PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
الجمهـــوريـــــة الجزائـريـــــة الديـمقـراطيـــــة الشـعبيـــــة
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
وزارة التعليـم العالـــي والبحـــث العلمـي
HIGHER SCHOOL OF COMPUTER SCIENCE AND DIGITAL TECHNOLOGIES-BEJAIA
المـــدرسة العـــليا في علـــوم وتكنـولوجيـات الاعـــلام الالـــي بجـــاية

Dissertation Submitted to the Department Of Computer Science in Partial Fulfillment of the Requirements for Master's Degree in Computer Science

Specialty: Artificial Intelligence and Data Sciences

Submitted By:

Asma BOUROUBA

Narima ALKAMA

Predictive Maintenance: Review of Current Methods and Techniques

Supervised by:

Dr. Meroua DAOUDI
ESTIN

Members of jury:

▪ **Pr.** Hamid KHERBACHI	President	ESTIN
▪ **Dr.** Youssef ELMIR	Examiner	ESTIN
▪ **Mr.** Amine MAMMASSE	Examiner	ESTIN
▪ **Mr.** Amine BECHAR	Examiner	ESTIN

Academic year: 2023/2024

Abstract

Achieving operational excellence is crucial for maintaining competitiveness in today's industrial landscape. Traditional maintenance strategies, both reactive and preventive, often fail to fully utilize the abundant data available. The emergence of Industry 4.0, emphasizing data acquisition and analytics, has introduced predictive maintenance, allowing for real-time insights into equipment health and proactive interventions. Business Intelligence (BI) systems play a central role in this shift by converting raw data into actionable insights, facilitating informed decision-making. This work examines the integration of Natural Language Processing (NLP), probabilistic models, and machine learning techniques in predictive maintenance, offering a comprehensive review of current methodologies. The research highlights how these advanced technologies can improve equipment reliability, reduce downtime, and optimize resource allocation, thereby enhancing production efficiency and profitability. The findings support a transition from traditional maintenance approaches to more proactive strategies, aligning with Industry 4.0 goals and fostering a data-driven, automated industrial environment.

Résumé

Assurer l'excellence opérationnelle est crucial pour rester compétitif dans le paysage industriel contemporain. Les stratégies de maintenance traditionnelles, réactives et préventives, ne parviennent souvent pas à exploiter pleinement les vastes quantités de données disponibles. L'avènement de l'Industrie 4.0, avec son accent sur l'acquisition et l'analyse de données, a introduit la maintenance prédictive, permettant des diagnostics en temps réel de l'état des équipements et des interventions proactives. Les systèmes de Business Intelligence (BI) jouent un rôle clé dans cette transformation en convertissant les données brutes en informations exploitables, facilitant ainsi la prise de décisions informées. Ce travail explore l'intégration du traitement du langage naturel (NLP), des modèles probabilistes et des techniques d'apprentissage automatique dans le cadre de la maintenance prédictive, offrant une analyse détaillée des méthodologies actuelles. La recherche démontre comment ces technologies avancées peuvent améliorer la fiabilité des équipements, réduire les temps d'arrêt et optimiser l'allocation des ressources, améliorant ainsi l'efficacité et la rentabilité de la production. Les résultats obtenus soutiennent une transition des approches de maintenance traditionnelles vers des stratégies plus proactives, alignées sur les objectifs de l'Industrie 4.0, favorisant un environnement industriel automatisé et axé sur les données.

Acknowledgements

First and foremost, we would like to express our deepest gratitude to Allah (God) for making this journey smooth and manageable. Without His blessings and guidance, none of this would have been possible.

We would also like to extend our heartfelt thanks to our family members for their unwavering support, love, and encouragement throughout this entire process. Their patience and understanding have been our greatest source of strength.

To our friends, thank you for always being there to lend a helping hand, provide a listening ear, and offer your support and motivation. Your presence has made this journey more enjoyable and less daunting.

Our deepest appreciation goes to our supervisor, Mrs. DAOUDI Meroua, for her invaluable guidance, continuous support, and insightful feedback. Her expertise and dedication have greatly contributed to the completion of this work.

Finally, we would like to thank the judges for accepting to evaluate our work. Your time, effort, and valuable input are greatly appreciated.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
BI	Business Intelligence
BIC	Bayesian Information Criterion
BIS	Business intelligence systems
BoW	Bag of Words
BN	Bayesian Network
BPTT	Backpropagation Through Time
CBoW	Continuous Bag of Words
CMMS	Computerized Maintenance Management Systems
CNN	Convolutional Neural Networks
CPD	Conditional Probability Distributions
CRM	Customer Relationship Management
DAG	Directed Acyclic Graph
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DL	Deep Learning
DSS	Decision Support System
DW	Data warehouse
EDA	Exploratory Data Analysis
EIS	Executive Information System
ERP	Enterprise Resource Planning
ETL	Extract, Transform, Load
FNN	Feedforward Neural Networks

GloVe	Global Vectors for Word Representation
GRU	Gated Recurrent Units
GUI	Graphical User Interface
HMM	Hidden Markov Model
IDF	Inverse Document Frequency
IG	Information Gain
IoT	Internet of Things
IT	Information Technology
KNN	K-Nearest Neighbors
KM	Knowledge Management
LDA	Latent Dirichlet Allocation
LDA	Linear Discriminant Analysis
LSA	Latent Semantic Analysis
LSTM	Long Short-Term Memory
MLE	Maximum Likelihood Estimation
ML	Machine Learning
NLP	Natural Language Processing
OLAP	Online analytical processing
PCA	Principal Component Analysis
PdM	Predictive Maintenance
RNN	Recurrent Neural Networks
RMSE	Root Mean Square Error
SBERT	Sentence-BERT
SCM	Social Content Management
SGD	Stochastic Gradient Descent
SQL	Structured Query Language
SVM	Support Vector Machines
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
UMAP	Uniform Manifold Approximation and Projection

General Introduction

In today's dynamic industrial landscape, maintaining a competitive edge hinges on achieving operational excellence. This necessitates effective maintenance management, a cornerstone of ensuring equipment reliability and availability while minimizing downtime and associated costs. However, traditional reactive and preventive maintenance strategies, while serving a purpose, have limitations. They often fail to fully exploit the vast potential of data readily available within industrial environments.

The rise of Industry 4.0, characterized by the integration of data acquisition and analytics, has ushered in a new era of predictive maintenance. By harnessing the power of this data, industries can gain real-time insights into equipment health, enabling proactive interventions before critical failures occur. This paradigm shift empowers companies to move from reactive repairs to preventative actions, maximizing equipment uptime, production efficiency, and overall profitability.

Central to this transformation is the implementation of Business Intelligence (BI) systems in industrial operations. BI facilitates the transformation of raw data into meaningful insights, enabling informed decision-making and strategic planning. The integration of BI tools helps industries not only to monitor and analyze performance metrics but also to predict future trends and behaviors, thus aligning maintenance activities with global business objectives. By leveraging BI, industries can enhance their operational efficiency, optimize resource allocation, and improve overall productivity.

Several challenges necessitate the focus on predictive maintenance. One of the primary issues is the high cost and inefficiency associated with unexpected equipment failures and unscheduled downtimes. Traditional maintenance approaches often result in either over-maintenance or under-maintenance, leading to wasted resources or increased risk of equipment breakdown. Predictive maintenance addresses these challenges by utilizing data-driven tech-

niques to anticipate and mitigate potential failures, ensuring timely and precise maintenance actions.

Different approaches have been explored for implementing predictive maintenance, ranging from statistical methods to advanced machine learning algorithms. Statistical approaches involve analyzing historical data to identify patterns and trends that could signal impending failures. Machine learning techniques leverage large datasets to train predictive models capable of recognizing complex failure patterns. Additionally, the use of NLP allows for the analysis of unstructured data, such as maintenance logs and operator notes, to extract valuable insights that might be missed by traditional methods.

This work explores the realm of predictive maintenance, with a particular focus on the application of NLP, probabilistic models, and machine learning techniques. We review and compare existing research studies that utilize these advanced technologies to analyze both structured and unstructured data. By examining the state-of-the-art developments in these areas, we aim to provide a comprehensive understanding of current methodologies and their effectiveness in predictive maintenance.

The remainder of this dissertation is structured as follows:

The first chapter, focuses on the field of industrial maintenance and need analysis, discussing various types of industrial prediction and tracing the evolution of industry from 1.0 to 4.0.

The second chapter, provides an overview of general concepts related to the topic, including the definition, types, and challenges of machine learning. It also introduces NLP and its techniques.

In the third chapter, we review significant related works in the area of predictive maintenance. This review is presented in a table summarizing each synthesized document, followed by a comparative analysis of these documents.

Conclusion, the dissertation concludes with a general summary and evaluation of the research conducted.

Chapter1

Definitions and Generalities

Contents

In this chapter, we explore basic concepts critical to understanding predictive maintenance in industrial settings. We begin with an overview of Business Intelligence (BI) and its role in transforming raw data into meaningful insights for decision-making. Next, we explore Computerized Maintenance Management Systems (CMMS), highlighting their importance in managing maintenance activities and tracking equipment performance. Finally, we examine Coswin 8i, a specific CMMS solution, discussing its features and capabilities in supporting predictive maintenance initiatives. This chapter sets the stage for the technical discussions in subsequent sections by establishing a clear context and framework for the study.

1.1 Business Intelligence

Business Intelligence (BI) is a set of concepts, methods, and processes that aim to improve business decisions, by using information from multiple sources in order to develop an accurate understanding of business dynamics[1]. According to [2], we can define BI as a system comprised of both technical and organizational elements that presents its users with historical information and analysis to enable effective decision making and management support, with the overall purpose of increasing organizational performance.

1.1.1 Business Intelligence Systems

Business Intelligence Systems (BIS) have emerged as a technological solution that offers data integration and key performance indicators to provide stakeholders at various organizational levels with valuable information for their decision-making [1]. The main tasks of a BI system are intelligent exploration, integration, aggregation, and multidimensional analysis of data gathered from various sources. Therefore, data is transformed from quantity to quality [2].

1.1.2 Main Objective of Business Intelligence in Companies

"*The high-level goal of BI is to help a business user turn business-related data into actionable knowledge. BI traditionally focused on reports, dashboards, and answering predefined questions*" [Beller and Barnett, 2009] [3]. Deep, exploratory, and interactive data studies utilizing Business Analytics, including data mining, predictive analytics, statistical analysis, and NLP tools, are now a major component of BI.

1.1.3 Commonly Used Technologies

BI technology has experienced rapid expansion and enhancement, enabling the resolution of increasingly complex business inquiries. From user-friendly querying tools to sophisticated OLAP and data mining instruments, a spectrum of capabilities has emerged. This technology encompasses various components such as data warehousing, Online Analytical Processing (OLAP), extraction, transformation, and loading (ETL) of data, data cleansing, information portals, data mining, business modeling, and more.

1.1.3.1 Data Warehouse

A data warehouse (DW) is a technology that supports the decision-making process and intelligent applications. These applications form the data warehouse cortex, utilizing the data warehouse as their intelligent machine [4]. Various technologies, such as Decision Support System (DSS), Executive Information System (EIS), Dashboard, BI, Knowledge Management (KM), Big Data, and more, fall under this umbrella. The goal of data warehousing is to extract, transform, and load data from different source systems into an integrated repository, the data warehouse [5].

1.1.3.1.1 Data Warehouse Schema

As mentioned in [4], three widely used data warehouse schemas are the star schema, the snowflake schema, and the fact constellation (or galaxy) schema.

- The star schema comprises a central fact table and one or more associated dimension tables, without any sub-dimension tables, as illustrated in figure 1.1.

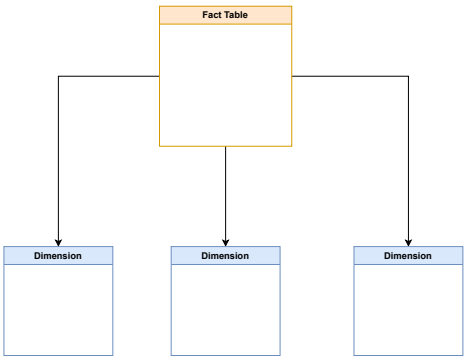

Figure 1.1: Data warehouse star schema

- The snowflake schema is a data warehouse schema that includes a single fact table and one or more dimension tables, each of which can have one or more sub-dimension tables, as shown in figure 1.2.

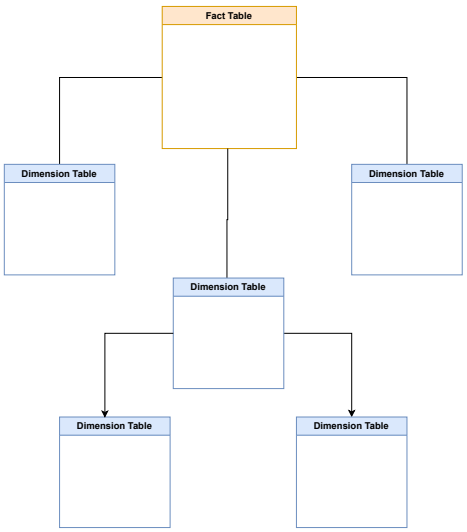

Figure 1.2: Data warehouse snowflake schema

- The fact constellation, or galaxy schema, is a data warehouse schema in which multiple fact tables share one or more dimension or sub-dimension tables, as depicted in figure 1.3.

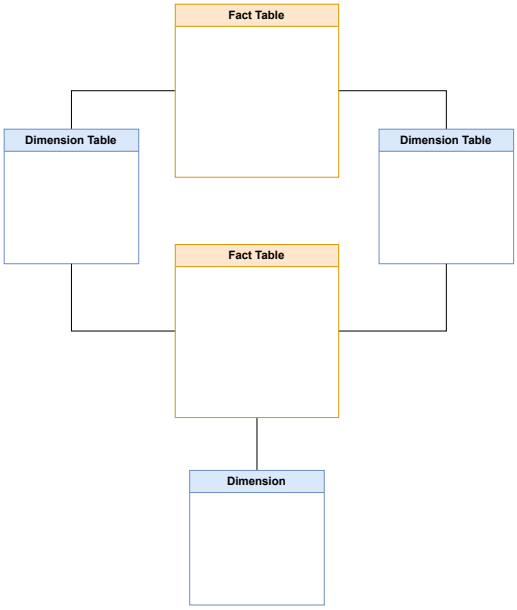

Figure 1.3: Data warehouse galaxy schema

According to Kimball et al. (2008) [6], an enterprise DW corresponds to the union of subject-oriented subsets called data marts, if the following conditions are met: each data mart must store granular data in dimensional models (i.e., with the lowest level of detail) and use conformed dimensions and facts (i.e. dimensions and facts share the same meaning across all data marts). Typically, a data mart is related to a single business process.

1.1.3.1.2 Data mart

Data marts serve as focused repositories within the data warehouse architecture. These departmentalized subsets provide pre-processed and integrated data tailored to the specific needs of a business unit or user group. Materialized or virtual in form, data marts enhance accessibility and usability for end-user analysis, leveraging traditional relational databases or OLAP tools for implementation [7].

1.1.3.2 ETL

ETL, short for Extract, Transform, Load, refers to the process of gathering data from various sources, converting it into a standardized format, and loading it into a target database or data warehouse. The ETL process involves several interconnected steps executed in sequence. the following figure 1.4 demonstrate the ETL process.

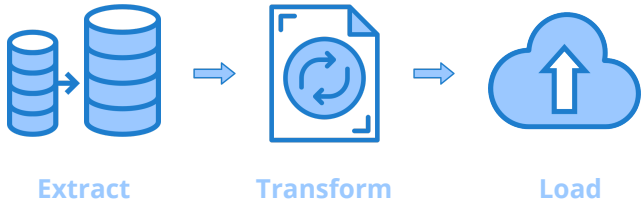

Figure 1.4: ETL process

1.1.3.2.1 Data extraction

Data extraction involves gathering useful data from multiple heterogeneous sources. The complexity of this process depends on the nature of the data sources. Various methods such as Full-extraction and Incremental-extraction are employed. Full-extraction resembles data migration or replication, whereas Incremental-extraction utilizes triggers, timestamps, or whole-table comparisons to extract only the updated data.

