Thèse de Doctorat

Mention : Sciences et technologies de l'information et de la communication
Spécialité : Informatique et applications

présentée à *l'École Doctorale en Sciences Technologie et Santé (ED 585)*

de l'Université du Littoral Côte d'Opale

par

Mohammad CHOAIB

pour obtenir le grade de Docteur de l'Université du Littoral Côte d'Opale

Vers des frameworks conversationnels explicables pour l'assistance à la configuration des systèmes cyber-physiques

Soutenue le 15 Decembre 2025, après avis des rapporteurs, devant le jury d'examen :

M. Cyril Fonlupt , Professeur, Université du Littoral Côte d'Opale	**Président**
M^me^. Kathia Marçal de Oliveira, Professeur, Université Polytechnique Hauts-de-France	**Rapporteure**
M. Abdeljalil Abbas-turki, Professeur, Université de Technologie de Belfort-Montbéliard	**Rapporteur**
M^me^. Habiba Chaoui, Professeur, École Nationale des Sciences Appliquées de Kénitra	**Examinatrice**
M. Moncef Garouani, Maître de Conférences, IRIT- Université Toulouse Capitole	**Invité**
M. Adeel Ahmad, Maître de Conférences, Université du Littoral Côte d'Opale	**Co-encadrant**
M. Mourad Bouneffa, Professeur, Université du Littoral Côte d'Opale	**Directeur de thèse**
M. Yasser Mohanna, Professeur, Université Libanaise	**Directeur de thèse**

PhD Thesis

Major : Information and Communication Sciences and Technologies
Specialization : Computer Science and Applications

Presented to the Doctoral School in Science, Technology and Health (ED 585)

of the l'Université du Littoral Côte d'Opale

by

Mohammad CHOAIB

to obtain the degree of Doctor of Philosophy from the Université du Littoral Côte d'Opale

Towards Explainable Conversational Frameworks for Cyber-Physical System Configuration Assistance

Defended on December 15, 2025, after the approval of the reviewers, in front of the examining board :

Mr. Cyril Fonlupt, Professor, Université du Littoral Côte d'Opale	**President**
Ms. Kathia Marçal de Oliveira, Professor, Université Polytechnique Hauts-de-France	**Reviewer**
Mr. Abdeljalil Abbas-turki, Professor, Université de Technologie de Belfort-Montbéliard	**Reviewer**
Ms. Habiba Chaoui, Professor, École Nationale des Sciences Appliquées de Kénitra	**Examiner**
Mr. Moncef Garouani, Maître de Conférences, IRIT- Université Toulouse Capitole	**Invited**
Mr. Adeel Ahmad, Maître de Conférences, Université du Littoral Côte d'Opale	**Co-Supervisor**
Mr. Mourad Bouneffa, Professor, Université du Littoral Côte d'Opale	**Thesis director**
Mr. Yasser Mohanna, Professor, Université Libanaise	**Thesis director**

Remerciements

First and foremost, I give thanks to God for providing me with the strength, patience, and perseverance required to complete this work. His guidance has accompanied me throughout this journey.

The completion of this thesis marks the culmination of years of work, reflection, and collaboration, and it would not have been possible without the contribution and support of many individuals. I would like to take this opportunity to express my heartfelt gratitude to all those who, in one way or another, made this journey achievable.

My deepest thanks go first to my supervisors Pr. Mourad Bouneffa and Pr. Yasser Mohanna, whom mentorship has guided me throughout this research. Their unwavering support, insightful advice, and academic generosity shaped the foundations of this work and constantly pushed me to broaden my perspective. Their patience, trust, and kindness created an environment in which I could grow both scientifically and personally. For Their encouragement during moments of doubt and his ability to illuminate the path forward in times of difficulty, I owe them my sincere gratitude.

I would also like to warmly thank my Co-supervisor Dr. Adeel Ahmad for his valuable input, availability, and constructive discussions. His advice and encouragement greatly contributed to shaping this work and helped me develop a clearer scientific perspective.

My sincere appreciation goes to Dr. Moncef Garouani for his support and help throughout these years. His willingness to assist, share ideas, and provide technical guidance made several difficult steps of this research much easier to navigate.

