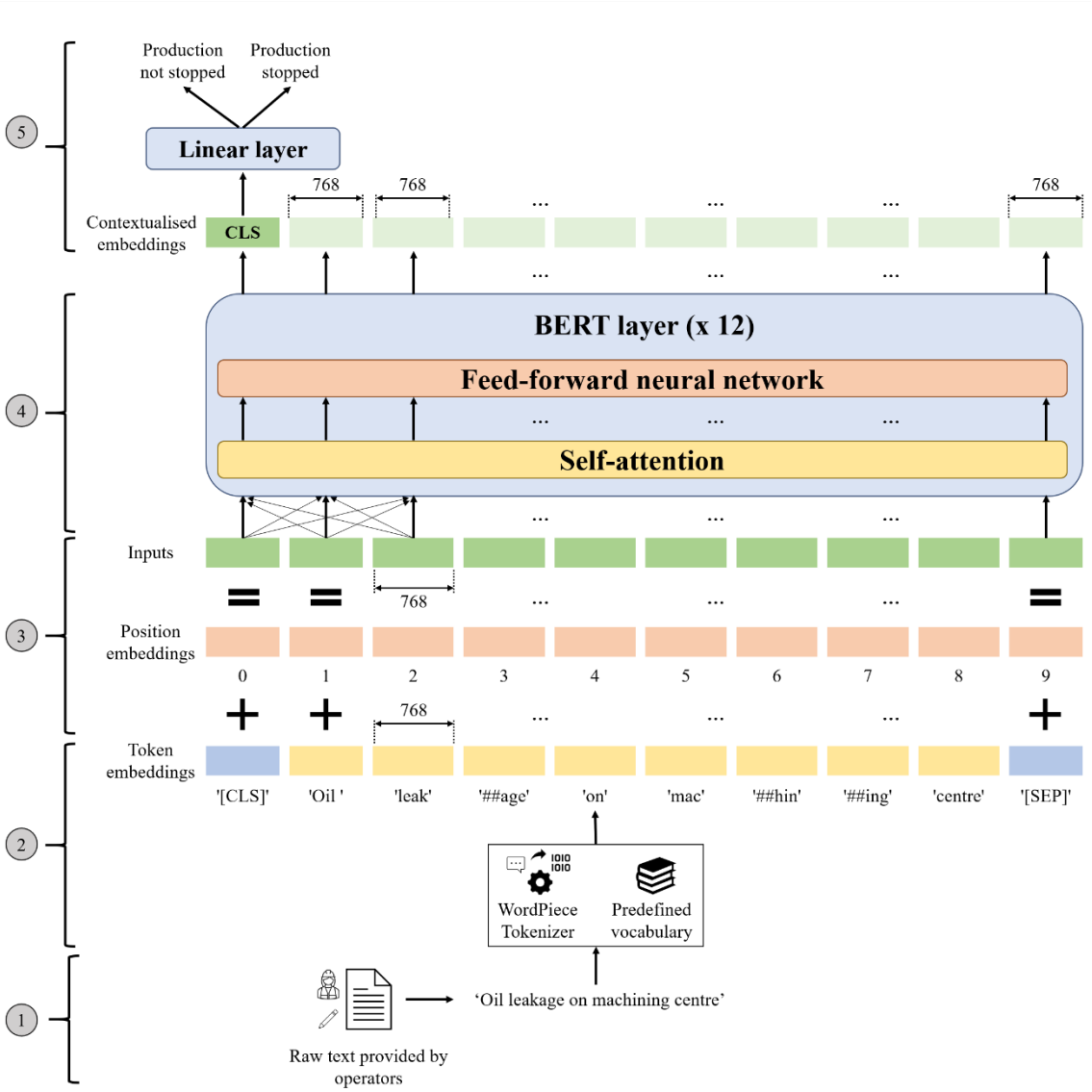


Figure 6.1 Example of a trained BERT for text classification.

Step 3 shows the creation of position embeddings, which enables BERT to account for word order in the text. The final input is the sum of the token and position embeddings. Segment embeddings are not shown in the figure nor explained because the applications using two separate documents are beyond the scope of this paper. Step 4 shows that BERT is a series of 12 stacked layers primarily containing a self-attention mechanism and a feed-forward neural network. The self-attention mechanism has the main task of refining the embeddings of each token with the context provided by the other words in the input. Thus, this will learn which words are more meaningful to others to modify their embeddings. For example, the token 'centre' would be expected to significantly affect the embeddings of tokens such as 'mac', '##hin', and '##ing', as 'centre' modifies the role of the term 'machining' from a verb to an

adjective. The crossing arrows between the inputs at the BERT layer indicate that BERT considers the context of words in bidirectionally to adjust their embeddings: a term will be affected by the words written before and after it. Finally, step 5 shows that the outputs of the 12 BERT layers are contextualised embeddings in a 768-dimensional space for each token. As the objective is to perform classification, only the embeddings for the [CLS] token are retained. These embeddings are then passed through a linear layer whose outputs can be used to determine the predicted class.

Finally, note that BERT and the transformers that are explored in this paper are called *autoencoding models* as they only rely on the encoder of the original transformer proposed by Vaswani et al. (2017), and the decoder is discarded.

After a brief explanation of attention mechanisms and transformers, it is essential to provide an overview of the approaches to handle class imbalance. According to Johnson and Khoshgoftaar (2019), there are three main families of techniques that mitigate the effect of class imbalance when training ML models:

1) Data-level methods: They modify the data distributions to reduce the class imbalance. Such methods include random oversampling (ROS) and random undersampling (RUS). While the former resamples data by replacing the minority classes up to a predefined level, the latter undersamples the majority classes. Artificial data generation through techniques such as SMOTE (Chawla et al., 2011) and data augmentation (Shorten and Khoshgoftaar, 2019) are also considered data-level methods.
2) Algorithm-level methods: These methods do not modify the data distribution. Instead, they increase the importance of minority classes in the loss function when performing training. For instance, when the class weights are assigned to each class in the dataset. Thus, wrongly predicting an observation of the minority class is penalised more than misclassifying an instance of the majority class.
3) Hybrid methods: These methods use a combination of the other two to reduce the effect of class imbalance.

6.2.2. Related work in predictive maintenance

In a previous study by Usuga Cadavid et al. (2020b), the authors explored the use of a transformer (i.e. CamemBERT) on a smaller variation of the dataset provided by Company A. The objective was to estimate the severity and breakdown duration from the description provided by the operators. The transformer was used in two different forms: fine-tuning and feature extraction. Fine-tuning involves adapting the existing weights of the model to the objective tasks, whereas feature extraction uses the model in its pre-trained form to extract embeddings for each document. Two classic ML models, random forest (RF) and logistic regression trained with term frequency-inverse document frequency (TF-IDF), were used for comparison. For these classic ML models, ROS and class weighting were used to mitigate the effect of class imbalance. For the transformer, only ROS was employed. The results suggested that fine-tuning the transformer model with ROS provided the best performance, surpassing the classic ML models.

In the aeronautics industry, the research by Kao et al. (2018, 2019) addressed the automatic recognition of parts mentioned in maintenance logs with the final objective of normalising them. Typically, operators do not use a standard language to name parts in their reports. For example, 418 spelling variants were observed for ten generic terms such as ‘circuit’ or ‘antenna’ in the dataset (Kao et al., 2019). Hence, these studies aimed to automatically discover part names when mentioned in a report (Kao et al., 2018) and their normalisation to a canonical form (Kao et al., 2019). Although these studies showed the advantages of valuing historical maintenance records, they mainly focused on data exploration and pre-processing. Also, the proposed approaches relied on some handcrafted rules that need to be adapted if the context changes, such as when using data from another company, and did not consider the case of class imbalance, as for rare part names. Therefore, their solution is unsuitable to the expected results of this paper, which are predicting severity, breakdown durations, and causes using methods able to tackle the class imbalance and needing less manual adaptations when the context changes.

Pham et al. (2020) explored handwriting recognition of reports in aeronautics. This is a challenging use case, as these reports are typically written with messy calligraphy, and they contain external noise such as validation stamps or symbols. To recognise the text, the authors employed a two-stage approach: first, they segmented the different words with a region-based

fully convolutional network (R-FCN), and second, they used a model called CTCSeq2Seq to perform text recognition. The framework proposed by this paper may inspire future work targeting companies where the use of information systems is rare, and paper reports are common. Thus, the work by Pham et al. (2020) can be used to digitise the reports to subsequently exploit the free-form texts through predictive models. Finally, Raheja et al. (2006) proposed a detailed framework for an application coupling data fusion and data mining in the context of condition-based maintenance. To illustrate the framework, the authors exemplified the key points in the scenario of a helicopter aft transmission and a gear crack. They suggested that harnessing historical data with data mining is valuable for condition-based maintenance as it enables the identification of meaningful variables and relationships. For instance, by aiding to construct a fault tree, the failure modes for critical components can be understood better. Although this study addressed the important challenge of performing data fusion, it lacked a case study exploiting a real industrial dataset to validate the proposed approach.

In the electric power industry, studies have primarily focused on the analysis of wind turbines using sensor data. This trend is probably because wind turbines are generally equipped with supervisory control and data acquisition (SCADA) systems and condition monitoring systems (CMSs), which facilitate the analysis as data are stored by default. In this domain, applications typically focused on automatically identifying abnormal functioning (e.g. 'healthy' or 'warning'): Koltsidopoulos Papatzimos et al. (2018) used a Gaussian support vector machine (SVM) trained on CMS and SCADA data to identify irregularities, Orozco et al. (2018) employed 948 GB of SCADA data to train a linear regression for estimating the component temperature in a wind turbine gearbox to identify possible malfunctioning, and Leahy et al. (2018) proposed a framework to automatically identify periods resulting in faults on stored SCADA data to then train an ensemble of decision trees to identify such periods in new scenarios. In this study, the authors chose to use RUS to mitigate the class imbalance. However, no other technique for class imbalance was assessed, which can be a shortcoming, as RUS may discard relevant information from the dataset, thus hurting the models' performance. Finally, in the domain of hydroelectric energy, Edwards et al. (2008) used 12 years of written maintenance reports of three pump stations in a dam to find clusters and identify whether a report would result in an expected or unexpected intervention. Although their approach used classic ML models such as decision trees and neural networks with TF-IDF and singular value decomposition, the authors had to heavily pre-process and clean the data by hand to achieve

exploitable results. For example, as the dataset was relatively small (842 records), spell-checking was performed manually in Excel. Nevertheless, this could be cumbersome when larger datasets are used.

From these papers in the electric power industry, one of the reasons that may contribute to the extensive usage of sensor data in predictive maintenance applications is the existence of specialised systems already installed in machines (e.g. SCADA systems) that ensure data availability. Nevertheless, it is critical when such systems are not previously installed, as companies must invest in sensors, infrastructure and skilled labour to develop and maintain them. In such cases, harnessing free-form text data may be an opportunity to value already collected data, as in the case of (Edwards et al., 2008). However, methods proposed in (Edwards et al. 2008) do not apply for this research, as performing manual spell-checking would be too time-consuming for the studied datasets containing thousands of observations. Thus, this paper needs to explore approaches that are less sensitive to spelling mistakes, reducing the burden of text pre-processing.

In the defence industry, Bruno et al. (2019) aimed to extract actions (e.g. 'fix') and objects (e.g. 'mechanical part') from approximately 10 million maintenance logs. The authors reported that some of the challenges of using free-text reports are that data are imbalanced, operators use non-standard words, descriptions tend to be too short (a median of 11 words), and that there may be conflicting labels (i.e. same text but different labels). To achieve their objective, the authors used an SVM using two different vectorisation strategies: TF-IDF for actions and Word2Vec for objects. Despite the promising results, TF-IDF and Word2Vec are techniques producing non-contextualised vector representations of texts that may be sensitive to spelling variations. Such limitation is to be addressed in this paper, as maintenance reports often contain spelling variations of words, and context should be considered to provide richer text vectorisations. Finally, the research by Nixon et al. (2018) focused on the use of sensor data from diesel engines of military assets to classify them into their respective failure modes. They employed a linear discriminant analysis–naïve Bayes classifier. The study also highlighted the fact that the dataset presented a class imbalance. For example, the authors mentioned that critical problems resulting in engine downtime represented only 5% of the dataset. However, no action was taken to tackle the effect of class imbalance. Indeed, this topic should be more frequently addressed in research, as it allows to improve the performance of ML models predicting rare events, which may be of the utmost importance in production.

Regarding industry-independent studies, Ansari (2020) aimed to perform knowledge discovery from text to support decision-making by creating an expert system that extracts elements such as total cost and required time to fix a problem from written maintenance reports. Although their approach was validated through a demonstrator, their application assumed the existence of a database containing the names and costs of parts, materials, and human actions associated with maintenance activities. This assumption may be difficult for large companies, where building and maintaining such a database may be too difficult. Sexton et al. (2017) stated that maintenance logs are collected; however, they are often not used for future diagnosis. To address this, their research aimed to automatically enrich reports written by operators by classifying the words into three categories: items, problems, and solutions. They employed an SVM using Word2Vec embeddings to perform the classification. Their approach required constant aid from industry experts to effectively tag the words into different categories, which may hinder the maintenance of the model if updates are required. Finally, Traini et al. (2020) trained a neural network with sensor data from a milling process to predict tool wear and remaining useful life. Although their study achieved excellent results, it was heavily oriented toward data fusion from different sources and pre-processing of sensor data, not considering other possible data forms such as text, tabular, and image data. Indeed, production systems rarely rely on only one type of data acquisition system or format. Hence, being able to perform data fusion from several sources with different natures is essential to create more capable systems.

Studies using sensor data often reported that the common obstacles are the high data volume, large variety of variables coming from multiple sources, and heavy pre-processing required to clean the data and generate labels to use supervised learning algorithms. The articles using text data all employed classic approaches such as TF-IDF, expert systems, Word2Vec, etc. They also presented heavy pre-processing pipelines to achieve exploitable data owing to their noisy nature. Furthermore, in some scenarios, these pipelines required specific knowledge in linguistics, such as in the processing loop presented by Kao et al. (2018) or participation from industry experts, as in Sexton et al. (2017).

Only four studies (Edwards et al., 2008; Leahy et al., 2018; Nixon et al., 2018; Bruno et al., 2019) mentioned the problem of class imbalance. Leahy et al. (2018) applied RUS to mitigate it, and Bruno et al. (2019) manually adjusted a threshold when training a TF-IDF vectorizer to solve the imbalance between action and object words.

We observed that papers on maintenance logs (both from sensors or written reports) tend to focus on data pre-processing (e.g. normalisation, generation of labels, and data fusion). Instead, comprehensive research aimed at mitigating the effect of class imbalance in maintenance or interpreting the results provided by ML models seems to be lacking in scientific literature, despite the imbalance being a classical characteristic of maintenance records. Thus, this paper proposes the use of transformers, which should lighten the necessity for heavy data pre-processing to focus on class imbalance mitigation and interpretability.

Although previous research conducted by Usuga Cadavid et al. (2020b) validated the superiority of fine-tuned transformers with ROS, this research extended the previous contribution by addressing the following points: First, only one dataset, a single transformer model, and a less exhaustive optimisation of classic ML models for comparison were originally employed to obtain the results. Thus, external validation is required by testing the same methodology on a wider variety of datasets, other transformers, and by using a more exhaustive optimisation of classic ML models. This is a critical step in generalising conclusions. Second, only ROS for transformers was employed to mitigate the effect of class imbalance. Other techniques, such as RUS and class weighting, must be tested. Finally, the interpretability of predictions and the extraction of patterns lacking in previous research are introduced in this paper. As stated by Ansari (2020), knowledge discovery from text data in maintenance is a topic that remains unexplored.

6.3. Materials and methods

This section characterises the three datasets employed for the study and the companies supplying them. Subsequently, a description is provided for the data pre-processing steps, the techniques to be used, and the training policy.

6.3.1. Characterisation of datasets and companies

The use of several datasets from different companies slightly reduces the risk of non-generalizable conclusions, but it requires an understanding of the differences between these companies. For confidentiality, we cannot provide the names of the companies or a detailed description of their activities. However, the framework proposed by Slack et al. (2007) was used to provide a typology of each industrial operation process. This framework encompasses four main characteristics: volume, variety, variation in demand, and visibility. Volume

measures the level or output rate from a process, variety assesses the diversity of products and processes, variation corresponds to how much demand for products or services varies over time, and visibility evaluates how much of the value-added activities occurs in the presence of the customer. For the characterisation used in this study, only the volume, variety, and variation in demand were used, as they are directly related to the production system.

Figure 6.2 shows this characterisation and the histograms for the length of the text descriptions for the three companies. Also, to better understand each company's dataset, Table 6.2 presents some descriptive statistics for the length of the text descriptions and the vocabulary size. The text descriptions were obtained from the maintenance logs through a simple string concatenation of the name of the concerned machine, maintenance symptoms, and other free-text details. This final concatenated string is denoted as the input sequence, and it served as the sole input for training the deep learning models. This decision was made to demonstrate that deep learning models require a few handcrafted variables to perform well in learning tasks. As the corpora were in French, the tokenization of the text descriptions to obtain the length statistics was performed using the CamemBERT tokenizer (Martin et al., 2019). This tokenizer was designed for the French language by Facebook AI Research, the French National Institute for Research in Computer Science and Automation (INRIA), and Sorbonne University. Finally, the vocabulary size for each dataset is also indicated. This enabled us to quantify the richness of the vocabulary employed by the operators of each company. To obtain this metric, the base tokenizer from SpaCy for the *fr_core_news_sm* pipeline was used to obtain the unigrams.

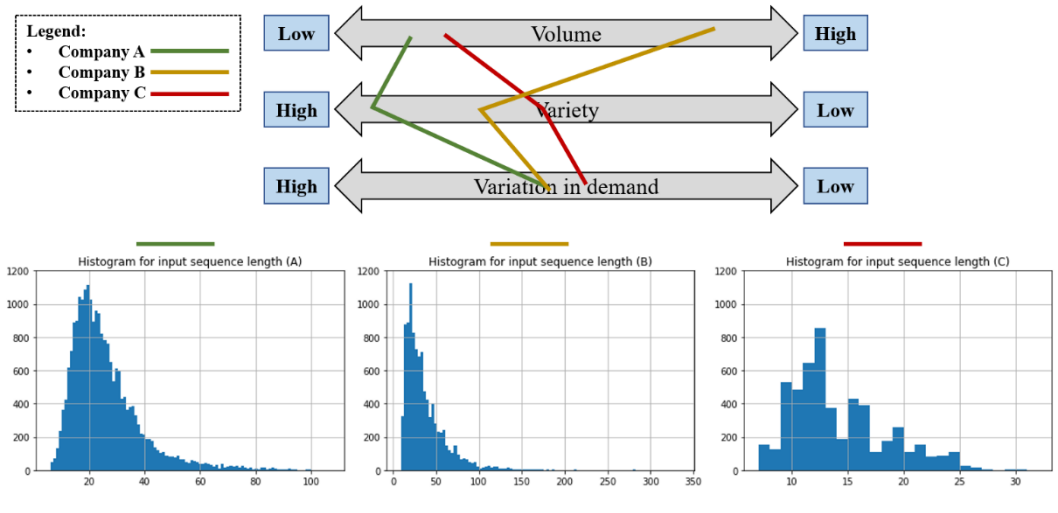


Figure 6.2 Characterisation of the three companies and their datasets. The tables below each histogram show the descriptive statistics for the input sequence lengths and the vocabulary size.

Company A		**Company B**		**Company C**	
Count	22709	Count	9993	Count	5317
Mean	26.2	Mean	36.4	Mean	13.6
St. dev.	12.6	St. dev.	24.4	St. dev.	4.3
Var. coeff.	0.48	Var. coeff.	0.67	Var. coeff.	0.31
Min length	6	Min length	9.0	Min length	7.0
Max length	107	Max length	335.0	Max length	32.0
Vocab. size	15264	Vocab. size	12550	Vocab. size	1134

Table 6.2 Descriptive statistics for the input sequence lengths and vocabulary sizes for each dataset.

Company A, belonging to the aeronautics industry, is characterised by low volumes, high product variety, and relatively controlled demand. Its maintenance logs were the largest dataset, with approximately 22000 reports. The distribution for the length of maintenance reports followed a right-skewed distribution, using, on average, 26.2 tokens to describe a problem. In addition, these maintenance logs contained the largest vocabulary, with 15264 tokens. Company B, in the electronics industry, has the highest production volume, medium product variety, and variation in demand. It provided the second largest dataset with the second largest vocabulary (12550 tokens). Although it contained the longest average description length with 36.4 tokens, it had the highest variability, with a variation coefficient of 0.67. This can also be observed with its severe skew to the right. Finally, Company C belongs to the aeronautics industry. It has relatively low production volumes, with a medium level of both variety and

variation in demand. It had the smallest dataset and vocabulary (1134 tokens) with the shortest average description length (13.6 tokens). However, it seemed to have the most stable description length, with a variation coefficient of 0.31.

6.3.2. Data pre-processing

6.3.2.1. Text pre-processing

In addition to cleaning some descriptions with an abnormal structure (e.g. several symbols stacked in a row or comments containing only a dot '.') and the creation of the input sequence through concatenation, no other text pre-processing tasks were conducted for the data used in the transformers. The choice of not further refining the data, such as part normalisation and spell-checking, aimed to check whether transformers can manage noisy text data with little requirement for pre-processing. After the creation of the input sequence, it was passed through a predefined tokenizer specific to each transformer. As shown in Figure 6.1, this tokenizer processed the text in a compliant manner with a predefined vocabulary, which was also specific to the transformer. Because transformers have their own predefined tokenizer and vocabulary, the creation of handcrafted text-processing pipelines is significantly simplified. For instance, this removed the burden of exploring different tokenizing strategies, removing stop words, performing lemmatization, and identifying n-grams.

As this study also compared the performance of transformers with respect to classic ML models, three tokenizers with their respective TF-IDF vectorizers were created to train the models:

1) Plain tokenizer (Plain_tokenizer): a base tokenizer that split the text based on rules such as white spaces or apostrophes.
2) Tokenizer with lemmatization (Lem_tokenizer): It was built based on the Plain_tokenizer. This tokenizer performed lemmatization, which consisted of transforming words into their canonical form or *lemma*. For example, the lemma for the word *cars* would be *car*.
3) Tokenizer with lemmatization and stop word removal (Lem_Stop_tokenizer): It was built based on the Lem_tokenizer. This tokenizer performed lemmatization and removed stop words, which are common words that have low semantic value.

Each of the three tokenizers also included a pre-processing step in which all non-word characters and hyperlinks were removed using regular expressions. The TF-IDF vectors were

limited to the 768 most relevant features. This choice was inspired by the size of the vectors produced by the last layer of the transformers evaluated in this study. Finally, an n-gram range from uni- to trigrams (1 to 3) was employed. For the tokenizers, a pipeline for French called *fr_core_news_sm* from the library SpaCy was used.

6.3.2.2. Target variables

A final pre-processing step was performed for the breakdown-duration prediction task. The distributions of breakdown durations are highly imbalanced in maintenance datasets, as most maintenance actions require less time to be performed compared with severe scenarios requiring a significant amount of time. Thus, to mitigate the class imbalance and following previous research (Usuga Cadavid et al., 2020b), the breakdown durations were grouped into clusters using k-means clustering. This facilitated the creation of more balanced classes of durations that were close to each other. To identify these clusters, the following steps were performed:

1) Identifying atypical values for the breakdown duration: This step was required to obtain more balanced classes using k-means, as this model is sensitive to outliers. Thus, values higher than the third quartile plus 1.5 times the interquartile range were considered as outliers. These values were excluded from the dataset and placed into a special cluster named the 'expert' cluster, meaning that they normally engage in high breakdown durations and must be assessed by an expert.
2) Training a k-means algorithm: The dataset without outliers was used to train a k-means algorithm. To select the value of k, silhouette diagrams were employed (Rousseeuw, 1987). These diagrams provided an overview of the cluster sizes as well as a measure of clustering quality. The cluster size was represented by the size of the bar, and the clustering quality was measured using the silhouette coefficient. This coefficient ranges from -1 to 1, where 1 means that the instances are well inside the cluster, 0 means that some instances are close to the boundaries, and -1 means that the instance should belong to another cluster (Géron, 2019). Aiming to obtain good quality clusters and tackle the effect of class imbalance, the chosen number of clusters was the one presenting a good silhouette coefficient while yielding similar cluster sizes, assessed through visual inspection of the silhouette diagrams.

3) Assigning clusters: The trained k-means algorithm was used to assign clusters to the dataset. Subsequently, the observations identified as atypical values in step 1 were added to an extra cluster called the ‘expert’ cluster.

Figure 6.3 shows the first two steps of the proposed data pre-processing for Company A (a) and the results of the second step for Companies B (b), and C (c). Table 6.3 presents the results for the third step for Companies A, B, and C. The obtained breakdown-duration ranges and relative size of each cluster are also provided.

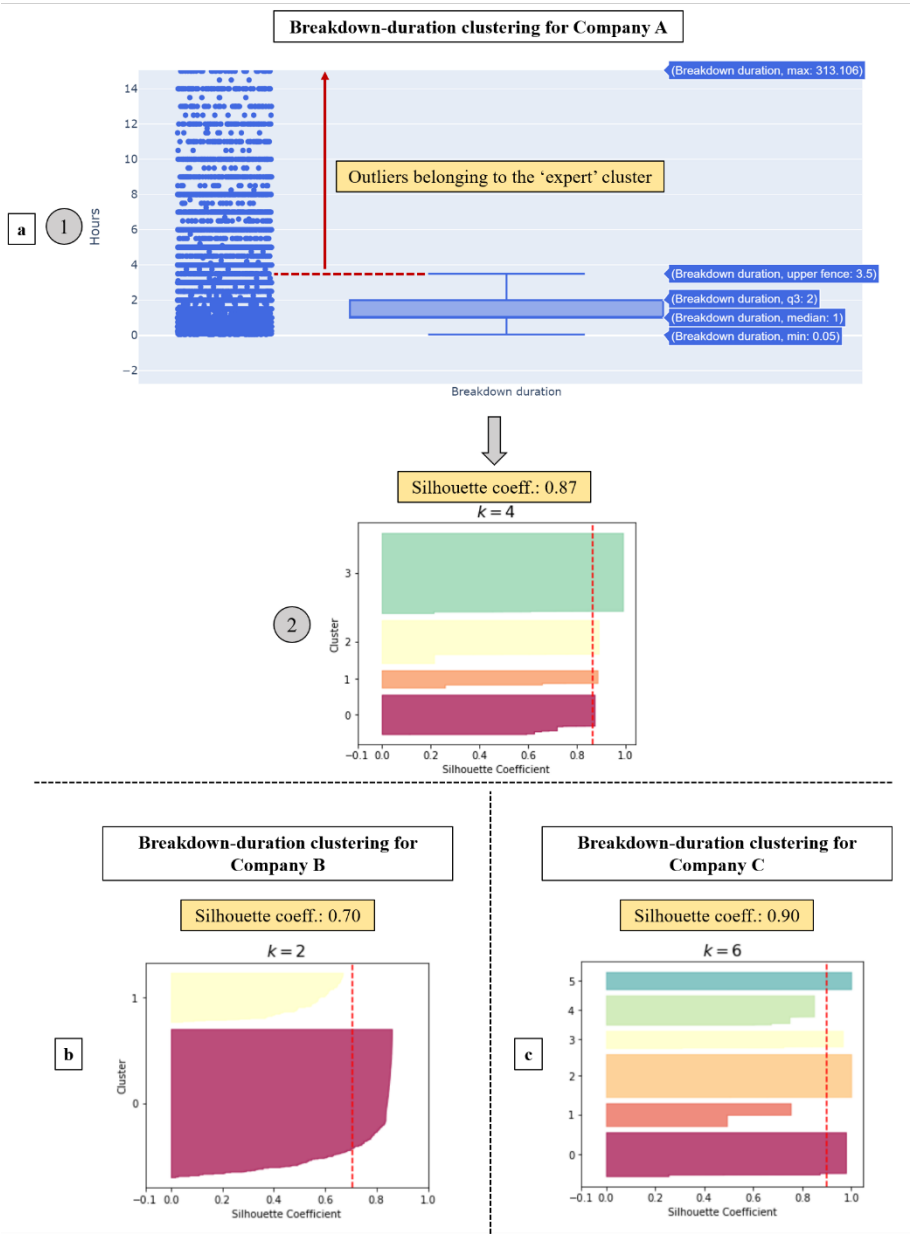


Figure 6.3 First two pre-processing steps for breakdown-duration clustering in Company A (a), and results of the second step for Companies B (b) and C (c).