1.1.3.2.2 Data transformation

Data transformation is about standardizing data from diverse organizational forms and formats into a consistent format. This step involves handling redundant, ambiguous, incomplete, or non-standard data to achieve uniformity in data granularity and format. Many ETL systems employ metadata-defined rules to facilitate this transformation process.

1.1.3.2.3 Data loading

Data loading entails moving the processed data from earlier steps into the data warehouse (DW). There are two main methods used: refreshing and updating. The refreshing method is applied when initially creating a DW to populate the database with data, whereas the updating method is used for maintaining the DW. Specific techniques for loading data may include the use of specialized loading tools or SQL commands.

1.1.3.3 OLAP and data mining

OLAP, or Online Analytical Processing, is a technology used for organizing large business databases and enabling users to analyze multidimensional data interactively from multiple perspectives. OLAP allows storing data in a multidimensional way, instead of a two-dimensional structure [8] OLAP systems facilitate complex analytical and ad-hoc queries swiftly, allowing users to gain insights into trends, patterns, and relationships within their data. In the BI

process, OLAP plays a crucial role in supporting decision-making by providing users with intuitive and flexible tools for data analysis, reporting, and visualization.

Data mining is the process of efficiently discovering valuable, non-obvious information and insights from large datasets without predefined questions. Data analysts uncover hidden trends, patterns, anomalies, and correlations, which are then presented to business stakeholders with recommendations. This helps reveal overlooked opportunities for decision-making and strategic planning, such as improving efficiency or targeting marketing efforts. If no new insights are found, the process is documented and closed [8].

In the BI process, data mining complements OLAP by providing deeper insights and predictive capabilities, helping organizations anticipate future trends and behaviors based on historical data. Together, OLAP and data mining form integral components of BI systems, enabling organizations to extract maximum value from their data assets and drive informed decision-making processes.

1.1.4 The five concepts of business intelligence

1.1.4.1 Data collection and integration

In the realm of business intelligence, data collection and integration are basic processes. This involves gathering data from various sources such as internal databases like ERP, CRM, and SCM systems, as well as external APIs and spreadsheets. Integration harmonizes these disparate data sources into a unified repository, employing techniques like ETL processes to ensure consistency. Subsequently, data cleaning rectifies errors and inconsistencies, while transformation prepares the data for analysis by aggregating, deriving metrics, and standardizing formats. The end result is a dataset presented in a consistent format suitable for analysis, facilitating reliable insights and informed decision-making.

1.1.4.2 Data analysis

Data analysis is integral to uncovering insights within business data. BI professionals utilize analytical tools and techniques to explore datasets, identifying patterns, trends, anomalies, and correlations. This involves the application of statistical methods and machine learning algorithms to extract valuable insights. By leveraging these approaches, businesses gain a deeper understanding of their data, enabling informed decision-making based on data-driven insights uncovered by BI professionals.

1.1.4.3 Data visualization for easy understanding

Data visualization is a crucial aspect of data analysis, involving the transformation of complex datasets into easily understandable visual formats such as charts, graphs, and dashboards. This process plays a vital role in helping stakeholders comprehend insights quickly and make informed decisions based on the presented information. By visually representing data, businesses can effectively communicate key findings and trends, facilitating better understanding and decision-making across various levels of the organization.

1.1.4.4 Reporting and Dashboards

Reporting and dashboards are essential components of BI, facilitating the communication of key insights and metrics to stakeholders. BI professionals create regular reports and interactive dashboards that showcase important performance indicators and metrics relevant to the organization. These reports offer a snapshot of business performance, allowing users to quickly assess the state of affairs. Moreover, interactive dashboards enable users to drill down into specific details and explore data dynamically, empowering them to make informed decisions based on real-time information. By providing easy access to relevant data, reporting and dashboards play a crucial role in driving organizational transparency and efficiency.

1.1.4.5 Decision support

BI aims to support strategic decision-making by providing timely and accurate insights aligned with organizational goals. By analyzing data effectively, BI helps optimize processes, identify opportunities, and address challenges. Through the strategic use of data-driven insights, organizations can make informed decisions that drive towards their objectives, fostering efficiency and maintaining competitiveness in the market.

At the end, Bi is not a one-off process, BI is a continuous process, a continuous journey from data to information to insight to decisions to business improvements [8] as depicted in the following figure 1.5:

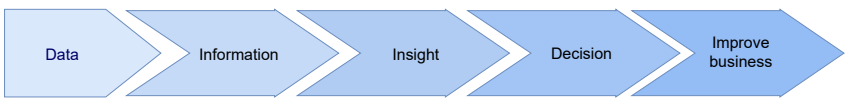

Figure 1.5: The Business Intelligence Cycle

1.1.5 Challenges of business intelligence

Research identifies several challenges in implementing Business Intelligence projects, which can be categorized into organizational and analytical challenges. Organizational challenges stem from issues related to structure, processes, administration, organizational culture, and management. On the other hand, analytical challenges are more technical in nature, involving skills and talent [9].

In the literature, there are a set of challenges that are particular to BI and keep them consistently raised as BI challenges. According to [10] these are the deepest challenges existing within BI :

1.1.5.1 Data

Masses of important information are unavailable or are unactionable ,beside the challenges that are mostly related to data and those that are related to lack of information about data (metadata) are grouped under this header. These challenges concern : Data quality issues, data governance challenges, data privacy and security, data acquisition challenges, lack of information, lack of business knowledge.

1.1.5.2 Skills

BI requires a diverse skill set from both team members and users. The BI team needs to possess more than just technical expertise; they also require business acumen and soft skills. Similarly, BI users need to be proficient in understanding their data, utilizing technology effectively, comprehending business operations, and exercising decision-making abilities.

1.1.5.3 Sponsorship

One of the challenges before the BI solution building phase which is the initiation phase, is securing support and endorsement from top management to proceed with the groundwork. This support isn't about project approval but rather about permission to develop a business case, grasp company goals, objectives, etc. Another related challenge is gaining access to the time of top management, which is typically difficult, and then persuading them in a brief period. Without support and backing from top management, middle management and others may be hesitant to cooperate and assist in the initiation phase. This is compounded by leadership's view of BI as a straightforward task, as they fail to grasp its complexities.

1.1.5.4 Alignment between BI, IT and business

Alignment between BI, IT, and business refers to the cohesive integration and synchronization of strategies, goals, and operations across these three domains [10]. It involves ensuring that BI initiatives and technologies are closely aligned with the overarching business objectives, and that IT infrastructure supports the effective deployment and utilization of BI solutions to meet business needs. Challenges in this alignment often stem from disparate priorities, communication gaps, and differing perspectives between BI, IT, and business stakeholders, which can hinder the seamless implementation and adoption of BI systems and limit their ability to deliver value to the organization.

1.1.5.5 Resistance to change

Resistance to change is a common challenge in BI initiatives, referring to the reluctance or opposition from individuals or groups within an organization to adopt new BI technologies, processes, or methodologies. This resistance can stem from various factors such as fear of the unknown, perceived threats to job roles or power dynamics, lack of understanding or training, and entrenched cultural norms. Overcoming resistance to change requires effective communication, stakeholder engagement, leadership support, and a clear articulation of the benefits and rationale behind the BI initiative to build trust, address concerns, and foster a culture of innovation and adaptability within the organization.

1.2 Industry evolution

1.2.1 Industry 1.0

Industry 1.0 marks the beginning of industrialization, where steam power and machines started replacing manual work. During this period, there were big developments in textiles, and iron and coal production also increased, along with improvements in transportation networks.

1.2.2 Industry 2.0

Industry 2.0 denotes the second phase of industrialization, featuring the emergence of electricity and further mechanization of manual tasks. This era witnessed significant growth

in the automobile and chemical industries, alongside the advent of transformative technologies like the telephone and radio.

1.2.3 Industry 3.0

Industry 3.0 signifies the third phase of industrialization, highlighted by the rise of computers and the automation of knowledge-based tasks. This period saw substantial advancements in the information technology sector, along with the expansion of the service industry. Additionally, notable innovations such as the internet and artificial intelligence emerged during this time.

1.2.4 Industry 4.0

Industry 4.0 denotes the fourth phase of industrialization, distinguished by the convergence of the physical, digital, and biological realms. This era is characterized by the rise of the Internet of Things (IoT), the growth of the biotechnology sector, and the development of advanced technologies such as quantum computing and nanotechnology.

1.3 Industrial Maintenance

Industrial maintenance plays a crucial role in manufacturing by significantly reducing machine failure time and minimizing costs, especially in the revolution of Industry 4.0 [11]. With the integration of advanced technologies like IoT sensors, predictive analytics, and automation, maintenance processes have become more proactive and efficient. Predictive maintenance allows for the early detection of potential equipment failures, enabling timely interventions to prevent costly breakdowns and unplanned downtime. This proactive approach not only enhances overall equipment effectiveness but also ensures a smoother production flow, supporting the objectives of Industry 4.0 for increased efficiency and competitiveness. Key aspects of industrial maintenance include preventive measures, predictive techniques, corrective actions, routine tasks, emergency repairs, asset management, and safety measures.

1.3.1 Industrial maintenance types

1.3.1.1 Preventive maintenance

Preventive maintenance involves setting a schedule which identifies or postpones the de-

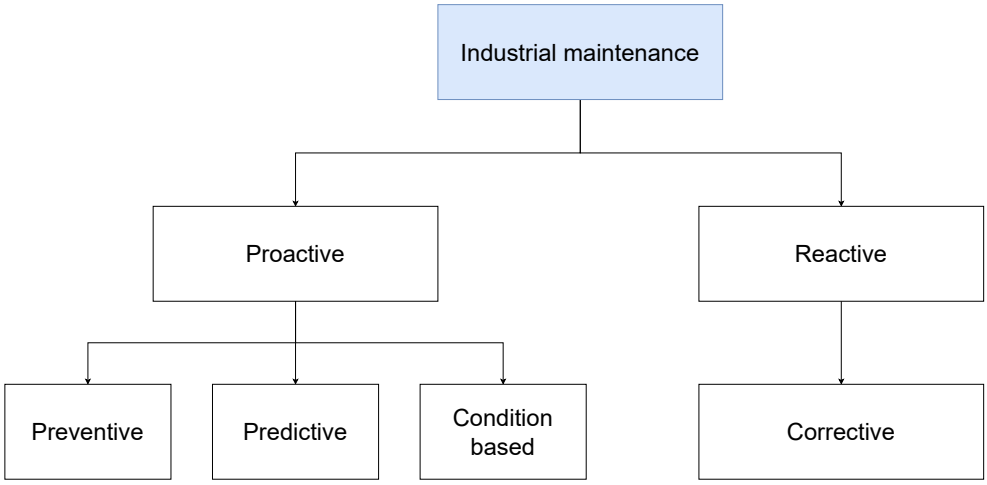

Figure 1.6: Maintenance types

cline of the system for the aim to extend equipment lifespan, and ensures the continuity of the production [12]. The challenging part about preventive maintenance is determining when to perform maintenance since the failure time is unknown, which makes early maintenance a waste of a useful machine life and cost, and late maintenance harmful especially to the safety of critical equipment [13].

1.3.1.2 Predictive maintenance

Predictive maintenance employs data analysis techniques to predict potential machine failures, enabling the issue to be addressed or maintained before significant wear occurs. What sets predictive maintenance apart from preventive maintenance is its ability to identify maintenance needs based on the current condition of the machine [14]. This technique allows organizations to optimize maintenance schedules, reduce unnecessary downtime, and allocate resources more effectively, ultimately improving overall operational efficiency and cost-effectiveness.

1.3.1.3 Corrective maintenance

Corrective maintenance is a maintenance type used to identify and correct the cause of system failures. It mainly focuses on the identification of breakdown causes from the failure phenomenon that contains one or more symptom failures [15]. To reduce production interruptions, it is imperative to promptly resolve unforeseen difficulties through corrective maintenance. However, depending only on corrective maintenance may result in more expenses because of unscheduled downtime, a larger chance of safety accidents, and possible equipment

damage. A common component of effective corrective maintenance strategies is root cause analysis, which identifies underlying issues and sets preventative measures in place to avoid recurring problems.

1.3.1.4 Condition-Based maintenance

Condition-Based maintenance (CBM) [15] is a type of preventive maintenance that is planned based on condition information. This approach relies on the monitoring of the actual condition of equipment to decide what maintenance needs to be done. It is a proactive method that involves performing maintenance actions before a failure occurs, using data related to the unit's condition, such as its age, usage, or state. Condition monitoring is essential for implementing CBM, as it involves interventions where the state of a unit is assessed, whether through continuous monitoring or periodic inspections. This allows for maintenance to be conducted based on the current condition and performance of the equipment, rather than on a fixed schedule, thereby potentially preventing failures and optimizing maintenance efforts. The following Table 1.1 explains the different between those maintenance strategies.

Table 1.1: Comparison between different maintenance types

Aspect	Preventive maintenance	Predictive maintenance	Corrective maintenance	Condition-based maintenance
Definition	Scheduled inspections, repairs, and replacements done before failures occur	Uses data analysis techniques to predict when maintenance is needed based on equipment condition	Repairs equipment after it has failed or malfunctioned	Monitors real-time condition data to schedule maintenance as needed
Approach	Proactive	Proactive	Reactive	Proactive
Timing	Scheduled intervals	Based on data-driven predictions	After equipment failure	Based on real-time equipment condition monitoring
Data utilization	Historical data and manufacturer recommendations	Real-time data analysis and monitoring	N/A (Response to failure)	Real-time condition data collected through sensors

Equipment Downtime	Minimal	Minimal	Can be significant	Variable, depends on the accuracy of condition monitoring and response time
Cost	Moderate, predictable	Moderate, variable depending on monitoring systems	Can be high, especially in terms of emergency repairs	Moderate to high, investments in sensors and monitoring systems
Resource Allocation	Planned and allocated resources based on schedule	Resources allocated based on data-driven predictions	Resources allocated in response to failure	Resources allocated based on real-time equipment condition
Benefit	Reduced downtime, prolonged equipment life, and increased safety	Minimized unplanned downtime, optimized maintenance, and cost savings	Prompt response to equipment failures and reduced disruption	Minimized downtime, optimized maintenance schedules

1.4 Computerized Maintenance Management System

Computerized Maintenance Management System (CMMS) is a software solution designed to help organizations manage and streamline maintenance operations effectively. CMMS software allows businesses to track and manage maintenance activities, schedule preventive maintenance tasks, manage work orders, track equipment inventory, and monitor maintenance costs. Implementing CMMS yields numerous advantages, including decreased costs, heightened productivity, and enhanced planning and scheduling capabilities. Nevertheless , selecting the most suitable one is not an easy task due to the large amount of CMMS available in the market [16].

1.5 Coswin 8i

Coswin 8i [17], a product by Siveco Group, is an advanced CMMS crafted to enhance the efficiency of maintenance operations across diverse industries. It encompasses a vast array of functionalities, such as asset management, maintenance scheduling, and workflow automation, all aimed at streamlining maintenance activities. The platform excels in managing inventories, handling procurement processes, and tracking work orders, thus covering every essential aspect of maintenance management. Moreover, Coswin 8i aids in regulatory compliance and provides sophisticated reporting tools, which are crucial for data-driven decision-making and strategic planning. Its integration capabilities with other enterprise systems, coupled with the use of mobile technologies, ensure real-time data accessibility and operational flexibility, positioning Coswin 8i as a comprehensive solution for contemporary maintenance management challenges.

1.6 Conclusion

In this introductory chapter, we have introduced fundamental concepts essential to our exploration of predictive maintenance in industrial contexts. By defining BI and briefly discussing CMMS, including a cursory overview of Coswin 8i, we have established a solid foundation. As we transition to the next chapter, focusing on Machine Learning and Natural Language Processing, we are primed to delve deeper into the technical aspects of predictive maintenance. This chapter sets the stage for our academic journey, providing a clear direction for further analysis and discussion.

Chapter2

Machine Learning and Natural Language Processing Overview

Contents

Machine Learning and Natural Language Processing are two interdisciplinary fields that have revolutionized various industries by enabling automated decision-making and the analysis of unstructured data. This second chapter provides an overview of the fundamentals of machine learning and Natural Language Processing, exploring their definitions, applications, and associated challenges.