My sincere thanks extend to the members of my thesis committee for dedicating their time to review and evaluate my work. Their insightful comments and constructive suggestions contributed significantly to improving the quality of this manuscript. I am truly honored by their participation and their contribution to my academic growth.

Special thanks to Université du Littoral Côte d'Opale (ULCO), and the Lebanese University for supporting this work with a scholarship grant.

I would like to extend my heartfelt thanks to Pr. Georges Stienne, whose support and friendship made a significant difference throughout my PhD. His guidance, encouragement, and kindness were invaluable to me, both academically and personally, and I am truly grateful for his presence during this journey.

My appreciation goes as well to the laboratory team, including its directors, administrative staff, and all research colleagues, for creating a welcoming, collaborative, and intellectually stimulating environment.

My deepest appreciation goes to my family, whose unconditional love and support have sustained me throughout this entire journey. Their patience, understanding, and constant encouragement gave me the strength and determination to persevere. No words can adequately express how grateful I am for their belief in me.

I would like to express my sincere appreciation to all my friends who accompanied me throughout this journey, each contributing in a meaningful way. To the friends who were by my side throughout the entire process and supported me from start to finish Dr. Hassan Yassine and Dr. Jawad Reslan, thank you for your unwavering presence and constant encouragement. To the friends who helped me at specific moments in the work, whether through discussions, advice, or assistance when it was most needed Dr. Hamza Issa, Dr. Sarah El-Hajj Chehade, and Mr. Ibrahim Cherri your support played an important role in the progress of this thesis. To the friends who were there for me, offering balance, friendship, and support Dr. Ali Amhaz, Dr. Hassan Aladdine, Dr. Ali Chamsedine, Dr. Mikhael Macary, Mr. Ralph Mattar, Mr. Hassan Khalife, Mr. Mostapha Youssef, Mr. Mohamad Atwi, and Dr. Hadi Elzein, I would like to say thank you from the bottom of my heart. To the friends who helped prepare for the final event, from rehearsals to organizational details Ms. Rita Arfoul, Ms. Safae Abouelouafa, and Ms. Sharanya Dattamandal your help and dedication ensured everything came together smoothly.

The most important, I would like to express my heartfelt gratitude to my wife, whose love, patience, and encouragement have been a constant source of strength. Thank you for standing beside me through every step of this journey and for believing in me even when I doubted myself. Your presence has made this accomplishment possible and profoundly meaningful.

Finally, I extend my heartfelt thanks to all the people who supported me in ways big and small, even if their names are not mentioned here. Your presence and kindness have been sincerely appreciated.

Abstract

In the era of Industry 4.0, the continuous evolution of Cyber-Physical Systems (CPS) design and their corresponding component configurations has made the process increasingly complex, cognitively demanding, and prone to human error. Researchers, engineers, domain experts, and also non-expert users face the challenge of transforming high-level, domain-specific requirements into precise technical specifications while dealing with heterogeneous component data, inconsistent schemas, and interoperability constraints. Traditional configuration tools rely on rigid forms or expert-driven rules that offer limited adaptability and almost no transparency in their recommendations.

This thesis introduces the Cyber-Physical Recommendation System (CPRS) an explainable conversational framework designed to assist users in CPS component selection and configuration through natural-language interaction and interpretable decision logic. CPRS combines semantic embeddings, rule-based validation, and explainable artificial intelligence (XAI) within a modular architecture that integrates classical information retrieval with large language model (LLM) generation. At its core, CPRS employs Sentence-BERT (SBERT) to represent user queries and component metadata within a shared semantic space, enabling accurate similarity-based retrieval. Through semantic retrieval and multi-criteria ranking, the system identifies compatible components and iteratively refines recommendations via interactive clarification. Experimental evaluation on a curated CPS sensor dataset demonstrates that CPRS achieves higher accuracy (0.85) and faster response times compared to Word2Vec- or BERT-based baselines, while maintaining more transparency of the reasoning process. User studies further reveal that 85% of participants found the system intuitive and context-aware, confirming the effectiveness of natural-language explanations in enhancing trust and usability.