Company A		
Cluster	**Range of values (hours)**	**Percentage of observations**
0	≤ 0.7	18.2%
1	2.5 – 3.5	8.0%
2	1.45 – 2.3	19.8%
3	0.75 – 1.4	37.1%
4 (‘expert’)	≥ 3.5	16.9%
Company B		
Cluster	**Range of values (minutes)**	**Percentage of observations**
0	≤ 5880	68.9%
1	6008 – 17305	22.8%
2 (‘expert’)	≥ 17305	8.3%
Company C		
Cluster	**Range of values (minutes)**	**Percentage of observations**
0	45 – 63	23.4%
1	150 – 180	12.2%
2	120 – 135	22.9%
3	210 – 270	9.2%
4	≤ 35	15.6%
5	78 – 90	9.2%
6 (‘expert’)	≥ 270	7.5%

Table 6.3 Results of the third pre-processing step for each company.

6.3.3. ML, imbalance mitigation and interpretation techniques to be used

6.3.3.1. ML techniques to be compared

As the text data used in this study were in French, two recent language-specific models were used: CamemBERT (Martin et al., 2019) and FlauBERT (Le et al., 2019). The superiority of language-specific models over multilingual models has been demonstrated in several studies (Le et al., 2019; Martin et al., 2019). Multilingual versions of BERT (mBERT) are not trained on as much data as their language-specific counterparts. For example, while the mBERT version for French was pre-trained on 57 GB of French text data, CamemBERT and FlauBERT were trained during hours on 138 and 71 GB of text, respectively.

Note that CamemBERT and FlauBERT are similar models based on the BERT architecture (CamemBERT is based on RoBERTa, which is based on BERT). A thorough review of the

differences between CamemBERT and FlauBERT is beyond the scope of this paper. However, their key distinctions can be summarised in the following two aspects:

1) Number of parameters: CamemBERT uses 110 million parameters versus 137 million for FlauBERT.
2) Tokenization strategy: CamemBERT uses a tokenization strategy called SentencePiece (Kudo and Richardson, 2018) with a vocabulary size of 32,000 sub-word tokens, whereas FlauBERT uses byte pair encoding (Sennrich et al., 2015) with 50,000 sub-word tokens. Both tokenization strategies seek to generate words and sub-words to reduce the risk of obtaining 'unknown' or 'out-of-vocabulary' words. For instance, tokenizing the phrase 'the leftmost panel was overheating' with byte pair encoding would result in the following tokens: ['the', 'left', '##most', 'panel', 'was', 'overhe', '##ating']. Thus, 'leftmost' and 'overheating' are split into sub-words. This enables the new words to be managed by breaking them into known fragments.

For further information on this subject, please refer to (Le et al., 2019), which presents the differences between BERT, RoBERTa, CamemBERT, and FlauBERT.

This study performed fine-tuning to adapt and compare these two state-of-the-art models on tasks related to maintenance to support decision-making in production. For each company, the hyperparameters used to perform the training were kept the same for both models to ensure a fair comparison. Furthermore, this study did not consider hyperparameter optimisation for transformers as the objective was to demonstrate that these models can achieve good performance without exhaustive optimisation. Nevertheless, future research will explore this topic. The hyperparameters are summarised in Table 6.4.

Global hyperparameters/company	**Hyperparameters for CamemBERT and FlauBERT**
Global hyperparameters	• Optimizer: AdamW (Loshchilov and Hutter, 2017) with a learning rate of $2 * 10^{-5}$ and an epsilon of $1 * 10^{-8}$ to prevent division by zero. • Epochs: 3 • Learning rate warm-up with zero warm-up steps
Specific for Company A	• Maximum input length: 128 • Batch size: 32

Global hyperparameters/company	Hyperparameters for CamemBERT and FlauBERT
Specific for Company B	• Maximum input length: 375 • Batch size: 16
Specific for Company C	• Maximum input length: 64 • Batch size: 32

Table 6.4 Hyperparameters employed for CamemBERT and FlauBERT in each company tasks.

The table shows that the maximum input length varied for each company. This corresponded to the maximum length of their respective input sequences and a security margin. The batch size was changed for Company B because a larger maximum input length depleted the memory from the RAM. Thus, reducing the batch size reduced the allocated memory at each iteration. Finally, these two models were implemented using the transformers library from Huggingface (Wolf et al., 2019) in PyTorch. This library also provides a tokenizer and vocabulary for each transformer.

For comparison, some classic ML models were trained, optimised, and compared with the results obtained using transformers. The following models were implemented: random forest (RF), AdaBoost (Ada), XGBoost (XGB), and a linear discriminant analysis–naïve Bayes classifier (LDA). While the first three are ensemble learning models, which are known to provide excellent results, LDA is a simple model that can serve as a baseline. To optimise these models, k-fold cross-validation with a grid search was used. Table 6.5 shows the hyperparameter space to be explored and the tokenizers considered for each model.

Model name	Tokenizers	Hyperparameter space
Random Forest	• Plain_tokenizer • Lem_tokenizer • Lem_Stop_tokenizer	• Number of estimators: [50, 100, 200]
AdaBoost	• Plain_tokenizer • Lem_tokenizer • Lem_Stop_tokenizer	• Base estimator: [Decision tree, Logistic regression, Random Forest] • Number of estimators: [50, 100, 200]
XGBoost	• Plain_tokenizer • Lem_tokenizer • Lem_Stop_tokenizer	• Number of estimators: [50, 100, 200] • Learning rate: [0.3, 0.5, 1.0] • Max depth: [3, 5]
LDA	• Plain_tokenizer	• Default values from Scikit-earn library (Pedregosa et al., 2011)

Table 6.5 Hyperparameter space and tokenization strategies explored during the optimization of each model.

6.3.3.2. Class imbalance mitigation techniques to be compared

Although previous research (Usuga Cadavid et al., 2020b) suggested that ROS provides excellent results, it is important to compare it with other techniques. ROS has been demonstrated to provoke overfitting to minority classes (Johnson and Khoshgoftaar, 2019). Additionally, it can dramatically increase the training time for strongly imbalanced datasets as it artificially increases the size of the training set. Hence, this study compared ROS with another data-level technique called RUS and an algorithm-level technique using class weights in the categorical cross-entropy loss function.

For ROS, the selected strategy was to resample the minority classes with a replacement from the training set until each of them achieved the number of instances of the largest class. For RUS, the majority classes were undersampled until the number of instances of the minority class was achieved. Finally, for the class weighting scheme, the weights for each class were calculated using Equation 6.1 proposed by the implementation in the scikit-learn library (Pedregosa et al., 2011) for Python.

$$w_k = \frac{M}{K * |C_k|} \tag{6.1}$$

In Equation 6.1, w_k represents the weight attributed to the categorical cross-entropy loss function for class k, M is the number of samples, K is the number of classes, and $|C_k|$ is the number of instances for class k. Thus, w_k will be higher for minority classes and lower for

majority classes, adapting the incurred penalty of the model. Finally, Equation 6.2 presents the weighted categorical cross-entropy, as represented by Ho and Wookey (2020).

$$L = -\frac{1}{M}\sum_{k=1}^{K}\sum_{m=1}^{M} w_k * y_m^k * \log(h_k) \quad (6.2)$$

In Equation 6.2, M is the number of samples, K is the number of classes, w_k is the weight for class k, y_m^k is the correct label for observation m belonging to class k, and h_k is the model's softmax output for class k.

These imbalanced mitigation schemes were implemented for both CamemBERT and FlauBERT. The unmodified plain versions of these two models were also fine-tuned. The code was adapted from a publicly available implementation of BERT fine-tuning by McCormick and Ryan (2019). For clarity, the following abbreviations are used:

1) CamemBERT with unweighted categorical cross-entropy loss: 1) Cam_Plain
2) CamemBERT using ROS: 2) Cam_ROS
3) CamemBERT using RUS: 3) Cam_RUS
4) CamemBERT with weighted categorical cross-entropy loss: 4) Cam_Class Weight
5) FlauBERT with unweighted categorical cross-entropy loss: 5) Flau_Plain
6) FlauBERT using ROS: 6) Flau_ROS
7) FlauBERT using RUS: 7) Flau_RUS
8) FlauBERT with weighted categorical cross-entropy loss: 8) Flau_Class Weight

Note that RUS models were not trained for cause prediction in Company B, as the strategy of undersampling the majority classes to the same level of minorities would result in classes containing very few instances.

For RF and Ada, ROS, RUS, and class weighting were also implemented. Only ROS and RUS were explored for the LDA and XGB.

6.3.3.3. Interpretation technique and insight extraction method

To enable the interpretability of the model, the LIME technique was employed. It was proposed by Ribeiro et al. (2016) and is a model-agnostic technique that enables local interpretability. In other words, it can be applied to any classifier or regressor to explain single predictions. LIME does not determine the global importance of the input features on outputs. Instead, it indicates,

for a single prediction, which inputs are the most meaningful to obtain a certain answer. LIME was selected because it enables the generation of simple interpretations as well as plots that are understandable by non-specialists. This corresponds with the objective of maintaining humans in the loop of decision support systems, which is a fundamental criterion for many companies to accept new technology-based solutions (Thomas et al., 2018a). Another recent technique for interpretation is SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017). However, its comparison with LIME will be explored in future research.

In addition to generating visualisations for predictions, insights can be generated for the production system using LIME's outputs. When used for classification, LIME outputs the extent to which a certain input contributes to class probability. In the context of this study, this can be harnessed to explore, for a given machine and situation, which are the words normally associated with high average probability contributions. As an illustration, the model obtained for severity prediction in Company A was employed to determine the most relevant words associated with production line stops for a particular machine. The mean probability contribution was used to determine the most relevant words. This is described by Equation 6.3.

$$\theta_{ikm} = \frac{1}{|D_{mk}|} \sum_{j=1}^{|D_{mk}|} p_{ijk}, where\ p_{ijk} = \begin{cases} p_{ijk}, & \hat{k} = k \\ 0, & otherwise \end{cases} \quad (6.3)$$

In Equation 6.3, θ_{ikm} is the mean probability contribution of word i to obtain the class of interest k in machine m, $|D_{mk}|$ is the total number of documents belonging to machine m and class k, where $D_{mk} \subseteq D_c$ and D_c is the set of documents for company c. Finally, p_{ijk} is the probability contribution of word i in document j to obtain the class of interest k. Note that p_{ijk} is set to zero if the model prediction is incorrect, that is, when the predicted class $\hat{k}$ is not equal to k. After obtaining the values for θ_{ikm}, they are sorted to obtain the words that contribute the most to the probabilities.

The contribution of this insight-extraction method is that it provides targeted global interpretability. Thus, local interpretations are generated with the base function of LIME, while global interpretations are obtained using the proposed method. Through *global interpretations*, this research identified the most meaningful inputs in a set of observations to obtain a particular result. This means that the proposed method can be adapted to any desired level of granularity by changing the set of observations. This may be useful for maintenance planners desiring to

understand what causes a certain scenario (e.g. severe machine breakdowns) in a specific machine or group of machines from written reports.

6.3.4. Training policy

For each company, the training set was split into 75% for training and 25% for testing. Subsequently, the training set was further split into 90%, which was effectively used to train the models (reduced training set) and 10% for the validation set, employed to compare and select the best model for each task. When the best model was selected, it was retrained in the full training set (before the reduction), and its performance was tested on the test set.

The Mathews correlation coefficient (MCC) was employed to measure the performance and select the best model. Although performance in classification is typically measured using the F1-score, this indicator has been observed to be misleading when assessing classifiers trained on imbalanced datasets (Hand and Christen, 2018). Instead, the MCC is frequently preferred (Delgado and Tibau, 2019). The MCC ranges from -1 to 1, where 1 corresponds to a perfect classifier, 0 corresponds to an average random prediction, and -1 is an inverse prediction.

For classic ML models, k-fold cross-validation with a grid search was employed. As mentioned earlier, this grid search explored a hyperparameter space for each model, several tokenization strategies, and various techniques to mitigate the effect of class imbalance. This resulted in many models being trained to determine an optimised setup.

Figure 6.4 summarises the training policy and number of classes in the training set for each task and quantifies the degree of class imbalance for each dataset using the imbalance ratio. This measure was selected as it was used in an extensive survey on class imbalance with deep learning performed by Johnson and Khoshgoftaar (2019). This ratio can be obtained using Equation 6.4:

$$\rho = \frac{max_k\{|C_k|\}}{min_k\{|C_k|\}} \tag{6.4}$$

In Equation 6.4, $max_k\{|C_k|\}$ and $min_k\{|C_k|\}$ are the number of instances of the majority and minority classes, respectively.

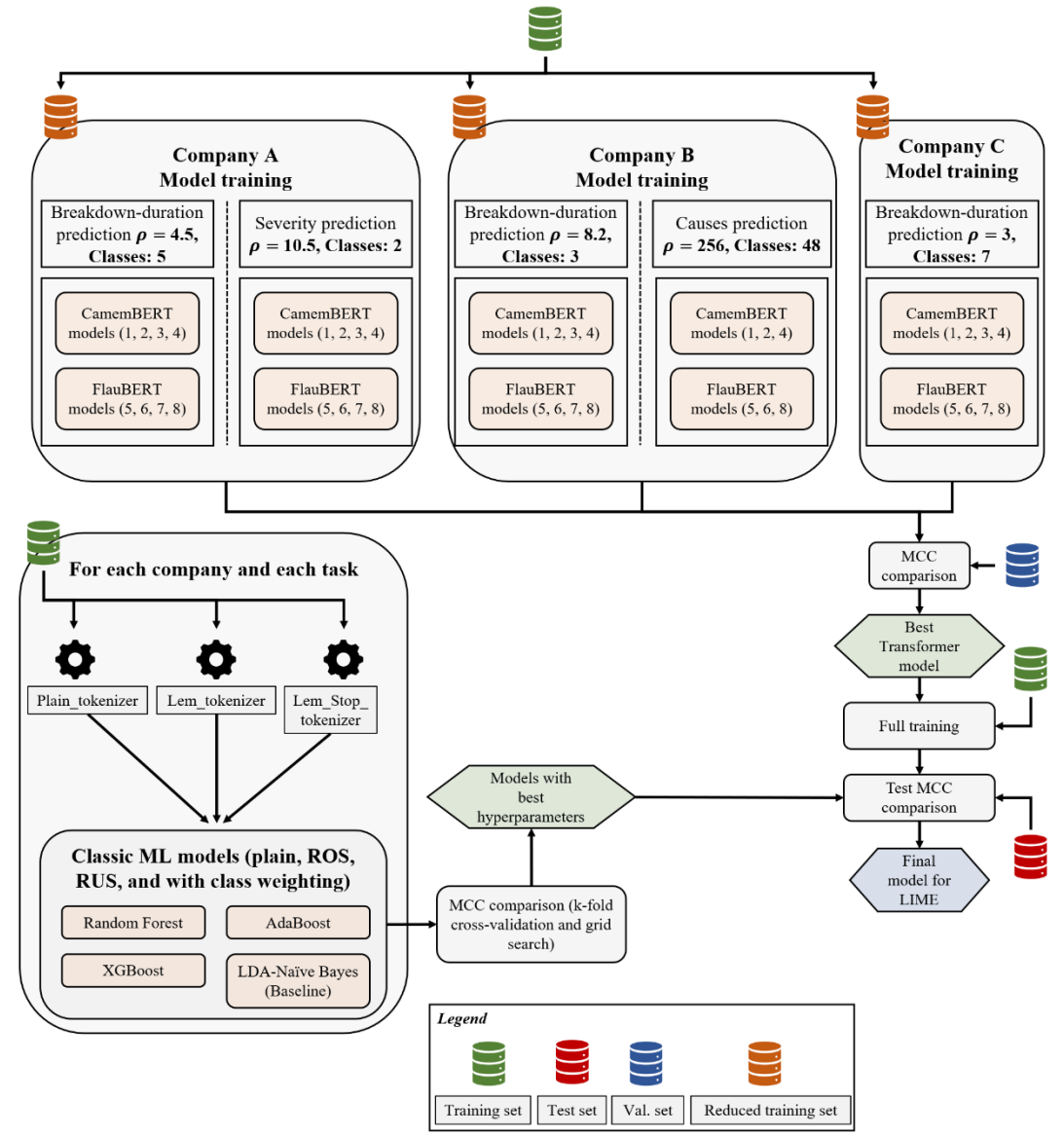


Figure 6.4 Training policy and model selection.

6.4. Results

The CamemBERT and FlauBERT models were implemented using PyTorch. To consider the uncertainty in the experiments and to provide a more reliable estimation of performance, the models were trained 20 times, and their MCC was measured. The best model was selected according to the median MCC, as the median was more robust to outliers. Finally, a t-test for the mean difference in MCC was used to perform multiple hypothesis testing between all models. Bonferroni correction was employed to control for the occurrence of false positives

when performing multiple hypothesis testing (Armstrong, 2014). An alpha level of 0.05 was used for these tests.

6.4.1. Model comparison and selection

Figure 6.5, Figure 6.6, and Figure 6.7 show the MCC box plots for Companies A, B, and C, respectively. The best model based on the median MCC is framed in a red box. The median is also provided. Finally, Table 6.6, Table 6.7, and Table 6.8 provide the results for the t-test for mean differences with Bonferroni correction. Only the hypothesis testing results for the best models are shown.

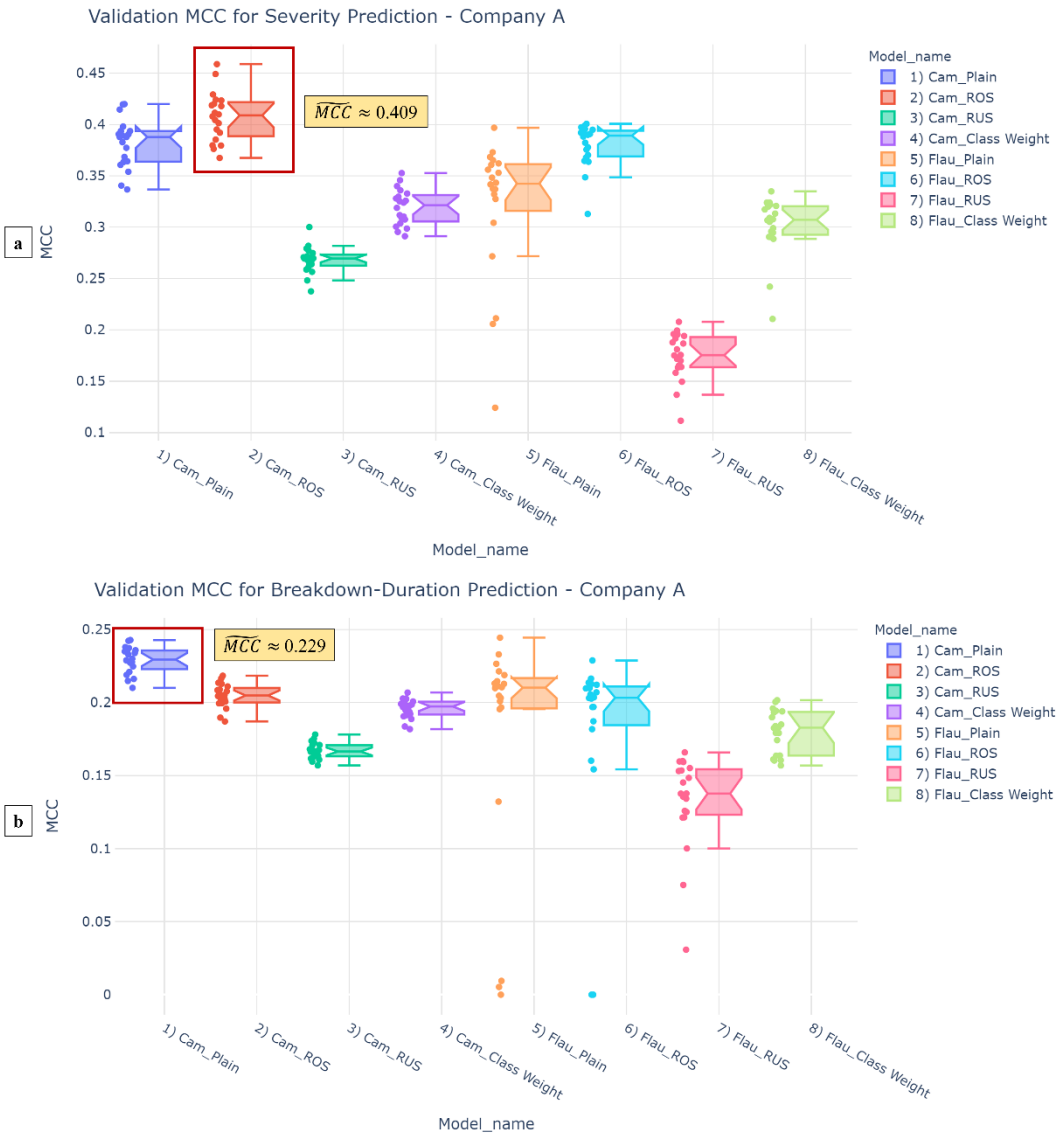


Figure 6.5 Validation MCC Box plots for (a) severity and (b) breakdown-duration prediction in Company A.

Company A		
Severity prediction (ρ= 10.5, Classes: 2)		
Best Model	**Compared to**	**p-corrected (Bonf.)**
2) Cam_ROS	1) Cam_Plain	0.04
2) Cam_ROS	3) Cam_RUS	0.00
2) Cam_ROS	4) Cam_Class Weight	0.00
2) Cam_ROS	5) Flau_Plain	0.00
2) Cam_ROS	6) Flau_ROS	0.01
2) Cam_ROS	7) Flau_RUS	0.00
2) Cam_ROS	8) Flau_Class Weight	0.00
Breakdown-duration prediction (ρ= 4.5, Classes: 5)		
Best Model	**Compared to**	**p-corrected (Bonf.)**
1) Cam_Plain	2) Cam_ROS	0.00
1) Cam_Plain	3) Cam_RUS	0.00
1) Cam_Plain	4) Cam_Class Weight	0.00
1) Cam_Plain	5) Flau_Plain	0.17
1) Cam_Plain	6) Flau_ROS	0.05
1) Cam_Plain	7) Flau_RUS	0.00
1) Cam_Plain	8) Flau_Class Weight	0.00

Table 6.6 T-test results with Bonferroni correction for Company A. P-values larger than 0.05 are highlighted in orange.

For Company A, the best models for severity prediction and breakdown-duration prediction were Cam_ROS and Cam_Plain, respectively. In severity prediction, the Cam_ROS mean MCC was statistically different from the mean MCC of other models. Nevertheless, in the breakdown-duration prediction, the plain version of CamemBERT was superior. Additionally, the mean MCC was not statistically different from the mean of Flau_Plain. This suggested that, for Company A data, using data pre-processing with k-means was seemingly sufficient to reduce the imbalance effect on the plain models compared with ROS. The decrease in ROS performance may have been caused by ROS, which tended to overfit the minority data, resulting in worse results in the validation.

The worst performance was obtained when using RUS. This was probably due to considerable information loss when undersampling the data. When the imbalance level was high (ρ= 10.5, Classes: 2) in severity prediction, class-weighting had results inferior to that of the ROS. However, for breakdown-duration prediction (ρ= 4.5, Classes: 5), class weighting results were similar to those of the ROS, even when 5 classes were considered. This implied that the effectiveness of class weighting decreased when it was employed for higher imbalance levels.

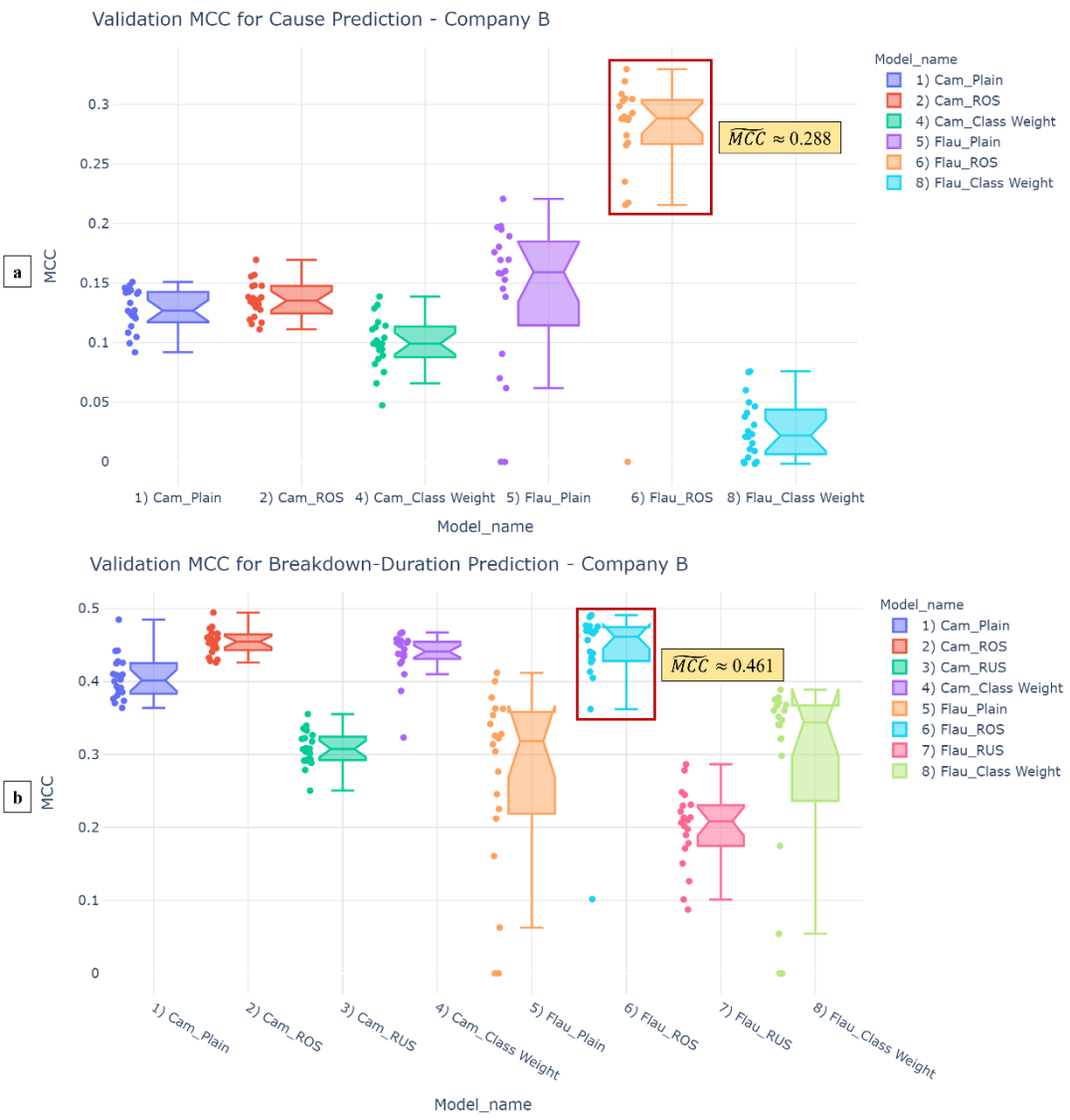


Figure 6.6 Validation MCC box plots for (a) cause and (b) breakdown-duration prediction in Company B.