The first part of this chapter delves into the basics of machine learning. Beginning with a definition of machine learning, we discuss its significance, particularly in the context of predictive maintenance, where it aids in anticipating equipment failures and optimizing maintenance schedules. We then explore the different types of machine learning, including supervised

learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Furthermore, we examine the machine learning pipeline and introduce probabilistic graphical models, such as Bayesian networks and Markov models.

The second part of the chapter focuses on the fundamentals of NLP. We define NLP and explore the challenges associated with processing and understanding human language, including issues of ambiguity and variability. Specifically, we delve into text mining and discuss various NLP techniques employed for tasks such as text preprocessing, feature extraction, and topic modeling.

Part one: Machine Learning

2.1 Definition of Machine Learning

Machine Learning (ML) is the most prevalent subfield of artificial intelligence (AI) that involves developing self-learning algorithms to gain insights from data to make predictions. Unlike traditional methods that rely on manual rule derivation and model building by humans through extensive data analysis, ML provides a more streamlined approach. It enables the extraction of knowledge from data to enhance predictive model performance over time, facilitating informed, data-driven decision-making [18].

Here is a slightly more general definition:

According to Arthur Samuel (1959) [19], Machine Learning is a "*field of study that gives computers the ability to learn without being explicitly programmed.*"

And a more engineering-oriented one:

Tom Mitchell (1997) provides the definition [19], "*A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.*"

2.2 Need of Machine Learning in predictive maintenance

Predictive maintenance, a cornerstone of modern industrial operations, hinges on the proactive identification and mitigation of equipment failures before they occur. In this context,

(ML emerges as an indispensable tool, offering a nuanced approach to prognostication and risk mitigation within industrial settings [20].

1. **Data-Driven Prognostication:** ML algorithms anticipate equipment failures by analyzing historical maintenance records, sensor data, and operational parameters, identifying subtle indicators of malfunctions. This early detection helps prevent costly downtimes and optimizes asset utilization.

2. **Dynamic Adaptation to Operational Variability:** ML techniques adapt to the dynamic nature of industrial environments by continuously refining predictive models with real-time data. This iterative learning ensures robust and effective predictive maintenance across varying operational conditions.

3. **Optimization of Maintenance Strategies:** Unlike traditional reactive maintenance, ML-driven predictive maintenance shifts to proactive risk mitigation. By forecasting equipment degradation and prioritizing maintenance activities, ML optimizes maintenance schedules, reduces operational disruptions, and enhances asset reliability.

4. **Precision in Fault Diagnosis and Root Cause Analysis:** ML algorithms excel in diagnosing faults and identifying root causes through pattern recognition and anomaly detection. This precision allows engineers to implement targeted remedial measures, minimizing downtime and extending asset lifespan.

5. **Continuous Improvement through Feedback Loops:** ML-driven predictive maintenance systems improve continuously through feedback loops. By leveraging historical performance data and maintenance outcomes, these systems enhance the accuracy and reliability of predictive models, achieving greater precision and effectiveness over time.

2.3 Types of Machine Learning

Machine Learning systems can be classified according to the amount and type of supervision they get during training. There are four major categories: supervised learning, unsupervised learning, semi supervised learning, and Reinforcement Learning [19].

We will learn about the fundamental differences between the three different learning types and, using conceptual examples, we will develop an intuition for the practical problem domains where these can be applied [18]. As described in figure 2.1.

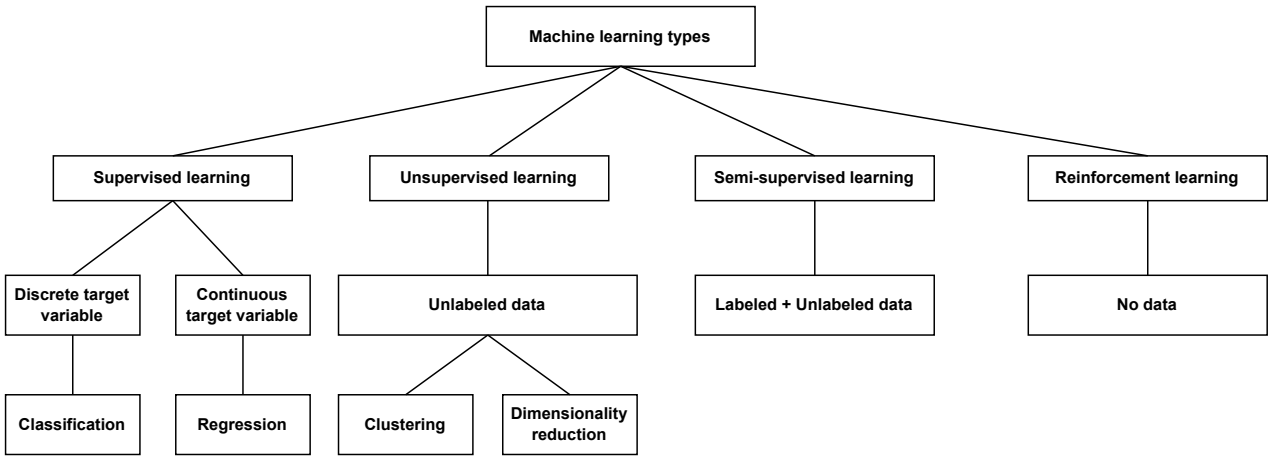

Figure 2.1: Types of Machine Learning

2.3.1 Supervised Learning

Supervised learning aims to learn a function that maps an input vector, denoted by X, to a corresponding output vector, denoted by Y. This output vector contains labels or tags that provide the desired explanation for each input example in X. Together, X and its corresponding label in Y form a training example. In essence, the training data consists of a collection of such training examples [21].

The objective is then to learn a model from labeled training data that allows us to make predictions about unseen or future data as illustrated in the figure 2.2 [18].

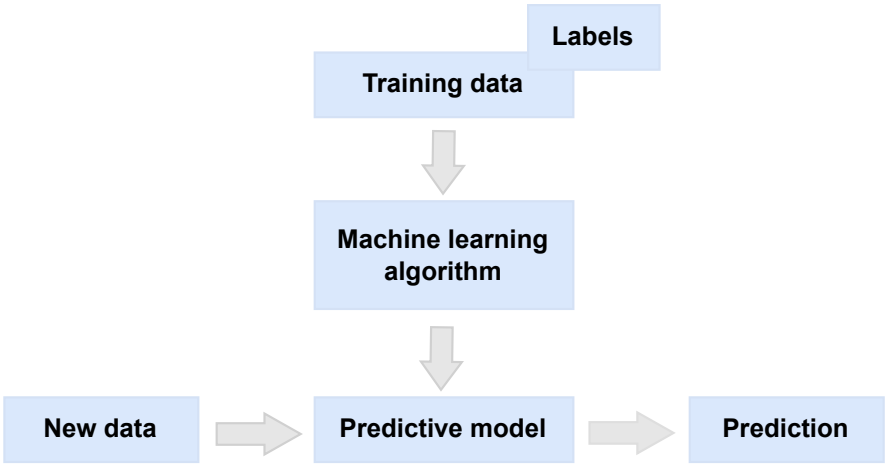

Figure 2.2: Supervised Learning workflow

Supervised learning can be broadly categorized into two main types based on the nature of the output variable: classification and regression.

2.3.1.1 Classification

Within supervised learning, classification tasks focus on predicting categorical class labels of new instances based on past observations. These labels represent discrete, unordered values that denote the group memberships of the instances, and they can include more than just binary categories. The predictive model learned by a supervised learning algorithm assigns any class label that was presented in the training dataset to a new, unlabeled instance [18].

For example, Figure 2.3 demonstrates a binary classification task with 20 training samples: 10 labeled as the negative class (circles) and 10 as the positive class (plus signs). Each sample is characterized by two values, x_1 and x_2. A supervised learning algorithm learns a decision boundary, depicted as a black dashed line, to separate these two classes and classify new data points based on their x_1 and $x_2$values.[18].

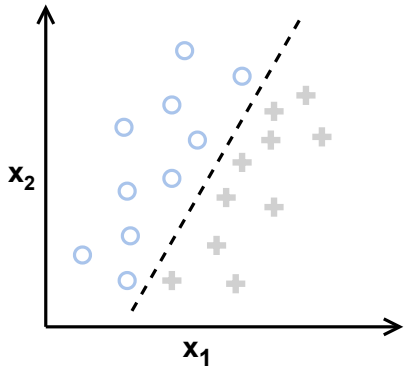

Figure 2.3: Binary classification

2.3.1.1.1 Support Vector Machines

Support Vector Machines (SVMs) are supervised learning models designed for classification and regression tasks. Their primary goal is to maximize the margin, which is the distance between the decision boundary (separating hyperplane) and the nearest training samples, called support vectors. This concept is illustrated in figure 2.4.
An SVM is defined by a linearly discriminant function of the form:

$$S(x) = w^T x + b$$

where **x** is the feature vector, **w** is the weight vector, and b is the bias term. The weight vector **w** determines the direction of the hyperplane, while the bias b adjusts its position.

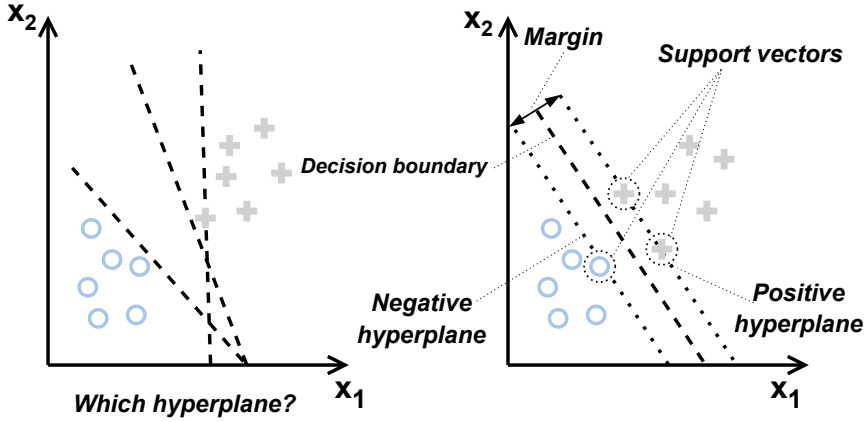

Figure 2.4: Maximum margin classification with support vector machines

Given a feature vector $\mathbf{x}$, the function $S(\mathbf{x})$ satisfies the following conditions:

1. $S(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b > 0$ if $\mathbf{x}$ is an instance in class C_1.

2. $S(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b < 0$ if $\mathbf{x}$ is an instance in class C_2.

During training, the weight vector $\mathbf{w}$ and the bias b are adjusted so that all instances of C_1 and C_2 lie on opposite sides of the hyperplane. The optimal hyperplane maximizes the margin, defined as $\frac{2}{\|\mathbf{w}\|}$, ensuring it is equidistant from the support vectors of both classes. The optimization problem is formulated as:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \forall i$$

where y_i is the class label of the i-th training sample (+1 for C_1 and -1 for C_2) and $\mathbf{x}_i$ is the corresponding feature vector [18, 21].

2.3.1.1.2 Logistic Regression

Logistic regression stands as a pivotal probabilistic-based statistical model extensively employed for addressing classification challenges within the domain of ML. Logistic regression usually uses the sigmoid function, to estimate probabilities, as defined by the equation below. It is particularly effective in scenarios where datasets have high dimensionality and can be linearly separated [19, 22].

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

2.3.1.1.3 Decision trees

Decision trees are non-parametric, supervised learning methods that provide a clear and understandable method for making predictions. The structure of a decision tree resembles a flowchart, where data is split based on certain criteria, leading to decisions that classify the data into distinct classes.

A decision tree is composed of internal decision nodes and terminal leaves. Each decision node represents a test on an attribute, with each branch representing the test's outcome. Beginning at the root node, the data is recursively split based on the feature that maximizes IG. This process continues until the leaves are pure, meaning all samples at a leaf node belong to the same class, or until no further splits can improve the classification. An example of a decision tree structure is given in the figure 2.5.

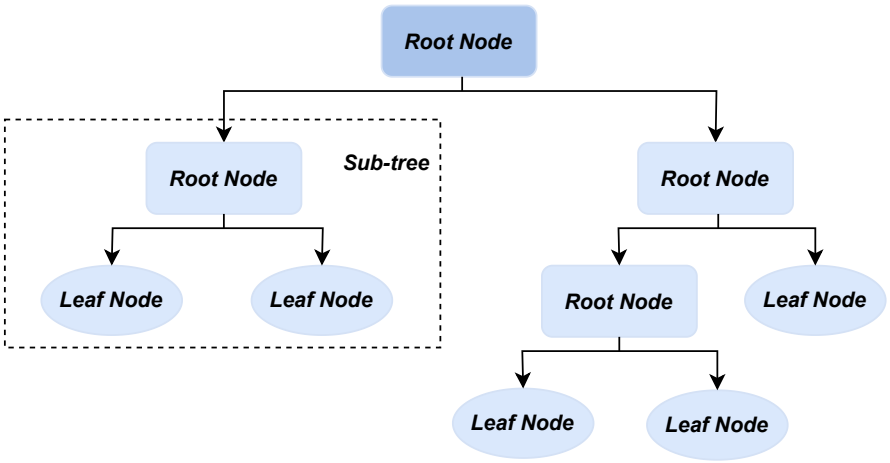

Figure 2.5: An example of a decision tree structure

To determine the most informative feature for splitting the data, decision trees use information gain, which is based on the concept of entropy from information theory. Entropy measures the uncertainty in a random variable, quantifying the expected value of the information contained in a message. The entropy $H(x)$ of a dataset is defined as:

$$H(x) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i)$$

Where $P(x_i)$ is the probability of class x_i.

Information gain is the reduction in entropy achieved by partitioning the dataset according to a given attribute. By choosing the attribute with the highest information gain, the decision tree effectively reduces uncertainty and enhances the purity of the nodes [18, 22].

2.3.1.1.4 Random Forests

Random Forest is an ensemble learning method used for both classification and regression tasks. It works by constructing multiple decision trees during training and outputs the mode of the classes for classification or the mean prediction for regression based on the individual trees. The algorithm involves two primary stages:

1. Creating the random forest by generating a collection of decision trees using a bagging approach, where multiple subsets of the original training data are created by through random sampling with replacement, and each subset is used to train a separate decision tree;

2. Making predictions by aggregating the outputs of all individual trees, with classification predictions determined by majority voting and regression predictions by averaging.

Employing parallel ensembling, Random Forest trains multiple decision trees independently in parallel on different data subsets, which minimizes overfitting and enhances the model's generalization ability [22, 23, 24].

2.3.1.1.5 K-Nearest Neighbors

The K-Nearest Neighbors (KNN) algorithm is a fundamental instance-based learning technique, often referred to as a lazy learning algorithm. Unlike traditional models that build an explicit internal model during training, KNN stores all training instances in an n-dimensional feature space and defers computation until a query is made. The algorithm operates on the principle of proximity to classify new data points. The steps involved in KNN are:

1. Select the number of neighbors k and the distance metric (commonly Euclidean distance).

2. Identify the k nearest neighbors of the sample to be classified.

3. Assign the class label based on a majority vote among the k nearest neighbors.

This process is illustrated by a scenario where a new data point is classified based on its proximity to its five nearest neighbors, as shown in the figure 2.6.

KNN is a versatile algorithm used for both classification and regression tasks. In classification, it assigns the class based on the majority vote of the k-nearest neighbors, while in regression, it predicts a value by averaging the values of the k-nearest neighbors [18, 21, 22].