The thesis presents several key contributions : a modular and explainable CPS recommendation pipeline; the integration of SBERT embeddings with a structured knowledge base; SHAP-based feature attribution for transparent reasoning; and the use of Mistral-7B for adaptive, user-centered response generation. To promote user trust and accountability, CPRS incorporates explainable AI techniques that justify configuration choices by highlighting influential parameters and decision factors. Experimental evaluations across representative CPS design scenarios demonstrate significant improvements in configuration efficiency, recommendation relevance, and interpretability compared to traditional, form-based approaches. Limitations related to

web-scraping fragility, embedding ambiguity, and static knowledge bases are acknowledged, with proposed future directions for robust data acquisition, domain-specific fine-tuning, multimodal interaction, and continual learning.

Keywords : recommendation systems ; cyber physical system ;decision support systems ;natural language processing ; conversational recommendation system.

Table des matières

Remerciements i

Abstract iii

Table des figures xi

Liste des tableaux xiii

Liste des Abbréviations xv

Liste des Symboles xvii

General Introduction 1

1 Introduction to Cyber Physical Systems : Understanding Complexities and Addressing Systemic Challenges 5

1.1 Introduction to Cyber–Physical Systems 6
1.2 Core Components of Cyber–Physical Systems 7
1.3 Architectural Models and Design Patterns 9
1.3.1 Modeling and Simulation 9
1.3.2 System Design and Engineering 9
1.3.3 Development Frameworks 10
1.3.4 Representative Architectures 10
1.4 Application Domains of Cyber–Physical Systems 11
1.4.1 Healthcare 11
1.4.2 Manufacturing 12
1.4.3 Transportation 12
1.5 Emerging Trends and Challenges 12
1.5.1 Artificial Intelligence and Machine Learning in CPS . . 13
1.5.2 Human–CPS Interaction 13
1.5.3 Regulatory and Ethical Considerations 13
1.5.4 From Challenges to CPS Implementation 14

1.5.5 Motivation for a Cyber–Physical Recommendation System . . . 14
1.6 Conclusion . . . 15

Bibliographie **17**

2 A Context-Aware Conversational Recommendation Framework for Cyber-Physical Systems Design and Integration **21**
2.1 Fundamentals of Recommendation Systems . . . 23
2.1.1 Core Objectives and Motivation . . . 23
2.1.2 Architectural Components . . . 24
2.1.3 Principal Recommendation Techniques . . . 24
2.1.4 Relevance to CPS Configuration . . . 27
2.2 Motivation and Research Problem . . . 28
2.3 Conversational recommendation systems . . . 29
2.3.1 Dialogue Systems for CPS . . . 30
2.3.2 Conversational Recommendation Systems for CPS . . . 31
2.3.3 IoT-AID : Proposed Framework . . . 31
2.4 Conclusion . . . 32

Bibliographie **35**

3 Architectural Framework, Design and Integration Details of IoT-AID's Conversational CPRS **37**
3.1 Proposed CPS Architectures and Connections . . . 38
3.2 Proposed Recommendation System . . . 39
3.3 Illustrative Example : Real-Time Patient Surveillance System . . . 41
3.4 Natural Language Processing Module . . . 42
3.4.1 Preprocessing and Text Normalization . . . 43
3.4.2 Prompt-Based Semantic Understanding . . . 43
3.4.3 Post-Processing and Data Analysis . . . 43
3.4.4 Integration and Continuous Refinement . . . 44
3.5 Conclusion . . . 44

Bibliographie **47**

4 Implementation Framework for IoT-AID : Leveraging Large Language Models in the Development of Conversational CPRS **49**
4.1 System Architecture and Module Integration . . . 50
4.1.1 Overview of IoT-AID Architecture . . . 51

4.1.2 Module Integration Workflow 51
4.2 NLP Pipeline Implementation 52
4.2.1 Overview of the NLP Workflow 53
4.2.2 Semantic Representation and Embedding Strategy . . . 55
4.2.3 Dialogue Context Integration 55
4.2.4 Outcome of the NLP Pipeline 55
4.3 Integration of Large Language Models in IoT-AID 56
4.3.1 Role of LLMs within the IoT-AID Framework 56
4.3.2 Model Selection and Adaptation Strategy 57
4.3.3 Embedding-Based Knowledge Retrieval 57
4.3.4 Explainable Recommendations through LLM Interpretability 58
4.4 Recommendation Engine Implementation 58
4.4.1 Data Acquisition and Structuring 59
4.4.2 Recommendation Model Architecture and Workflow . 61
4.4.3 Results and Analysis 63
4.5 CPRS Text classification 64
4.5.1 DataSet Processing 64
4.5.2 Data Preprocessing 66
4.5.3 Feature Extraction 67
4.6 Conclusion 67