Company B		
Cause prediction (ρ= 256, Classes: 48)		
Best Model	**Compared to**	**p-corrected (Bonf.)**
6) Flau_ROS	1) Cam_Plain	0.00
6) Flau_ROS	2) Cam_ROS	0.00
6) Flau_ROS	4) Cam_Class Weight	0.00
6) Flau_ROS	5) Flau_Plain	0.00
6) Flau_ROS	8) Flau_Class Weight	0.00
Breakdown-duration prediction (ρ= 8.2, Classes: 3)		
Best Model	**Compared to**	**p-corrected (Bonf.)**
6) Flau_ROS	1) Cam_Plain	1.00
6) Flau_ROS	2) Cam_ROS	1.00
6) Flau_ROS	3) Cam_RUS	0.00
6) Flau_ROS	4) Cam_Class Weight	1.00
6) Flau_ROS	5) Flau_Plain	0.00
6) Flau_ROS	7) Flau_RUS	0.00
6) Flau_ROS	8) Flau_Class Weight	0.00

Table 6.7 T-test results with Bonferroni correction for Company B. P-values larger than 0.05 are highlighted in orange.

For Company B, the best model was Flau_ROS in both scenarios. This indicated that FlauBERT seemingly adapted to Company B's maintenance log text structure. Even in scenarios like cause prediction with severe imbalance levels and a relatively high number of classes (ρ= 256, Classes: 48), ROS dominated all the models. For breakdown-duration prediction, the mean MCC for Flau_ROS was not statistically different from that of Cam_Plain. This supported the observation of Company A's results: pre-processing with k-means the durations may be sufficient to mitigate the effect of class imbalance.

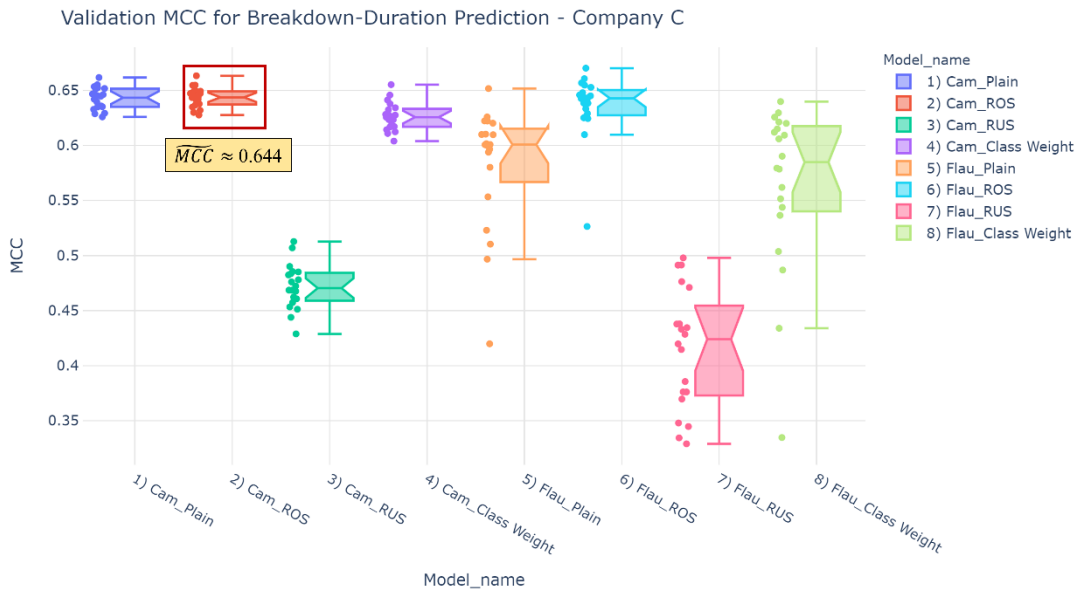


Figure 6.7 Validation MCC box plots for breakdown-duration prediction in Company C.

Company C		
Breakdown-duration prediction (ρ= 3, Classes: 7)		
Best Model	**Compared to**	**p-corrected (Bonf.)**
2) Cam_ROS	1) Cam_Plain	1.00
2) Cam_ROS	3) Cam_RUS	0.00
2) Cam_ROS	4) Cam_Class Weight	0.00
2) Cam_ROS	5) Flau_Plain	0.00
2) Cam_ROS	6) Flau_ROS	1.00
2) Cam_ROS	7) Flau_RUS	0.00
2) Cam_ROS	8) Flau_Class Weight	0.00

Table 6.8 T-test results with Bonferroni correction for Company C. P-values larger than 0.05 are highlighted in orange.

For Company C, the best model for the breakdown-duration prediction was Cam_ROS. Although this last result confirmed the superiority of ROS when addressing class imbalance, the t-test suggested that the mean MCC for Cam_ROS was not statistically different from that of Flau_ROS and Cam_Plain. This again suggested that the pre-processing step with k-means was sufficient to provide results using a plain model that are as excellent as when using ROS.

The comparison between CamemBERT and FlauBERT suggested that both models are competitive in terms of performance, as suggested by Le et al. (2019). This may be because they share a similar architecture. Thus, we recommend exploring both models when operating with French corpora. We observed that using the pre-processing step with k-means to reduce

the class imbalance for predicting breakdown duration seemed to result in plain models providing similar results as ROS. This may be useful because it avoids increasing the training time by unnecessary training models employing ROS. Nevertheless, ROS should be preferred when the extra computation cost is bearable, as it seems to globally improve performance in imbalanced classification, both for severe imbalance levels and a high number of classes. For the tasks in which no pre-processing was performed (i.e. severity and cause prediction), ROS seemed to provide statistically significant improvements in the quality of classification of the model.

Class weighting seemed to be generally inferior to ROS and to plain models in some scenarios. Nevertheless, further research may be conducted to study the behaviour of other loss functions and class weighting strategies. Finally, the results indicated that RUS provides the worst results for all configurations.

6.4.2. Test results

When the best transformer for each company and task was observed, it was evaluated on the test set. In addition, the best classic ML model and baseline were evaluated. Figure 6.8, Figure 6.9, and Figure 6.10 show the results for the test set performance in Companies A, B, and C, respectively. The previously presented validation results for transformers are also presented for comparison. The dashed red and black lines indicate the MCC test for the best classic ML model and the baseline, respectively.

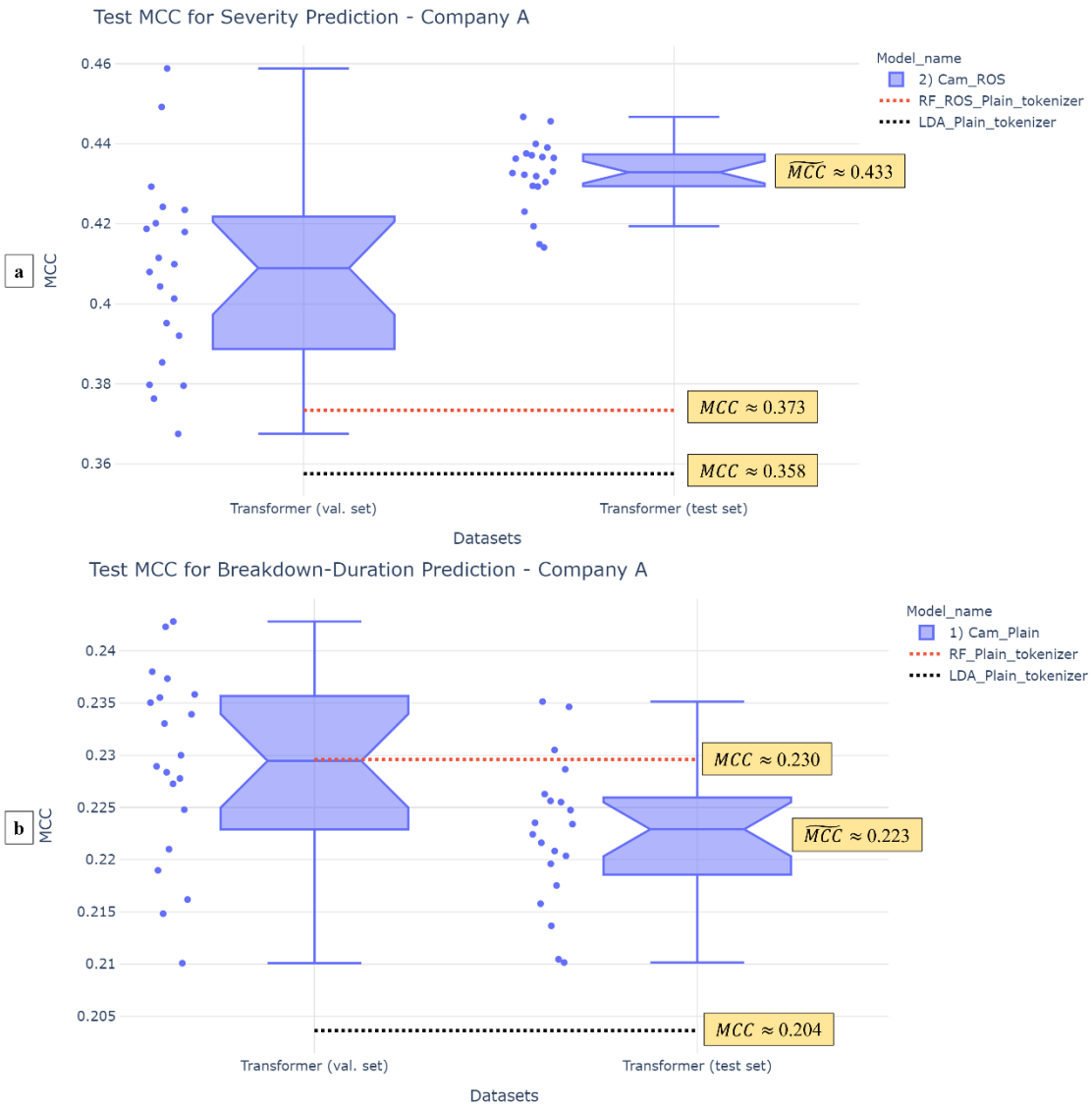


Figure 6.8 Test MCC box plots for (a) severity and (b) breakdown-duration prediction in Company A.

For Company A, transformers exhibited superior results for severity prediction. In breakdown-duration prediction, the classic ML model (RF using the Plain_tokenizer) exhibited better results than the transformer, but the difference between their MCCs was relatively small. Furthermore, Figure 6.8(b) shows that the transformer had signs of overfitting for breakdown-duration prediction, as the test performance was lower than the validation performance. If overfitting was corrected, the performance of the transformer could be improved to the same level as that of the RF.

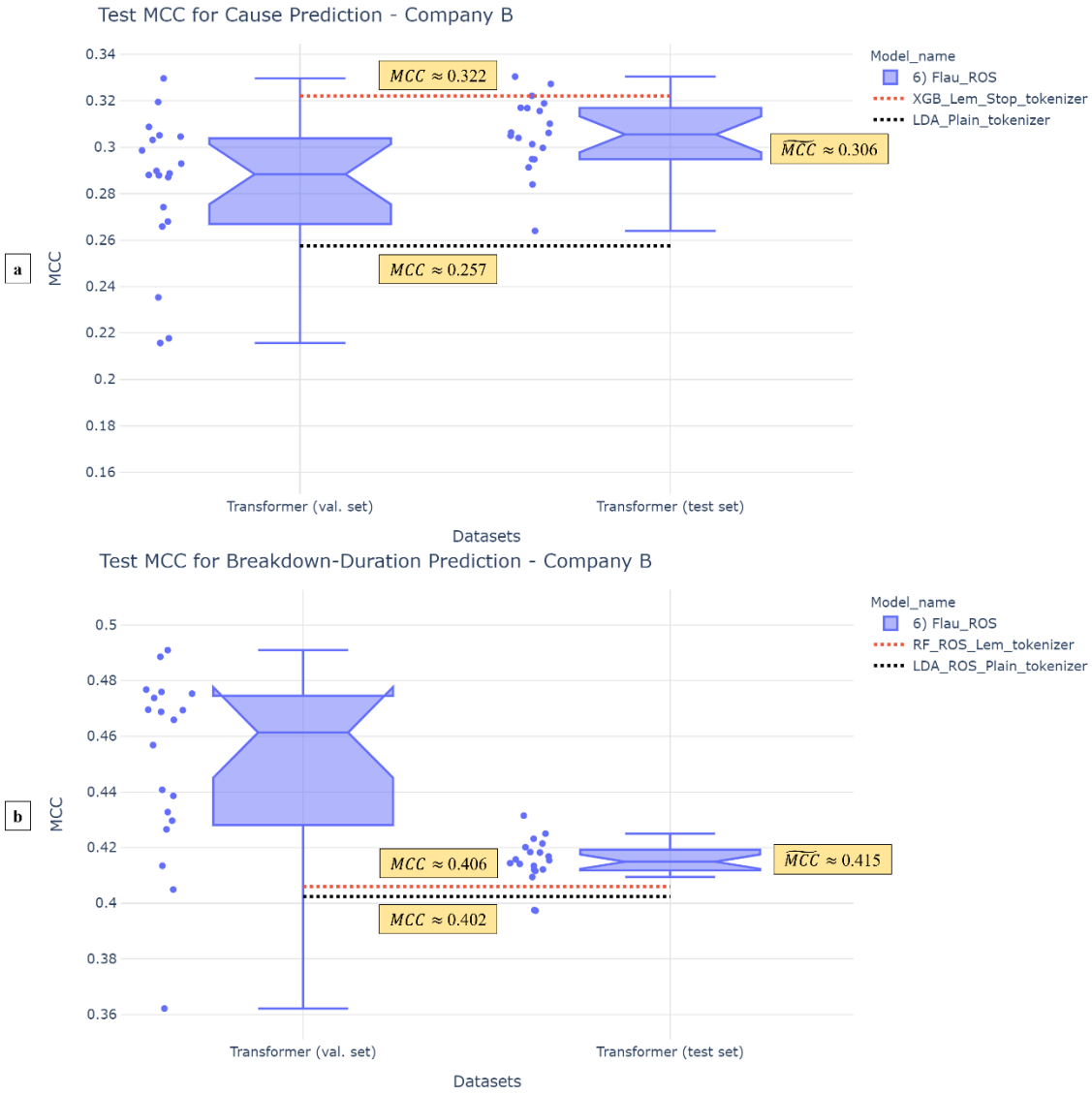


Figure 6.9 Test MCC box plots for (a) cause and (b) breakdown-duration prediction in Company B.

For Company B, models also exhibited similar performances: classic ML models were slightly better for cause prediction using an XGB with the Lem_Stop_tokenizer, while transformers were superior for the breakdown-duration prediction. Nevertheless, for the latter task, the transformer exhibited signs of overfitting.

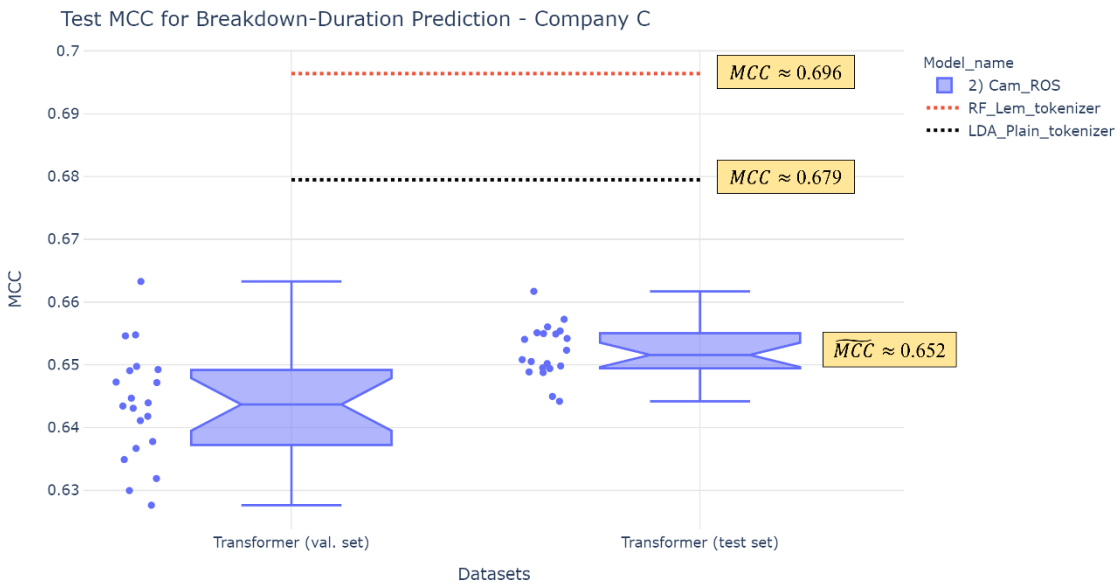


Figure 6.10 Test MCC box plots for breakdown-duration prediction in Company C.

For Company C, both the baseline and classic ML models yielded much better MCCs than the transformer. However, as shown in Table 6.2, Company C had a rather small vocabulary size (1134 tokens) and stable descriptions lengths, with a variation coefficient of 0.31. Its maintenance logs were more similar to standard codes than actual texts produced by humans with more semantic content. This may be because Company C was also the company with the lowest variety in its products and processes. As transformers are pre-trained in large corpora of text produced by humans, fine-tuning them may provide worse results than classic ML models in scenarios in which the vocabulary is limited, and texts present a more standard structure.

Generally, when transformers were surpassed, they only exhibited a slightly worse performance than the classic ML models (except for the breakdown-duration prediction in Company C). When compared with the heavy optimisation process used for classic ML models with grid search and several tokenization strategies, using transfer learning for transformers seemed to easily achieve excellent results with less effort. Further research should focus on the optimisation of transformers for these three tasks.

6.4.3. Model interpretation

Figure 6.11 shows the results for the interpretation using LIME and the proposed method to extract insights. A particular machine was selected from the test set, and predictions were performed for each of the 24 reports, resulting in a production stop. This machine is a type of

pump that enables fluids to be sucked from a tank. In Figure 6.11, two out of the 24 prediction examples are shown. Some sections in the maintenance log text were hidden (denoted as '[H]') because of confidentiality reasons. Figure 6.11 also shows the results of the insight extraction method for the top ten words that contributed the most to the probability of a stop in production. Finally, the translation to English for each maintenance log is provided below each text.

Figure 6.11 (a) A first example of LIME interpretation graphs. (b) A second example of LIME interpretation graphs. (c) Words obtained using the insight extraction method for a given machine regarding problems stopping the production process.

Figure 6.11 (a) and (b) show that LIME enables a straightforward visualisation of the inputs that the model considers as most important for each prediction. Additionally, these two examples provide a clear view of why operating with maintenance logs represents a challenge.

First, they typically contain highly heterogeneous writing styles, such as the inexistent word ‘KC’, which is used as a phonetic abbreviation for ‘cassé’ (broken, in French). Second, operators do not always respect the standards proposed to fill maintenance logs. For example, while Figure 6.11 (b) correctly provides the symptoms of the problem, Figure 6.11 (a) describes the problem itself. Additionally, operators tend to repeat the same information in all form fields instead of providing more detailed information on a problem, which is why phrases are repeated, although written differently.

Note that even if example in Figure 6.11 (a) contains the inexistent word ‘KC’, CamemBERT managed to recognise that this word contributed as much to a halt in the production as the word ‘broken’ (‘cassé’). In addition, although LIME does not consider n-grams, it still recognises groups of words that may convey a larger meaning. For instance, in Figure 6.11 (b), both the words ‘pas’ (does not) and ‘aspire’ (suck) were considered very important for the prediction, which was coherent, as they are occurred together in the text, indicating a malfunctioning in the aspiration system. Nevertheless, future research must adapt the method to consider more than just unigram representations of texts.

Regarding the insight extraction method, the identified top words provided an idea of the common problems observed in the pump: it was frequently subjected to clogging and long cycle times. This was indicated by the words ‘clogged’ (‘bouchée’, ‘bouché’) and the coupling of ‘time’ (‘temps’) and ‘cycle’ (‘cycle’). This information may be useful for maintenance managers who are willing to automatically identify the causes of common machine breakdowns from descriptions provided by operators.

6.5. Conclusion, limitations, and perspectives

6.5.1. Implications

The results of this research are targeted for use by researchers as well as industry practitioners willing to improve the integration of MES with high-level planners and schedulers to better react to unexpected events on the shopfloor. As stated by Saenz De Ugarte et al. (2009), such integration is not only unexplored, but it is also mandatory to be able to respond to production uncertainties.

Not all industries and companies are ready to deploy, maintain, and exploit IoT systems. Thus, an interesting opportunity may be to harness already collected data in the form of free-form text to contribute to the integration between MES and other information systems. However, free-form text data in maintenance present skewed distributions, resulting in class imbalance, and it often requires the design of time-consuming pre-processing pipelines. Thus, this research can be used as a reference to explore transformer models in maintenance, as they generally provide practical results with less implementation effort. More precisely, this paper contributes to industrial and theoretical implications. The industrial implications are:

1) For managers willing to exploit free-form text data collected in the shopfloor: According to the recent review from Montero Jimenez et al. (2020), few studies in predictive maintenance use free-form text data due to their highly unstructured nature, even if the information contained in logs can be used to improve the maintenance process. Indeed, companies may prefer to change the free-form text data inputs to predefined taxonomies using drop-down menus. However, predefined options often lack flexibility, may be cumbersome to fill, and can be difficult to adapt without losing already collected records. Instead, this research showed that it is possible to effectively exploit free-form text data from maintenance logs presenting imbalanced distributions, providing the benefits of both rich inputs and predictive capabilities. Furthermore, results suggested that good performance can be achieved even with minimal domain-specific text pre-processing when using transformers. Hence, the methods and techniques employed in this paper can be extended to other contexts with less effort than by using handcrafted rules or classic ML techniques.
2) For managers willing to create systems mixing inputs from diverse sources and natures: some of the common challenges when employing data from several sources are data fusion and feature extraction to generate meaningful variables for ML models. Indeed, combining information from multiple sources is necessary for predictive maintenance, as data collection tends to be scattered across different entities and levels (Raheja et al., 2006). An example of this for sensor data is the study performed by Traini et al. (2020), where one of the main parts of the proposed framework for tool condition monitoring exclusively focused on data pre-processing and feature engineering to achieve data fusion. Nevertheless, real-life production

environments may not only rely on sensors to collect data, as they can also include free-form text inputs, photographs, business data, etc. Hence, finding ways to generate meaningful feature representations from multiple sources is vital when training ML models. This research suggested that contextualised embeddings produced by fine-tuned transformers can effectively serve as numerical representations for free-form texts. Therefore, managers willing to integrate free-form text data with other inputs may find transformers suitable for vectorising texts.

3) For managers willing to develop ML systems to accelerate decision-making while keeping the human in the loop: exploiting human inputs to support decisions while allowing them to act from their knowledge is not only vital to ensure the success of new tools, but also a domain with growing interest (Schreck et al., 2008; Sahu et al., 2020). By harnessing free-form text data from operators, humans can be better included as their inputs are considered as provided, and minor changes are done to their way of working. Indeed, it is important to avoid adding constraints to operators when implementing new systems, as their acceptance and inclusion is vital for the successful adoption of new technologies (Schreck et al., 2008; Thomas et al., 2018a). Hence, this paper showed that it is possible to characterise maintenance issues from highly unstructured text inputs by determining the expected breakdown duration, severity, and causes, to support human decision-making. Furthermore, such support provides enough flexibility to let humans act from their knowledge and make more informed decisions to perform tasks such as production rescheduling or choosing the most appropriate technician to solve a particular issue. Finally, the lack of digital culture and skills has been identified as one of the most critical problems in adopting I4.0 (Ivanov et al., 2020). For managers willing to tackle the effects of such insufficiencies, this research suggested that transformers and LIME effectively enable the interpretation of predictions and insight extraction, fostering the acceptance of new tools by operators that may be reluctant to trust ML model's outputs. Indeed, interpretability helps to achieve impartiality in decision-making by detecting possible biases, design more robust ML models, and ensure that only meaningful variables influence the outputs (Barredo Arrieta et al., 2020).

The theoretical contributions are:

1) For researchers and practitioners exploring class imbalance solutions for maintenance logs: ROS proved to be the most suitable method to mitigate class imbalance in various datasets. Although ROS increases the computational requirements in training time and dataset size, it should be preferred when such extra computational cost is bearable. Otherwise, class weighting also yielded good yet inferior results, avoiding the additional computational cost. Finally, RUS should be avoided when exploiting maintenance logs, as the information loss resulting from discarding data hurt the models' performance in all datasets. Nevertheless, it is worth mentioning that there is no 'best' or 'worst' class imbalance mitigation technique, as the performance greatly depends on the context (Johnson and Khoshgoftaar, 2019).
2) For researchers and practitioners needing to find categories in numerical variables that mitigate class imbalance: this study proposed an original method employing k-means, silhouette coefficients, and silhouette diagrams to establish classes in numerical variables (i.e. breakdown durations). This is useful when classification is preferred to regression, as ranges of values can provide further insight than just single value estimations. The proposed method provided excellent results when mitigating class imbalance, achieving results comparable to ROS in multiple datasets.
3) For researchers and practitioners willing to find techniques to exploit free-form text data from maintenance logs: results in this research suggested that transformers typically need less handcrafted text pre-processing to achieve superior or equivalent results when compared to classic ML models. This was observed for cases where maintenance logs were similar to natural language produced by humans with a rich vocabulary, typos, and spelling variations. However, for cases where data resembled predefined codes with a small vocabulary size and few word variations, traditional ML models performed better. Hence, researchers and practitioners may find these results helpful when evaluating what technique to use, depending on the characteristics of their dataset.

6.5.2. Limitations of the study and future research

This study had five main limitations: First, only one interpretation technique (i.e. LIME) was employed, while other more recent techniques such as SHAP (Lundberg and Lee, 2017) can provide more suitable interpretations of the model. Additionally, the proposed insight extraction method only considers unigrams when scoring relevant words, reducing interpretation clarity. Second, only one algorithm-based technique (i.e. weighted categorical cross-entropy) was compared, while other loss functions proposed in the literature, such as the focal loss (Lin et al., 2017), class correction loss (Li et al., 2019), and mean false error loss (Wang et al., 2016), may further improve the performance in imbalanced classification. Third, this research did not study the optimisation of the transformer hyperparameters such as the learning rate, number of epochs, regularisation to control overfitting, etc. Fourth, this study only used language-specific models, as the texts used were fully in French. This may result in a loss in the model classification quality for companies with bilingual maintenance reports. Finally, the data pre-processing performed using k-means for the breakdown-duration prediction task provides ranges of values that may not be the most suitable for model interpretation by maintenance planners. Thus, approaches that enable the creation of balanced clusters compliant with the advice of industry experts should be used.