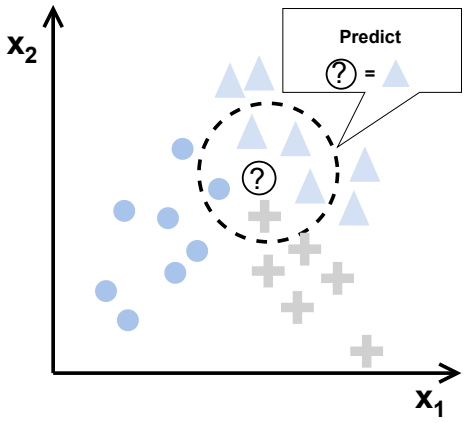

Figure 2.6: An example of a KNN scenario

2.3.1.1.6 Naive Bayes

Naive Bayes (NB) is a probabilistic algorithm grounded in Bayes' theorem, assuming independence between pairs of features. It is effective for both binary and multi-class classification tasks and is widely used in real-world applications like document classification and spam filtering. The algorithm utilizes Bayes' theorem to calculate the posterior probability of an event A given a prior probability of event B, expressed as:

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)}$$

where $P(A)$ and $P(B)$ are the probabilities of observing events A and B independently, $P(A|B)$ is the conditional probability of A given B, and $P(B|A)$ is the conditional probability of B given A [21, 24].

2.3.1.1.7 Neural Networks

Neural networks (NNs) are a basic technology in the field of AI, designed to mimic the way the human brain processes information. Comprising layers of interconnected nodes, or neurons, these networks can learn and make decisions by identifying patterns in data. By training on large datasets, neural networks can improve their performance, making them highly effective for applications that require adaptive learning and high accuracy. Key types of neural networks include Feedforward Neural Network (FNN), the simplest form where data moves in one direction from input to output; Convolutional Neural Network (CNN), specialized for processing grid-like data such as images using convolutional layers; RNN, designed for sequential data like time series or natural language, with connections forming cycles to retain information across steps; and advanced types of Recurrent Neural Networks (RNNs) like

Long Short-Term Memory (LSTM) and Gated Reccurent Units (GRU), which address the limitations of traditional RNNs by better handling long-term dependencies [21, 22].

2.3.1.2 Regression

Another type of supervised learning is the prediction of continuous outcomes using predictor variables, known as regression analysis. The goal here is to find a relationship between those variables that allows us to predict an outcome [18, 22].

The figure 2.7 exemplifies the concept of linear regression, a common regression technique which involves modeling the relationship between a single predictor variable X and a continuous response variable Y with a straight line. This line is defined by the equation:

$$Y = a + bX + e$$

Where a is the intercept, b the slope, and e the error term. The goal is to find the optimal a and b that minimize the prediction error.

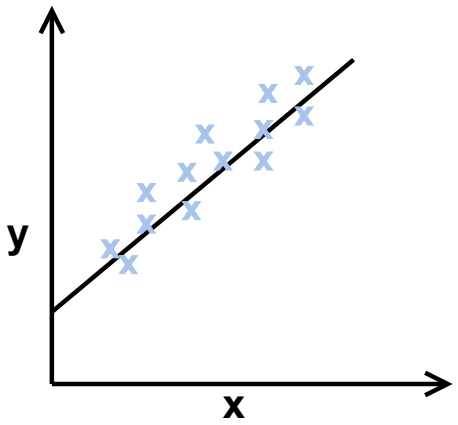

Figure 2.7: Linear regression

Multiple linear regression extends this concept to multiple predictors, forming an equation:

$$Y = a + b_1X_1 + b_2X_2 + \ldots + b_nX_n + e$$

For capturing non-linear relationships and modeling more complex patterns, polynomial regression is used, where the relationship is expressed as:

$$Y = b_0 + b_1X + b_2X^2 + \ldots + b_nX^n + e$$

2.3.2 Unsupervised Learning

In supervised learning, the aim is to learn a mapping from the input to an output whose correct values are provided by a supervisor. In contrast, unsupervised learning lacks a supervising signal, and we only have input data. The aim is to find the regularities in the input and group the observations into different groups in such a way that the data of each subset share common characteristics [18].

The most common unsupervised learning tasks are clustering and dimensionality reduction.

2.3.2.1 Clustering

Clustering is an exploratory data analysis technique that allows us to organize a pile of information into meaningful subgroups (clusters) without having any prior knowledge of their group memberships. Each cluster that may arise during the analysis defines a group of objects that share a certain degree of similarity but are more dissimilar to objects in other clusters, which is why clustering is also sometimes called "unsupervised classification" [18].

The figure 2.8 illustrates how clustering can be applied to organizing unlabeled data into three distinct groups based on the similarity of their features x_1 and x_2.

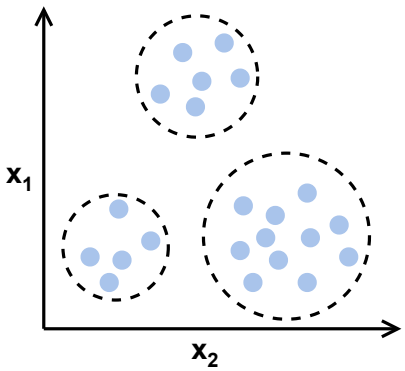

Figure 2.8: Clustering

2.3.2.1.1 Hierarchical Clustering

Hierarchical clustering aims to partition a given set of observations into clusters, with the number of clusters typically unknown. Starting with each data point in its own cluster, the algorithm iteratively merges or splits clusters based on a distance metric, resulting in a dendrogram—a tree-like diagram that reveals the structure and relationships within the

data. This method can be agglomerative, merging clusters from the bottom up, or divisive, splitting clusters from the top down, with agglomerative methods being more commonly used. Determining the optimal number of clusters is challenging and often involves analyzing similarity values and conducting parametric studies [22, 23, 25].

2.3.2.1.2 K-means Clustering

The k-means clustering algorithm is a widely utilized method in unsupervised learning, particularly for partitioning data into clusters of similar variance. Its primary objective is to minimize inertia, also referred to as within-cluster sum-of-squares, by assigning data points to centroids iteratively. A drawback of this method, common to centroid-based clustering, is the necessity to predefine the number of clusters (k). Despite this limitation, its popularity stems from its simplicity and scalability with large datasets.

Formally, the algorithm segregates a dataset X with N data points into K disjoint clusters C_k, each described by cluster centroids μ_k, not restricted to being actual data points. The objective is to minimize inertia, represented mathematically as:

$$\min_{\mathbf{C},\mu} \sum_{k=1}^{K} \sum_{x \in C_k} \|x - \mu_k\|^2$$

The algorithm is composed of the following steps as illustrated in the figure 2.9 [25]:

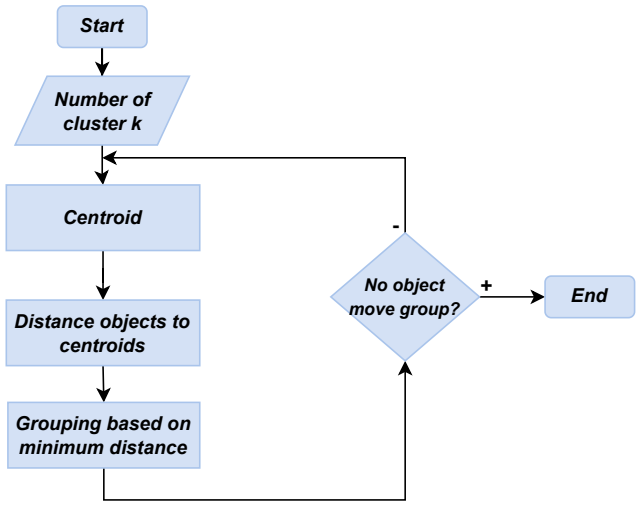

Figure 2.9: K-means algorithm

1. Randomly select k initial centroids from the dataset.

2. Assign each data point to the cluster with the nearest centroid.

3. Recalculate the centroids of the clusters based on the mean of the data points assigned to each cluster.

4. Repeat steps 2 and 3 until the centroids no longer change significantly or until the maximum number of iterations is reached.

2.3.2.1.3 Density-Based Spatial Clustering of Applications with Noise

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a widely-used algorithm in data mining and Machine Learning for density-based clustering. Unlike k-means, DBSCAN does not require the number of clusters to be specified in advance and can identify clusters of arbitrary shapes and sizes. It defines clusters as continuous regions of high density, effectively identifying both dense regions and outliers in noisy datasets.

DBSCAN operates by identifying the number of instances within a small distance ε of each instance to define its neighborhood. If an instance has at least *min_ samples* instances (including itself) in its ε-neighborhood, it is marked as a core instance, indicating a dense region. All instances within a core instance's neighborhood are assigned to the same cluster, potentially creating long sequences of connected core instances. Instances that are not core instances and lack a core instance in their neighborhood are classified as anomalies [18, 19].

2.3.2.2 Dimensionality Reduction

Another subfield of unsupervised learning is dimensionality reduction, particularly useful for managing high-dimensional data that challenges storage and computational resources. This technique aids in feature preprocessing by removing noise that can degrade predictive performance and compressing data onto a smaller subspace while retaining most relevant information. It also facilitates data visualization, allowing high-dimensional datasets to be projected onto lower-dimensional spaces for easier interpretation via scatter plots or histograms.

Principal Component Analysis (PCA) is the most popular dimensionality reduction algorithm. PCA simplifies datasets by transforming them into a lower-dimensional space, retaining as much original information as possible. It identifies the hyperplane closest to the data and projects the data onto it, aligning with the directions of maximum variance known as principal components. This method helps preserve significant information while reducing dimensional complexity, making it a valuable preprocessing step for Machine Learning algorithms.

PCA involves standardizing the data, computing the covariance matrix, finding eigenvectors and eigenvalues, sorting them, selecting the top k eigenvectors, and projecting the data onto the new coordinate system. This process results in a lower-dimensional representation that facilitates further data analysis and feature extraction, enhancing the efficacy of Machine Learning models and simplifying data interpretation [18, 21].

2.3.3 Semi Supervised Learning

Semi-supervised learning is a Machine Learning approach that falls between supervised and unsupervised learning. It involves training an algorithm on a dataset that contains both labeled and unlabeled data. The goal is to improve learning accuracy by leveraging the vast amounts of unlabeled data along with the relatively smaller amount of labeled data [19, 21]. This concept is depicted in the figure 2.10

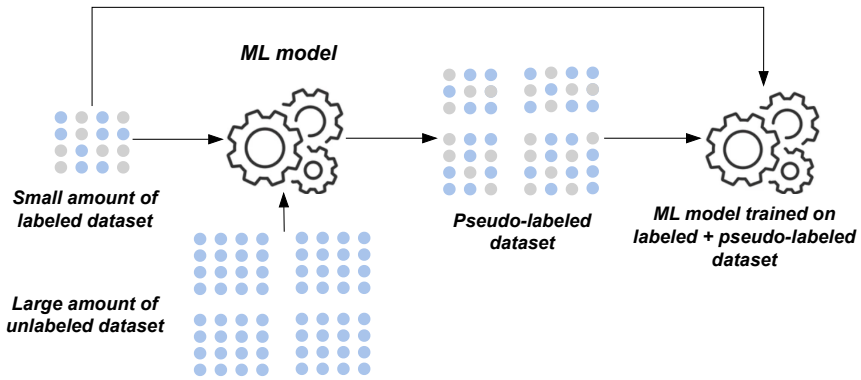

Figure 2.10: Semi-supervised learning

2.3.4 Reinforcement Learning

Reinforcement Learning (RL) is a very different beast. The learning system, called an agent in this context, interacts with its environment, selecting actions and receiving rewards in return (see figure 2.11). The agent's objective is to autonomously learn the optimal strategy, termed a policy, to maximize cumulative rewards over time. Unlike supervised learning, where correct labels are provided, reinforcement learning relies on a feedback mechanism based on rewards, representing how well actions align with a predefined reward function. This iterative process allows the agent to learn a sequence of actions that lead to maximum reward accumulation, either through trial-and-error exploration or deliberative planning [18, 19].

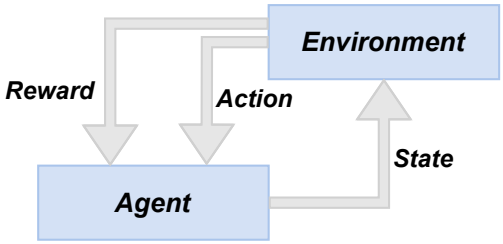

Figure 2.11: Reinforcement Learning

2.4 Machine Learning pipeline

In this section, we will discuss other important parts of a Machine Learning system accompanying the learning algorithm. The diagram below in the figure 2.12 shows a typical workflow diagram for using ML in predictive modeling, which we will discuss in the following subsections [18].

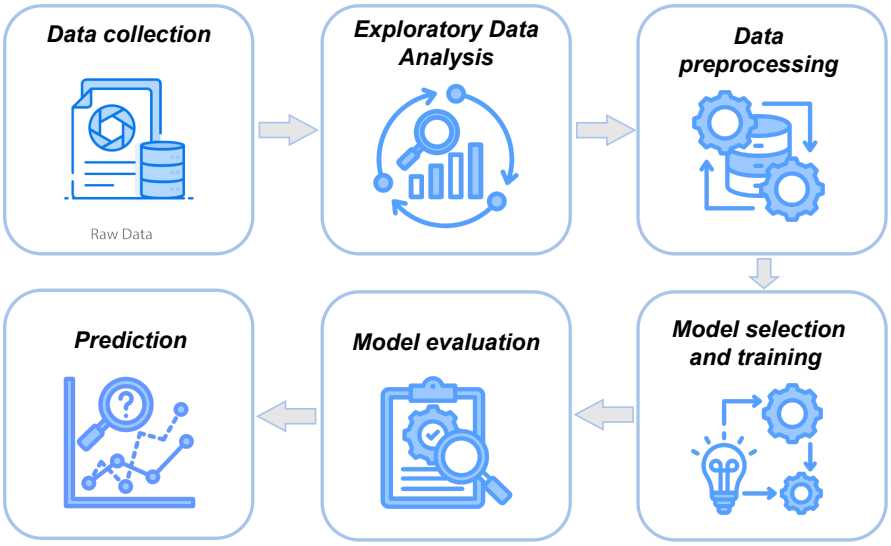

Figure 2.12: Machine Learning pipeline

2.4.1 Data collection

Data collection is the foundational step in any ML pipeline, involving the gathering of raw data from various sources to serve as the input for subsequent analysis and model building. This data can come from numerous origins, including databases, Application Programming Interfaces (APIs), web scraping, sensor readings, and more. Effective data collection requires careful planning to ensure that the data is both relevant and of high quality, addressing the specific needs of the problem at hand [20].

2.4.2 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is the process of understanding, interpreting, and analyzing the data. There are various ways to understand the data, it may be statistical, graphical, etc. In order to establish a foundation for further analysis, EDA seeks to spot patterns, anomalies, relationships, and other aspects of the data [20].

2.4.3 Data preprocessing

Data preprocessing involves preparing raw data for analysis and modeling by cleaning, transforming, and organizing it into an appropriate format, as shown in Figure 2.13. This step is crucial in any machine learning application because the accuracy and quality of the results are highly dependent on the quality of the data used. To ensure high standards of data quality and result accuracy, data preprocessing is repeated multiple times as new data becomes available [20].

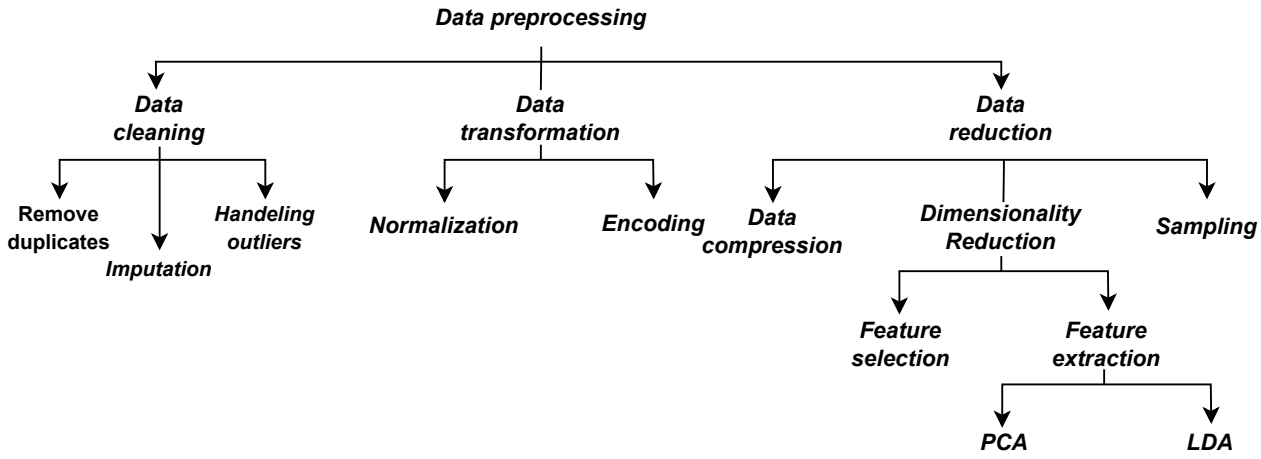

Figure 2.13: Data preprocessing

2.4.3.1 Data cleaning

Data cleaning, also known as data cleansing or data scrubbing, is the critical process of inspecting and correcting errors in data to ensure its reliability and accuracy. Similar to verifying facts before making important decisions, data cleaning involves tasks such as removing duplicates, filling in missing information, correcting errors, and handling outliers to maintain consistent data formats. These steps are vital in the data science process, as even minor inconsistencies can significantly impact the accuracy of statistical models and analysis

results. Consequently, data cleaning is an indispensable step in preparing data for meaningful analysis, ensuring that outcomes are precise and reliable [20].