Bibliographie **69**

5 Implementation and Evaluation of the Cyber-Physical Recommendation System **71**
5.1 General System Architecture 72
5.2 Workflow and Module Descriptions 77
5.2.1 Preprocessing and Embedding Preparation 77
5.2.2 Intent Classification 79
5.2.3 Information Sufficiency Module 80
5.2.4 Similarity-Based Retrieval 83
5.2.5 Explainability Module via SHAP 84
5.2.6 Response Generation (Mistral-7B) 86
5.2.7 Knowledge Base Construction 88
5.2.8 Data Acquisition 88
5.2.9 Data Structuring 89
5.2.10 Embedding Preparation 90
5.2.11 Advantages of the Hybrid Approach 91

5.2.12 Transparency and User Feedback 91
5.3 Implementation and Results 93
5.3.1 Implementation Environment 93
5.3.2 Embedding Preparation 94
5.3.3 Explainability and Response Generation 94
5.3.4 Dataset and Experimental Setup 95
5.3.5 Qualitative Results : User eXperience Study 98
5.3.6 Discussion of Results 100

6 Research Contributions, Implications and Originality 103
6.1 Contributions 104
6.1.1 Modular and Explainable CPS Recommender Architecture 104
6.1.2 Integration of Semantic Embeddings with a Curated Knowledge Base 104
6.1.3 SHAP-Based Explainability for Transparent Recommendations 105
6.1.4 User-Centered Interaction through Sufficiency Checks and Feedback 105
6.1.5 Balanced Trade-off Between Efficiency and Interpretability 105
6.1.6 Practical Domain-Specific Advantage for CPS 106
6.2 Publications & Contributions 106
6.2.1 Automated Decision Support Framework for IoT : Towards a Cyber-Physical Recommendation System," Proceedings of ICEIS 2023 107
6.2.2 Towards a Cyber-Physical Recommendation System : Automated Decision Support Framework for IoT," Journées IA-Optim 2023 107
6.2.3 IoT-AID : An Automated Decision Support Framework for IoT," ICEIS Post-Proceedings 2024 108
6.2.4 IoT Sensor Selection in Cyber-Physical Systems : Leveraging Large Language Models as Recommender Systems," Proceedings of CoDIT 2024 109
6.2.5 IoT-AID : Leveraging XAI for Conversational Recommendations in Cyber-Physical Systems," Proceedings of the ICEIS 2025 Workshop 110

6.2.6 IoT-AID : An Explainable Conversational Agent for Assisting Cyber-Physical System Configuration,Journées IA-Optim 2025 . . . 111
6.2.7 Towards an Explainable Cyber-Physical Recommender System," 2025 Proceedings. . . . 112

7 General Conclusion & Research Perspectives 115
7.1 Conclusion . . . 117
7.2 Limitations and Future Directions . . . 119
7.2.1 Current Limitations . . . 120
7.2.2 Future Directions . . . 121
7.3 Research Perspectives . . . 122
7.3.1 Early Stage : Scientific Consolidation and Publication . . . 122
7.3.2 Mid Stage : System Enhancement and Domain Specialization . . . 122
7.3.3 Late Stage : Industrial Deployment and Continual Learning . . . 123

Résumé 125
Introduction générale . . . 126
Motivation et problématique de recherche . . . 126
Approche proposée . . . 127
Objectifs et contributions de la thèse . . . 128
Organisation du manuscrit . . . 128
Chapitre 1 : Introduction aux systèmes cyber–physiques . . . 129
Chapitre 2 : Un cadre de recommandation conversationnelle contextuelle pour la conception et l'intégration des systèmes cyber–physiques . . . 131
Chapitre 3 : Cadre architectural, conception et détails d'intégration du CPRS conversationnel IoT-AID . . . 136
Chapitre 4 : Cadre d'implémentation de IoT-AID et intégration des grands modèles de langage dans le développement d'un CPRS conversationnel . . . 140
Chapitre 5 : Implémentation et évaluation du système de recommandation cyber–physique . . . 146
7.4 Contributions . . . 149
7.4.1 A Modular and Explainable Conversational CPS Recommender Architecture . . . 150