After highlighting the limitations of the study, future research will focus on the following four axes:

1) Interpretability: Review and compare other recent model-agnostic and transformer-specific interpretation techniques to identify their advantages and drawbacks. In addition, the insight extraction method should be extended to consider more than just unigrams.
2) Algorithm-level techniques: Compare the performance of other loss functions observed in the literature, as well as the interest of a better tuning of class weights.
3) Transformer hyperparameters: Study the effects of hyperparameter optimisation in transformers and verify whether there are ranges for the hyperparameters that provide excellent results in data coming from maintenance logs.
4) Multilingual corpora: Assess the classification performance of language-specific models in bilingual corpora, particularly when compared with standard large transformer models dedicated to English.

6.5.3. Conclusion

This research explored the use of two state-of-the-art deep learning models for NLP (i.e. CamemBERT and FlauBERT) to harness predictions from texts in maintenance logs to support decision-making in production processes. Three actual datasets from different companies were employed to train the models into three different tasks: severity prediction, breakdown-duration prediction, and cause prediction. In addition, these datasets had a class imbalance. To mitigate this effect, data pre-processing, data-level (i.e. ROS and RUS), and algorithm-level techniques were employed. For comparison, four classic ML models were trained for these tasks. Finally, an interpretation technique called LIME was employed to provide an explanation of individual predictions and to propose a method that enables the extraction of insights from a set of maintenance reports of a particular machine.

The results suggested that the classification performances of CamemBERT and FlauBERT were similar. Regarding class imbalance, using data pre-processing seemed to be sufficient to solve this problem, particularly when working with numerical data, such as in breakdown-duration prediction. For other scenarios in which data pre-processing is not possible, ROS provides the best results for a wide range of imbalance levels and number of classes. RUS yielded the poorest results, probably due to information loss when undersampling the dataset. Finally, class weighting with categorical cross-entropy loss did not provide reasonable results compared with ROS and the plain models. This technique seems to be sensitive to high imbalance levels, which harmed the global classification quality of the model.

Compared with classic ML models, transformers yielded a superior performance in two use cases: severity prediction for Company A and breakdown-duration prediction for Company B. For the three remaining use cases, classic ML models were only slightly better than transformers. Nevertheless, classic ML models require optimisation and more exhaustive text pre-processing through tokenization. However, the transformer hyperparameters were not optimised and the text pre-processing burden was lower, suggesting that they can achieve excellent results with low implementation effort. The only exception was breakdown-duration prediction in Company C, where classic ML models achieved considerably better results. This may be due to the highly standardised maintenance reports from this company, which were closer to predefined codes than to NLP data produced by humans.

Regarding interpretation, using LIME enables the generation of local explanations for individual predictions and the proposal of a method for extracting insights from a set of maintenance reports. This may be valuable for maintenance managers willing to assess the quality of individual predictions and provide an overview of what may cause a given problem on a particular machine.

The findings of this study suggest that even if maintenance logs from companies are highly unstructured, heterogeneous, and imbalanced, transformer models and techniques to solve the class imbalance may aid in harnessing value supporting decision-making and interpretability.

Acknowledgements

This work was financially supported by a partnership between the company iFAKT France SAS and the Association Nationale de la Recherche et de la Technologie (ANRT) under Grant 2018/1266.

7. Chapter 7: Article 4 - Artificial Data Generation with Language Models for Imbalanced Classification in Maintenance

Name of the book series: Studies in Computational Intelligence – Springer book series

Workshop dates: 27-28 January 2021

Status: In Press

Authors: Juan Pablo Usuga-Cadavid, Bernard Grabot , Samir Lamouri, Arnaud Fortin

Corresponding author: Juan Pablo Usuga-Cadavid

Abstract: Harnessing data that comes from maintenance logs may help improve production planning and control in manufacturing companies. However, maintenance logs can contain highly unstructured text data, presenting imbalanced distributions. This hinders the training of Machine Learning (ML) models, as they tend to poorly perform when identifying the underrepresented classes. Thus, this study uses a recent language model called GPT-2 to generate artificial maintenance reports. These artificial samples are employed to mitigate the class imbalance when training a Deep Learning (DL) architecture named CamemBERT. To carry out the experiments, an industrial dataset is used to train eleven DL models with different approaches to tackle class imbalance. Findings suggest that mixing random over-sampling with artificial samples improves the performance of classifiers when trained on imbalanced datasets. Finally, results imply that using nucleus sampling when generating artificial text sequences with language models ameliorates the quality of produced data.

Keywords: natural language processing, language model, maintenance, deep learning, class imbalance, artificial data, industry 4.0

7.1. Introduction

Valuing data coming from maintenance logs may provide several advantages to performing better production planning and control. For instance, by adapting a production schedule to unexpected disturbances, the engaged delivery dates can still be respected (Usuga Cadavid et al., 2019). Machine Learning (ML) has been extensively used in production planning and control research to improve manufacturing systems in the framework of Industry 4.0 (I4.0) (Usuga Cadavid et al., 2020a). In fact, ML offers a way to harness data from diverse sources such as information systems, equipment sensors, products, customers, etc. to support decision making (Tao et al., 2018; Usuga Cadavid et al., 2020a).

Despite the potential advantages provided by ML, the quality of the learning process strongly depends on the dataset employed. In applications such as fraud detection, disease diagnosis or image recognition, data distributions may be strongly skewed towards one of the classes (Johnson and Khoshgoftaar, 2019). For example, in the case of a rare disease diagnosis, there will be few examples of patients having a certain disease compared to the number of healthy patients. This naturally induced class imbalance is denominated intrinsic imbalance; conversely to extrinsic imbalance, which occurs when the imbalance is artificially introduced by external factors (Johnson and Khoshgoftaar, 2019). Maintenance logs can also present intrinsic imbalance. For example, few issues will lead to a halt in the production process, while the vast majority will not cripple it.

Class imbalance may strongly hurt the performance of ML models, as the learning process tends to be disproportionately influenced by the Overrepresented Class (OC). Thus, the model fails to correctly detect the Underrepresented Class (UC) in most of the cases. This may be unacceptable in some contexts, where not identifying the UC can lead to severe consequences. For instance, not detecting that a production problem will cripple the production line can strongly disrupt the manufacturing process.

Maintenance logs normally contain free-form text data manually provided by technicians. These reports describe the symptoms of events like machine breakdowns and provide guidance to understand the issue. Nevertheless, even if the textual reports encapsulate meaningful information to train ML models, they are highly unstructured: they commonly contain typos, abbreviations, and they may be strongly influenced by jargon. Hence, this research focuses on the use of a recent language model called GPT-2 (Radford et al., 2018) to generate artificial descriptions of maintenance reports leading to a production halt. The objective will be to use these artificially generated reports to reduce the effect of class imbalance when training a state-of-the-art Deep Learning (DL) model. Such a model will seek to determine whether a maintenance report corresponds to an issue that blocks the production. The task of classifying maintenance reports from their description will be handled as a classification problem in supervised learning. Following the nomenclature used by Wang and Jiang (2018), problems that stop the production process are named dominant disturbances, while others are called recessive disturbances.

The remainder of this article is organized as follows: Section 2 provides details about the necessary background and related work. Section 3 presents the employed dataset, tested techniques, and training policies. Section 4 presents the results and discussion. Finally, section 5 concludes this study and provides perspectives on future work.

7.2. Background and Related Work

7.2.1. Background

7.2.1.1. Handling Class Imbalance with Data-level Techniques

According to Johnson and Khoshgoftaar (2019), the techniques that mitigate the effect of class imbalance can be grouped into three categories: data-level, algorithm-level, and hybrid approaches. Data-level techniques modify the training set distribution to reduce the level of imbalance. Algorithm-level methods modify the way ML algorithms perform learning by, for instance, assigning a higher importance to the UCs. Finally, hybrid approaches combine the latter two strategies. This study will focus on the data-level approach, leaving the other two for future research.

Two common techniques employed in the data-level approach are Random Over-Sampling (ROS) and Random Under-Sampling (RUS). ROS randomly resamples the set of UCs with replacement until the training set is nearly balanced. Conversely, RUS randomly removes observations from the set of OCs until achieving balance.

Both ROS and RUS have been extensively compared in the scientific literature. Nevertheless, no sampling method is guaranteed to perform best across all of the domains (Johnson and Khoshgoftaar, 2019). In fact, each method has its own advantages and shortcomings: while ROS has proven to better mitigate a class imbalance, it may greatly increase the requirements in terms of computing power and memory usage due to an increase in data. Additionally, it may cause overfitting to the oversampled classes (Wang et al., 2016). On the other hand, RUS has outperformed ROS in some scenarios and reduces the training time, but it may discard meaningful information in the training set when excluding observations.

Other techniques such as SMOTE (Chawla et al., 2011) and data augmentation (Shorten and Khoshgoftaar, 2019) focus on generating artificial samples for the UC instead of resampling

from the already existing observations. They have proven to greatly improve the performance of ML algorithms, especially of DL models, which are prone to overfitting.

As stated by Johnson and Khoshgoftaar (2019), most of the research that has been done on DL with a class imbalance has targeted Convolutional Neural Networks (CNNs) and image data for computer vision applications. Thus, this research focuses on the use of data-level approaches to tackle class imbalance in the field of Natural Language Processing (NLP) with DL. More specifically, this is done through the use of recent transformed-based models using attention mechanisms (Vaswani et al., 2017), which have greatly improved the state of the art in NLP.

7.2.1.2. Transformer-based Architectures in NLP

When working in NLP, choosing how to vectorize text inputs into numeric representations exploitable by ML is important. Since their introduction in 2013 (Mikolov et al., 2013), word embeddings obtained through models such as Word2Vec, GloVe or Fasttext have been used extensively. Word embeddings are vector representations of text obtained, for instance, through neural networks. These vectors have improved the state of the art in NLP with respect to older techniques that rely on weighting strategies such as TF-IDF.

Despite the advantages provided by approaches such as Word2Vec, the vectors produced are non-contextualized embeddings. This means that the polysemy of words is ignored. Put differently, a certain word will have the same vector representation no matter its usage, which may be harmful for terms whose meaning depends on context.

To solve this, DL models relying on attention mechanisms like ELMo (Peters et al., 2018), BERT (Devlin et al., 2018), and GPT-2 (Radford et al., 2018) have been developed. These architectures are normally called transformed-based models. They are normally pre-trained on several gigabytes of text data to learn meaningful feature representations of language. Also, they produce contextualized word embeddings and use more robust tokenization strategies, such as Byte-Pair encoding (Sennrich et al., 2015), which may better handle typos, acronyms, or abbreviations. Thus, this research will focus on these architectures. More specifically, on GPT-2 and a modified version of BERT adapted to French, which is called CamemBERT (Martin et al., 2019).

GPT-2 is a language model: it can be employed, among other tasks, to probabilistically generate the next words from a given text input. Hence, it will be used to artificially generate

maintenance log reports describing problems blocking the production process and reduce the class imbalance. Once these artificial texts are generated, the CamemBERT model will be trained to perform classification. The aim will be to classify whether a description of an issue will lead to a halt in the production.

7.2.2. Related Work

Mitigating class imbalance problems when training ML and DL models have been an important yet understudied topic in recent research (Masko and Hensman, 2015; Johnson and Khoshgoftaar, 2019). Most scientific production in the domain has focused on CNNs and image data, leaving significant research gaps regarding other DL architectures and data types (Johnson and Khoshgoftaar, 2019). Such is the case for NLP. This section summarizes related work concerning the use of NLP in datasets containing class imbalance. For each study, the imbalance ratio ρ, as used in (Johnson and Khoshgoftaar, 2019), is provided. This ratio is estimated as presented in Equation 7.1. If several datasets were used, the highest imbalance ratio is reported.

$$\rho = \frac{max_i\{|C_i|\}}{min_i\{|C_i|\}} \tag{7.1}$$

In Equation 7.1, C_i is the whole set of observations of class i. Thus, $max_i\{|C_i|\}$ and $min_i\{|C_i|\}$ represent the maximum and minimum class sizes, respectively. For instance, if the largest class has 10000 observations and the smallest has 10 observations, $\rho = 1000$.

In the context of social network security, Wu et al. (2020) focused on the task of recognizing bots on Twitter. As the number of bots is fewer than the number of human accounts, the training set presented a class imbalance of $\rho \approx 4.3$. To tackle this imbalance, a data-level approach was used: a modified generative adversarial neural network (Goodfellow et al., 2014) was employed to produce artificial observations and train a neural network. The approach outperformed other data-level techniques such as ROS, SMOTE, and ADASYN.

In the field of biomedical research, Deepika et al. (2019) explored the task of classifying texts containing descriptions about drug-drug pairs, drug-adverse effects pairs and drug-disease pairs, which is a multi-class classification problem. To mitigate the class imbalance ($\rho \approx 26.3$), authors used a data-level approach, i.e. SMOTE, and different corpora from different data sources to train a CNN.

Regarding software development, Nnamoko et al. (2019) targeted the bug severity prediction from text reports. The dataset used contained seven levels of bug severity and presented an imbalance of $\rho \approx 45.5$. Authors employed an algorithm-level approach to reduce the disparities between class-sizes and train several FastText (Bojanowski et al., 2016) classifiers: They developed a hierarchical tree-like architecture to train several binary models. The first model was trained on the largest class versus the other classes altogether. Then, after discarding the largest class, the second model was trained on the second largest class, versus the remaining classes. This process was repeated until only two classes were left. This approach was compared with a standard training, resulting in similar performance.

In the study performed by Kato and Tsuda (2018), the aim was to identify the most important factors contributing to the perception of *quality* for a brand. To achieve this, they employed a logistic regression to classify companies between *top brands*. As top brands were less frequent, the imbalance level was $\rho \approx 7.2$, which was corrected through RUS.

Finally, Wang et al. (2016) assessed the use of two new loss functions to train deep neural networks in imbalanced datasets: the mean false error and mean squared false error. This algorithmic-level approach was then tested on several datasets of both image and text data containing different levels of imbalance. Regarding NLP, the most severe case concerned document classification for the Newsgroup dataset with an imbalance level of (nearly) $\rho = 20$. Carried experiments compared the performance of neural networks trained with the proposed loss functions to models trained with the mean squared error. Findings suggested that using the new loss functions achieved better results.

Despite the recurrent use of data-level approaches, no other study has employed language models to generate artificial text samples and to reduce class imbalance. Furthermore, transformed-based models to perform classification were not used either. To the best of the authors' knowledge, this is the first study using transformed-based models to both generate and classify maintenance logs containing free-form text descriptions.

7.3. Methods and Materials

7.3.1. Employed Dataset

The employed dataset comes from the maintenance logs of a company whose industry and name will not be mentioned for confidentiality purposes. Each maintenance log contained the description of the symptoms, the name of the equipment concerned, the importance level of the equipment, and the type of disturbance (recessive or dominant). From these inputs, the equipment name and symptoms are free-text comments, which means that two technicians reporting the same problem on the same machine may not produce the same description. Finally, the importance level was a categorical variable containing three possible values: “essential”, “important”, and “secondary”. The initial dataset contained around 26000 observations. After cleaning the data, 22709 records were kept.

As transformed-based models can handle unstructured text sequences including typos, abbreviations, etc., the choice was to create two new variables (i.e. *Text seed* and *Issue description*) by concatenating the already existing inputs:

1) Text seed: this variable concatenates the equipment name and importance level. It will be used as text seed to generate the artificial samples with the language model.
2) Issue description: this variable concatenates the equipment name, importance level, and symptoms. It will be used to predict the type of disturbance with CamemBERT.

Figure 7.1 illustrates the variables created through a toy example. Finally, the imbalance level between recessive and dominant disturbances is $\rho \approx 10.6$, being the recessive disturbances the largest class.

Original inputs

Equipment description	Symptoms	Equipment importance	Type of disturbance
Manual lathe	Ways dirty, tailstock difficult to move	Secondary	Recessive

New variable creation

Text seed	Issue description	Type of disturbance
Manual lathe. Secondary	Manual lathe. Secondary. Ways dirty, tailstock difficult to move	Recessive

Figure 7.1 Creation of the Text seed and Issue description from initial variables.

7.3.2. Techniques Tested

The proposed method in this study has two main modules: a data generation module and an issue classification module.

The data generation module will use GPT-2, which is a recent language model proposed and pre-trained on around 40GB of text by the authors of (Radford et al., 2018). There are four model sizes available: small (124M parameters), medium (355M parameters), large (774M parameters), and extra-large (1558M parameters). For this study, the small and medium architectures will be fine-tuned, compared, and the best one will be selected. The library employed to use GPT-2 is the one proposed by Woolf (2019).

To generate the artificial maintenance descriptions, two main hyperparameters were explored: the temperature and nucleus sampling. The temperature determines how much randomness will be introduced into the language model choices: the higher the temperature, the higher the randomness. Hence, language models with higher temperatures will tend to create more *creative* text sequences. Nucleus sampling, proposed by Holtzman et al. (2019), helps avoid generating incoherent words by setting a threshold of P. Hence, the cumulative probability distribution is computed for all of the tokens, starting with the most likely ones. After it reaches P, all the of other tokens, which are less likely to be generated, are discarded.

The issue classification module will use CamemBERT, a transformed-based model inspired from the RoBERTa architecture (Liu et al., 2019). CamemBERT was pre-trained by the authors of (Martin et al., 2019) on 138GB of uncompressed text in French employing several GPUs for

17 hours. The implementation uses the subclass of CamemBERT, called *CamemBertForSequenceClassification* available in (Huggingface, 2019), and the code was based on the example proposed by McCormick and Ryan (2019). The following subsection details the training policies of each module.

7.3.3. Training Policies

First, the initial dataset is split into 75% for training and the remainder for the test. Then, the training set is further split into 10% for validation and 90% for actual training.

The data generation module will be exclusively trained on the *Issue descriptions* leading to dominant disturbances. Two model sizes will be compared: the small and medium sized models. As suggested by Woolf (2019), the lower the average training loss, the better. Thus, each model will be trained during 2400 steps and their average loss will be compared. When the best model is chosen, it will use the *Text seeds* to generate the artificial text samples leading to dominant disturbances.

Using hyperparameters advised in (Woolf, 2019), the following text generation strategies will be employed: temperature of 0.7 and no nucleus sampling (T0.7-P0), random temperature following $U(0.7, 1)$ and no nucleus sampling (TRnd-P0), temperature of 0.7 and nucleus sampling with a threshold of 0.9 (T0.7-P0.9), and random temperature following $U(0.7, 1)$ and nucleus sampling with a threshold of 0.9 (TRnd-P0.9).

The issue classification module will be trained on the training set balanced through the four following strategies: ROS, RUS, artificial data coming from each of the four text generation strategies, and 50% of ROS plus 50% of artificial data. Furthermore, a model trained on the training set with no modifications will be also assessed. The validation set will serve to fine tune the hyperparameters of each model and to select the best one. Then, the best model will be retrained by mixing the training and validation sets and by following the best class balancing strategy. Finally, its performance will be measured with the test set. The eleven models that will be compared are summarized in Figure 7.2.

For comparison purposes, the Matthews Correlation Coefficient (MCC) will be used. Recent research has suggested that the F1-score may not be suitable to assess the quality of classifiers in imbalanced datasets (Hand and Christen, 2018). Instead, the MCC is preferred (Delgado and Tibau, 2019). The MCC ranges from -1 to 1, where 1 represents a perfect classifier.

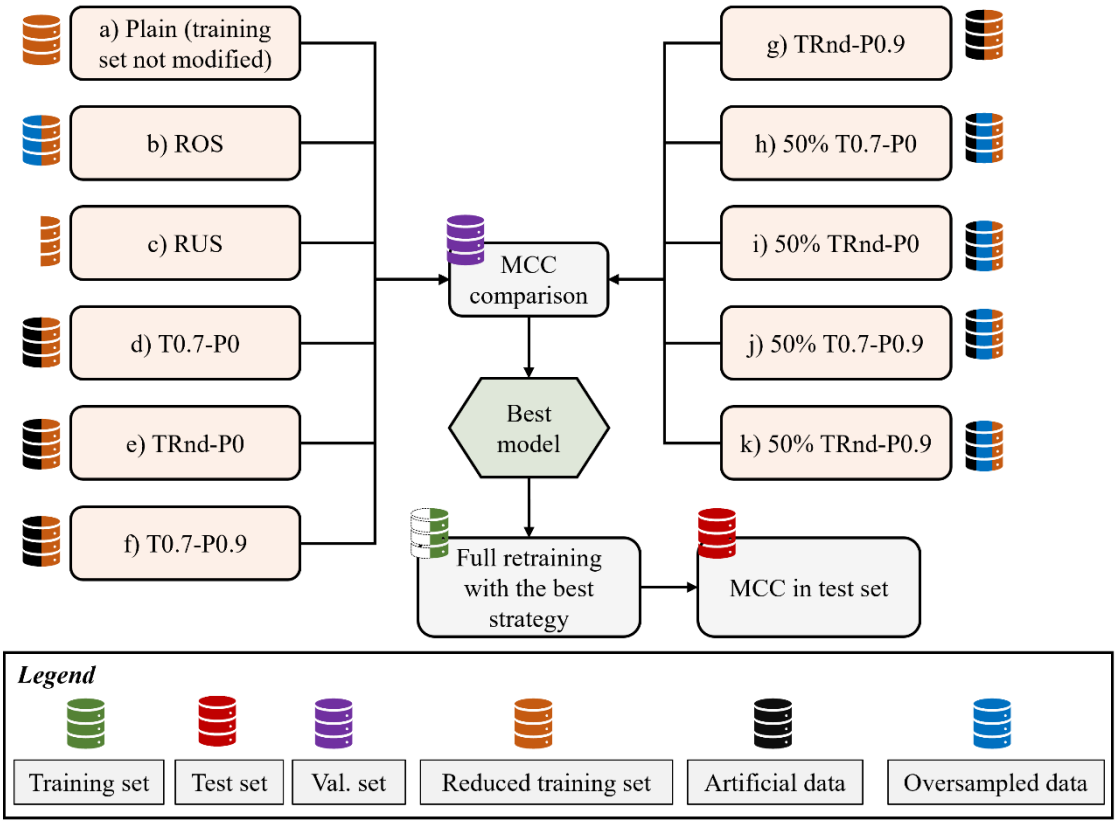


Figure 7.2 Training policy for the issue classification module.

7.4. Results

The models were trained using a GPU Tesla P100-PCIE-16GB. GPT-2 was trained using TensorFlow, while the CamemBERT models used PyTorch. As using GPUs introduces randomness, the experiments were run several times: five times for each of the two GPT-2 models and 20 times for each of the CamemBERT models.

7.4.1. Results for the Data Generator Module with GPT-2

Figure 7.3 shows the mean of the average training loss across all five runs and the 2400 training steps for the small and medium model.

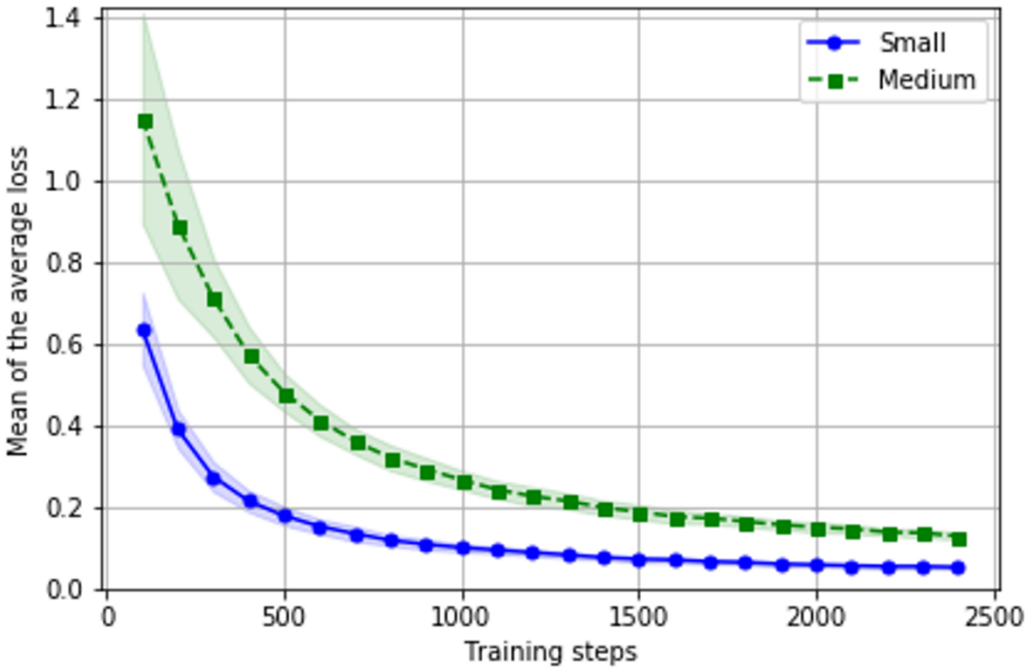


Figure 7.3 Mean average training loss for the small (blue) and medium (green) GPT-2 model.

Findings suggest that the language model that better fits the text descriptions for dominant disturbances is the small model. This may indicate that when language models have relatively little data to learn (around 1300 examples), smaller models perform better. Thus, the small architecture was used to generate the artificial data for the following experiments. Finally, it is worth noting that the GPT-2 model used from (Woolf, 2019) was originally designed for English. Thus, the generated texts dropped all of the uniquely French characters.

7.4.2. Results for the Issue Classification Module with CamemBERT

Figure 7.4 shows the box plots for the validation MCC for each of the eleven models.

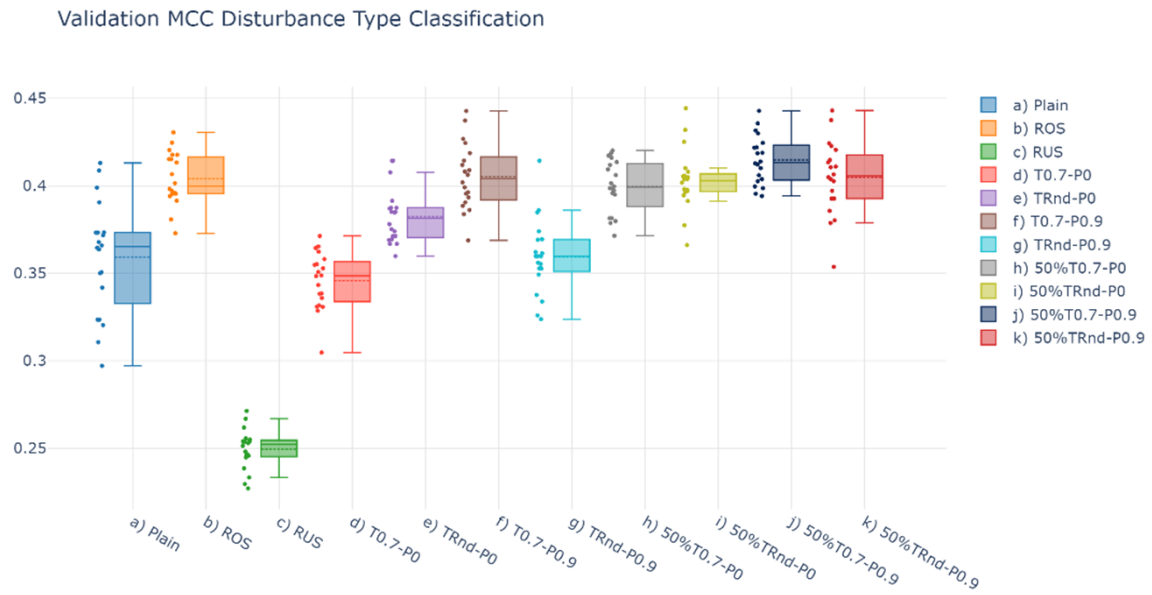


Figure 7.4 Validation MCC for the eleven CamemBERT models.