2.4.3.2 Data transformation

Data transformation is a critical process that involves reorganizing or altering the format of data to improve its accessibility and comprehensibility, making it suitable for informed decision-making. Common tasks in data transformation include converting text categories into numbers, scaling data uniformly, combining multiple data pieces into one, and restructuring data for easier manipulation.

Normalization is a key aspect of data transformation, involving techniques such as min-max normalization, which scales data to a range of 0 to 1, making it suitable for certain ML algorithms. Z-score normalization standardizes data to a mean of zero and a standard deviation of one, reducing sensitivity to outliers and ensuring compatibility with algorithms that require normally distributed data. Decimal scaling normalization adjusts data so that the largest absolute value equals 1, simplifying the scaling process.

Encoding is another essential task, converting categorical data into numerical formats that machines can understand. Techniques like one-hot encoding, ordinal encoding, and label encoding help in transforming categories into unique numerical values, facilitating their use in ML models. The selection of an encoding technique depends on the nature of the data and the requirements of the specific algorithm [18, 20].

2.4.3.3 Data reduction

Data reduction is a technique employed to make large datasets more manageable while retaining essential information and eliminating superfluous data. This process includes various methods [20, 26]:

- **Dimensionality reduction:** This method simplifies data by reducing the number of features or variables, which is particularly useful for high-dimensional datasets. Techniques include feature selection, which retains the most relevant variables, and feature extraction, which creates new variables that capture essential information. The best known and most widely used feature extraction methods are PCA and LDA, which are both linear projection methods, unsupervised and supervised respectively.

- **Sampling:** Instead of using the entire dataset, a representative subset is chosen, either randomly or systematically. This reduces the data size, making it easier to handle and analyze while ensuring critical information is maintained.

- **Data compression:** This method decreases data size by removing redundant information. Lossless compression retains all original data, while lossy compression removes less critical information, beneficial for large datasets where some data loss is acceptable.

The choice of data reduction technique depends on the nature of the data and the research objectives. Dimensionality reduction is useful for simplifying complex datasets, sampling reduces the dataset size for quicker analysis, and data compression minimizes storage needs. Properly implemented, data reduction ensures efficient data handling without compromising the integrity of the analysis.

To determine whether our ML algorithm not only performs well on the training set but also generalizes well to new data, we also want to randomly divide the dataset into a separate training and test set. We use the training set to train and optimize our ML model, while we keep the test set until the very end to evaluate the final model.

2.4.4 Selecting and training a predictive model

Selecting and training a predictive model is crucial in the machine learning workflow, requiring careful consideration of various algorithms and hyperparameters. Each ML algorithm has inherent biases, and no single model is universally superior without assumptions about the task. Thus, it is essential to compare multiple algorithms to identify the best-performing model [26].

2.4.4.1 Choosing performance metrics

Before comparing models, selecting a performance metric is critical. For classification tasks, a common metric is classification accuracy, defined as the proportion of correctly clas sified instances. Other metrics, such as precision, recall, F1 score, and ROC-AUC, may also be relevant depending on the problem specifics. Similarly, for regression tasks, metrics such as mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and R-squared (R2) score are used to assess predictive performance. Meanwhile, in clustering tasks, metrics like silhouette score and Davies-Bouldin index provide insights into the quality of clustering results, evaluating how well data points are grouped together.

2.4.4.2 Model selection techniques

To estimate a model's performance on unseen data without using the test set during model selection, cross-validation techniques are employed. Cross-validation involves partitioning the training dataset into training and validation subsets, enabling the estimation of the model's generalization performance. Common methods include k-fold cross-validation, which divides the dataset into k subsets and trains the model k times, each time using a different subset as the validation set.

2.4.4.3 Hyperparameter optimization

Model performance can be significantly improved by fine-tuning hyperparameters—parameters not learned from the data but set prior to training. Techniques like grid search, random search, and Bayesian optimization help identify the optimal hyperparameters. Hyperparameter tuning is essential because the default parameters provided by software libraries may not be optimal for specific tasks.

2.4.5 Evaluating models and predicting unseen data instances

After we have selected a model that has been fitted on the training dataset, we can use the test dataset to estimate how well it performs on this unseen data to estimate the generalization error. If we are satisfied with its performance, we can now use this model to predict new, future data. It is important to note that the parameters for the previously mentioned procedures—such as feature scaling and dimensionality reduction—are solely obtained from the training dataset, and the same parameters are later re-applied to transform the test dataset, as well as any new data samples—the performance measured on the test data may be over optimistic otherwise [18].

2.5 Deep Learning

Deep Learning, an advanced subset of Machine Learning,has emerged as a powerful paradigm for processing text, image, and speech data. It primarily relies on artificial neural networks, whose training has been greatly facilitated by the abundant availability of data and computational resources. The term "deep" in Deep Learning refers to the multiple layers within neural network architectures, enabling complex learning processes directly through the

network itself. Among the prominent algorithms within Deep Learning are Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), which serve as foundational frameworks for various applications [27].

2.5.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs or ConvNets) have gained widespread acclaim not only in computer vision for image classification tasks but also in text classification. As a cornerstone of deep learning, CNNs are meticulously crafted to decipher the spatial arrangements of pixels in an input image or textual data through layers of feature detectors. The fundamental concept revolves around mastering convolution kernels to seamlessly transform data, ensuring invariance to translation. Moreover, the depth of features dynamically adjusts as more filters are applied across successive layers [18, 28].

2.5.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are essentially feedforward neural networks with feedback loops, allowing them to propagate information through time. In RNNs, neurons activate for a limited period before temporarily deactivating, subsequently activating other neurons at later time points. Despite their success in tasks like speech recognition and language translation, RNNs encounter challenges during training due to the temporal nature of sequential data. To address this issue, LSTM networks were introduced. LSTMs enhance RNNs by incorporating memory states for each layer, enabling the retention and utilization of information from previous inputs over extended sequences. This advancement resolves gradient issues, improving the network's capability to handle sequential data effectively [18, 27, 29].

2.6 Probabilistic Graphical Models

Probabilistic graphical models are a rich framework for encoding probability distributions over complex domains. These models provide a way to visualize the structure of a probabilistic model and to design and build complex models in a modular fashion. Two prominent types of probabilistic graphical models are Bayesian Networks and Markov Models.

2.6.1 Bayesian network

Bayesian Networks (BNs), also known as Belief Networks or Bayes Nets, are probabilistic graphical models used to represent knowledge about an uncertain domain. They provide a structured way to model the conditional dependencies among a set of random variables. In a Bayesian Network, each node represents a random variable, and each directed edge (arc) between nodes represents a conditional dependency between the connected variables [23, 25, 26].

2.6.1.1 Structure of a Bayesian Network

A Bayesian Network is composed of nodes and directed arcs, where nodes correspond to random variables X, each associated with a probability $P(X)$. If there is a directed arc from node X_i to node X_j, it indicates that X_i has a direct influence on X_j. This influence is quantified by the conditional probability $P(X_j \mid X_i)$. The network forms a Directed Acyclic Graph (DAG), meaning there are no cycles in the graph. The structure of the network is defined by the nodes and the directed edges, while the parameters of the network are the conditional probability distributions (CPDs) associated with these edges. The joint probability distribution over all variables $X_1, X_2, \ldots, X_n$ can be factored using the chain rule of probability, taking into account the conditional dependencies:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid \text{Pa}(X_i))$$

where $\text{Pa}(X_i)$ denotes the set of parent nodes of X_i.

2.6.1.2 Inference in Bayesian Networks

Inference involves computing the probability of certain variables given known values of other variables. For a set of variables X with evidence E, the posterior probability is given by:

$$P(X \mid E) = \frac{P(X, E)}{P(E)}$$

This can be achieved using techniques such as:

- **Variable Elimination:** This method involves summing out the irrelevant variables from the joint distribution:

$$P(X_i \mid E) = \sum_{Y} \prod_{j=1}^{n} P(X_j \mid \text{Pa}(X_j))$$

where Y is the set of non-evidence and non-query variables.

- **Belief Propagation:** Used for exact inference in tree-structured networks, belief propagation involves passing messages between nodes to update beliefs about variable distributions.

2.6.1.3 Learning in Bayesian Networks

2.6.1.3.1 Parameter learning

Parameter learning in Bayesian Networks involves estimating the conditional probability distributions from data. Parameters can be learned using Maximum Likelihood Estimation (MLE) or Bayesian estimation.

Maximum Likelihood Estimation (MLE) is a statistical method used to estimate the parameters of a model, in this case, a Bayesian Network. The goal of MLE is to find the set of parameters that maximize the likelihood of the observed data given the model.

Given a dataset D with N instances, the MLE for the parameters of a conditional probability distribution $P(X_i \mid \text{Pa}(X_i))$ is:

$$\hat{\theta} = \arg\max_{\theta} \prod_{i=1}^{N} P(x_i \mid \text{Pa}(x_i); \theta)$$

2.6.1.3.2 Structure learning

Structure learning involves discovering the optimal graph structure from data using search-and-score methods or constraint-based methods. Search-and-score methods evaluate different structures using a scoring function such as the Bayesian Information Criterion (BIC):

$$\text{BIC}(G \mid D) = \log P(D \mid G) - \frac{\log N}{2} \cdot |G|$$

where $|G|$ is the number of parameters in the network.

2.6.2 Markov Models

Markov Models are stochastic models that describe a sequence of possible events where the probability of each event depends only on the state attained in the previous event. They are particularly useful for modeling time-dependent phenomena.

2.6.2.1 Markov Chains

A Markov chain is a sequence of random variables $X_1, X_2, \ldots$ with proper Markov properties, where the next state depends on the current state and the transition probability, that is, the conditional probability does not depend on the states or values before k:

$$P(X_{k+1} = s \mid X_k = s_k, X_{k-1} = s_{k-1}, \ldots, X_1 = s_1) = P(X_{k+1} = s \mid X_k = s_k)$$

Such chains can have stable probability distributions as $k \to \infty$, which will forget their initial states. Suppose we wish to estimate the states of a random parameter θ; two important things are the current values and the transition probability [23].

2.6.2.2 Hidden Markov models

Hidden Markov Models (HMMs) extend Markov Chains by incorporating hidden states that generate observable events, making them powerful tools for modeling sequences where the system is not directly observable. Given a sequence of observations, HMMs aim to infer the underlying dynamical system, resulting in a model of the process. The three basic problems in HMMs are [21]:

1. Scoring problem, which finds the probability of an observed sequence given the HMM;
2. Alignment problemwhich determines the most likely sequence of hidden states from an observation sequence;
3. Training problem: which creates an HMM from a set of related training sequences.

2.7 Main Challenges of Machine Learning

The primary challenges in ML revolve around selecting an appropriate learning algorithm and training it effectively with data. Issues can arise from both "bad algorithm" choices and "bad data" quality. Below, we discuss common data-related challenges and algorithm-related pitfalls [18, 19].

2.7.1 Challenges with Data

2.7.1.1 Insufficient Quantity of Training Data

Unlike human learning, which can occur with minimal examples, ML algorithms typically require vast amounts of data to perform well. Simple problems may need thousands of exam-

ples, while complex tasks like image or speech recognition might demand millions. Without adequate data, models struggle to generalize from the training set to new, unseen instances.

2.7.1.2 Nonrepresentative Training Data

For a model to generalize effectively, the training data must be representative of the entire population of cases. If the training data is biased or lacks diversity, the model will likely perform poorly on new data. For instance, a training set missing certain countries would not be fully representative, affecting the model's predictions.

2.7.1.3 Poor-Quality Data

Errors, outliers, and noise in the training data can significantly hinder a model's ability to learn. Cleaning the data is crucial to enhance performance.

2.7.1.4 Irrelevant Features

The presence of irrelevant or redundant features can degrade model performance. Effective feature engineering is essential, involving feature selection and feature extraction.

2.7.2 Challenges with Algorithms

Under this section, we will explore the two main issues that can arise with algorithm selection: overfitting and underfitting. To illustrate these concepts and the ideal balance between them, refer to Figure 2.14, which demonstrates underfitting, overfitting, and a well-compromised model.

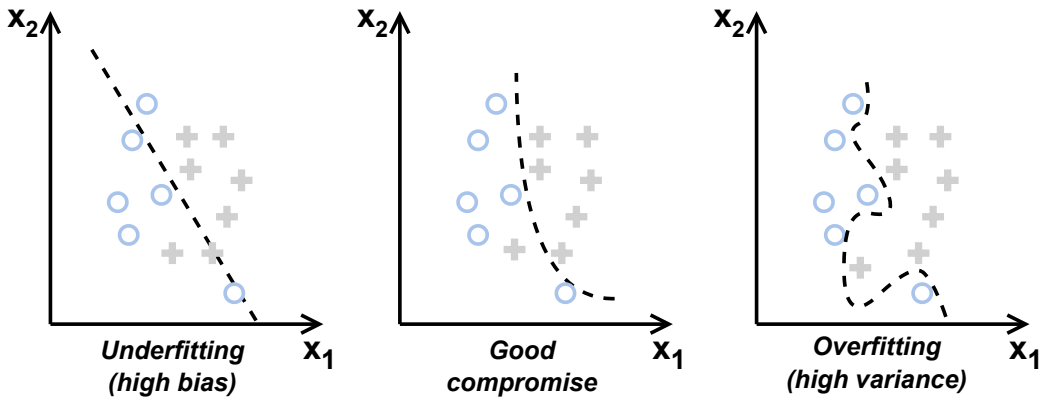

Figure 2.14: Overfitting and underfitting

2.7.2.1 Overfitting the Training Data

Overfitting occurs when a model learns the training data too well, capturing noise and outliers, which hinders its performance on new data. This is akin to overgeneralizing based on limited experiences. Overfitting can be mitigated by using more generalized models, cross-validation, and regularization techniques.

2.7.2.2 Underfitting the Training Data

Underfitting happens when a model is too simplistic to capture the underlying data patterns, leading to poor performance even on training data. This can be addressed by:

- Selecting more complex models with additional parameters.
- Improving feature engineering to provide the algorithm with better data.
- Reducing model constraints, such as easing regularization.

Part two: Natural Language Processing

2.8 Definition of Natural Language Processing

Natural Language Processing (NLP) is a subfield of computer science and AI dedicated to enabling computers to understand and interpret human languages similarly to how humans do. This includes tasks such as sentiment analysis, speech recognition, and automated response generation.

The field has rapidly advanced, leading to significant AI breakthroughs and practical applications like customer service chatbots, mobile phone auto-correct, and virtual assistants such as Cortana and Siri.

NLP systems are typically designed as pipelines, with multiple stages that progressively transform raw language data into refined outputs, ultimately aiming to achieve human-like language processing capabilities in computers [29, 30].

2.9 Applications of Natural Language Processing

Table 2.1 provides a comprehensive overview of various applications of NLP [29].