7.4.2 Semantic Retrieval Grounded in a Curated CPS Knowledge Base . . . 150
7.4.3 Integration of SHAP for Explainable CPS Recommendations . . . 150
7.4.4 User-Centered Interaction via Information Sufficiency and Feedback . . . 151
7.4.5 Efficient and Interpretable Semantic Recommendation Pipeline . . . 151
7.4.6 A Domain-Specific Conversational Framework for CPS Configuration . . . 151
Chapitre 7 : Conclusion générale et perspectives de recherche . . . 152

A Publications **155**
A.1 Journals . . . 155
A.2 Conferences . . . 155
A.3 Communications . . . 156

B Annex B **157**

Table des figures

1.1 Three–layer CPS architecture proposed by [14]. 8
1.2 Conceptual architecture illustrating interaction between cyber, physical, and communication spheres. 10
1.3 Proposed CPS connection concept adapted from [3]. 11

2.1 User–based collaborative filtering process. 25
2.2 Content–based filtering principle. 25
2.3 Hybrid recommendation architecture. 26
2.4 An architecture for a task-oriented dialog system [18]. 31

3.1 Global architecture of the proposed Conversational CPS Recommendation System. 40
3.2 Example XML file describing a humidity sensor. 41
3.3 Illustrative example : Home surveillance for elderly patients. . 42

4.1 Overall system architecture and module interaction of IoT-AID's Conversational CPRS. 52
4.2 NLP pipeline for semantic understanding and query formation in IoT-AID. 54
4.3 ETL Pipeline for Data Collection and Structuring 60
4.4 Sensors Dataset 61
4.5 Fine-tuning LLMs 62
4.6 CPRS Architecture. 65
4.7 Entities Dataset 66

5.1 General Architecture of the Cyber-Physical Recommendation System. 74

Liste des tableaux

5.1 Performance comparison of embedding systems. 96

Liste des Abbréviations

AI	**A**rtificial **I**ntelligence
API	**A**pplication **P**rogramming **I**nterface
BERT	**B**idirectional **E**ncoder **R**epresentations from **T**ransformers
CNN	**C**onvolutional **N**eural **N**etwork
CPS	**C**yber–**P**hysical **S**ystem
CPRS	**C**yber–**P**hysical **R**ecommendation **S**ystem
CSV	**C**omma–**S**eparated **V**alues
DL	**D**eep **L**earning
EDA	**E**xploratory **D**ata **A**nalysis
ECA	**E**vent–**C**ondition–**A**ction
GPT	**G**enerative **P**re–trained **T**ransformer
IDE	**I**ntegrated **D**evelopment **E**nvironment
IoT	**I**nternet **o**f **T**hings
IoT–AID	**I**nternet **o**f **T**hings – **A**ssisted **I**ntelligent **D**esign
JSON	**J**ava**S**cript **O**bject **N**otation
KB	**K**nowledge **B**ase
LLM	**L**arge **L**anguage **M**odel
LIME	**L**ocal **I**nterpretable **M**odel–agnostic **E**xplanations
LSTM	**L**ong **S**hort–**T**erm **M**emory
ML	**M**achine **L**earning
MQTT	**M**essage **Q**ueuing **T**elemetry **T**ransport
NLP	**N**atural **L**anguage **P**rocessing
NER	**N**amed **E**ntity **R**ecognition
RAG	**R**etrieval–**A**ugmented **G**eneration
RNN	**R**ecurrent **N**eural **N**etwork
SBERT	**S**entence–**B**ERT
SHAP	**SH**apley **A**dditive **exP**lanations
SVM	**S**upport **V**ector **M**achine
TF–IDF	**T**erm **F**requency–**I**nverse **D**ocument **F**requency
UI	**U**ser **I**nterface
XAI	**E**xplainable **A**rtificial **I**ntelligence

Liste des Symboles

C	Number of classes or components
L	Loss function or objective function
E	Error or energy measure
R	Regularization term or resistance (in context)
t	Time variable
τ	Time constant
λ	Regularization or weighting parameter
σ	Standard deviation or noise term
θ	Model parameter or weight vector
η	Learning rate
$\mathcal{L}$	Loss function or optimization objective
$\mathcal{D}$	Dataset
$\mathcal{M}$	Model function
$\mathbf{x}$	Input feature vector
$\mathbf{y}$	Target or output vector
$\hat{y}$	Predicted output
∇	Gradient operator
μ	Mean value
k	Number of nearest neighbors or top results
d	Dimension of feature or embedding vector
A	Accuracy metric
P	Precision metric
R_s	Recall metric
F_1	F1-score
$\cos(\theta)$	Cosine similarity measure