Table 7.1 provides further detail on the results, including more common metrics. It presents the average accuracy (Acc.), specificity (S1), sensitivity (S2), and MCC. In this case, the S1 and S2 measure the percentage of recessive and dominant disturbances that were correctly classified, respectively. For each measure, the highest value is highlighted in bold. Models are displayed by decreasing MCC.

Model name	Acc.	S1	S2	MCC
j) 50%T0.7-P0.9	0.902	0.930	0.554	**0.415**
f) T0.7-P0.9	0.927	0.972	0.373	0.405
k) 50%TRnd-P0.9	0.895	0.921	0.571	0.405
b) ROS	0.895	0.922	0.565	0.404
i) 50%TRnd-P0	0.901	0.931	0.537	0.403
h) 50%T0.7-P0	0.903	0.934	0.522	0.400
e) TRnd-P0	0.920	0.962	0.394	0.382
g) TRnd-P0.9	0.918	0.963	0.365	0.359
a) Plain	**0.931**	**0.985**	0.251	0.359
d) T0.7-P0	0.919	0.966	0.339	0.346
c) RUS	0.710	0.708	**0.734**	0.250

Table 7.1 Average validation accuracy, specificity, sensitivity, and MCC.

Results suggest that using the 50%T0.7-P0.9 (j) approach increases the average MCC. This approach mixed 50% of the resampled observations of the UC with 50% of the artificial examples generated with a stable temperature and nucleus sampling. With respect to the plain

model (a), the sensitivity is greatly improved, which means that more dominant disturbances are correctly detected. When compared to ROS (b), results are similar in terms of specificity and sensitivity. However, model j globally improves the results of the classifier, which are observed through a superior MCC.

The results obtained with RUS (c) yielded the best sensitivity, meaning that this is the best model to detect dominant disturbances. Nevertheless, the information loss produced by excluding observations severely penalizes the global performance of the classifier. This also means that the model will fail to detect more recessive disturbances in maintenance, which is not advantageous, either.

The fact that the Plain model (a) achieves the best accuracy and specificity shows why these measures are not well suited to evaluate ML models in imbalanced datasets: the classifier will mainly learn the OC, which will boost its accuracy, even if it has bad performance when detecting the UC.

The findings indicate that nucleus sampling is beneficial to generate meaningful artificial samples. In fact, three out of four models using it achieved good performance, reaching the top 3 MCCs among all of the models. Finally, the fact that employing artificial samples achieved good results even when using a GPT-2 model that was not adapted to French suggests that further improvements could be done with this technique.

The performance of model 50%T0.7-P0.9 (j) is then assessed using the test set. Results are shown in Table 7.2.

Model name	Acc.	S1	S2	MCC
j) 50%T0.7-P0.9	0.890	0.920	0.566	0.419

Table 7.2 Average test accuracy, specificity, sensitivity, and MCC.

Model j performance in the test set is close to the one presented in the validation set. This suggests that CamemBERT did not overfit the training data and could generalize to the task. This validates the performance of the proposed approach.

7.5. Conclusion and Future Work

This study explored the use of language models to artificially generate maintenance descriptions and reduce the class imbalance problem when classifying between dominant and recessive

disturbances in an industrial dataset. The approach used two state-of-the-art models in NLP: GPT-2 and CamemBERT. GPT-2 was employed to generate the artificial data, while CamemBERT was trained as a classifier to detect whether a maintenance issue would block the production process by analyzing its description. Two versions of GPT-2 were compared: a small and a medium version. The former provided better training performance. Also, the influence of the temperature and nucleus sampling when generating the artificial samples with GPT-2 was assessed. Results suggested that employing nucleus sampling improves the quality of the generated data.

Regarding CamemBERT, the best model was achieved by reducing the class imbalance with a mixture of real and artificial data. Such data was generated by keeping a constant temperature of 0.7 and using a threshold for nucleus sampling equal to 0.9. Test performance validated the results and suggested that there was no apparent over-fitting.

Future work will focus on four key aspects: first, the proposed approach is to be compared with algorithmic-level techniques, as increasing the amount of data may not be suitable for applications using massive datasets. In fact, such techniques may further improve the results without increasing the data volume. Secondly, the mix between real and artificial data was arbitrarily set to 50% in this study. This is to be studied to find relatively good values for this mix. Thirdly, the approach is to be validated using several industrial datasets. Finally, using a version of GPT-2 adapted to French may increase the effectiveness of the approach. This will be further explored in future work.

8. Chapter 8: Using an alternative loss function to tackle class imbalance in natural language processing

This chapter is based on the results of a study presented at the *17th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2021*, conducted from 7 to 9 June 2021. The paper was presented, and its status was 'in press' for publication in the conference proceedings.

8.1. Motivation of the study

In Chapter 6 and Chapter 7 we explain the three types of methods to tackle the effect of class imbalance, according to Johnson and Khoshgoftaar (2019), which are data-level, algorithm-level, and hybrid approaches. This chapter explores in detail an alternative algorithm-level method, which was applied to the data employed in Chapter 6 for Company A in the use case of severity prediction. The aim of this study was to assess the advantages and shortcomings of this method. The previous chapters of this thesis primarily focus on exploring better ways to use data-level methods while limiting the study of algorithm-level approaches to simply employing class weighting for classic ML models or the weighted categorical CE for neural networks. Table 8.1 lists the data-level and algorithm-level methods used in each chapter.

	Chapter 5	**Chapter 6**	**Chapter 7**
Data-level methods	• ROS • Pre-processing outputs with k-means and silhouette diagrams	• ROS • RUS • Pre-processing outputs with k-means and silhouette diagrams	• ROS • RUS • Artificial data generated using language models • Artificial data plus ROS
Algorithm-level methods	• Class weighting for classic ML models • WCE for neural networks	• Class weighting for classic ML models • WCE for transformers	

Table 8.1 Methods to mitigate the class imbalance used in each chapter

Table 8.1 shows that the exploration of algorithm-level methods was limited in past chapters. Nevertheless, these methods may provide advantages over the tested data-level methods. Some of these advantages are:

1) As ROS artificially increases the observations of minority classes through resampling, it increases the size of the training set. For certain configurations where

the imbalance is too high or there are several classes to resample, this may be prohibitive because of the larger dataset size and higher training time. Thus, algorithm-level methods can be used to adjust the penalty for the minority classes or modify the way ML can learn to avoid these issues.

2) When using ROS, classes with few instances may have the same observations repeated several times in the resampled dataset, which increases the risk of overfitting. Wang et al. (2016) reported that using ROS may tend to overfit the model to minority classes. The use of algorithm-level methods can avoid this issue.
3) Using artificial data generation with language models, as described in Chapter 7, provided good results. However, in cases where there are several minority classes, training the dedicated language model for each class and generating the data may be time consuming. Moreover, it may happen that not all classes have sufficient data to train the language model accurately. Algorithm-level methods avoid these issues as they do not require manual resampling to increase the dataset size.
4) Algorithm-level methods do not discard data, as it occurs in the case of RUS, thereby avoiding information loss.
5) Pre-processing the outputs with k-means and silhouette diagrams, as described in Chapter 5 and Chapter 6, proved to reduce the effect of class imbalance. However, this method can only be applied if outcomes are numerical, such as breakdown durations, excluding cases where outputs are categorical, such as in severity or cause prediction. Thus, algorithm-level methods can be employed for any output, for example, by weighting some of the observations more than others.

Chapter 6 only explores WCE for training transformers to tackle the class imbalance. The WCE is a common loss function employed when training neural networks in imbalanced datasets. This is a modified version of the CE loss. In Chapter 6, Equation 6.2 presents the formulation of WCE. The CE is obtained from Equation 6.2 when $w_k = 1$ for all classes, which implies that all categories have the same importance when training the ML model.

Although the results from Chapter 6 suggested that the WCE helps in reducing the class imbalance, this loss function has a significant limitation, i.e. it is less sensitive to differences between easy and hard misclassified observations. To overcome this shortcoming, Lin et al. (2017) proposed the FL, which allows the down-weighting of easy examples to focus on difficult observations. Moreover, the FL accepts class weighting, as in the WCE.

8.2. Related work

Several studies have proposed the use of algorithm-level methods to reduce the effect of class imbalance.

Nnamoko et al. (2019) aimed to predict the severity of software bugs based on their descriptions. However, the most common bug type had approximately 45 times more instances than the least common type, leading to a class imbalance. To solve this issue, they proposed an algorithm to train a hierarchical classification tree that starts learning to classify the largest class versus the rest. Then, it discards the largest class and learns to classify the second-largest class against the remaining categories. This process is repeated until only two classes remain. The results of this model provided similar results to those of standard training.

In the automotive industry, Fathy et al. (2021) proposed a hybrid approach using cost-sensitive learning along with artificial data generation to train the XGBoost algorithm to tackle class imbalance. Although their results are promising, their data consisted of sensor readings. Therefore, their synthetic data generation approach may be difficult to extend to our free-form text data from maintenance logs. Additionally, in Chapter 6, we compared the performance of XGBoost with certain techniques to reduce the class imbalance for the data from Company A in severity prediction. The results suggest that the transformer models demonstrated superior performance. Thus, we opted to focus on transformers.

Other studies used alternative loss functions to reduce the class imbalance. For example, Wang et al. (2016) proposed a novel function called *mean square false error* in the 20 Newsgroup dataset and obtained promising results. Iikura et al. (2021) utilised the FL in NLP to determine whether two sentences belong to the same paragraph to perform segmentation of texts. The results yielded better performance than using CE and WCE. Finally, inspired by the FL, Li et al. (2019) proposed *class corrections loss*, an adaptation of the FL for multi-class scenarios. The authors applied this loss in a neural network to perform emotion recognition and achieved satisfactory results.

As it appears that other authors have highlighted the benefits of FL (Li et al., 2019; Iikura et al., 2021), we believe that this approach may provide interesting results when applied in our case consisting of free-form text data obtained from maintenance logs. Hence, this chapter focuses on the study of the application of FL to CamemBERT with the same hyperparameters

employed in the study presented in Chapter 6. This research is similar to the article by Iikura et al. (2021), which exploited data obtained from literature novels. However, we employed the data obtained from maintenance, which may be more challenging as they present less quality. This is because operators tend to provide short descriptions of maintenance issues and care less about the writing style. We further explored the behaviour of the FL when varying one of its hyperparameters to identify patterns.

8.3. Proposed approach

8.3.1. Dataset and ML technique

For the data, we reused the maintenance logs from Company A and the data pre-processing steps employed in Chapter 6. The transformer model used was CamemBERT (Martin et al., 2019), as it provided the best results for severity prediction.

8.3.2. Techniques to be compared for class imbalance mitigation

This study aimed to understand the behaviour of the FL and assess its performance when compared to other loss functions. In this case, we chose the CE and WCE for comparison because they are common choices for training neural networks. Additionally, they were used in the other chapters of this thesis. The following equation presents the calculation of FL.

$$FL = -\sum_{k=1}^{K}\sum_{m=1}^{M} w_k * (1-h_k)^{\gamma} * y_m^k * \log(h_k) \quad (8.1)$$

In Equation 8.1, M is the number of samples, K is the number of classes, w_k is the weight for class k estimated using Equation 6.1, y_m^k is the correct label for observation m belonging to class k, h_k is the model softmax output for class k, and γ is the *focusing parameter* with $\gamma \geq 0$.

In Equation 8.1, the term $(1-h_k)^{\gamma}$ is called the modulating factor. It allows the FL to consider how hard a specific observation is to be classified. Recall that h_k is the output from the softmax function; thus, $0 \leq h_k \leq 1$. Hence, for easy examples, the model is confident about its prediction, which leads to $h_k \to 1$, resulting in a low modulating factor and loss value. For cases where hard examples are encountered, $h_k \to 0$, yielding a high modulating factor and high loss. In this way, the FL helps the model target hard examples and learns to classify them. Finally, γ modifies the impact of the modulating factor, where higher values of γ assigns more importance

to hard examples. In this study, we explored the influence of the focusing parameter by varying it in the range $\gamma \in [0.5, 1, 2, 3, \ldots, 9]$.

By elaborately studying the FL and comparing it to more classic approaches that tackle class imbalance, such as the WCE, we can understand whether it is worth spending time optimising the extra hyperparameter γ. In fact, the FL adds more hyperparameters to the model, which may be an extra burden for the data scientist creating it.

8.3.3. Training policy

The following three variations of CamemBERT were trained:

1) Cam_Plain: In this variation, CamemBERT model was fine-tuned by employing the CE. This is the baseline model, with no modifications or techniques to address the class imbalance.
2) Cam_Class: Here, CamemBERT model was fine-tuned using the WCE.
3) FL_Cam_γ: In this model, CamemBERT was fine-tuned using the FL. To further explore the influence of the FL, we varied the values of γ. For the proposed nomenclature, if $\gamma = 6$, the name of the model is FL_Cam_6.

The dataset was split into 75% for training (full training set) and 25% for testing. As the objective was also to test several values for the focusing parameter γ to choose the best model, the training set was further split into 90% for effective training (training set for FL) and 10% for validation of these variations with the FL.

With the training set for FL, we trained 10 models, each with a different focusing parameter, i.e. $\gamma \in [0.5, 1, 2, 3, \ldots, 9]$. Their performance was assessed in the validation set to select the best configuration. Then, the best model and the models using the CE and WCE were trained in the full training set. Finally, their performances were evaluated and compared using the test set. The chosen metric for model evaluation and selection was the MCC. Moreover, specificity and sensitivity are provided for the sake of analysis. In this case, the specificity and sensitivity measure the ratio of the correctly classified observations regarding the not stopping (majority class) and stopping (minority class) of the production, respectively.

8.3.4. Further understanding the learning process through data visualisation

To better understand the influence of the different loss functions in the learning process of the transformer, we employed a dimensionality reduction technique to visualise the embeddings produced by each of the three variations of CamemBERT. In fact, CamemBERT outputs a 768-dimensional representation for each document in the dataset. Thus, it is necessary to reduce the dimensions to enable data visualisation. Furthermore, the embeddings generated with an untrained CamemBERT were visualised to observe the influence of fine-tuning in transfer learning.

To perform dimensionality reduction, we chose PCA, as it is a well-known technique typically applied in ML research. Generally, PCA determines orthogonal axes, accounting for the most significant amount of variance. If the number of projected axes is lower than the original dimension of the dataset, PCA performs data compression (Jolliffe, 2011; Géron, 2019).

8.4. Results

We trained each variation of CamemBERT 20 times, and the best model was chosen based on the median MCC.

8.4.1. Choosing the best focusing parameter γ

Figure 8.1 shows the results for the 10 variations of CamemBERT with several focusing parameters. The boxplots for MCC (a), specificity (b), and sensitivity (c) are provided for each variation. The best model is framed in a red box.

From Figure 8.1 (a), it can be noted that when the focusing parameter is increased, the global performance of the model increases up to a point where the model focuses too much on difficult observations. After this point, the MCC degrades. This is probably because easy but numerous observations are ignored, contributing less to the learning process. This tendency is better understood by observing Figure 8.1 (b) and (c), i.e. the sensitivity tends to keep increasing while the specificity is reduced after reaching a maximum.

Figure 8.1 shows that the best choice for the focusing parameter is around $\gamma = 6$, where the highest global performance is achieved, with good results in specificity and sensitivity. However, it appears that the focusing parameter has a significant influence on the performance of the model. This may be a drawback of the FL, as introducing an extra hyperparameter to the

model can increase the burden of the data scientist when designing the model. Hence, further research should explore whether there are ranges for γ where superior global performance is achieved. For instance, based on Figure 8.1, it can be suggested that there is only a marginal performance variation for $\gamma \in [2, 6]$. For the following comparison with the CE and WCE, the model using $\gamma = 6$ is employed.

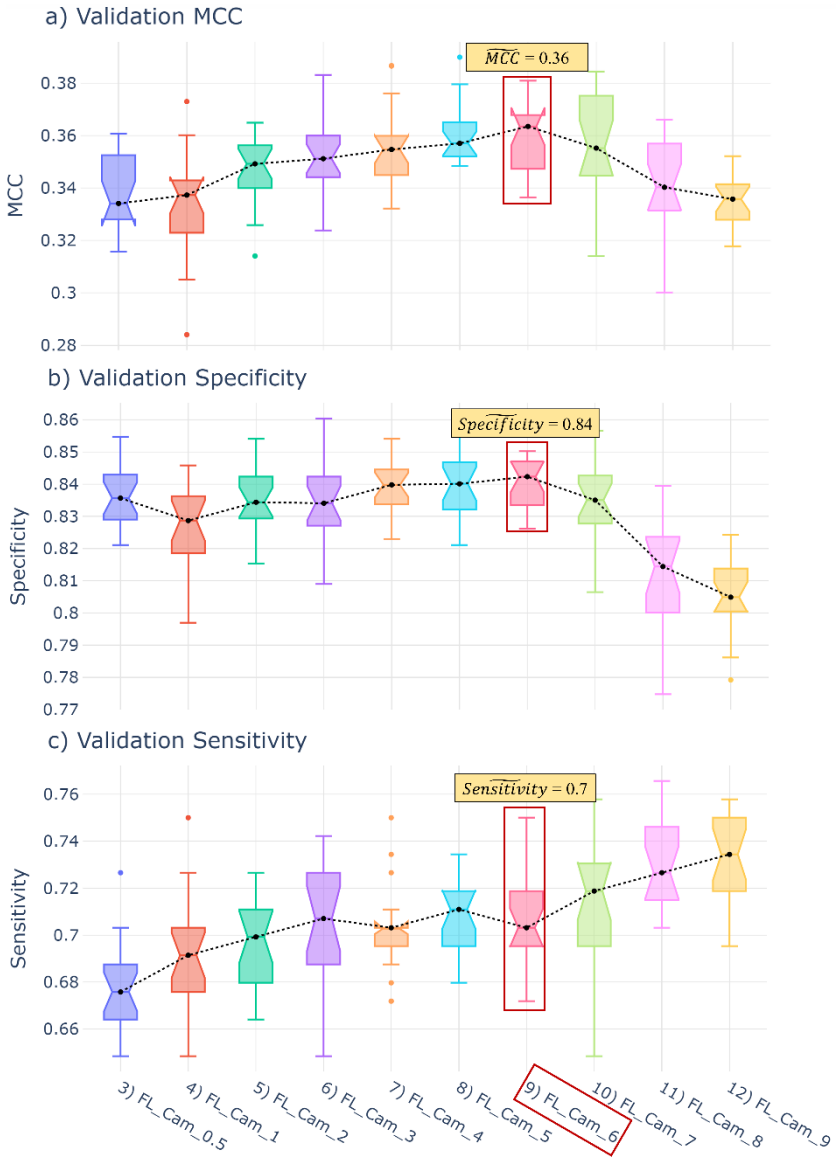


Figure 8.1 Validation of MCC (a), specificity (b), and sensitivity (c) for different focusing parameter values

8.4.2. Comparing the cross-entropy, weighted cross-entropy, and focal loss

Figure 8.2 shows the box plots for the MCC (a), specificity (b), and sensitivity (c), and each variation of CamemBERT when using the CE, WCE, and FL with $\gamma = 6$.

From Figure 8.2 (a), it can be observed that the best model based on the median MCC is the baseline model trained with the CE. It achieved superior performance when measured by the MCC, which is a balanced metric for imbalanced classification. The fact that the model employing the FL did not achieve a higher MCC score is probably because it poorly identifies instances from the majority class, thereby hurting its global performance (Figure 8.2 (b)). However, this model achieved the best sensitivity, implying that it has an excellent capacity for detecting instances of the minority class (Figure 8.2 (c)). Additionally, the sensitivity was better than that of the model using the WCE. This suggests that the FL is more capable of improving the detection of minority classes than typical approaches employed in ML, such as the WCE. Therefore, if it is essential to successfully detect minority classes for a particular use case, the FL should be considered among the options.

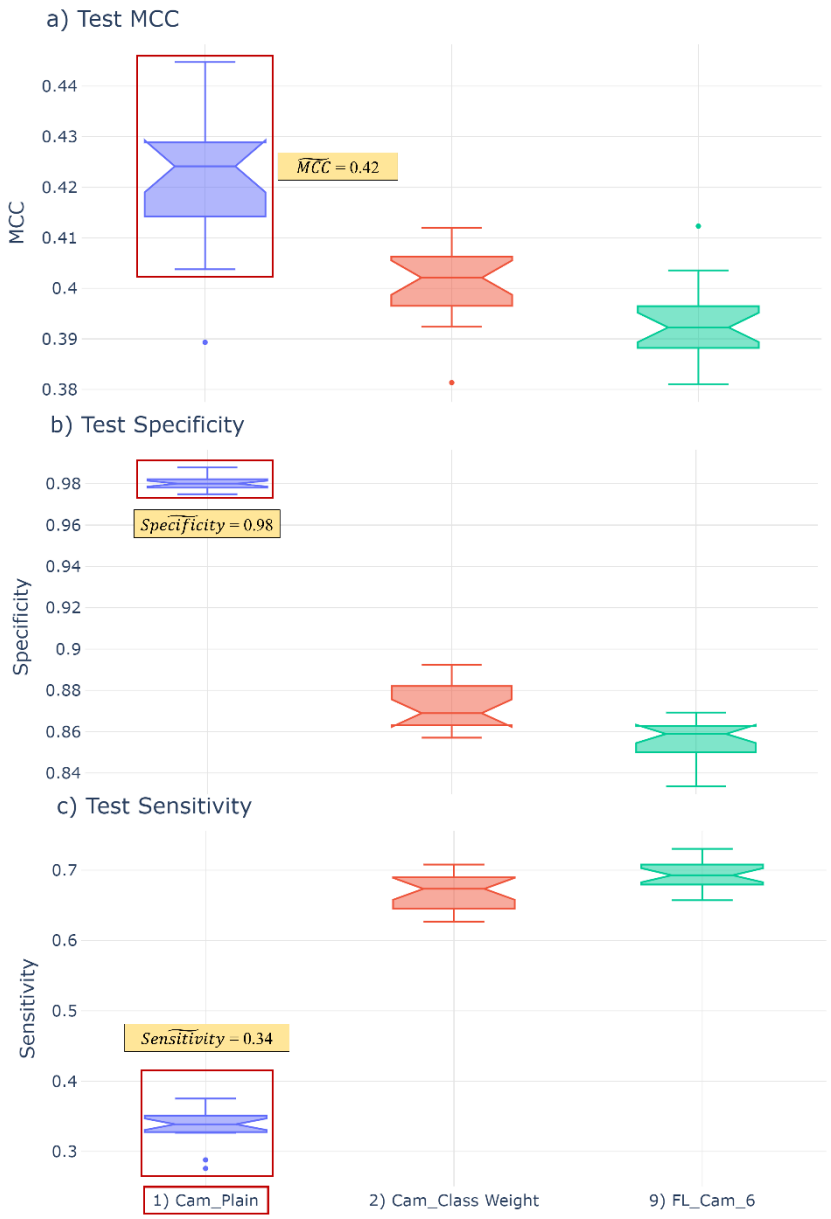


Figure 8.2 Test MCC (a), specificity (b), and sensitivity (c) for different loss functions

8.4.3. Visualising the learnt embeddings

Figure 8.3 shows, for each variation of CamemBERT, the kernel density plots for the embedding representations of the dataset mapped to a two-dimensional space with PCA. Additionally, the two predicted classes are displayed in different colours for analysis.

An ideal classifier learns embedding representations that do not overlap, achieving a perfect distinction between the two classes. Considering this, Figure 8.3(c) shows why the model using the FL achieves the best performance when identifying the minority class: the observations of

the minority class are in a high-density zone and further apart than those of the majority class. Moreover, Figure 8.3(b) shows that the learnt embeddings for the two classes using the WCE tend to be separated. However, the separation is less marked when compared to that using the FL. Figure 8.3(a) shows that when no method is used to tackle the class imbalance, the model attempts to separate the classes. However, it fails to generate high-density zones containing the minority class, resulting in low sensitivity. Conversely, high-density zones for the majority class are effectively created, which explains the high specificity. Finally, Figure 8.3(d) shows what happens when no fine-tuning is performed; in this case, the model does not learn to differentiate the classes. Therefore, fine-tuning is necessary.

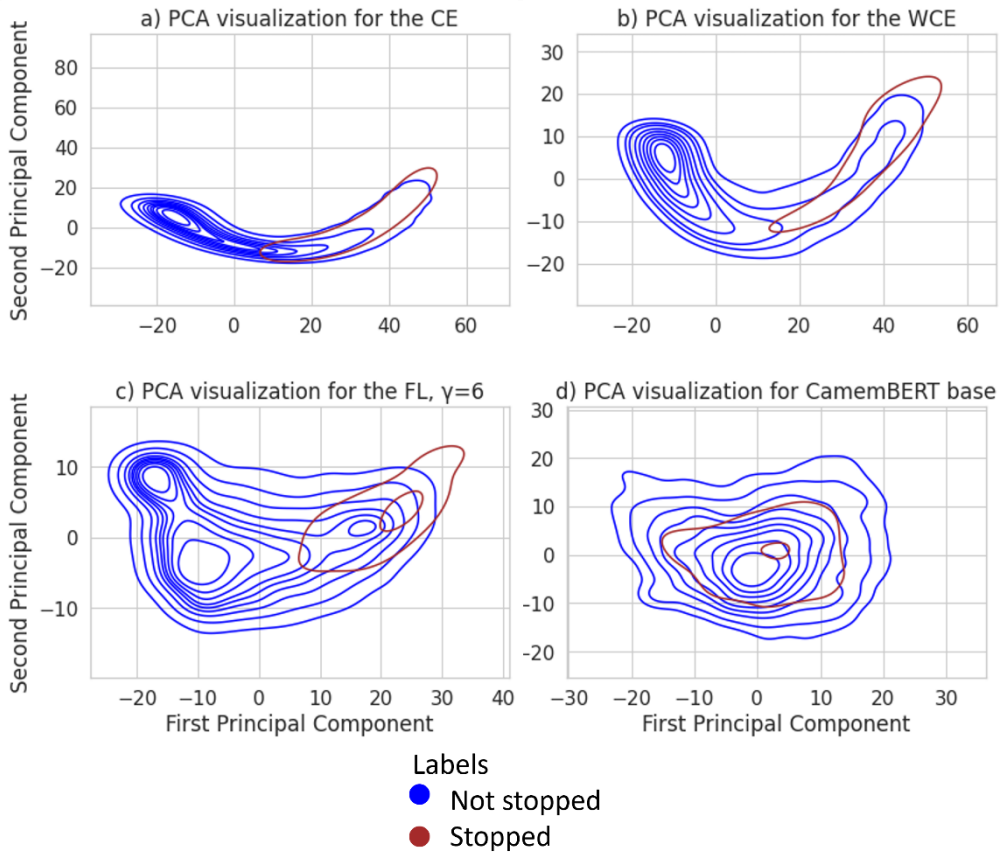


Figure 8.3 Kernel density plots for CamemBERT using cross-entropy (a), weighted cross-entropy (b), focal loss with γ=6 (c), and with no fine-tuning (d)

8.5. Conclusion

This chapter explored an alternative algorithm-level approach to tackle the class imbalance in maintenance datasets, i.e. the FL. To assess the effect of the proposed approach, we employed data from previous research consisting of maintenance logs containing free-form text data

comments left by operators. The objective was to learn whether these comments led to a severe issue, thereby blocking production.

The ML models that were trained were transformers, as they previously provided excellent results when working with NLP. Three transformer variations were trained to validate the proposed approach using the following three different loss functions: CE, WCE, and FL. Moreover, PCA was employed to allow two-dimensional visualisations of the produced embeddings of each transformer with different loss functions.