Table 2.1: Applications of Natural Language Processing

Search	Web	Documents	Autocomplete
Editing	Spelling	Grammar	Style
Dialog	Chatbot	Assistant	Scheduling
Writing	Index	Concordance	Table of contents
Email	Spam filter	Classification	Prioritization
Text mining	Summarization	Knowledge extraction	Medical diagnoses
Law	Legal inference	Precedent search	Subpoena classification
News	Event detection	Fact checking	Headline composition
Attribution	Plagiarism detection	forensics	Style coaching
Sentiment analysis	Morale monitoring	Product review triage	Customer care
Behavior prediction	Finance	Election forecasting	Marketing
Creative writing	Movie scripts	Poetry	Song lyrics

2.10 Challenges of Natural Language Processing

NLP presents a challenging problem domain due to the inherent ambiguity and creativity of human language. This section explores these characteristics in detail [27, 31].

2.10.1 Ambiguity

Ambiguity refers to the uncertainty of meaning in language. Human languages are naturally ambiguous, with words and sentences often having multiple interpretations depending on the context. Ambiguity can occur at the word, sentence, or meaning level.

2.10.2 Common Knowledge

Human language relies heavily on common knowledge—facts assumed to be known by all. For instance, distinguishing between "man bit dog" and "dog bit man" requires understanding that the former is unlikely. Encoding such common knowledge into computational models remains a significant challenge in NLP.

2.10.3 Creativity

Language's creative aspect, encompassing various styles, dialects, and genres, adds complexity. Poetry exemplifies the creative use of language, which is difficult for machines to comprehend.

2.10.4 Diversity Across Languages

The lack of direct vocabulary mappings between most languages complicates NLP solutions. A model effective for one language might fail for another, necessitating either language-agnostic solutions, which are conceptually difficult, or separate solutions for each language, which are labor-intensive.

2.11 Text Mining

Text mining involves the automated extraction of information from large volumes of unstructured text to uncover previously unknown knowledge. Unlike traditional web searches, which aim to retrieve known information, text mining induces new, obscure insights that are not readily discernible through individual reading of existing unstructured text documents. While unstructured text is easily manipulated by humans, it poses significant challenges for computer programs. To address these challenges, text mining first converts unstructured text into structured data, which is then analyzed using quantitative methods.

Text mining techniques are widely applied across various domains, including industry, academia, web applications, and the internet. These techniques are integral to applications such as search engines, customer relationship management systems, email filtering, product recommendation analysis, fraud detection, and social media analytics. They facilitate opinion mining, feature extraction, sentiment analysis, predictive modeling, and trend analysis. The methods employed in text mining are adaptations of long-established techniques in computational linguistics, particularly in the automatic classification of the semantic content of texts [32].

Various methods, including dictionary lookup and ML, are employed for textual entity identification. Information extraction and parsing techniques further dissect text to identify relationships among entities and extract structured information [33].

2.12 Natural Language Processing Techniques for Text Mining

2.12.1 Text Preprocessing

Text mining heavily relies on preprocessing techniques to enhance the quality of data and facilitate subsequent analysis. The following preprocessing techniques are commonly employed

[27, 30].

2.12.1.1 Tokenization

In computer processing, bodies of text are treated as single string objects, regardless of punctuation. To enable the computer to analyze each word individually, we need to segment this unified text into distinct tokens, representing words or characters for further evaluation. This process, known as word tokenization, is fundamental in NLP and serves as a type of document segmentation [30].

As the initial step in an NLP pipeline, tokenization significantly influences subsequent processing stages. A tokenizer divides unstructured natural language text into discrete elements, facilitating counting and analysis of token occurrences within a document. These token counts can be directly utilized as numerical representations of the text, suitable for ML applications. They serve as features in ML pipelines, triggering various actions, responses, or complex decision-making processes [29].

2.12.1.2 Stop word removal

Stop words are frequently occurring words in a language (e.g., "the," "a," "and") that often carry little semantic meaning. Removing stop words helps reduce noise in the data and improves the efficiency of subsequent analysis [29].

2.12.1.3 Text normalization

Text normalization aims to standardize the text data to reduce the number of distinct tokens by combining similar-meaning tokens into a single, normalized form, which can improve the performance of NLP models. Key techniques for text normalization include [29]:

- Case Normalization: Converting all characters in the text to a uniform case, typically lowercase, to ensure that words are treated equally regardless of their original casing.

- Stemming: Stemming involves reducing words to their base or root form by removing suffixes. This technique groups words with similar meanings under a common stem, although the resulting stems may not always be valid words.

- Lemmatization: Lemmatization reduces words to their base or dictionary form, known as a lemma, considering the word's meaning. Unlike stemming, lemmatization uses a

knowledge base to ensure that only words with similar meanings are consolidated into a single token. This process is more accurate than stemming and is preferable for most applications.

2.12.2 Feature extraction

After text preprocessing, the next critical step in the NLP pipeline is feature extraction. Feature extraction, also known as feature engineering or text representation, involves transforming the preprocessed text into a structured format that ML algorithms can interpret and utilize. This process converts textual data into numerical representations while preserving the essential information and relationships within the text. The goal of feature extraction is to capture the characteristics of the text into a numeric vector that can be understood by ML algorithms.

Feature extraction can be approached differently depending on the methodology. In a classical NLP and traditional ML pipeline, feature extraction techniques like Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and n-grams are commonly used. These methods focus on representing text based on the frequency and presence of words and phrases. Conversely, in a deep learning (DL) pipeline, feature extraction often involves word embeddings, such as Word2Vec, GloVe, and FastText, which capture semantic relationships between words [31].

2.12.2.1 Bag of Word

The Bag-of-Words (BoW) model is a straightforward and commonly used feature extraction algorithm in NLP. The essence of the BoW model is to count the frequency of each unique word in a document, disregarding the order or context in which the words appear. This process can be likened to counting items in a bag; the algorithm records the number of each item without concern for their sequence or arrangement. The BoW model operates as follows [29, 30]:

1. Vocabulary Creation: A vocabulary of unique tokens (words) is created from the entire set of documents.

2. Feature Vector Construction: Each document is represented as a feature vector containing the counts of how often each word from the vocabulary appears in the document.

A variant of the BoW model is the Continuous Bag-of-Words (CBoW), which predicts a target word based on its surrounding context, rather than predicting the next word in a sequence as done in the Skip-Gram model.

2.12.2.2 TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) builds on the BoW model by providing a more nuanced measure of a word's importance in a document relative to its prevalence across all documents. The TF-IDF value helps to highlight words that are significant in specific documents while down weighting common words that are less informative. The TF-IDF calculation involves two main components [17, 29, 30, 31]:

- **Term Frequency :** TF measures how often a term or word appears in a given document. To normalize these counts, especially since different documents in the corpus may vary in length, the number of occurrences is divided by the total number of terms in the document. The term frequency $TF(t, d)$ of a term t in a document d is defined as:

$$TF(t, d) = \frac{\text{Number of occurrences of term } t \text{ in document } d}{\text{Total number of terms in document } d}$$

- **Inverse Document Frequency :** IDF measures the importance of a term across the entire corpus. While TF assigns equal weight to all terms, common terms (e.g., "is," "are," "am") are not as informative. IDF addresses this by weighing down frequently occurring terms and emphasizing rare terms. The inverse document frequency IDF(t) of a term t is calculated as:

$$IDF(t) = \log \frac{\text{Total number of documents in the corpus}}{\text{Number of documents containing term } t}$$

The TF-IDF score for a term t in a document d is the product of its term frequency and inverse document frequency:

$$TF - IDF(t, d) = TF(t, d) \times IDF(t)$$

2.12.2.3 N-grams

An N-gram is a contiguous sequence of characters or words extracted from a given text, used extensively in NLP tasks such as text classification. At the character level, a unigram consists of a single character, a bigram includes two characters, and this pattern continues

for longer sequences. Similarly, at the word level, N-grams represent sequences of n words. Unlike the traditional Bag of Words model, which ignores word order, N-grams incorporate the sequential nature of words or phrases, capturing contiguous and sequential word tokens [34].

2.12.2.4 Word Embedding

Traditional feature engineering strategies for textual data, such as the Bag of Words model, including term frequencies, TF-IDF, and N-Grams, are effective but have limitations. These models treat text as a collection of unstructured words, thereby losing valuable information like semantics, structure, sequence, and context of words within a document. This shortcoming motivates the exploration of more sophisticated models that capture such information, leading to the development of word embeddings, which provide vector representations of words.

Word embeddings transform the context of each word into dense vectors, capturing semantic meanings and relationships. These embeddings map words or phrases to vector space, allowing for the measurement of similarity or dissimilarity between them. This transformation facilitates various NLP applications, such as spell checking and sentiment analysis, by providing a continuous vector space where individual words are projected based on their context [27, 30, 34].

2.12.2.4.1 Word2Vec

Word2Vec, introduced by Google in 2013, is a predictive deep learning-based model designed to generate high-quality, distributed, and continuous dense vector representations of words that capture contextual and semantic similarities. This unsupervised model processes massive textual corpora to create a vocabulary and generate dense word embeddings for each word in the vector space. These embeddings, often specified in size by the user, result in a much lower-dimensional space compared to the high-dimensional sparse vector space of traditional Bag of Words models.

Word2Vec employs two primary model architectures to create word embeddings: the Continuous Bag of Words (CBOW) model and the Skip-Gram model. Both models, introduced by Tomas Mikolov et al., use a simple neural network to capture the weights of the hidden layer, which represent the word embeddings. Despite utilizing neural network architecture,

Word2Vec remains computationally efficient and straightforward.

The CBOW model predicts a target word from its surrounding context words. For example, in the sentence "The cat sat on the dirty mat," CBOW would predict the word "mat" using the context words "the," "cat," "sat," "on," and "dirty." Conversely, the Skip-Gram model predicts context words using a target word. Using the same example, Skip-Gram would predict the context words "the," "cat," "sat," "on," and "dirty" based on the target word "mat."

Word2Vec has significantly advanced NLP by allowing for continuous representation of semantically similar words, where words sharing similar contexts also share similar meanings. This model reduces the size of the encoding space and compresses word representations to desired vector lengths. The dense vectors produced by Word2Vec enable the exploration of interesting mathematical relationships between word vectors, such as the analogy "king - man = queen - woman" [28, 34].

2.12.2.4.2 Global Vectors

The GloVe (Global Vectors) model, introduced by Jeffrey Pennington, Richard Socher, and Christopher Manning in 2014, is an advanced method for creating dense vector representations of words. Unlike the Skip-Gram and CBOW models, which train on isolated windows of text, GloVe captures global statistical information by training on aggregated word-to-word co-occurrence counts across the entire corpus. This approach combines the strengths of matrix factorization methods, like Latent Semantic Analysis (LSA), and predictive models like Word2Vec, to produce word embeddings that maintain meaningful sub-structures in vector space.

GloVe begins by constructing a large word-context co-occurrence matrix, where each element represents the frequency of a word appearing with a specific context. This matrix is then factorized to approximate the original co-occurrence counts, typically using methods such as Stochastic Gradient Descent (SGD) to minimize reconstruction error. The resulting word vectors capture semantic relationships and analogies more effectively than either traditional count-based models or isolated context predictive models [30, 34].

2.12.2.4.3 FastText

Introduced by Facebook in 2016, the FastText model is an enhancement of the Word2Vec model, aimed at improving word representations and text classification tasks. Detailed in the

paper "Enriching Word Vectors with Subword Information" by Mikolov et al., FastText offers a comprehensive framework for learning word embeddings. Unlike Word2Vec, which treats words as single entities, FastText considers each word as a bag of character n-grams. For instance, the word "whisper" would be represented by n-grams such as "wh," "whi," "his," "isp," and "per." This approach allows FastText to handle rare and morphologically complex words more effectively [29, 34].

2.12.3 Topic modeling

Topic modeling is an unsupervised learning method used to discover the underlying themes or categories present in a collection of documents. This method assumes that each document is a mixture of topics, with one or more topics dominating. It is particularly useful when the categorization of documents is unknown, as it helps in identifying the latent topics across the entire corpus.

Topic models extract information from text bodies to identify recurring themes. The central idea is that certain topics will appear more frequently in relevant documents and less so in irrelevant ones. This can be beneficial for tasks such as keyword generation for improved searchability or for shorthand summarization. Common applications of topic modeling include document clustering, organizing large text collections, and text classification.

Given a large collection of documents, topic modeling operationalizes the intuition of summarizing the corpus by identifying key terms or "topics" without prior knowledge. Unlike rule-based text mining approaches, topic modeling leverages statistical and mathematical techniques to discover latent semantic structures within the text data.

To provide a comprehensive understanding of topic modeling, we will explore several key techniques that are widely used in this field, including Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA), and BERT (Bidirectional Encoder Representations from Transformers)[28, 29, 34].

2.12.3.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a generative probabilistic model introduced in 2003 by David Blei, Andrew Ng, and Michael I. Jordan. It addresses the limitations of TF-IDF by capturing the semantics and contextual position of words in text. LDA models each document as a mixture of topics and each topic as a distribution over words.

To understand the mechanics of LDA, it is essential to explore its fundamental assumptions and the processes involved in topic generation and word assignment [29, 30]:

1. Document Length: The number of words N in a document follows a Poisson distribution.
2. Topic Distribution: Each document has a topic distribution drawn from a Dirichlet distribution.
3. Word Assignment:
    - Each word in a document is assigned a topic from a multinomial distribution based on θ.
    - Words are then drawn from a topic-specific multinomial distribution.

2.12.3.2 Latent Semantic Analysis

Latent Semantic Analysis (LSA) utilizes Singular Value Decomposition (SVD), a long-established technique for dimension reduction. SVD breaks down a matrix into three simpler matrices, aiding in applications like matrix inversion. In the context of text processing, SVD can decompose a term-document matrix into three matrices, which, when truncated and recombined, provide a reduced representation of the original data.

LSA captures the essential "latent semantics" of documents by retaining the most significant features and eliminating noise. This involves aligning new dimensions with the maximum variance in word frequencies, effectively identifying the combinations of words that account for the most variation. This technique is similar to PCA used in other fields [29].

2.12.3.3 BERT

BERT (Bidirectional Encoder Representations from Transformers), introduced by Google in 2018, has become a revolutionary tool in NLP. It significantly outperforms previous models in various NLP tasks and is widely adopted in both academic research and commercial applications. BERT is a pretrained model that captures the contextual meaning of words, distinguishing between different usages based on context, unlike earlier models like word2vec. This makes it especially useful in complex text mining tasks where language nuances are crucial.

BERT utilizes transformers, specifically focusing on the encoder part, which employs multi-head attention to understand word relationships in a sentence. This bidirectional approach allows BERT to consider both preceding and succeeding words to determine meaning. Fine-tuning pretrained BERT models for specific tasks enhances their performance by leveraging vast amounts of pre-existing knowledge while adapting to new data.

In topic modeling, BERT can be used to create sophisticated models like BERTopic, which combines contextual word embeddings with clustering algorithms such as Uniform Manifold Approximation and Projection (UMAP) and Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN). BERTopic offers a novel approach to topic modeling by leveraging the power of BERT embeddings. These embeddings capture rich semantic information about the text. BERTopic then utilizes a class-based TF-IDF weighting scheme to further refine the representation of each document. To facilitate efficient clustering, the high-dimensional BERT embeddings are projected into a lower-dimensional space using the UMAP technique. This dimensionality reduction allows for efficient clustering of documents based on their thematic similarity [29].

BERTopic generates topic representations through three main steps [35]:

1. Document Embedding: Each document is transformed into an embedding representation using the Sentence-BERT (SBERT) framework, which converts sentences and paragraphs into dense vector representations with a pretrained language model.

2. Dimensionality Reduction: The dimensionality of these embeddings is reduced using techniques like UMAP to improve the clustering process. The embeddings are primarily used to cluster similar documents but not directly for generating topics.

3. Topic Extraction: Using a custom-based variation of TF-IDF, topic representations are extracted from the document clusters. This assumes that documents containing the same topic are semantically similar.

This multi-step process allows BERTopic to effectively identify and represent topics in a corpus, making it a powerful tool for text analysis.

2.13 Conclusion

This chapter has provided a comprehensive overview of the fundamental concepts related to Machine Learning and Natural Language Processing. These two innovative fields hold significant promise in advancing the capabilities of predictive maintenance by transforming unstructured textual data into actionable insights, thereby enhancing the efficiency and reliability of maintenance operations.