General Introduction

The rise of Industry 4.0 has brought Cyber-Physical Systems (CPS) to the forefront of technological advancements. By combining computational intelligence with physical processes, CPS allows for real-time monitoring, adaptive control, and autonomous decision-making in fields like manufacturing, healthcare, transportation, and energy. In these systems, physical devices such as sensors, actuators, and machines connect with cyber components that handle computation, analytics, and decision logic. They are connected through reliable, low-latency communication protocols. This close connection enables factories to quickly adjust production settings based on sensor data, hospitals to continuously monitor patient health, and transportation systems to dynamically reroute vehicles in response to congestion or hazards.

Despite their potential for change, CPSs are complicated to design and configure. A single CPS project may involve hundreds of components that interact with one another, ranging from low-power embedded sensors to high-performance cloud-based analytics services. Engineers and system designers need to choose from extensive options of hardware and software, often made by different vendors, documented in various formats, and governed by different technical standards. Each choice has to meet multiple requirements. For example, they must balance energy efficiency with data throughput or maximize fault tolerance while sticking to a strict budget. As CPS moves into critical areas, the risks of misconfiguration become more serious, increasing the need for precision, clarity, and reliability.

Motivation and Research Gap

The process of turning high-level user objectives into a precise CPS configuration remains a significant challenge. In practice, engineers often begin with informal, domain-specific descriptions such as “monitor vibration and temperature in a high-humidity environment with limited power supply.”

Translating such statements into a concrete bill of materials, a validated architecture, and a set of communication and processing specifications is labor-intensive and error-prone. Current approaches rely heavily on domain experts who must manually navigate product datasheets, reconcile heterogeneous measurement units, and verify interoperability across layers of the CPS architecture. This dependency on scarce expertise is a bottleneck for scalability.

The diversity of data sources compounds the difficulty. Component specifications are published in disparate formats : accelerometer vendors may provide calibration curves as CSV files, network hardware manufacturers often publish latency tables in XML, and sensor environmental ratings might be described only in PDF datasheets. Even within the same format, attributes may be labeled inconsistently one vendor may use “sampling rate” while another uses “data acquisition frequency” and units may differ. Without a unified schema and semantic alignment, automated retrieval and compatibility checking become infeasible.

Moreover, existing CPS design tools are often rigid, requiring users to fill predefined forms or select from fixed menus. They offer little flexibility in handling vague or evolving requirements, and they rarely provide explanations for their recommendations. This creates a gap in user trust and limits adoption, especially in safety-critical contexts where engineers must justify every design decision to regulatory authorities or stakeholders.

Proposed Approach

This thesis addresses these challenges by proposing the *Cyber-Physical Recommendation System* (CPRS), an intelligent, conversational framework designed to assist engineers in translating high-level objectives into optimized CPS configurations. The CPRS engages users through natural language dialogue, enabling them to express requirements in their own words. Through semantic parsing and domain-specific knowledge representation, the system interprets these requirements, identifies candidate components, and iteratively refines its recommendations through interactive clarification.

The CPRS is underpinned by a structured, semantically-enriched knowledge base compiled from heterogeneous sources including XML, CSV, pdf, and web-extracted metadata. Schema normalization and attribute-driven alignment ensure that component attributes are consistently represented, enabling accurate compatibility checks and multi-criteria ranking. The retrieval

process leverages semantic embeddings tuned to the CPS domain, allowing the system to match user intents with relevant components even when exact keyword matches are absent.

A key innovation is the integration of an interactive clarification loop. When the system detects ambiguity or missing constraints for example, an unspecified environmental tolerance or communication range it prompts the user with targeted questions. This conversational refinement process minimizes the cognitive load on the user while ensuring that the final configuration meets both functional and contextual requirements.