The results suggested that the FL did not provide the best global results, as it presented mediocre performance when learning the majority class. Nevertheless, it achieved the best performance when detecting the minority class. This may be advantageous for cases in which not detecting instances belonging to the minority class may lead to severe consequences. Furthermore, using the FL introduces an extra hyperparameter, leading to extra work when designing the model.

Comparing the results in Figure 8.2 with the results obtained in Chapter 6, it is clear that the FL can still not surpass the performance obtained while using ROS to tackle the class imbalance in severity prediction for Company A. Therefore, for cases where the extra computational cost and training time are bearable, ROS should still be preferred. However, if we compare the results with those obtained for RUS in Chapter 7, FL is the best approach to achieve good detection of minority classes while mitigating the degradation of global performance. In Chapter 7, RUS achieved a mean sensitivity of 0.73 versus 0.69 achieved for the FL. However, using the FL achieves an MCC of 0.39 versus 0.25 for RUS, suggesting that the utilisation of FL results in a good trade-off between global performance and capacity to detect the minority class.

Future research avenues should focus on the following two axes:

1) Reviewing and testing other available loss functions for imbalance classification.
2) Assessing the behaviour of the FL with respect to the focusing parameter on several datasets. This would allow the generalisation of conclusions and exploring a range for the focusing parameter for which the FL provides good results, thereby easing the burden on hyperparameter tuning.

9. Conclusion and discussion

9.1. On the proposed approach and the results

The primary objective of this thesis was to harness maintenance data to better react to production disturbances. In a world where manufacturing systems are subjected to uncertainty, predicting and reacting effectively to unexpected events is crucial for continuous improvement and ensuring efficient production systems. To achieve this, data-driven models, and more specifically, ML models, were explored to support decision making when encountering production disturbances occurring from maintenance.

To achieve this goal, our first specific objective was to understand the research gaps, opportunities, and trends regarding the application of ML in production planning and control in recent scientific literature. Then, we performed a literature review covering the period between 2011 and 2019, focusing on ML in production planning and control. Additionally, as research on PdM is rapidly producing new contributions, we performed a second study to update the results of the first research. This study focused on PdM with ML. In both studies, some common research gaps were identified. First, companies are interested in valuing historical data collected in their information systems. However, this may be challenging because of unstructured form of the data, coming from subjective data sources, such as free-form text comments. Second, data imbalance is a common issue observed in manufacturing. However, recent ML research appears to overlook this problem, limiting the capabilities of models when detecting rare events. Third, the problem of concept drift remains a rarely explored issue. Manufacturing systems change over time, and ML models should adapt to these changes. However, recent research seldom questions when to retrain the model or detect concept drift. Finally, transfer learning appears as a solution to tackle complex use cases when data are scarce. The literature review also revealed certain desired characteristics of PdM systems in I4.0. Among these, generating knowledge from data has attracted significant attention, while the inclusion of humans in the loop has rarely been addressed despite its frequently mentioned importance.

After identifying the research trends and gaps, our second objective was to evaluate the technical feasibility of ML models to target some of the previously identified research avenues and help to react to production disturbances. To achieve this goal, we decided to focus on using maintenance logs to value historical data, harness free-form text data from maintenance reports to consider highly unstructured inputs, and tackle the issues derived from class imbalance. Therefore, the first study was conducted. Based on the results obtained from this study, it was

identified that transformers may provide interesting opportunities for handling free-form text data, especially when they are fine-tuned. Moreover, techniques to mitigate the effects of class imbalance improved the performance of ML models. This study also highlighted some research avenues that need to be addressed in future work. Notably, alternative approaches for mitigating class imbalance, enabling knowledge generation from highly unstructured maintenance logs, and validating the results in other datasets to generalise conclusions must be explored.

Our third objective was to explore solutions to overcome some of the identified challenges through a literature review and technical feasibility test. Therefore, we focused on two different research paths. The first research path aimed to generalise the conclusions obtained when addressing the second objective and proposes a method to generate knowledge from highly unstructured data. The second research path explored alternative techniques to mitigate the class imbalance when using maintenance datasets containing free-form text. Specifically, we explored artificial data generation through language models and used a loss function (i.e. FL) that was initially proposed for computer vision applications.

The first research path showed that transformers achieve excellent results with low implementation and data pre-processing efforts, even when compared to classic ML models such as random forests, AdaBoost, or XGBoost. However, we identified that for cases where free-form text data from maintenance logs present a limited vocabulary, classic ML models demonstrate superior performance. While considering the knowledge generation method, we can effectively derive insights from a group of maintenance reports containing free-form text. However, the technique should be improved to consider groups of words instead of just unigrams.

The second research path showed that generating artificial data with language models further reduced the class imbalance, validating the approach. However, this approach was time consuming because of the training of a DL model to generate synthetic data. Moreover, the study concerning the FL suggests that it improves the capability of the model to detect the minority class, but the performance when detecting the majority class is reduced.

As a general conclusion for this third objective, we provide the following recommendations:

1) When harnessing the data obtained from maintenance logs containing free-form text data, the vocabulary size appears to be a critical factor influencing the choice of the

ML models. For cases where the data present a limited vocabulary that is closer to predefined codes than to actual natural language, it is preferable to employ classic ML models. However, transformers should be used when the data have a richer vocabulary.

2) Although the results with the FL did not provide the best global results (MCC), it was the best approach to detect minority classes while avoiding degradation of the overall performance that may happen with RUS. Hence, for cases where an acceptable global performance is desired, but particular emphasis is placed on detecting instances of the minority class, the FL should be employed.
3) Generally, when extra computation and training times are acceptable, ROS should be preferred in case of class imbalance. Additionally, if the effort to generate artificial data is bearable, it should also be considered as a good option for class imbalance. Indeed, these were the methods with the best mitigation of class imbalance in maintenance logs.

9.2. Limitations

One of the main limitations of this thesis is that it only considered ML, a particular type of data-driven model, to perform all the studies. A more realistic approach is to use a multi-model method, where data-driven, knowledge-based, and physics-based models interact to address more effectively production disturbances and reduce the limitations of single models. For instance, the class imbalance problem is typically observed in data-driven models. In contrast, knowledge-based models may be less sensitive to this issue, as they depend on hard-coded rules by humans. This idea is constantly mentioned by Montero Jimenez et al. (2020), and it should further drive the development of PdM applications.

Although the challenge of knowledge generation was explored through interpretability, this topic was barely explored in this thesis. Based on several limitations, we highlight the following:

1) We did not explore any other interpretability method to generate knowledge from our data-driven models. This should be addressed in the future, as there is a vast resource of recent literature regarding the interpretation of ML models. For instance,

(Barredo Arrieta et al., 2020) provided a thorough taxonomy and review of interpretability for artificial intelligence.

2) Being able to interpret the results of an ML model is different from understanding them. This work did not study whether the proposed interpretations of the models were aligned with the explanations provided by maintenance technicians and operators. Therefore, this should be addressed before further exploration of knowledge generation to avoid spurious correlations.
3) Our proposed approach to interpreting predictions using free-form text data from maintenance logs is still limited to unigrams. Unigrams tend to convey only little information, as they are isolated words that should be put into context to better interpret the results.

To integrate the humans in the loop of ML models, we decided to employ their free-form text descriptions *as provided*. Thus, no modifications are done to the way operators work. However, no research has been conducted to verify the perception of operators in the developed model.

The way we addressed the use cases has a significant limitation regarding the dynamics of maintenance issues, i.e. we only employed the initial report of symptoms provided by the operator to predict the final state or repercussions of the problem. However, certain scenarios may be more dynamic, as the issue can be processed using a series of evaluations in which various stakeholders add information about the topic. Thus, our research did not consider cases where there can be intermittent information arriving through time which can modify the output of our models.

The nature of the employed data was also a limitation of this study. Indeed, we targeted using free-form text data to develop PdM models, while in reality, inputs can have all possible formats. For example, it may be common to find systems that can report maintenance issues through free-form text data, photos taken by the operator, and records of the operating parameters of the concerned equipment. In such cases, all the provided data may be valuable for accurate predictions.

Finally, the last major limitation was that we did not explore hybrid methods to mitigate class imbalance. Our studies focused on either data- or algorithm-level strategies. However, hybrid methods can provide promising results that harness the advantages of both data- and algorithm-level approaches while reducing their shortcomings.

9.3. Perspectives

The following five axes for perspectives and research avenues were identified:

1) Understand the use of NLP, interpretability, and knowledge generation methods for production planning and control through literature reviews: Although research employing NLP in production planning and control is scarce, this thesis showed that some authors worked on this topic. Thus, future research should assess the maturity, usage, and trends of NLP in manufacturing. This is also the case for interpretability and knowledge generation methods, which are attracting increasing interest in fundamental research in ML. Therefore, it would be valuable to show how these techniques have been applied to production planning and control.
2) Explore the influence of big data on ML systems applied to manufacturing: An update on the literature review performed in Chapter 2 shows that it is rare to find papers focusing on the influence of the 5Vs (velocity, veracity, volume, variety, and value (Zhou et al., 2017)) of big data on the development of systems relying on ML models. However, manufacturing systems produce more data, and real-time predictions are required. This implies that industrial needs will converge towards tackling the 5Vs. Furthermore, this is an opportunity to explore cloud technologies applied to ML systems in manufacturing, which appears to be underexplored despite the relevance of this technology for future applications.
3) Explore alternative methods to tackle class imbalance: Improving the techniques to generate realistic synthetic observations faster and more easily is essential. Indeed, data in manufacturing are usually imbalanced, and collecting more data to tackle this imbalance may not be feasible in certain contexts. Thus, better methods for creating artificial data can strongly benefit future applications. Moreover, exploring new loss functions and algorithms that can reduce class imbalance is essential to support cases where it is challenging to generate artificial data.
4) Develop methodologies and strategies to detect and correct the concept drift issue: The dynamic behaviour in manufacturing systems has rarely been tackled in recent ML research. Future research should focus on automatically detecting when the trained model is obsolete for the current state of the system and choosing the

appropriate data to be retrained. Indeed, avoiding the concept drift issue is mandatory to create realistic systems that can adapt to varying conditions.

5) Extend the usage of free-form text data to two cases: First, the case where new information is obtained intermittently. This is the case for maintenance issues that are evaluated by several stakeholders, where new inputs are received each time a stakeholder performs an assessment. Second, cases where both structured and unstructured data are required to perform predictions, as in cases where images, texts, and sensor readings are recorded to report a maintenance problem.

List of publications

Articles in peer-reviewed JCR Journals (ISI Web of Science):

1) **One of 2020's Top Downloaded JIM Research Articles:** Usuga Cadavid, J. P., Lamouri, S., Grabot, B., Pellerin, R. and Fortin, A. (2020) Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0, Journal of Intelligent Manufacturing, 31(6), pp. 1531–1558. doi: 10.1007/s10845-019-01531- 7
2) Usuga Cadavid, J. P., Grabot, B., Lamouri, S., Pellerin, R. and Fortin, A. (2020) Valuing free-form text data from maintenance logs through transfer learning with CamemBERT, Enterprise Information Systems, (In Press), pp. 1–29. doi: 10.1080/17517575.2020.1790043
3) Usuga-Cadavid, J. P., Lamouri, S., Grabot, B. and Fortin, A. (2021) Using Deep Learning to Value Free-Form Text Data for Predictive Maintenance, International Journal of Production Research, (In Press). doi: 10.1080/00207543.2021.1951868.

Book chapters:

1) Nguyen, A., Usuga-Cadavid, J. P., Lamouri, S., Grabot, B. and Pellerin, R. (2021) Understanding Data-Related Concepts in Smart Manufacturing and Supply Chain Through Text Mining, Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future. SOHOMA 2020. Studies in Computational Intelligence, vol 952. Edited by L. S. Borangiu T., Trentesaux D., Leitão P., Cardin O. Paris, France: Springer, Cham. doi: 10.1007/978-3-030-69373-2_37.
2) Usuga-Cadavid, J. P., Grabot, B., Lamouri, S. and Fortin, A. (In Press) Artificial Data Generation with Language Models for Imbalanced Classification in Maintenance, SOHOMA Latina 2020. Bogota, Colombia.

Peer-reviewed international conferences:

1) **1st place of the '*Best PhD paper award*.'** Usuga-Cadavid, J. P., Lamouri, S., Grabot, B. and Fortin, A. (2021) Etude de la Perte Focale pour le Traitement du Langage Naturel en Maintenance dans l'Industrie 4.0, in CIGI QUALITA 2021 - Conférence Internationale Génie Industriel Qualita.

2) Usuga-Cadavid, J. P., Lamouri, S., Grabot, B. and Fortin, A. (In Press) Exploring the Influence of Focal Loss on Transformer Models for Imbalanced Maintenance Data in Industry 4.0, in INCOM 2021 - 17th IFAC Symposium on Information Control Problems in Manufacturing. Budapest, Hungary.
3) Usuga Cadavid, J. P., Lamouri, S., Grabot, B., Pellerin, R. and Fortin, A. (2019) Estimation of Production Inhibition Time Using Data Mining to Improve Production Planning and Control, in 2019 International Conference on Industrial Engineering and Systems Management (IESM). Shanghai, China, 25-27 September 2019, pp. 1–6. doi: 10.1109/IESM45758.2019.8948129.
4) **1st place of the '*LAURENT VILLENEUVE Young Reseacher award*.'** Usuga Cadavid, J. P., Lamouri, S., Grabot, B. and Fortin, A. (2019) L'Apprentissage Automatique dans la planification et le contrôle de la production : un état de l'art, in CIGI QUALITA 2019 - Conférence Internationale Génie Industriel Qualita.
5) Usuga Cadavid, J. P., Lamouri, S., Grabot, B. and Fortin, A. (2019) Machine Learning in Production Planning and Control: A Review of Empirical Literature, IFAC-PapersOnLine, 52(13), pp. 385–390. doi: https://doi.org/10.1016/j.ifacol.2019.11.155.
6) Usuga Cadavid, J. P., Lamouri, S. and Grabot, B. (2018) Trends in machine learning applied to demand & sales forecasting: A review, in ILS 2018 - Information Systems, Logistics and Supply Chain, Proceedings, pp. 307–316. Available at: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85050979282&partnerID=40&md5=563b9723e88a308975db42343494fc09.

References

Acernese, A., Del Vecchio, C., Tipaldi, M., Battilani, N. and Glielmo, L. (2020) Condition-based maintenance: an industrial application on rotary machines, *Journal of Quality in Maintenance Engineering*, ahead-of-p(ahead-of-print). doi: 10.1108/JQME-10-2019-0101.

Aissani, N., Bekrar, A., Trentesaux, D. and Beldjilali, B. (2012) Dynamic scheduling for multi-site companies: A decisional approach based on reinforcement multi-agent learning, *Journal of Intelligent Manufacturing*, 23(6), pp. 2513–2529. doi: 10.1007/s10845-011-0580-y.

Alammar, J. (2018) *The Illustrated Transformer*. Available at: https://jalammar.github.io/illustrated-transformer/ (Accessed: 9 March 2021).

Aliev, K. and Antonelli, D. (2021) Proposal of a monitoring system for collaborative robots to predict outages and to assess reliability factors exploiting machine learning, *Applied Sciences (Switzerland)*, 11(4), pp. 1–20. doi: 10.3390/app11041621.

Altaf, M. S., Bouferguene, A., Liu, H., Al-Hussein, M. and Yu, H. (2018) Integrated production planning and control system for a panelized home prefabrication facility using simulation and RFID, *Automation in Construction*, 85, pp. 369–383. doi: 10.1016/j.autcon.2017.09.009.

Ansari, F. (2020) Cost-based text understanding to improve maintenance knowledge intelligence in manufacturing enterprises, *Computers and Industrial Engineering*, 141. doi: 10.1016/j.cie.2020.106319.

Armstrong, R. A. (2014) When to use the Bonferroni correction, *Ophthalmic & physiological optics : the journal of the British College of Ophthalmic Opticians (Optometrists)*, 34(5), pp. 502–508. doi: 10.1111/opo.12131.

Ayvaz, S. and Alpay, K. (2021) Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time, *Expert Systems with Applications*, 173(November 2020), p. 114598. doi: 10.1016/j.eswa.2021.114598.

Barredo Arrieta, A. *et al.* (2020) Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, *Information Fusion*, 58, pp. 82–115. doi: https://doi.org/10.1016/j.inffus.2019.12.012.

Beel, J., Gipp, B., Langer, S. and Breitinger, C. (2016) Research-paper recommender systems:

a literature survey, *International Journal on Digital Libraries*, 17(4), pp. 305–338. doi: 10.1007/s00799-015-0156-0.

Bergmann, S., Feldkamp, N. and Strassburger, S. (2016) Approximation of dispatching rules for manufacturing simulation using data mining methods, in *2015 Winter Simulation Conference*. Huntington Beach, USA, pp. 2329–2340. doi: 10.1109/WSC.2015.7408344.

Bi, J. and Zhang, C. (2018) An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme, *Knowledge-Based Systems*, 158, pp. 81–93. doi: 10.1016/j.knosys.2018.05.037.

Bird, S. and Loper, E. (2004) NLTK: The Natural Language Toolkit, in *Proceedings of the ACL Interactive Poster and Demonstration Sessions*. Barcelona, Spain: Association for Computational Linguistics, pp. 214–217.

Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y. M. (2020) YOLOv4: Optimal Speed and Accuracy of Object Detection, *arXiv e-prints*, p. arXiv:2004.10934.

Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T. (2016) Enriching Word Vectors with Subword Information, *arXiv e-prints*, p. arXiv:1607.04606.

Bondielli, A. and Marcelloni, F. (2019) A data-driven approach to automatic extraction of professional figure profiles from Résumés, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11871 LNCS, pp. 155–165. doi: 10.1007/978-3-030-33607-3_18.

Borangiu, T., Răileanu, S., Berger, T. and Trentesaux, D. (2015) Switching mode control strategy in manufacturing execution systems, *International Journal of Production Research*, 53(7), pp. 1950–1963. doi: 10.1080/00207543.2014.935825.

Borith, T., Bakhit, S., Nasridinov, A. and Yoo, K. H. (2020) Prediction of machine inactivation status using statistical feature extraction and machine learning, *Applied Sciences (Switzerland)*, 10(21), pp. 1–18. doi: 10.3390/app10217413.

Bruno, N., Jun, T. and Tessier, H. (2019) Natural language processing and classification methods for the maintenance and optimization of US weapon systems, in *2019 Systems and Information Engineering Design Symposium, SIEDS 2019*. Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/SIEDS.2019.8735587.

Cai, B., Liu, H. and Xie, M. (2016) A real-time fault diagnosis methodology of complex systems using object-oriented Bayesian networks, *Mechanical Systems and Signal Processing*, 80, pp. 31–44. doi: 10.1016/j.ymssp.2016.04.019.

Cao, X. C., Chen, B. Q., Yao, B. and He, W. P. (2019) Combining translation-invariant wavelet frames and convolutional neural network for intelligent tool wear state identification, *Computers in Industry*, 106, pp. 71–84. doi: 10.1016/j.compind.2018.12.018.

Carvajal Soto, J. A., Tavakolizadeh, F. and Gyulai, D. (2019) An online machine learning framework for early detection of product failures in an Industry 4.0 context, *International Journal of Computer Integrated Manufacturing*, 32(4–5), pp. 452–465. doi: 10.1080/0951192X.2019.1571238.

Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. da P., Basto, J. P. and Alcalá, S. G. S. (2019) A systematic literature review of machine learning methods applied to predictive maintenance, *Computers and Industrial Engineering*, 137(April), p. 106024. doi: 10.1016/j.cie.2019.106024.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2011) SMOTE: Synthetic Minority Over-sampling Technique, *arXiv e-prints*, p. arXiv:1106.1813.

Chen, C., Xia, B., Zhou, B. hai and Xi, L. (2015) A reinforcement learning based approach for a multiple-load carrier scheduling problem, *Journal of Intelligent Manufacturing*, 26(6), pp. 1233–1245. doi: 10.1007/s10845-013-0852-9.

Chen, L. (2019) A Classification Framework for Online Social Support Using Deep Learning, in Nah, F. F.-H. and Siau, K. (eds) *HCI in Business, Government and Organizations. Information Systems and Analytics*. Cham: Springer International Publishing, pp. 178–188.

Chicco, D. and Jurman, G. (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation, *BMC Genomics*, 21(1), p. 6. doi: 10.1186/s12864-019-6413-7.

Chiu, M. C., Tsai, C. De and Li, T. L. (2020) An Integrative Machine Learning Method to Improve Fault Detection and Productivity Performance in a Cyber-Physical System, *Journal of Computing and Information Science in Engineering*, 20(2), pp. 1–12. doi: 10.1115/1.4045663.

Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. (2014) Empirical Evaluation of Gated

Recurrent Neural Networks on Sequence Modeling, *arXiv e-prints*, p. arXiv:1412.3555.

Cohen, J. (1960) A Coefficient of Agreement for Nominal Scales, *Educational and Psychological Measurement*, 20(1), pp. 37–46. doi: 10.1177/001316446002000104.

Curatolo, N., Lamouri, S., Huet, J. C. and Rieutord, A. (2014) A critical analysis of Lean approach structuring in hospitals, *Business Process Management Journal*, 20(3), pp. 433–454. doi: 10.1108/BPMJ-04-2013-0051.

Dalzochio, J., Kunst, R., Pignaton, E., Binotto, A., Sanyal, S., Favilla, J. and Barbosa, J. (2020) Machine learning and reasoning for predictive maintenance in Industry 4.0: Current status and challenges, *Computers in Industry*, 123, p. 103298. doi: 10.1016/j.compind.2020.103298.

Deepika, S. S., Saranya, M. and Geetha, T. V (2019) Cross-Corpus Training with CNN to Classify Imbalanced Biomedical Relation Data, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Edited by V. S. S. V. S. M. Metais E. Meziane F., 11608 LNCS, pp. 170–181. doi: 10.1007/978-3-030-23281-8_14.

Delgado, R. and Tibau, X.-A. (2019) Why Cohen's Kappa should be avoided as performance measure in classification, *PLOS ONE*, 14(9), pp. 1–26. doi: 10.1371/journal.pone.0222916.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *arXiv e-prints*, p. arXiv:1810.04805.

Diaz-Rozo, J., Bielza, C. and Larrañaga, P. (2017) Machine Learning-based CPS for Clustering High throughput Machining Cycle Conditions, in *45th SME North American Manufacturing Research Conference*. Los Angeles, USA, pp. 997–1008. doi: 10.1016/j.promfg.2017.07.091.

Ding, K. and Jiang, P. (2018) RFID-based Production Data Analysis in an IoT-enabled Smart Job-shop, *IEEE/CAA Journal of Automatica Sinica*, 5(1), pp. 128–138. doi: 10.1109/JAS.2017.7510418.

Dinis, D., Barbosa-Póvoa, A. and Teixeira, Â. P. (2019) Valuing data in aircraft maintenance through big data analytics: A probabilistic approach for capacity planning using Bayesian networks, *Computers and Industrial Engineering*, 128, pp. 920–936. doi: 10.1016/j.cie.2018.10.015.

Dolgui, A., Bakhtadze, N., Pyatetsky, V., Sabitov, R., Smirnova, G., Elpashev, D. and Zakharov, E. (2018) Data Mining-Based Prediction of Manufacturing Situations Data Mining-Based, in *16th IFAC Symposium on Information Control Problems in Manufacturing*. Bergamo, Italy: Elsevier B.V., pp. 316–321. doi: 10.1016/j.ifacol.2018.08.302.

Doltsinis, S., Ferreira, P. and Lohse, N. (2014) An MDP model-based reinforcement learning approach for production station ramp-up optimization: Q-learning analysis, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(9), pp. 1125–1138. doi: 10.1109/TSMC.2013.2294155.

Edwards, B., Zatorsky, M. and Nayak, R. (2008) Clustering and classification of maintenance logs using text data mining, in *Conferences in Research and Practice in Information Technology Series*, pp. 193–199. Available at: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84870484659&partnerID=40&md5=f34d58eba076a357439884fae90dc336.

Fathy, Y., Jaber, M. and Brintrup, A. (2021) Learning with Imbalanced Data in Smart Manufacturing: A Comparative Analysis, *IEEE Access*, 9, pp. 2734–2757. doi: 10.1109/ACCESS.2020.3047838.

Fotuhi, F., Huynh, N., Vidal, J. M. and Xie, Y. (2013) Modeling yard crane operators as reinforcement learning agents, *Research in Transportation Economics*, 42(1), pp. 3–12. doi: 10.1016/j.retrec.2012.11.001.

Gao, X., Shang, C., Jiang, Y., Huang, D. and Chen, T. (2014) Refinery Scheduling with Varying Crude: A Deep Belief Network Classification and Multimodel Approach, *AIChE Journal*, 60(7), pp. 2525–2532. doi: 10.1002/aic.

Garengo, P., Biazzo, S. and Bititci, U. S. (2005) Performance measurement systems in SMEs: A review for a research agenda, *International Journal of Management Reviews*, 7(1), pp. 25–47. doi: 10.1111/j.1468-2370.2005.00105.x.

Géron, A. (2019) *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. O'Reilly Media, Inc.

Ghasemi, A., Yacout, S. and Ouali, M. S. (2007) Optimal condition based maintenance with imperfect information and the proportional hazards model, *International Journal of Production*

Research, 45(4), pp. 989–1012. doi: 10.1080/00207540600596882.

Lo Giudice, P., Musarella, L., Sofo, G. and Ursino, D. (2019) An approach to extracting complex knowledge patterns among concepts belonging to structured, semi-structured and unstructured sources in a data lake, *Information Sciences*, 478, pp. 606–626. doi: 10.1016/j.ins.2018.11.052.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014) Generative Adversarial Networks, *arXiv e-prints*, p. arXiv:1406.2661.

Grabot, B. (2020) Rule mining in maintenance: Analysing large knowledge bases, *Computers and Industrial Engineering*, 139, pp. 1–15. doi: 10.1016/j.cie.2018.11.011.

Gupta, P. and Vardhan, S. (2016) Optimizing OEE, productivity and production cost for improving sales volume in an automobile industry through TPM: A case study, *International Journal of Production Research*, 54(10), pp. 2976–2988. doi: 10.1080/00207543.2016.1145817.

Gyulai, D., Pfeiffer, A., Nick, G., Gallina, V., Sihn, W. and Monostori, L. (2018a) Lead time prediction in a flow-shop environment with analytical and machine learning approaches, in *16th IFAC Symposium on Information Control Problems in Manufacturing*. Bergamo, Italy, pp. 1029–1034. doi: 10.1016/j.ifacol.2018.08.472.

Gyulai, D., Pfeiffer, A., Bergmann, J. and Gallina, V. (2018b) Online lead time prediction supporting situation-aware production control, in *6th CIRP Global Web Conference – Envisaging the future manufacturing, design, technologies and systems in innovation era*, pp. 190–195. doi: 10.1016/j.procir.2018.09.071.

Gyulai, D., Kádár, B. and Monosotori, L. (2015) Robust production planning and capacity control for flexible assembly lines, in *15th IFAC Symposium onInformation Control Problems inManufacturing*. Ottawa, Canada: Elsevier Ltd., pp. 2312–2317. doi: 10.1016/j.ifacol.2015.06.432.

Gyulai, D., Kádár, B. and Monostori, L. (2014) Capacity planning and resource allocation in assembly systems consisting of dedicated and reconfigurable lines, in *8th International Conference on Digital Enterprise Technology*. Stuttgart, Germany: Elsevier B.V., pp. 185–191.

doi: 10.1016/j.procir.2014.10.028.

Habib Zahmani, M. and Atmani, B. (2018) Extraction of Dispatching Rules for Single Machine Total Weighted Tardiness using a Modified Genetic Algorithm and Data Mining, *International Journal of Manufacturing Research*, 13(1), pp. 1–25. doi: 10.1504/ijmr.2018.10007544.