In the next chapter, we will delve into the extensive body of research and studies that have been conducted in the realm of predictive maintenance. Effective maintenance strategies are crucial for operational success, and understanding the latest advancements in this field is key to implementing proactive and sustainable practices. To gain a comprehensive understanding of the current state of the field, we will review and analyze various approaches, methodologies, and technologies employed in predictive maintenance.

Chapter3

State of the Art

Contents

In recent years, industries worldwide have increasingly embraced predictive maintenance strategies, driven by technological advancements and the need for operational efficiency. With the rise of sophisticated machinery, the emphasis has shifted toward reducing downtime and prolonging asset lifespan. This trend has sparked a growing interest in leveraging advanced techniques, particularly artificial intelligence, to proactively identify and address potential faults and failures. Researchers have responded to this demand by exploring a variety of methodologies and algorithms, ranging from traditional statistical approaches to cutting-edge artificial intelligence techniques. These efforts aim to refine predictive maintenance models, focusing on aspects like data collection, feature engineering, and algorithm sophistication.

This chapter aims to illuminate the current state of predictive maintenance within industrial contexts, offering insights into methodologies, datasets, and performance metrics. Through comparative analysis, it seeks to guide future research directions and contribute to the advancement of predictive maintenance practices.

3.1 Related Works

To explore predictive maintenance, various techniques have been implemented, including Machine Learning (ML), Natural Language Processing (NLP), and statistical methods. Authors have conducted numerous studies to assess the effectiveness of these approaches in predicting and preventing equipment failures. By categorizing works into ML and NLP

approaches, researchers aim to better understand the strengths and limitations of different predictive maintenance techniques.

3.1.1 Machine Learning based approaches

Machine Learning has emerged as a powerful tool for predictive maintenance, offering sophisticated algorithms capable of analyzing complex datasets to forecast equipment failures and optimize maintenance schedules. Several notable studies have contributed to this field:

Abbasi et al. [36], have developed a Graphical User Interface (GUI) application tailored for predictive maintenance data analytics within the oil and gas sector, leveraging Multiple Linear Regression. The application is devised to mitigate the challenges associated with elevated maintenance expenditures and unplanned downtime by forecasting equipment failures based on real-time operational parameters. Methodologically, it involves data importation from Excel spreadsheets, construction of predictive models employing Multiple Linear Regression, and subsequent normalization of forecasted data. Evaluation entails employing RMSE to assess the accuracy of the predictive model across training and validation datasets. The ability to prognosticate future equipment conditions facilitates proactive maintenance interventions, thereby enhancing equipment reliability and curbing maintenance costs.

Wang et al. [37] employed Long Short-Term Memory Recurrent Neural Networks (LSTM RNN) to address the challenge of capturing long-term dependencies in time series data. LSTM-RNNs, an advanced form of Recurrent Neural Networks (RNNs), feature intricate cell structures with gating mechanisms that selectively retain and forget information. This design overcomes conventional RNNs' limitations in handling extended sequences. Utilizing mem ory cells and specialized gates (input, forget, and output), LSTM-RNNs effectively manage information over long sequences, making them suitable for tasks requiring accurate long-term predictions. The model is trained using Backpropagation Through Time (BPTT) with labeled data to forecast maintenance timing based on historical field data. This approach highlights LSTM-RNN's strong predictive capabilities in enabling proactive maintenance strategies for high-speed railway power equipment, thereby improving equipment reliability and mainte nance efficiency.

Tang et al. [38], proposed an advanced predictive maintenance system tailored for nuclear power steam turbine units, prioritizing safety and operational efficiency. Through the integration of intelligent early warning mechanisms and predictive maintenance strategies, par-

ticularly targeting mechanical equipment like steam turbines, the system exceeds industry standards by incorporating cutting-edge technologies such as the LSTM-Bayes signal anomaly recognition method. Novel early warning parameters including abnormal probability and remaining useful life augment maintenance decision-making, while real-time monitoring and data analysis enable proactive fault detection, reducing safety risks and maintenance costs. This system not only enhances operational safety and productivity in nuclear power plants but also holds promise for optimizing maintenance practices and improving operational performance across various industrial sectors.

3.1.2 Natural language processing based approaches

While traditional maintenance analysis often prioritizes numerical data, a wealth of valuable information resides within textual maintenance records. Recognizing this potential, recent research has increasingly explored text mining techniques to extract meaningful features from such unstructured data. This approach leverages feature extraction, clustering, and classification algorithms to unlock the insights hidden within these records.

Naqvi et al. [39]. proposed a Technical Language Processing (TLP) pipeline for semantic search in industrial text using the transformer-based Large Language Model, BERT. A thorough quantitative and qualitative assessment indicates that LLMs like BERT can efficiently pro cess complex industrial text without normalization. The approach utilizes a case base to store numerical representations of text. BERT is adapted to learn these representations through unsupervised domain fine-tuning. The second part of the methodology utilizes the Case Based Reasoning (CBR) cycle, which consists of four steps: Retrieve, Reuse, Revise, and Retain. This approach was validated on two case studies—excavators and aviation mainte nance records—where only light preprocessing was applied by uppercasing the text. Next, these records were passed to the BERT model and fine-tuned using TSDAE, SimCSE, and CT. The CBR was then applied to retrieve similar maintenance records for excavator and aviation systems. Quantitatively, precision-based analysis showed high accuracy in retrieving relevant records for excavators. Qualitatively, pattern-based analysis demonstrated the sys tem's ability to extract valuable insights from complex maintenance texts. Same fine-tuning techniques were applied to aviation records to predict similar cases. A comparative analysis of training time showed each technique's efficiency. The methodology identified common aviation maintenance issues, such as engine leakages and cylinder problems, providing insights into

prevalent challenges. Results from the excavator case study show that TSDAE outperformed SimCSE and CT in all system/subsystem categories, with an average precision of 0.97. The second-best score was achieved by the model developed using CT, with a precision of 0.63, consistent with the excavator case study results.

Ansari et al. [40]. This paper introduces a scalable AI methodology for enhancing automated processes and extracting knowledge from textual data in DSB, aiming to improve Overall Equipment Effectiveness (OEE). The approach includes early downtime predictions through reducing the Mean Failure Detection Time (MFDT), word recommendations for documen tation, and selecting the best-fit maintenance technician. In an automotive use-case, the methodology led to a 6.7% increase in uptime and a 97.3% rise in Maintenance Fault Detec tion Times (MFDT) under 60 minutes, resulting in a 5.3% boost in OEE. This contrasts with a 2.2% OEE improvement without using the methodology.

Sala et al. [41] utilized refined Natural Language Processing (NLP) techniques, including advanced text-mining methods such as topic modeling, sentiment analysis, and entity recognition, to improve the analysis of maintenance service reports. These methodologies were tailored to extract valuable insights from the textual data, identify patterns, and uncover underlying relationships within the reports. In the case study of an Italian manufacturing company producing bottling machines, these refined NLP techniques were applied to understand intervention distribution, extract key information on common maintenance tasks, identify weaknesses in installed assets, pinpoint issues in the maintenance service delivery process and asset components, and formulate improvement plans. By leveraging these advanced NLP techniques, the company derived actionable strategies for enhancing maintenance service delivery and asset design. The analysis results led to significant improvements in operational efficiency, cost-effectiveness, and overall maintenance service quality, demonstrating the effectiveness of the refined NLP approaches in generating valuable insights for industrial asset and service enhancement.

In his paper, Ansari et al. [42] employ text processing techniques, including sentiment analysis, opinion indexing, and association measuring, to extract quantitative insights from maintenance reports in manufacturing. In the first use case, 124 reports pertaining to a specific machine were analyzed. Keywords linked to maintenance activities were identified, and the Association Measuring Index (AMI) was used to quantify their relationships, demonstrating effectiveness at a specific threshold. In the second use case, 350 out of 900 reports from a

production line were evaluated. This analysis identified the most significant keywords for each machine, with higher AMI values indicating stronger associations. These results underscore the practical application and effectiveness of text processing techniques in extracting valuable information from maintenance reports and quantifying keyword associations with maintenance activities in manufacturing.

3.1.3 Probabilistic based approach

Probabilistic models provide a robust framework for predictive maintenance by capturing the uncertainties inherent in equipment degradation processes. This section highlights key studies that utilize probabilistic approaches to enhance predictive maintenance strategies.

Liang et al. [43] introduce a hidden Markov model with auto-correlated observations (HMM-AO) to improve the prediction accuracy of remaining useful life (RUL) and optimize maintenance decisions for manufacturing systems. The study also addresses the aging power transformer population by proposing a continuous-time Markov chain (CTMC) model, which optimizes transformer maintenance using inspection and continuous monitoring data such as dissolved gas analysis (DGA) and frequency response analysis (FRA). The CTMC model categorizes transformer conditions into five states: healthy, aged, defective, faulty, and failure, considering various subsystems and deterioration types. By analyzing transition rates and integrating real-time monitoring data, the model aims to extend transformer lifespans, reduce costs, and enhance reliability and availability in power distribution and transmission networks.

Chen et al. [44], present a novel approach to predicting the remaining useful life (RUL) of manufacturing systems by combining traditional Hidden Markov Models (HMMs) with autocorrelated observations. This enhancement allows for a more accurate representation of degradation processes over time. The methodology includes a parameter estimation algorithm based on the Expectation-Maximization (EM) method, which accounts for autocorrelated observations, and two RUL prediction methods: a state-based method and an observation-based method. The model's effectiveness is demonstrated through a case study using an LED degradation dataset, showing improved prediction accuracy and optimized maintenance strategies compared to conventional methods.

In the field of predictive maintenance for industrial equipment, Burmeister et al. [45] explore the use of interpretable machine learning models, particularly Classification Trees and Bayesian Networks, to predict maintenance needs based on production data. The study aims

to enable proactive maintenance and cost savings by addressing critical errors identified during manufacturing. Using a score-based learning approach for model development and feature selection, the research highlights the importance of understanding the relationship between production errors and potential failures for effective root cause analysis and early detection. While demonstrating the feasibility of using production data for predictive maintenance, the study also addresses challenges related to data availability, model complexity, and scalability. These findings underline the need for robust data collection and validation procedures to ensure the methodology's effectiveness in industrial environments.

Zhong et al. [46] propose a dynamic risk analysis model for oil pipelines that incorporates imperfect maintenance to optimize risk control measures. The model combines Fault Tree Analysis (FST), Dynamic Bayesian Networks (DBN), and expert judgments to address missing data and capture complex system state relationships. It models multi-state degradation with time-dependent rates affected by operating conditions, environmental factors, and external stresses. Key applications include Remaining Useful Life (RUL) prediction, maintenance optimization, and failure diagnosis. The impact of imperfect maintenance is assessed using the virtual age and improved factor models, which calculate failure and maintenance rates to determine state transitions under various scenarios. A case study validates the model's robustness, demonstrating its effectiveness in enhancing maintenance strategies and improving system reliability and availability.

3.1.4 Hybrid modeling based approach

Hybrid modeling approaches combine various techniques to leverage their individual strengths, resulting in more robust predictive maintenance models. This section highlights significant research employing hybrid methods for maintenance prediction and classification.

Yang et al. [47], propose a novel framework that integrates two key models, which were applied on a dataset of maintenance records collected from excavator buckets. The first one is a CNN-based records clustering model. The CNN likely employs convolutional filters to scan the text for relevant features. Pooling layers then downsample these features, and fully connected layers combine them to generate a numerical representation that captures the overall meaning of the record. These numerical representations are then used by k-means, the clustering technique, to yield groups of similar maintenance interventions, which are assumed to refer to similar degradation states. Trying this model with several numbers of clusters

from 2 to 10, the silhouette score was 80% for 4 clusters which is the highest score but only 2 clusters out of the 4 clusters were believed to represent distinct equipment degradation states. Considering cluster 1 relates to partially damaged components whereas cluster 2 relates to severely damaged components, the stochastic model establishes a correspondence between the maintenance records clusters 1 and 2 and the component degradation states. By analyzing the timing of maintenance interventions within each chosen cluster, the model can estimate the equipment's degradation rate and predict future maintenance needs. This allows for the development of customized maintenance strategies tailored to the specific condition of each equipment component.

Ahadh et al. [48], have introduced a novel, semi-supervised, domain-independent approach for automating the classification of accident reports, significantly reducing the need for extensive human intervention typically required by supervised learning methods. Traditional text classification relies on pre-classified documents, demanding substantial human effort for curation. In contrast, their methodology leverages a keyword-based approach to text classification. It involves automatically extracting domain-specific keywords from literature sources such as handbooks, glossaries, and journal papers, capturing the essence of documents for effective text mining. These keywords are then utilized to generate relevant topics that organize accident reports into meaningful categories. This process facilitates the automated labeling of documents, allowing for the creation of predictive models with minimal human involvement. The proposed system not only enhances efficiency by saving time and resources but also maintains domain independence, making it applicable across various fields. Ultimately, this approach demonstrates a significant advancement in automating document analysis, offering a scalable solution for reducing manual intervention in text classification tasks.

Valcamonico et al. [49], have combined NLP techniques with Bayesian networks to analyze textual reports of process safety events (PSEs) in hydrocarbon production assets. It begins by developing a Bayesian network model that integrates expert knowledge with information extracted from PSE reports. To facilitate this, a taxonomy is created to map terms from the reports to Influencing Factors (IFs), thereby organizing and categorizing the information. An NLP-based model is then used to define the structure of the Bayesian network and estimate Conditional Probability Tables (CPTs), automating hyperparameter settings to improve efficiency. This model allows for tracking the evolution of PSE probabilities over time and identifying significant influencing factors. The methodology is validated through application

to a real repository of PSE reports from hydrocarbon plants, showcasing its ability to identify critical factors influencing the severity of PSE consequences and supporting decision-making to enhance safety and reliability in the oil and gas industry.

3.2 Comparative Study of Predictive Maintenance Approaches

Previously, we have presented the main prediction approaches in the field of maintenance. In the Table 3.1, we will carry out a comparative study of the approaches proposed above according to the following five factors:

- **Paper:** presents the paper.
- **Methodology:** indicates the methodology followed in the associated paper.
- **Used techniques:** describes the specific techniques or models applied for predictive maintenance.
- **Forces:** presents the main advantages of the approach.
- **Limitations:** shows the disadvantages of the techniques used.

Table 3.1: Comparative Study of Predictive Maintenance Approaches

Paper	**Methodology**	**Used techniques**	**Forces**	**Limitations**
[36]	Building of Predictive Model based on Multiple Linear Regression on normalized data	Multiple Linear Regression	Modeling of relationships between multiple parameters, providing a comprehensive approach to predictive maintenance.	Setting appropriate thresholds and normalization parameters requires expertise to ensure effective and accurate system operation.

[37]	Predictive and proactive maintenance strategies combination for high-speed railway power equipment. It involves two key components: the Sample Generator and the Maintenance Predictor powered by LSTM-RNN.	LSTM-RNN	- The LSTM-RNN-based Maintenance Predictor can accurately predict maintenance timing based on historical field data. - The Sample Generator creates sample data based on physical degradation models, ensuring a continuous supply of data for training and testing the Maintenance Predictor.	- The effectiveness of the approach relies heavily on the availability and quality of historical field data for training the LSTM-RNN model; insufficient or inaccurate data may impact prediction accuracy. - Implementation into existing maintenance practices for high-speed railway power equipment Challenges
[38]	Capture long-term dependencies with LSTM models and use Bayesian methods for anomaly recognition.	LSTM, bayesian methods	- The LSTM-Bayes signal anomaly recognition method improves fault detection accuracy and prediction performance.	- The system's effectiveness relies heavily on the availability and quality of equipment data. - The LSTM model requires adequate training data and periodic retraining to maintain accuracy and reliability, which can be resource-intensive.
[39]	TLP Pipeline using BERT.	BERT, CBR, TSDAE, SimCSE, CT.	- It gives a High accuracy (0.97 with TSDAE). - Semantic clustering with BERT. - It does not require custom preprocessing, only lowercasing. - It works with unlabeled data	- Hard to evaluate, there is no ground truth available for performance comparison. - Variations in acronyms, non-standard writing, and assumed familiarity with equipment make processing records challenging. - The complexities and noise in the industrial text.