Hammami, Z., Mouelhi, W. and Said, L. Ben (2016) A self adaptive neural agent based decision support system for solving dynamic real time scheduling problems, in *10th International Conference on Intelligent Systems and Knowledge Engineering*. Taipei, Taiwan, pp. 494–501. doi: 10.1109/ISKE.2015.79.

Hammami, Z., Mouelhi, W. and Ben Said, L. (2017) On-line self-adaptive framework for tailoring a neural-agent learning model addressing dynamic real-time scheduling problems, *Journal of Manufacturing Systems*, 45, pp. 97–108. doi: 10.1016/j.jmsy.2017.08.003.

Hand, D. and Christen, P. (2018) A note on using the F-measure for evaluating record linkage algorithms, *Statistics and Computing*, 28(3), pp. 539–547. doi: 10.1007/s11222-017-9746-6.

Harding, J. A., Shahbaz, M., Srinivas and Kusiak, A. (2006) Data Mining in Manufacturing: A Review, *Journal of Manufacturing Science and Engineering-Transactions of the ASME*, 128(4), pp. 969–976. doi: 10.1115/1.2194554.

Heger, J., Branke, J., Hildebrandt, T. and Scholz-Reiter, B. (2016) Dynamic adjustment of dispatching rule parameters in flow shops with sequence-dependent set-up times, *International Journal of Production Research*, 54(22), pp. 6812–6824. doi: 10.1080/00207543.2016.1178406.

Hirschberg, J. and Manning, C. D. (2015) Advances in natural language processing, *Science*, 349(6245), pp. 261 LP – 266. doi: 10.1126/science.aaa8685.

Ho, Y. and Wookey, S. (2020) The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling, *IEEE Access*, 8, pp. 4806–4813. doi: 10.1109/ACCESS.2019.2962617.

Holtzman, A., Buys, J., Du, L., Forbes, M. and Choi, Y. (2019) The Curious Case of Neural Text Degeneration, *arXiv e-prints*, p. arXiv:1904.09751.

Hosseini, S. and Barker, K. (2016) A Bayesian network model for resilience-based supplier

selection, *International Journal of Production Economics*, 180, pp. 68–87. doi: 10.1016/j.ijpe.2016.07.007.

Hosseini, S. and Ivanov, D. (2019) A new resilience measure for supply networks with the ripple effect considerations: a Bayesian network approach, *Annals of Operations Research*. doi: 10.1007/s10479-019-03350-8.

Hosseini, S., Ivanov, D. and Dolgui, A. (2019) Review of quantitative methods for supply chain resilience analysis, *Transportation Research Part E: Logistics and Transportation Review*, 125(December 2018), pp. 285–307. doi: 10.1016/j.tre.2019.03.001.

Huang, B., Wang, W., Ren, S., Zhong, R. Y. and Jiang, J. (2019) A proactive task dispatching method based on future bottleneck prediction for the smart factory, *International Journal of Computer Integrated Manufacturing*, 32(3), pp. 278–293. doi: 10.1080/0951192X.2019.1571241.

Huggingface (2019) *Transformers: State-of-the-art Natural Language Processing for TensorFlow 2.0 and PyTorch*. Available at: https://github.com/huggingface/transformers (Accessed: 1 February 2020).

Hyndman, R. J. and Fan, Y. (1996) Sample Quantiles in Statistical Packages, *The American Statistician*, 50(4), pp. 361–365. doi: 10.1080/00031305.1996.10473566.

Iikura, R., Okada, M. and Mori, N. (2021) Improving bert with focal loss for paragraph segmentation of novels, in *Advances in Intelligent Systems and Computing*. L'Aquila, Italy, 7-10 October 2020, pp. 21–30. doi: 10.1007/978-3-030-53036-5_3.

Ireland, R. and Liu, A. (2018) Application of data analytics for product design: Sentiment analysis of online product reviews, *CIRP Journal of Manufacturing Science and Technology*, 23, pp. 128–144. doi: 10.1016/j.cirpj.2018.06.003.

Ivanov, D., Tang, C. S., Dolgui, A., Battini, D. and Das, A. (2020) Researchers' perspectives on Industry 4.0: multi-disciplinary analysis and opportunities for operations management, *International Journal of Production Research*, 0(0), pp. 1–24. doi: 10.1080/00207543.2020.1798035.

Ji, W. and Wang, L. (2017) Big data analytics based fault prediction for shop floor scheduling, *Journal of Manufacturing Systems*, 43, pp. 187–194. doi: 10.1016/j.jmsy.2017.03.008.

Jiang, S. long, Liu, M., Lin, J. hua and Zhong, H. xing (2016) A prediction-based online soft scheduling algorithm for the real-world steelmaking-continuous casting production, *Knowledge-Based Systems*, 111, pp. 159–172. doi: 10.1016/j.knosys.2016.08.010.

Johnson, J. M. and Khoshgoftaar, T. M. (2019) Survey on deep learning with class imbalance, *Journal of Big Data*, 6(1), p. 27. doi: 10.1186/s40537-019-0192-5.

Jolliffe, I. (2011) Principal Component Analysis, in Lovric, M. (ed.) *International Encyclopedia of Statistical Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1094–1096. doi: 10.1007/978-3-642-04898-2_455.

Jordan, M. I. and Mitchell, T. M. (2015) Machine learning: Trends, perspectives, and prospects, *Science*, 349(6245), pp. 255–260. doi: 10.1126/science.aac4520.

Jurafsky, D. and Martin, J. H. (2009) *Speech and Language Processing (2nd Edition)*. USA: Prentice-Hall, Inc.

Jurkovic, Z., Cukor, G., Brezocnik, M. and Brajkovic, T. (2018) A comparison of machine learning methods for cutting parameters prediction in high speed turning process, *Journal of Intelligent Manufacturing*, 29(8), pp. 1683–1693. doi: 10.1007/s10845-016-1206-1.

Jurman, G., Riccadonna, S. and Furlanello, C. (2012) A comparison of MCC and CEN error measures in multi-class prediction, *PloS one*. 2012/08/08, 7(8), pp. e41882–e41882. doi: 10.1371/journal.pone.0041882.

Kao, A., Niraula, N. B. and Whyatt, D. (2018) PANDA - Discovering Part Name in Noisy Text Data, in *2018 IEEE International Conference on Prognostics and Health Management, ICPHM 2018*. Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/ICPHM.2018.8448708.

Kao, A., Niraula, N. B. and Whyatt, D. (2019) Part name normalization, in *2019 IEEE International Conference on Prognostics and Health Management, ICPHM 2019*. Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/ICPHM.2019.8819386.

Karen, S. J. (1972) A statistical interpretation of term specificity and its application in retrieval, *Journal of Documentation*, 28(1), pp. 11–21. doi: 10.1108/eb026526.

Kartal, H., Oztekin, A., Gunasekaran, A. and Cebi, F. (2016) An integrated decision analytic

framework of machine learning with multi-criteria decision making for multi-attribute inventory classification, *Computers and Industrial Engineering*, 101, pp. 599–613. doi: 10.1016/j.cie.2016.06.004.

Kato, T. and Tsuda, K. (2018) A Management Method of the Corporate Brand Image Based on Customers' Perception, *Procedia Computer Science*, 126, pp. 1368–1377. doi: https://doi.org/10.1016/j.procs.2018.08.088.

Keyvanpour, M. and Serpush, F. (2019) ESLMT: A new clustering method for biomedical document retrieval, *Biomedizinische Technik*. doi: 10.1515/bmt-2018-0068.

Khader, N. and Yoon, S. W. (2018) Online control of stencil printing parameters using reinforcement learning approach, in *28th International Conference on Flexible Automation and Intelligent Manufacturing*. Columbus, USA: Elsevier B.V., pp. 94–101. doi: 10.1016/j.promfg.2018.10.018.

Khalid, W., Albrechtsen, S. H., Sigsgaard, K. V., Mortensen, N. H., Hansen, K. B. and Soleymani, I. (2021) Predicting maintenance work hours in maintenance planning, *Journal of Quality in Maintenance Engineering*, 27(2), pp. 366–384. doi: 10.1108/JQME-06-2019-0058.

Kho, D. D., Lee, S. and Zhong, R. Y. (2018) Big Data Analytics for Processing Time Analysis in an IoT-enabled manufacturing Shop Floor, in *46th SME North American Manufacturing Research Conference*. Texas, USA: Elsevier B.V., pp. 1411–1420. doi: 10.1016/j.promfg.2018.07.107.

Kiangala, K. S. and Wang, Z. (2020) An Effective Predictive Maintenance Framework for Conveyor Motors Using Dual Time-Series Imaging and Convolutional Neural Network in an Industry 4.0 Environment, *IEEE Access*, 8, pp. 121033–121049. doi: 10.1109/ACCESS.2020.3006788.

Kim, H. and Lim, D.-E. (2018) Deep-Learning-Based Storage-Allocation Approach to Improve the AMHS Throughput Capacity in a Semiconductor Fabrication Facility, in *Communications in Computer and Information Science*. Springer Singapore, pp. 232–240. doi: 10.1007/978-981-13-2853-4.

Kim, S. and Nembhard, D. A. (2013) Rule mining for scheduling cross training with a heterogeneous workforce, *International Journal of Production Research*, 51(8), pp. 2281–

2300. doi: 10.1080/00207543.2012.716169.

Kingma, D. P. and Ba, J. (2014) Adam: A Method for Stochastic Optimization, *arXiv e-prints*, p. arXiv:1412.6980.

Kitchenham, B. (2004) *Procedures for Performing Systematic Reviews*. Department of Computer Science, Keele University, UK.

Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M. and Linkman, S. (2010) Systematic literature reviews in software engineering-A tertiary study, *Information and Software Technology*, 52(8), pp. 792–805. doi: 10.1016/j.infsof.2010.03.006.

Koltsidopoulos Papatzimos, A., Dawood, T. and Thies, P. R. (2018) Data Insights from an Offshore Wind Turbine Gearbox Replacement, in Tande J.O.G. Kvamsdal T., M. M. (ed.) *Journal of Physics: Conference Series*. Institute of Physics Publishing. doi: 10.1088/1742-6596/1104/1/012003.

Kosmopoulos, D. I., Doulamis, N. D. and Voulodimos, A. S. (2012) Bayesian filter based behavior recognition in workflows allowing for user feedback, *Computer Vision and Image Understanding*, 116(3), pp. 422–434. doi: 10.1016/j.cviu.2011.09.006.

Kretschmer, R., Pfouga, A., Rulhoff, S. and Stjepandić, J. (2017) Knowledge-based design for assembly in agile manufacturing by using Data Mining methods, *Advanced Engineering Informatics*, 33, pp. 285–299. doi: 10.1016/j.aei.2016.12.006.

Kruger, G. H., Shih, A. J., Hattingh, D. G. and Van Niekerk, T. I. (2011) Intelligent machine agent architecture for adaptive control optimization of manufacturing processes, *Advanced Engineering Informatics*, 25(4), pp. 783–796. doi: 10.1016/j.aei.2011.08.003.

Kudo, T. and Richardson, J. (2018) SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing, *arXiv e-prints*, p. arXiv:1808.06226.

Kumar, A., Shankar, R. and Thakur, L. S. (2018) A big data driven sustainable manufacturing framework for condition-based maintenance prediction, *Journal of Computational Science*, 27, pp. 428–439. doi: 10.1016/j.jocs.2017.06.006.

Kusiak, A. (2017) Smart manufacturing must embrace big data, *Nature*, 544(7648), pp. 23–25.

doi: 10.1038/544023a.

Kusiak, A. (2019) Fundamentals of smart manufacturing: A multi-thread perspective, *Annual Reviews in Control*, 47, pp. 214–220. doi: 10.1016/j.arcontrol.2019.02.001.

Lai, L. K. C. and Liu, J. N. K. (2012) WIPA: Neural network and case base reasoning models for allocating work in progress, *Journal of Intelligent Manufacturing*, 23(3), pp. 409–421. doi: 10.1007/s10845-010-0379-2.

Lai, X., Shui, H. and Ni, J. (2018) A Two-Layer Long Short-Term Memory Network for Bottleneck Prediction in Multi-Job Manufacturing Systems, in *13th International Manufacturing Science and Engineering Conference*. Texas, USA, p. V003T02A014. doi: 10.1115/msec2018-6678.

Le, H. *et al.* (2019) FlauBERT: Unsupervised Language Model Pre-training for French, *arXiv e-prints*, p. arXiv:1912.05372.

Le, Q. and Mikolov, T. (2014) Distributed Representations of Sentences and Documents, in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. JMLR.org (ICML'14), pp. II–1188–II–1196.

Leahy, K., Gallagher, C., O'Donovan, P., Bruton, K. and O'Sullivan, D. T. J. (2018) A robust prescriptive framework and performance metric for diagnosing and predicting wind turbine faults based on SCADA and alarms data with case study, *Energies*, 11(7). doi: 10.3390/en11071738.

Lehmann, C., Goren Huber, L., Horisberger, T., Scheiba, G., Sima, A. C. and Stockinger, K. (2020) Big Data architecture for intelligent maintenance: a focus on query processing and machine learning algorithms, *Journal of Big Data*, 7(1). doi: 10.1186/s40537-020-00340-7.

Lemieux, A. A., Lamouri, S., Pellerin, R. and Tamayo, S. (2015) Development of a leagile transformation methodology for product development, *Business Process Management Journal*, 21(4), pp. 791–819. doi: 10.1108/BPMJ-02-2014-0009.

Leng, J., Chen, Q., Mao, N. and Jiang, P. (2018) Combining granular computing technique with deep learning for service planning under social manufacturing contexts, *Knowledge-Based Systems*, 143, pp. 295–306. doi: 10.1016/j.knosys.2017.07.023.

Li, D. C., Chen, C. C., Chen, W. C. and Chang, C. J. (2012a) Employing dependent virtual samples to obtain more manufacturing information in pilot runs, *International Journal of Production Research*, 50(23), pp. 6886–6903. doi: 10.1080/00207543.2011.631603.

Li, L., Zijin, S., Jiacheng, N. and Fei, Q. (2013) Data-based scheduling framework and adaptive dispatching rule of complex manufacturing systems, *International Journal of Advanced Manufacturing Technology*, 66(9–12), pp. 1891–1905. doi: 10.1007/s00170-012-4468-6.

Li, R., Si, Q., Fu, P., Lin, Z., Wang, W. and Shi, G. (2019) A multi-channel neural network for imbalanced emotion recognition, in *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*. Portland, OR, USA, 4-6 November 2019, pp. 353–360. doi: 10.1109/ICTAI.2019.00057.

Li, X., Duan, F., Loukopoulos, P., Bennett, I. and Mba, D. (2018) Canonical variable analysis and long short-term memory for fault diagnosis and performance estimation of a centrifugal compressor, *Control Engineering Practice*, 72(January), pp. 177–191. doi: 10.1016/j.conengprac.2017.12.006.

Li, X., Wang, J. and Sawhney, R. (2012b) Reinforcement learning for joint pricing, lead-time and scheduling decisions in make-to-order systems, *European Journal of Operational Research*, 221(1), pp. 99–109. doi: 10.1016/j.ejor.2012.03.020.

Liao, Q. (2018) Study of SVM-based intelligent dispatcher for parallel machines scheduling with sequence-dependent setup times, in *6th International Conference on Mechanical, Automotive and Materials Engineering, CMAME 2018*. Hong Kong: IEEE, pp. 46–50. doi: 10.1109/CMAME.2018.8592381.

Lieber, D., Stolpe, M., Konrad, B., Deuse, J. and Morik, K. (2013) Quality prediction in interlinked manufacturing processes based on supervised & unsupervised machine learning, in *46th CIRP Conference on Manufacturing Systems 2013*. Setúbal, Portugal: Elsevier B.V., pp. 193–198. doi: 10.1016/j.procir.2013.05.033.

Lin, T.-Y., Goyal, P., Girshick, R., He, K. and Dollár, P. (2017) Focal Loss for Dense Object Detection, *arXiv e-prints*, p. arXiv:1708.02002.

Lingitz, L., Gallina, V., Ansari, F., Gyulai, D., Pfeiffer, A. and Sihn, W. (2018) Lead time prediction using machine learning algorithms: A case study by a semiconductor manufacturer,

in *51st CIRP Conference on Manufacturing Systems*. Stockholm, Sweden, pp. 1051–1056. doi: 10.1016/j.procir.2018.03.148.

Liu, C., Tang, D., Zhu, H. and Nie, Q. (2021) A Novel Predictive Maintenance Method Based on Deep Adversarial Learning in the Intelligent Manufacturing System, *IEEE Access*, 9, pp. 49557–49575. doi: 10.1109/ACCESS.2021.3069256.

Liu, Y. *et al.* (2019) RoBERTa: A Robustly Optimized BERT Pretraining Approach, *arXiv e-prints*, p. arXiv:1907.11692.

Llave, M. R. (2018) Data lakes in business intelligence: Reporting from the trenches, in *CENTERIS/ProjMAN/HCist 2018*. Lisbon, Portugal: Elsevier B.V., pp. 516–524. doi: 10.1016/j.procs.2018.10.071.

Loshchilov, I. and Hutter, F. (2017) Decoupled Weight Decay Regularization, *arXiv e-prints*, p. arXiv:1711.05101.

Lubosch, M., Kunath, M. and Winkler, H. (2018) Industrial scheduling with Monte Carlo tree search and machine learning, in *51st CIRP Conference on Manufacturing Systems*. Stockholm, Sweden: Elsevier B.V., pp. 1283–1287. doi: 10.1016/j.procir.2018.03.171.

Lundberg, S. and Lee, S.-I. (2017) A Unified Approach to Interpreting Model Predictions, *arXiv e-prints*, p. arXiv:1705.07874.

Lv, S., Zheng, B., Kim, H. and Yue, Q. (2018a) Data Mining for Material Feeding Optimization of Printed Circuit Board Template Production, *Journal of Electrical and Computer Engineering*, 2018. doi: 10.1155/2018/1852938.

Lv, Y., Qin, W., Yang, J. and Zhang, J. (2018b) Adjustment mode decision based on support vector data description and evidence theory for assembly lines, *Industrial Management and Data Systems*, 118(8), pp. 1711–1726. doi: 10.1108/IMDS-01-2017-0014.

Lynn, T., Endo, P. T., Rosati, P., Silva, I., Santos, G. L. and Ging, D. (2019) A comparison of machine learning approaches for detecting misogynistic speech in urban dictionary, in *2019 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (Cyber SA 2019)*. Oxford, United Kingdom: IEEE, pp. 1–8. doi: 10.1109/CyberSA.2019.8899669.

Ma, Y., Qiao, F., Zhao, F. and Sutherland, J. (2017) Dynamic Scheduling of a Semiconductor Production Line Based on a Composite Rule Set, *Applied Sciences*, 7(10), p. 1052. doi: 10.3390/app7101052.

Madhusudanan, N., Gurumoorthy, B. and Chakrabarti, A. (2017) Automatic expert knowledge acquisition from text for closing the knowledge loop in PLM, *International Journal of Product Lifecycle Management*, 10(4), pp. 301–325. doi: 10.1504/IJPLM.2017.090327.

Madhusudanan, N., Gurumoorthy, B. and Chakrabarti, A. (2019) From natural language text to rules: knowledge acquisition from formal documents for aircraft assembly, *Journal of Engineering Design*, 30(10–12), pp. 417–444. doi: 10.1080/09544828.2019.1630804.

Maghrebi, M., Shamsoddini, A. and Waller, S. T. (2016) Fusion based learning approach for predicting concrete pouring productivity based on construction and supply parameters, *Construction Innovation*, 16(2), pp. 185–202.

Mahdavinejad, M. S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P. and Sheth, A. P. (2018) Machine learning for internet of things data analysis: a survey, *Digital Communications and Networks*, 4(3), pp. 161–175. doi: 10.1016/j.dcan.2017.10.002.

Manns, M., Wallis, R. and Deuse, J. (2015) Automatic proposal of assembly work plans with a controlled natural language, in *9th CIRP Conference on Intelligent Computation in Manufacturing Engineering*. Capri, Italy, pp. 345–350. doi: 10.1016/j.procir.2015.06.079.

Manupati, V. K., Anand, R., Thakkar, J. J., Benyoucef, L., Garsia, F. P. and Tiwari, M. K. (2013) Adaptive production control system for a flexible manufacturing cell using support vector machine-based approach, *International Journal of Advanced Manufacturing Technology*, 67(1–4), pp. 969–981. doi: 10.1007/s00170-012-4541-1.

Martin, L., Muller, B., Ortiz Suárez, P. J., Dupont, Y., Romary, L., Villemonte de la Clergerie, É., Seddah, D. and Sagot, B. (2019) CamemBERT: a Tasty French Language Model, *arXiv e-prints*, p. arXiv:1911.03894. Available at: https://ui.adsabs.harvard.edu/abs/2019arXiv191103894M.

Masko, D. and Hensman, P. (2015) *The Impact of Imbalanced Training Data for Convolutional Neural Networks*. KTH ROYAL INSTITUTE OF TECHNOLOGY.

McCann, B., Bradbury, J., Xiong, C. and Socher, R. (2017) Learned in Translation:

Contextualized Word Vectors, *arXiv e-prints*, p. arXiv:1708.00107.

McCormick, C. (2020) *The Inner Workings of BERT eBook*. Available at: https://www.chrismccormick.ai/the-bert-collection (Accessed: 9 March 2021).

McCormick, C. and Ryan, N. (2019) *BERT Fine-Tuning Tutorial with PyTorch*. Available at: https://mccormickml.com/2019/07/22/BERT-fine-tuning/ (Accessed: 1 February 2020).

Mikołajczyk, A. and Grochowski, M. (2018) Data augmentation for improving deep learning in image classification problem, in *2018 International Interdisciplinary PhD Workshop, IIPhDW 2018*. Swinoujście, Poland: IEEE, pp. 117–122. doi: 10.1109/IIPHDW.2018.8388338.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J. (2013) Distributed Representations of Words and Phrases and Their Compositionality, in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. Red Hook, NY, USA: Curran Associates Inc. (NIPS'13), pp. 3111–3119.

Mitchell, T. (1997) Machine Learning, in. McGraw-Hill, p. 2.

Moeuf, A., Pellerin, R., Lamouri, S., Tamayo-Giraldo, S. and Barbaray, R. (2018) The industrial management of SMEs in the era of Industry 4.0, *International Journal of Production Research*, 56(3), pp. 1118–1136. doi: 10.1080/00207543.2017.1372647.

Montero Jimenez, J. J., Schwartz, S., Vingerhoeds, R., Grabot, B. and Salaün, M. (2020) Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics, *Journal of Manufacturing Systems*, 56(May), pp. 539–557. doi: 10.1016/j.jmsy.2020.07.008.

Mori, J. and Mahalec, V. (2015) Planning and scheduling of steel plates production. Part I: Estimation of production times via hybrid Bayesian networks for large domain of discrete variables, *Computers and Chemical Engineering*, 79, pp. 113–134. doi: 10.1016/j.compchemeng.2015.02.005.

Muñoz, E. and Capón-García, E. (2019) Systematic approach of multi-label classification for production scheduling, *Computers and Chemical Engineering*, 122, pp. 238–246. doi: 10.1016/j.compchemeng.2018.08.020.

Nazir, Q. and Shao, C. (2021) Online tool condition monitoring for ultrasonic metal welding

via sensor fusion and machine learning, *Journal of Manufacturing Processes*, 62(August 2020), pp. 806–816. doi: 10.1016/j.jmapro.2020.12.050.

Nguyen, A., Usuga-Cadavid, J. P., Lamouri, S., Grabot, B. and Pellerin, R. (2021) *Understanding Data-Related Concepts in Smart Manufacturing and Supply Chain Through Text Mining, Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future. SOHOMA 2020. Studies in Computational Intelligence, vol 952.* Edited by L. S. Borangiu T., Trentesaux D., Leitão P., Cardin O. Paris, France: Springer, Cham. doi: 10.1007/978-3-030-69373-2_37.

Nixon, S., Weichel, R., Reichard, K. and Kozlowski, J. (2018) A machine learning approach to diesel engine health prognostics using engine controller data, in Bregon A., O. M. (ed.) *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM*. Prognostics and Health Management Society. Available at: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85071453547&partnerID=40&md5=597bb684555e4987ace04156ade1e229.

Nnamoko, N., Cabrera-Diego, L. A., Campbell, D. and Korkontzelos, Y. (2019) Bug Severity Prediction Using a Hierarchical One-vs.-Remainder Approach, in Metais E. Meziane F., V. S. S. V. S. M. (ed.) *24th International Conference on Applications of Natural Language to Information Systems*. Salford, UK, 26–28 June 2019: Springer Verlag, pp. 247–260. doi: 10.1007/978-3-030-23281-8_20.

Orozco, R., Sheng, S. and Phillips, C. (2018) Diagnostic Models for Wind Turbine Gearbox Components Using SCADA Time Series Data, in *2018 IEEE International Conference on Prognostics and Health Management, ICPHM 2018*. Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/ICPHM.2018.8448545.

Ortega, M., Ivorra, E., Juan, A., Venegas, P., Martínez, J. and Alcañiz, M. (2021) Mantra: An effective system based on augmented reality and infrared thermography for industrial maintenance, *Applied Sciences (Switzerland)*, 11(1), pp. 1–26. doi: 10.3390/app11010385.

Ou, X., Chang, Q., Arinez, J. and Zou, J. (2018) Gantry Work Cell Scheduling through Reinforcement Learning with Knowledge-guided Reward Setting, *IEEE Access*, 6, pp. 14699–14709. doi: 10.1109/ACCESS.2018.2800641.

Ou, X., Chang, Q. and Chakraborty, N. (2019) Simulation study on reward function of

reinforcement learning in gantry work cell scheduling, *Journal of Manufacturing Systems*, 50, pp. 1–8. doi: 10.1016/j.jmsy.2018.11.005.

Palombarini, J. and Martínez, E. (2012) SmartGantt - An intelligent system for real time rescheduling based on relational reinforcement learning, *Expert Systems with Applications*, 39(11), pp. 10251–10268. doi: 10.1016/j.eswa.2012.02.176.

Pan, S. J. and Yang, Q. (2010) A survey on transfer learning, *IEEE Transactions on Knowledge and Data Engineering*, 22(10), pp. 1345–1359. doi: 10.1109/TKDE.2009.191.

Panicucci, S. *et al.* (2020) A cloud-to-edge approach to support predictive analytics in robotics industry, *Electronics (Switzerland)*, 9(3), pp. 1–22. doi: 10.3390/electronics9030492.

Pedregosa, F. *et al.* (2011) Scikit-Learn: Machine Learning in Python, *Journal of Machine Learning Research*, 12, pp. 2825–2830.

Pennington, J., Socher, R. and Manning, C. D. (2014) GloVe: Global Vectors for Word Representation, in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543. Available at: http://www.aclweb.org/anthology/D14-1162.

Perez, L. and Wang, J. (2017) The Effectiveness of Data Augmentation in Image Classification using Deep Learning, *arXiv e-prints*, p. arXiv:1712.04621.

Peters, M. E., Ammar, W., Bhagavatula, C. and Power, R. (2017) Semi-supervised sequence tagging with bidirectional language models, *arXiv e-prints*, p. arXiv:1705.00108.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L. (2018) Deep contextualized word representations, *arXiv e-prints*, p. arXiv:1802.05365.

Pham, H. *et al.* (2020) Robust Handwriting Recognition with Limited and Noisy Data, in *Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR*. Institute of Electrical and Electronics Engineers Inc., pp. 301–306. doi: 10.1109/ICFHR2020.2020.00062.

Powers, D. M. W. (2011) Evaluation: From Precision, Recall and F-Measure To Roc, Informedness, Markedness & Correlation, *Journal of Machine Learning Technology*, 2(1), pp. 37–63.

Priore, P., Ponte, B., Puente, J. and Gómez, A. (2018) Learning-based scheduling of flexible

manufacturing systems using ensemble methods, *Computers and Industrial Engineering*, 126, pp. 282–291. doi: 10.1016/j.cie.2018.09.034.

Qu, S., Chu, T., Wang, J., Leckie, J. and Jian, W. (2015) A centralized reinforcement learning approach for proactive scheduling in manufacturing, in *IEEE International Conference on Emerging Technologies and Factory Automation*. Luxembourg, Luxembourg. doi: 10.1109/ETFA.2015.7301417.

Qu, S., Wang, J., Govil, S. and Leckie, J. O. (2016a) Optimized Adaptive Scheduling of a Manufacturing Process System with Multi-skill Workforce and Multiple Machine Types: An Ontology-based, Multi-agent Reinforcement Learning Approach, in *49th CIRP Conference on Manufacturing Systems (CIRP-CMS 2016)*. Stuttgart, Germany: Elsevier B.V., pp. 55–60. doi: 10.1016/j.procir.2016.11.011.

Qu, S., Jie, W. and Shivani, G. (2016b) Learning adaptive dispatching rules for a manufacturing process system by using reinforcement learning approach, in *IEEE International Conference on Emerging Technologies and Factory Automation*. Berlin, Germany. doi: 10.1109/ETFA.2016.7733712.

Quatrini, E., Costantino, F., Di Gravio, G. and Patriarca, R. (2020) Machine learning for anomaly detection and process phase classification to improve safety and maintenance activities, *Journal of Manufacturing Systems*, 56(June), pp. 117–132. doi: 10.1016/j.jmsy.2020.05.013.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I. (2018) Language Models are Unsupervised Multitask Learners. Available at: https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

Radford, A. and Salimans, T. (2018) *Improving Language Understanding by Generative Pre-Training, OpenAI*. Available at: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

Raheja, D., Llinas, J., Nagi, R. and Romanowski, C. (2006) Data fusion/data mining-based architecture for condition-based maintenance, *International Journal of Production Research*, 44(14), pp. 2869–2887. doi: 10.1080/00207540600654509.

Rainer, C. (2013) *Data Mining as Technique to Generate Planning Rules for Manufacturing*

Control in a Complex Production System, *Robust Manufacturing Control*. Edited by K. Windt. Heidelberg, Germany: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-30749-2.

Razmi-Farooji, A., Kropsu-Vehkaperä, H., Härkönen, J. and Haapasalo, H. (2019) Advantages and potential challenges of data management in e-maintenance, *Journal of Quality in Maintenance Engineering*, 25(3), pp. 378–396. doi: 10.1108/JQME-03-2018-0018.

Reboiro-Jato, M., Glez-Dopazo, J., Glez, D., Laza, R., Gálvez, J. F., Pavón, R., Glez-Peña, D. and Fdez-Riverola, F. (2011) Using inductive learning to assess compound feed production in cooperative poultry farms, *Expert Systems with Applications*, 38(11), pp. 14169–14177. doi: 10.1016/j.eswa.2011.04.228.

Reuter, C., Brambring, F., Weirich, J. and Kleines, A. (2016) Improving Data Consistency in Production Control by Adaptation of Data Mining Algorithms, in *9th International Conference on Digital Enterprise Technology*. Nanjing, China, pp. 545–550. doi: 10.1016/j.procir.2016.10.107.

Ribeiro, M. T., Singh, S. and Guestrin, C. (2016) 'Why Should I Trust You?''': Explaining the Predictions of Any Classifier', *arXiv e-prints*, p. arXiv:1602.04938.

Rogers, A., Kovaleva, O. and Rumshisky, A. (2020) A Primer in BERTology: What we know about how BERT works, *arXiv e-prints*, p. arXiv:2002.12327.

Rostami, H., Blue, J. and Yugma, C. (2018) Automatic equipment fault fingerprint extraction for the fault diagnostic on the batch process data, *Applied Soft Computing*, 68, pp. 972–989. doi: 10.1016/j.asoc.2017.10.029.

Rousseeuw, P. J. (1987) Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics*, 20, pp. 53–65. doi: https://doi.org/10.1016/0377-0427(87)90125-7.

Ruessmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P. and Harnisch, M. (2015) Industry 4.0: The Future of Productivity and Growth in Manufacturing, *The Boston Consulting Group, Vol. 9*, pp. 1–20.

Ruiz-Sarmiento, J. R., Monroy, J., Moreno, F. A., Galindo, C., Bonelo, J. M. and Gonzalez-Jimenez, J. (2020) A predictive model for the maintenance of industrial machinery in the context of industry 4.0, *Engineering Applications of Artificial Intelligence*, 87(January 2019),

p. 103289. doi: 10.1016/j.engappai.2019.103289.

Saenz De Ugarte, B., Artiba, A. and Pellerin, R. (2009) Manufacturing execution system - A literature review, *Production Planning and Control*, 20(6), pp. 525–539. doi: 10.1080/09537280902938613.

Sahebjamnia, N., Tavakkoli-Moghaddam, R. and Ghorbani, N. (2016) Designing a fuzzy Q-learning multi-agent quality control system for a continuous chemical production line - A case study, *Computers and Industrial Engineering*, 93, pp. 215–226. doi: 10.1016/j.cie.2016.01.004.

Sahu, C. K., Young, C. and Rai, R. (2020) Artificial intelligence (AI) in augmented reality (AR)-assisted manufacturing applications: a review, *International Journal of Production Research*, pp. 1–57. doi: 10.1080/00207543.2020.1859636.

Schreck, G., Lisounkin, A. and Krüger, J. (2008) Knowledge modelling for rule-based supervision and control of production facilities, *International Journal of Production Research*, 46(9), pp. 2531–2546. doi: 10.1080/00207540701738128.

Schuh, G., Reuter, C., Prote, J. P., Brambring, F. and Ays, J. (2017a) Increasing data integrity for improving decision making in production planning and control, *CIRP Annals - Manufacturing Technology*, 66(1), pp. 425–428. doi: 10.1016/j.cirp.2017.04.003.

Schuh, G., Prote, J. P., Luckert, M. and Hünnekes, P. (2017b) Knowledge Discovery Approach for Automated Process Planning, in *50th CIRP Conference on Manufacturing Systems Knowledge*. Taichung, Taiwan, pp. 539–544. doi: 10.1016/j.procir.2017.03.092.

Schuh, G., Reinhart, G., Prote, J. P., Sauermann, F., Horsthofer, J., Oppolzer, F. and Knoll, D. (2019) Data mining definitions and applications for the management of production complexity, in *52nd CIRP Conference on Manufacturing Systems*. Ljubljana, Slovenia: Elsevier B.V., pp. 874–879. doi: 10.1016/j.procir.2019.03.217.

Sennrich, R., Haddow, B. and Birch, A. (2015) Neural Machine Translation of Rare Words with Subword Units, *arXiv e-prints*, p. arXiv:1508.07909.

Sexton, T., Brundage, M. P., Hoffman, M. and Morris, K. C. (2017) Hybrid datafication of maintenance logs from AI-assisted human tags, in *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*. Boston, United States, pp. 1769–1777. doi: 10.1109/BigData.2017.8258120.

Shahzad, A. and Mebarki, N. (2012) Data mining based job dispatching using hybrid simulation-optimization approach for shop scheduling problem, *Engineering Applications of Artificial Intelligence*, 25(6), pp. 1173–1181. doi: 10.1016/j.engappai.2012.04.001.

Sharp, M., Ak, R. and Hedberg, T. (2018) A survey of the advancing use and development of machine learning in smart manufacturing, *Journal of Manufacturing Systems*, 48, pp. 170–179. doi: 10.1016/j.jmsy.2018.02.004.

Sharp, M., Sexton, T. and Brundage, M. P. (2017) Toward Semi-autonomous Information: Extraction for Unstructured Maintenance Data in Root Cause Analysis, *IFIP Advances in Information and Communication Technology*. Edited by L. H. T. K.-D. K. D. Riedel R. von Cieminski G., 513, pp. 425–432. doi: 10.1007/978-3-319-66923-6_50.

Shiue, Y. R., Guh, R. S. and Tseng, T. Y. (2012) Study on shop floor control system in semiconductor fabrication by self-organizing map-based intelligent multi-controller, *Computers and Industrial Engineering*, 62(4), pp. 1119–1129. doi: 10.1016/j.cie.2012.01.004.

Shiue, Y. R., Lee, K. C. and Su, C. T. (2018) Real-time scheduling for a smart factory using a reinforcement learning approach, *Computers and Industrial Engineering*, 125(101), pp. 604–614. doi: 10.1016/j.cie.2018.03.039.

Shorten, C. and Khoshgoftaar, T. M. (2019) A survey on Image Data Augmentation for Deep Learning, *Journal of Big Data*, 6(1), p. 60. doi: 10.1186/s40537-019-0197-0.

Slack, N., Chambers, S. and Johnston, R. (2007) *Operations management*. 5th ed. Prentice Hall/Financial Times.

Solti, A., Raffel, M., Romagnoli, G. and Mendling, J. (2018) Misplaced product detection using sensor data without planograms, *Decision Support Systems*, 112, pp. 76–87. doi: 10.1016/j.dss.2018.06.006.

Soualhi, M., El Koujok, M., Nguyen, K. T. P., Medjaher, K., Ragab, A., Ghezzaz, H., Amazouz, M. and Ouali, M. S. (2021) Adaptive prognostics in a controlled energy conversion process based on long- and short-term predictors, *Applied Energy*, 283(June 2020), p. 116049. doi: 10.1016/j.apenergy.2020.116049.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014) Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *J. Mach. Learn. Res.*, 15(1), pp.

1929–1958.

Stein, N., Meller, J. and Flath, C. M. (2018) Big data on the shop-floor: sensor-based decision-support for manual processes, *Journal of Business Economics*, 88(5), pp. 593–616. doi: 10.1007/s11573-017-0890-4.

Stricker, N., Kuhnle, A., Sturm, R. and Friess, S. (2018) Reinforcement learning for adaptive order dispatching in the semiconductor industry, *CIRP Annals - Manufacturing Technology*, 67(1), pp. 511–514. doi: 10.1016/j.cirp.2018.04.041.

Subramaniyan, M., Skoogh, A., Muhammad, A. S., Bokrantz, J., Johansson, B. and Roser, C. (2020) A data-driven approach to diagnosing throughput bottlenecks from a maintenance perspective, *Computers and Industrial Engineering*, 150(February). doi: 10.1016/j.cie.2020.106851.

Sullivan, G., Pugh, R., Melendez, A. P. and Hunt, W. D. (2010) *Operations & Maintenance Best Practices - A Guide to Achieving Operational Efficiency (Release 3)*. doi: 10.2172/1034595.

Talhi, A., Fortineau, V., Huet, J. C. and Lamouri, S. (2017) Ontology for cloud manufacturing based Product Lifecycle Management, *Journal of Intelligent Manufacturing*, 30(5), pp. 1–22. doi: 10.1007/s10845-017-1376-5.

Tao, F., Qi, Q., Liu, A. and Kusiak, A. (2018) Data-driven smart manufacturing, *Journal of Manufacturing Systems*, 48, pp. 157–169. doi: 10.1016/j.jmsy.2018.01.006.

Thomas, A., Noyel, M., Zimmermann, E., Suhner, M.-C., Bril El Haouzi, H. and Thomas, P. (2018a) Using a classifier ensemble for proactive quality monitoring and control: The impact of the choice of classifiers types, selection criterion, and fusion process, *Computers in Industry*, 99(March), pp. 193–204. doi: 10.1016/j.compind.2018.03.038.

Thomas, T. E., Koo, J., Chaterji, S. and Bagchi, S. (2018b) Minerva: A reinforcement learning-based technique for optimal scheduling and bottleneck detection in distributed factory operations, in *10th International Conference on Communication Systems and Networks*. Bengaluru, India, pp. 129–136. doi: 10.1109/COMSNETS.2018.8328189.

Tian, G., Zhou, M. and Chu, J. (2013) A chance constrained programming approach to determine the optimal disassembly sequence, *IEEE Transactions on Automation Science and*

Engineering, 10(4), pp. 1004–1013. doi: 10.1109/TASE.2013.2249663.

Tong, Y., Li, J., Li, S. and Li, D. (2016) Research on Energy-Saving Production Scheduling Based on a Clustering Algorithm for a Forging Enterprise, *Sustainability*, 8(2), p. Article number 136. doi: 10.3390/su8020136.

Tony Arnold, J. R., Chapman, S. N. and Clive, L. M. (2012) Introduction to Materials Management, in. Pearson, p. 118.

Traini, E., Bruno, G. and Lombardi, F. (2020) Tool condition monitoring framework for predictive maintenance: a case study on milling process, *International Journal of Production Research*. doi: 10.1080/00207543.2020.1836419.

Tranfield, D., Denyer, D. and Smart, P. (2003) Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review, *British Journal of Management*, 14, pp. 207–222.

Trappey, A. J. C., Chen, P. P. J., Trappey, C. V. and Ma, L. (2019) A machine learning approach for solar power technology review and patent evolution analysis, *Applied Sciences (Switzerland)*, 9(7), p. Article number 1478. doi: 10.3390/app9071478.

Tuncel, E., Zeid, A. and Kamarthi, S. (2014) Solving large scale disassembly line balancing problem with uncertainty using reinforcement learning, *Journal of Intelligent Manufacturing*, 25(4), pp. 647–659. doi: 10.1007/s10845-012-0711-0.

Usuga Cadavid, J. P., Lamouri, S., Grabot, B., Pellerin, R. and Fortin, A. (2019) Estimation of Production Inhibition Time Using Data Mining to Improve Production Planning and Control, in *2019 International Conference on Industrial Engineering and Systems Management (IESM)*. Shanghai, China, 25-27 September 2019, pp. 1–6. doi: 10.1109/IESM45758.2019.8948129.

Usuga Cadavid, J. P., Lamouri, S., Grabot, B., Pellerin, R. and Fortin, A. (2020a) Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0, *Journal of Intelligent Manufacturing*, 31(6), pp. 1531–1558. doi: 10.1007/s10845-019-01531-7.

Usuga Cadavid, J. P., Grabot, B., Lamouri, S., Pellerin, R. and Fortin, A. (2020b) Valuing free-form text data from maintenance logs through transfer learning with CamemBERT, *Enterprise Information Systems*, (In Press), pp. 1–29. doi: 10.1080/17517575.2020.1790043.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I. (2017) Attention is All you Need, in Guyon, I. et al. (eds) *Advances in Neural Information Processing Systems 30*. Long Beach, CA, USA, 4-9 December 2017: Curran Associates, Inc., pp. 5998–6008. Available at: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Wang, C. and Jiang, P. (2018) Manifold learning based rescheduling decision mechanism for recessive disturbances in RFID-driven job shops, *Journal of Intelligent Manufacturing*, 29(7), pp. 1485–1500. doi: 10.1007/s10845-016-1194-1.

Wang, C. and Jiang, P. (2019) Deep neural networks based order completion time prediction by using real-time job shop RFID data, *Journal of Intelligent Manufacturing*, 30(3), pp. 1303–1318. doi: 10.1007/s10845-017-1325-3.

Wang, C. L., Rong, G., Weng, W. and Feng, Y. P. (2015) Mining scheduling knowledge for job shop scheduling problem, in *15th IFAC Symposium on Information Control Problems in Manufacturing*. Ottawa, Canada: Elsevier Ltd., pp. 800–805. doi: 10.1016/j.ifacol.2015.06.181.

Wang, H., Jiang, Z., Zhang, X., Wang, Y. and Wang, Y. (2017) A fault feature characterization based method for remanufacturing process planning optimization, *Journal of Cleaner Production*, 161, pp. 708–719. doi: 10.1016/j.jclepro.2017.05.178.

Wang, H. X. and Yan, H. Sen (2016) An interoperable adaptive scheduling strategy for knowledgeable manufacturing based on SMGWQ-learning, *Journal of Intelligent Manufacturing*, 27(5), pp. 1085–1095. doi: 10.1007/s10845-014-0936-1.

Wang, J., Yang, J., Zhang, J., Wang, X. and Zhang, W. (2018a) Big data driven cycle time parallel prediction for production planning in wafer manufacturing, *Enterprise Information Systems*, 12(6), pp. 714–732. doi: 10.1080/17517575.2018.1450998.

Wang, J., Ma, Y., Zhang, L., Gao, R. X. and Wu, D. (2018b) Deep learning for smart manufacturing: Methods and applications, *Journal of Manufacturing Systems*, 48, pp. 144–156. doi: 10.1016/j.jmsy.2018.01.003.

Wang, J., Zhang, J. and Wang, X. (2018c) Bilateral LSTM: A two-dimensional long short-term memory model with multiply memory units for short-term cycle time forecasting in re-entrant manufacturing systems, *IEEE Transactions on Industrial Informatics*, 14(2), pp. 748–758. doi:

10.1109/TII.2017.2754641.

Wang, S., Liu, W., Wu, J., Cao, L., Meng, Q. and Kennedy, P. J. (2016) Training deep neural networks on imbalanced data sets, in *Proceedings of the International Joint Conference on Neural Networks*. Vancouver, BC, Canada, 24-29 July: Institute of Electrical and Electronics Engineers Inc., pp. 4368–4374. doi: 10.1109/IJCNN.2016.7727770.

Waschneck, B., Bauernhansl, T., Knapp, A. and Kyek, A. (2018) Optimization of global production scheduling with deep reinforcement learning, in *51st CIRP Conference on Manufacturing Systems*. Stockholm, Sweden, pp. 1264–1269. doi: 10.1016/j.procir.2018.03.212.

Wauters, T., Verbeeck, K., Verstraete, P., Vanden Berghe, G. and De Causmaecker, P. (2012) Real-world production scheduling for the food industry: An integrated approach, *Engineering Applications of Artificial Intelligence*, 25(2), pp. 222–228. doi: 10.1016/j.engappai.2011.05.002.

Wolf, T. *et al.* (2019) HuggingFace's Transformers: State-of-the-art Natural Language Processing, *arXiv e-prints*, p. arXiv:1910.03771.

Woolf, M. (2019) *gpt-2-simple*. Available at: https://github.com/minimaxir/gpt-2-simple (Accessed: 30 April 2020).

Wu, B., Liu, L., Yang, Y., Zheng, K. and Wang, X. (2020) Using improved conditional generative adversarial networks to detect social bots on Twitter, *IEEE Access*, 8, pp. 36664–36680. doi: 10.1109/ACCESS.2020.2975630.

Wu, W., Ma, Y., Qiao, F. and Gu, X. (2015) Data mining based dynamic scheduling approach for semiconductor manufacturing system, in *34th Chinese Control Conference*. Hangzhou, China, pp. 2603–2608. doi: 10.1109/ChiCC.2015.7260038.

Wu, Y. *et al.* (2016) Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, *arXiv e-prints*, p. arXiv:1609.08144.

Xanthopoulos, A. S., Kiatipis, A., Koulouriotis, D. E. and Stieger, S. (2017) Reinforcement Learning-Based and Parametric Production-Maintenance Control Policies for a Deteriorating Manufacturing System, *IEEE Access*, 6, pp. 576–588. doi: 10.1109/ACCESS.2017.2771827.

Yang, Z., Zhang, P. and Chen, L. (2016) RFID-enabled indoor positioning method for a real-time manufacturing execution system using OS-ELM, *Neurocomputing*, 174, pp. 121–133. doi: 10.1016/j.neucom.2015.05.120.

Yeh, D. Y., Cheng, C. H. and Hsiao, S. C. (2011) Classification knowledge discovery in mold tooling test using decision tree algorithm, *Journal of Intelligent Manufacturing*, 22(4), pp. 585–595. doi: 10.1007/s10845-009-0321-7.

Yuan, B., Wang, L. and Jiang, Z. (2014) Dynamic parallel machine scheduling using the learning agent, in *2013 IEEE International Conference on Industrial Engineering and Engineering Management*. Bangkok, Thailand, pp. 1565–1569. doi: 10.1109/IEEM.2013.6962673.

Zellner, G. (2011) A structured evaluation of business process improvement approaches, *Business Process Management Journal*, 17(2), pp. 203–237. doi: 10.1108/14637151111122329.

Zhai, S., Gehring, B. and Reinhart, G. (2021) Enabling predictive maintenance integrated production scheduling by operation-specific health prognostics with generative deep learning, *Journal of Manufacturing Systems*, (March). doi: 10.1016/j.jmsy.2021.02.006.

Zhang, W., Yang, D. and Wang, H. (2019) Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey, *IEEE Systems Journal*, 13(3), pp. 2213–2227. doi: 10.1109/JSYST.2019.2905565.

Zhang, Z., Zheng, L., Hou, F. and Li, N. (2011) Semiconductor final test scheduling with Sarsa(λ, k) algorithm, *European Journal of Operational Research*, 215(2), pp. 446–458. doi: 10.1016/j.ejor.2011.05.052.

Zhang, Z., Zheng, L., Li, N., Wang, W., Zhong, S. and Hu, K. (2012) Minimizing mean weighted tardiness in unrelated parallel machine scheduling with reinforcement learning, *Computers and Operations Research*, 39(7), pp. 1315–1324. doi: 10.1016/j.cor.2011.07.019.

Zhong, R. Y., Huang, G. Q., Dai, Q. Y. and Zhang, T. (2014) Mining SOTs and dispatching rules from RFID-enabled real-time shopfloor production data, *Journal of Intelligent Manufacturing*, 25(4), pp. 825–843. doi: 10.1007/s10845-012-0721-y.

Zhong, R. Y., Newman, S. T., Huang, G. Q. and Lan, S. (2016) Big Data for supply chain

management in the service and manufacturing sectors: Challenges, opportunities, and future perspectives, *Computers and Industrial Engineering*, 101, pp. 572–591. doi: 10.1016/j.cie.2016.07.013.

Zhou, L., Pan, S., Wang, J. and Vasilakos, A. V. (2017) Machine learning on big data: Opportunities and challenges, *Neurocomputing*, 237(January), pp. 350–361. doi: 10.1016/j.neucom.2017.01.026.

Zhou, P., Guo, D. and Chai, T. (2018) Data-driven predictive control of molten iron quality in blast furnace ironmaking using multi-output LS-SVR based inverse system identification, *Neurocomputing*, 308, pp. 101–110. doi: 10.1016/j.neucom.2018.04.060.

Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S. and Li, G. P. (2020) Predictive maintenance in the Industry 4.0: A systematic literature review, *Computers and Industrial Engineering*, 150(April 2019), p. 106889. doi: 10.1016/j.cie.2020.106889.

Appendixes

Appendix A

The following paragraph details the details of the selection criteria to determine the paper sample to be used in the text mining study.

To carry the study, Scopus, ScienceDirect, and IEEE were queried with the following string chain in *titles*, *abstracts*, and *keywords*: (‘machine learning’ OR ‘data analytics’ OR ‘big data analytics’ OR ‘data mining’ OR ‘artificial intelligence’ OR ‘data engineering’ OR ‘data management’) AND (‘supply chain’ OR ‘Industry 4.0’ OR ‘smart manufacturing’). Articles published between 2011 and March 2020 were considered (R1). Only papers labelled as ‘Review Articles’, ‘Research Articles’ or ‘Book Chapters’ in ScienceDirect (Rsd) were kept. Then, results are merged, and duplicates are removed, resulting in a sample containing 3858 papers.

Juan Pablo Usuga Cadavid

Contribution to the development of a methodology coupling natural language processing and machine learning to react to production disturbances

Résumé : Dans l'ère de l'industrie 4.0, exploiter les données stockées dans les systèmes d'information est un axe d'amélioration des systèmes de production. En effet, ces bases de données contiennent des informations pouvant être utilisées par des modèles d'apprentissage automatique (AA) permettant de mieux réagir aux futures perturbations de la production. Dans le cas de la maintenance, les données sont fréquemment récupérées au moyen de rapports établis par les opérateurs. Ces rapports sont souvent rédigés en utilisant des champs de saisie en textes libres avec comme résultats des données non structurées et complexes : elles contiennent des irrégularités comme des acronymes, des jargons, des fautes de frappe, etc. De plus, les données de maintenance présentent souvent des distributions statistiques asymétriques : quelques évènements arrivent plus souvent que d'autres. Ce phénomène est connu sous le nom de « déséquilibre de classes » et peut entraver l'entraînement des modèles d'AA, car ils ont tendance à mieux apprendre les évènements les plus fréquents, en ignorant les plus rares. Enfin, la mise en place de technologies de l'industrie 4.0 doit assurer que l'être humain reste inclus dans la boucle de prise de décision. Si cela n'est pas respecté, les entreprises peuvent être réticentes à adopter ces nouvelles technologies. Cette thèse se structure autour de l'objectif général d'exploiter des données de maintenance pour mieux réagir aux perturbations de la production. Afin de répondre à cet objectif, nous avons utilisé deux stratégies. D'une part, nous avons mené une revue systématique de la littérature pour identifier des tendances et des perspectives de recherche concernant l'AA appliqué à la planification et au contrôle de la production. Cette étude de la littérature nous a permis de comprendre que la maintenance prédictive peut bénéficier de données non structurées provenant des opérateurs. Leur utilisation peut contribuer à l'inclusion de l'humain dans l'application de nouvelles technologies. D'autre part, nous avons abordé certaines perspectives identifiées au moyen d'études de cas utilisant des données issues de systèmes de productions réels. Ces études de cas ont exploité des données textuelles fournies par les opérateurs qui présentaient des déséquilibres de classes. Nous avons exploré l'utilisation de techniques pour mitiger l'effet des données déséquilibrées et nous avons proposé d'utiliser une architecture récente appelée « transformer » pour le traitement automatique du langage naturel.

Mots clés : Apprentissage automatique, Traitement automatique du langage naturel, Industrie 4.0, Apprentissage profond, Maintenance.

Abstract: In the age of Industry 4.0 (I4.0), exploiting data stored in information systems offers an opportunity to improve production systems. Datasets stored in these systems may contain patterns that machine learning (ML) models can recognise to react more effectively to future production disturbances. In the case of industrial maintenance, data are frequently collected through reports provided by operators. However, such reports are often provided using free-form text fields, resulting in complex unstructured data; therefore, they may contain irregularities such as acronyms, jargon, and typos. Furthermore, maintenance data often present asymmetrical distributions, where certain events occur more frequently than others. This phenomenon is known as class imbalance, and it can hinder the training of ML models as they tend to recognise the more frequent events better, ignoring rarer incidents. Finally, when implementing I4.0 technologies, the inclusion of humans in the decision-making process must be ensured. Otherwise, companies may be reluctant to adopt new technologies. The work presented in this thesis aims to tackle the general objective of harnessing maintenance data to react more effectively to production disturbances. To achieve this, we employed two strategies. First, we performed a systematic literature review to identify the research trends and perspectives regarding the use of ML in production planning and control. This literature analysis allowed us to understand that predictive maintenance may benefit from the unstructured data provided by operators. Additionally, their usage can contribute to the inclusion of humans in the implementation of new technologies. Second, we addressed some of the identified research gaps through case studies that employed data from real production systems. These studies harnessed the free-form text data provided by operators and presented class imbalance. Hence, the proposed case studies explored techniques to mitigate the effect of imbalanced data; moreover, we also suggested the use of a recent architecture for natural language processing called transformer.

Keywords: Machine learning, Natural language processing, Industry 4.0, Deep learning, Maintenance.