[40]	Scalable AI Methodology for Automated Processes.	AI techniques, MFDT reduction.	- Significant OEE improvement (5.3%). - Comprehensive strategy (Downtime prediction, Word recommendations for documentation, Selection of the best-fit maintenance technician).	- challenging computational efficiency, processing speed, and scalability with large log file datasets. - complexe data preprocessing. - It involves addressing compatibility, data synchronization, and system interoperability challenges for effective predictive maintenance.
[41]	Advanced NLP Techniques for Maintenance Reports.	Topic modeling, sentiment analysis, entity recognition.	- Effective insight extraction. - Tangible improvements in operational efficiency and service quality.	- Limited generalizability (case study specific to one industry). - Requires significant technical expertise.
[42]	Text Processing for Quantitative Insights	Sentiment analysis, opinion indexing, AMI.	- The technique reveals concealed patterns and sentiments within maintenance texts, boosting knowledge intelligence. - It transforms text into cost-related metrics, enhancing maintenance planning. - The method supports decision-making through a unified Text Understanding Map.	- Sensitivity to AMI thresholds. - Poor data quality in manufacturing reduces the effectiveness of the method. - The complexity of results from advanced KDT can make interpretation difficult. - Reliance on subjective factors such as readability affects the reliability of the analysis.

[43]	Development of a continuous-time Markov chain model which integrates information from periodic inspections and continuous monitoring to optimize maintenance.	CTMC	- The method accurately models power transformer deterioration by combining cause-effect analysis and a continuous-time Markov chain model. - Optimization of maintenance strategies by integrating information from both periodic inspections and continuous monitoring	- Obtaining comprehensive transformer malfunction data is difficult. - Implementing cause-effect analysis and continuous-time Markov chain modeling is complex. - Accurate assumptions about transition rates and maintenance costs are critical for optimization outcomes.
[44]	Development of HMM-AO to model degradation processes and predict the remaining RUL of manufacturing systems.	HMMs, EM Algorithm and Multivariate Gaussian Distributions	- Improved Prediction Accuracy by considering the autocorrelation of observations. - Adaptability to Missing Data and Noise. - Ability to capture the temporal evolution of degradation processes and account for autocorrelated observations.	- Complexity of Parameter Estimation, especially with the consideration of auto-correlated observations. - Assumption of Autocorrelation which may not always hold true for all manufacturing systems.
[45]	Use of classification Trees and Bayesian Networks to develop models for predicting critical errors and root causes.	Bayesian Networks, Classification Trees Cross Validation and F1 score.	- The use of interpretable models such as Classification Trees and Bayesian Networks allows for a clear understanding of the factors influencing maintenance needs, enabling root cause analysis.	- The effectiveness of the methodology relies on the availability and quality of production data. Inadequate or incomplete data may limit the accuracy of predictive models.

[46]	CNN-based Records Clustering Model with a multi-state degradation model.	CNN,k-means clustering.	- Gives a High precision (silhouette score of 80%). - Automatically extract information from maintenance records without expert intervention - does not require an expert.	- An exhaustive test of the number of clusters to maximize the silhouette score and identify significant clusters. - The complexity of developing a stochastic multi-state degradation model, including parameter estimation for transition rates and failure probabilities.
[47]	Modeling of multi-state degradation with time-dependent rates affected by operating conditions, environmental factors, and external stresses.	Fault Tree Analysis (FST), DBN, and Markov model.	- The integration of FTA, DBN, and expert judgments allows for a comprehensive assessment of the risks associated with oil pipelines. - Handeling uncertainties and fuzziness in the risk assessment process. - DBN facilitate dynamic risk assessment by considering the temporal evolution of system states, degradation processes, and maintenance activities.	- Managing and interpreting the interactions between different components of the model could be challenging. - Data Requirements. - Biases or inaccuracies in expert opinions could impact the reliability of the results.
[48]	Proposition of an automated, semi-supervised, domain-independent approach for analyzing accident reports with a keyword-based approach to text classification	Keyword Extraction Algorithm, LDA and accuracy metric.	- The Semi-Supervised Approach requires minimal manual intervention and does not rely on labeled accident reports for training, making it applicable across various domains.	- The topics generated by the algorithm may not always be easily interpretable by humans, especially if the initial assignment of words to topics is random. - Domain-specific keyword Selection may require careful curation and refinement.

[49]	Combination of Natural Language Processing (NLP) and Bayesian Networks (BNs) to estimate the probabilities of Process Safety Events (PSEs) severity in hydrocarbon production assets.	TF-IDF, LDA, CNN, BERT and Bayesian Networks.	- Enhanced risk assessment integrates NLP and Bayesian Networks (BNs). - NLP techniques extract valuable information from textual reports, revealing factors influencing PSE severity often missed by traditional methods. - BNs facilitate probabilistic estimation of PSE severity, offering a nuanced understanding of the likelihood and impact of different events.	- The methodology may require thorough validation and testing across different hydrocarbon production assets to ensure its generalizability and reliability in diverse operational contexts. - Data Quality required.

3.3 Conclusion

In recent years, the field of predictive maintenance has advanced significantly, driven by technological innovations and the need for greater operational efficiency. This chapter highlights key developments in leveraging machine learning, natural language processing, and probabilistic models to forecast equipment failures and optimize maintenance schedules. Notable approaches include the use of LSTM-RNNs for long-term predictions, transformer-based models like BERT for analyzing maintenance texts, and hybrid models combining multiple techniques. These methodologies have demonstrated substantial improvements in equipment reliability, cost reduction, and operational safety across various industries, guiding future research and enhancing maintenance practices.

General Conclusion

In conclusion, this dissertation emphasizes the transformative impact of predictive maintenance within the framework of Industry 4.0, focusing on the integration of advanced data analytics to enhance operational excellence in industrial settings. The research demonstrates that by leveraging Natural Language Processing (NLP), probabilistic models, and machine learning techniques, industries can transition from traditional reactive and preventive maintenance strategies to more proactive and predictive approaches. This shift ensures not only higher equipment reliability and availability but also minimizes downtime and associated costs, ultimately boosting production efficiency and profitability.

The comprehensive review of existing research studies and methodologies within this dissertation illustrates the effectiveness of these advanced technologies in analyzing both structured and unstructured data for predictive maintenance. The ability to process and interpret large volumes of data allows for the early detection of potential equipment failures, enabling timely and informed decision-making. This proactive approach to maintenance not only enhances the lifespan of industrial assets but also optimizes resource allocation and reduces unexpected disruptions in production processes.

Furthermore, the insights gained from this research provide a robust framework for implementing predictive maintenance strategies, paving the way for more intelligent and efficient industrial operations. The adoption of such advanced maintenance techniques aligns with the goals of Industry 4.0, promoting a more data-driven and automated industrial environment.

Bibliography

[1] A. Popovič, R. Hackney, P. S. Coelho, and J. Jaklič, "Towards business intelligence systems success: Effects of maturity and culture on analytical decision making," *Decision support systems*, vol. 54, no. 1, pp. 729–739, 2012.

[2] W. Yeoh and A. Koronios, "Critical success factors for business intelligence systems," *Journal of computer information systems*, vol. 50, no. 3, pp. 23–32, 2010.

[3] R. T. Ng, P. C. Arocena, D. Barbosa, and G. Carenini, *Perspectives on Business Intelligence*. Morgan & Claypool Publishers, 2013.

[4] H. L. H. S. Warners and R. Randriatoamanana, "Datawarehouser: A data warehouse artist who have ability to understand data warehouse schema pictures," in *2016 IEEE Region 10 Conference (TENCON)*, pp. 2205–2208, IEEE, 2016.

[5] A. L. Antunes, E. Cardoso, and J. Barateiro, "Incorporation of ontologies in data warehouse/business intelligence systems-a systematic literature review," *International Journal of Information Management Data Insights*, vol. 2, no. 2, p. 100131, 2022.

[6] R. Kimball, M. Ross, W. Thorthwaite, B. Becker, and J. Mundy, *The data warehouse lifecycle toolkit*. John Wiley & Sons, 2008.

[7] D. L. Moody and M. A. Kortink, "From enterprise models to dimensional models: a methodology for data warehouse and data mart design.," in *DMDW*, p. 5, 2000.

[8] A. K. VK, *Business Intelligence Demystified: Understand and Clear All Your Doubts and Misconceptions About BI (English Edition)*. BPB Publications, 2021.

[9] B. Ramesh and A. Ramakrishna, "Unified business intelligence ecosystem: a project management approach to address business intelligence challenges," in *2018 Portland International Conference on Management of Engineering and Technology (PICMET)*, pp. 1–10, IEEE, 2018.

[10] P. R. Clavier, H. H. Lotriet, and J. J. van Loggerenberg, "Business intelligence challenges in the context of goods-and service-dominant logic," in *2012 45th Hawaii International Conference on System Sciences*, pp. 4138–4147, IEEE, 2012.

[11] K. Zhong, T. Jackson, A. West, and G. Cosma, "Natural language processing approaches in industrial maintenance: A systematic literature review," *Procedia Computer Science*, vol. 232, pp. 2082–2097, 2024.

[12] F. Longo, L. Nicoletti, and A. Padovano, "Smart operators in industry 4.0: A human-centered approach to enhance operators' capabilities and competencies within the new smart factory context," *Computers & industrial engineering*, vol. 113, pp. 144–159, 2017.

[13] A. Sahli, R. Evans, and A. Manohar, "Predictive maintenance in industry 4.0: Current themes," *Procedia CIRP*, vol. 104, pp. 1948–1953, 2021.

[14] Y. Wang, C. Deng, J. Wu, Y. Wang, and Y. Xiong, "A corrective maintenance scheme for engineering equipment," *Engineering Failure Analysis*, vol. 36, pp. 269–283, 2014.

[15] B. De Jonge and P. A. Scarf, "A review on maintenance optimization," *European journal of operational research*, vol. 285, no. 3, pp. 805–824, 2020.

[16] D. Meira, I. Lopes, and C. Pires, "Selection of computerized maintenance management systems to meet organizations' needs using ahp," *Procedia Manufacturing*, vol. 51, pp. 1573–1580, 2020.

[17] "Siveco." https://www.siveco.com/en/cmms-software/coswin-8i. Accessed: 2024-06-01.

[18] S. Raschka and V. Mirjalili, "Python machine learning packt publishing ltd," 2015.

[19] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow.* " O'Reilly Media, Inc.", 2022.

[20] P. N. Mahalle, P. P. Hujare, and G. R. Shinde, *Predictive Analytics for Mechanical Engineering: A Beginners Guide*. Springer, 2023.

[21] M. Mohammed, M. B. Khan, and E. B. M. Bashier, *Machine learning: algorithms and applications*. Crc Press, 2016.

[22] I. Sarker, "Machine learning: algorithms, real-world applications and research directions. sn comput sci 2: 160," 2021.

[23] X.-S. Yang, *Introduction to algorithms for data mining and machine learning*. Academic press, 2019.

[24] J. Alzubi, A. Nayyar, and A. Kumar, "Machine learning from theory to algorithms: an overview," in *Journal of physics: conference series*, vol. 1142, p. 012012, IOP Publishing, 2018.

[25] Y. Zhang, *New advances in machine learning*. BoD–Books on Demand, 2010.

[26] E. Alpaydın, "Introduction to machine learning," 2010.

[27] P. Goyal, S. Pandey, and K. Jain, "Deep learning for natural language processing," *New York: Apress*, 2018.

[28] R. Arumugam and R. Shanmugamani, *Hands-On Natural Language Processing with Python: A practical guide to applying deep learning architectures to your NLP applications*. Packt Publishing Ltd, 2018.

[29] H. Hapke, C. Howard, and H. Lane, *Natural Language Processing in Action: Understanding, analyzing, and generating text with Python*. Simon and Schuster, 2019.

[30] T. Beysolow, "Applied natural language processing with python," 2018.

[31] S. Vajjala, B. Majumder, A. Gupta, and H. Surana, *Practical natural language processing: a comprehensive guide to building real-world NLP systems*. O'Reilly Media, 2020.

[32] G. Nota, A. Postiglione, and R. Carvello, "Text mining techniques for the management of predictive maintenance," *Procedia Computer Science*, vol. 200, pp. 778–792, 2022.

[33] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, and M. Data, "Practical machine learning tools and techniques," in *Data mining*, vol. 2, pp. 403–413, Elsevier Amsterdam, The Netherlands, 2005.

[34] D. Sarkar, *Text analytics with Python: a practitioner's guide to natural language processing*. Springer, 2019.

[35] A. Abuzayed and H. Al-Khalifa, "Bert for arabic topic modeling: An experimental study on bertopic technique," *Procedia computer science*, vol. 189, pp. 191–194, 2021.

[36] T. Abbasi, K. H. Lim, N. Rosli, I. Ismail, and R. Ibrahim, "Development of predictive maintenance interface using multiple linear regression," in *2018 International Conference on Intelligent and Advanced System (ICIAS)*, pp. 1–5, IEEE, 2018.

[37] Q. Wang, S. Bu, and Z. He, "Achieving predictive and proactive maintenance for high-speed railway power equipment with lstm-rnn," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6509–6517, 2020.

[38] J. Tang, D. You, F. Li, and Y. Cheng, "Development of predictive maintenance system for nuclear power turbine unit," in *2023 2nd International Conference on Artificial Intelligence and Computer Information Technology (AICIT)*, pp. 1–4, IEEE, 2023.

[39] S. M. R. Naqvi, M. Ghufran, C. Varnier, J.-M. Nicod, K. Javed, and N. Zerhouni, "Unlocking maintenance insights in industrial text through semantic search," *Computers in Industry*, vol. 157, p. 104083, 2024.

[40] F. Ansari, L. Kohl, J. Giner, and H. Meier, "Text mining for ai enhanced failure detection and availability optimization in production systems," *CIRP Annals*, vol. 70, no. 1, pp. 373–376, 2021.

[41] R. Sala, F. Pirola, G. Pezzotta, and S. Cavalieri, "Nlp-based insights discovery for industrial asset and service improvement: an analysis of maintenance reports," *IFAC-PapersOnLine*, vol. 55, no. 2, pp. 522–527, 2022.

[42] F. Ansari, "Cost-based text understanding to improve maintenance knowledge intelligence in manufacturing enterprises," *Computers & Industrial Engineering*, vol. 141, p. 106319, 2020.

[43] Z. Liang and A. Parlikad, "A markovian model for power transformer maintenance," *International Journal of Electrical Power & Energy Systems*, vol. 99, pp. 175–182, 2018.

[44] Z. Chen, Y. Li, T. Xia, and E. Pan, "Hidden markov model with auto-correlated observations for remaining useful life prediction and optimal maintenance policy," *Reliability Engineering & System Safety*, vol. 184, pp. 123–136, 2019.

[45] N. Burmeister, R. D. Frederiksen, H. Esben, P. Nielsen, *et al.*, "Exploration of production data for predictive maintenance of industrial equipment: A case study," *IEEE Access*, 2023.

[46] W. Zhong, J. Cai, Y. Song, T. Liang, J. Zhang, and Z. Gao, "Risk evolution of crude oil pipeline under periodic maintenance based on dynamic bayesian network," *Journal of Loss Prevention in the Process Industries*, vol. 87, p. 105229, 2024.

[47] Z. Yang, P. Baraldi, and E. Zio, "A novel method for maintenance record clustering and its application to a case study of maintenance optimization," *Reliability Engineering & System Safety*, vol. 203, p. 107103, 2020.

[48] A. Ahadh, G. V. Binish, and R. Srinivasan, "Text mining of accident reports using semi-supervised keyword extraction and topic modeling," *Process safety and environmental protection*, vol. 155, pp. 455–465, 2021.

[49] D. Valcamonico, P. Baraldi, E. Zio, L. Decarli, A. Crivellari, and L. La Rosa, "Combining natural language processing and bayesian networks for the probabilistic estimation of the severity of process safety events in hydrocarbon production assets," *Reliability Engineering & System Safety*, vol. 241, p. 109638, 2024.