To enhance transparency and user trust, the CPRS incorporates explainable AI techniques. Recommendations are accompanied by justifications that identify the most influential parameters, constraints, or past user preferences that shaped the suggestion. By surfacing this reasoning, the system enables engineers to validate decisions quickly and defend them in documentation or regulatory audits.

Research Objectives and Contributions

The overarching objective of this research is to reduce CPS configuration time, improve recommendation relevance, and increase user confidence through transparency. The specific goals are :

1. To develop a unified, semantically-enriched knowledge representation for CPS components that supports compatibility checking and semantic retrieval.
2. To implement a natural language interface that translates free-form requirements into precise, machine-interpretable constraints.
3. To design an interactive clarification mechanism that handles incomplete or ambiguous specifications efficiently.
4. To integrate explainable recommendation capabilities that clearly articulate the rationale behind each configuration decision.
5. To evaluate the system across representative CPS design scenarios, measuring performance in terms of efficiency, accuracy, and user satisfaction.

The main contributions of this thesis include the modular architecture of the CPRS, the construction of a curated and semantically-annotated CPS

component knowledge base, and the implementation of an interactive, explainable recommendation workflow tailored to the CPS domain. Experimental evaluation on realistic configuration tasks demonstrates the system's ability to streamline the design process and deliver relevant, trustworthy recommendations.

Thesis Structure

Chapter 1 introduces the foundations of Cyber-Physical Systems (CPS), their components, architectures, application domains, and key challenges. Chapter 2 presents the principles of recommendation systems and conversational recommendation frameworks, motivating their relevance for CPS configuration. Chapter 3 describes the architectural design of the proposed IoT-AID conversational recommendation system, including its components and integration flow. Chapter 4 details the implementation framework of IoT-AID, with a focus on the NLP pipeline, semantic embeddings, and large language model integration. Chapter 5 reports the full system implementation and presents both quantitative experiments and qualitative user studies used to evaluate the prototype. Chapter 6 consolidates the research contributions and discusses their scientific and practical implications. Chapter 7 concludes the thesis and outlines the main limitations and future research directions.

Chapitre 1

Introduction to Cyber Physical Systems : Understanding Complexities and Addressing Systemic Challenges

The emergence of *Industry 4.0* has fundamentally transformed the industrial landscape through the convergence of digital technologies, intelligent automation, and ubiquitous connectivity. This revolution is driven by the integration of the Internet of Things (IoT), Artificial Intelligence (AI), cloud computing, and data analytics into traditional production systems. At the core of this transformation lie Cyber–Physical Systems (CPS) complex, intelligent, and interconnected systems that tightly couple computational and physical processes through communication networks.

CPS enable real–time monitoring, autonomous control, and adaptive decision–making across a broad range of application domains, from manufacturing and healthcare to transportation and energy. By bridging the gap between the physical and digital worlds, CPS provide the technological foundation for smart factories, intelligent infrastructure, and data-driven services. Their defining characteristic is the continuous feedback loop between sensing, computation, and actuation, where data collected from the physical environment are processed in the cyber domain and translated into control actions that influence the physical processes in real time.

Despite their immense potential, designing and deploying CPS remain challenging. These systems are inherently multidisciplinary, combining mechanical, electrical, computational, and communication components that must interact seamlessly under strict performance and safety constraints. The diversity of hardware and software elements, the need for reliable real-time communication, and the growing scale of data exchange introduce significant

complexity. Furthermore, many industrial stakeholders and domain specialists lack the expertise required to configure CPS components efficiently or to optimize them for specific operational objectives. This creates a pressing need for tools and frameworks that can simplify CPS design, ensure interoperability, and support informed decision-making through intelligent assistance.

This chapter provides a foundational understanding of Cyber–Physical Systems, emphasizing their conceptual framework, architectural models, and application scope. It begins with an overview of CPS principles and components, highlighting the integration between cyber, physical, and communication layers. The subsequent sections examine representative CPS architectures, discuss key application domains such as manufacturing, healthcare, and transportation, and analyze emerging trends such as AI-enabled CPS, human–CPS collaboration, and ethical design considerations. The chapter concludes with a discussion on the complexities and systemic challenges of CPS deployment, setting the stage for the introduction of the proposed *Cyber–Physical Recommendation System* (CPRS) in later chapters.

The remainder of this chapter is organized as follows. Section 1.1 introduces the fundamental concepts and motivations behind CPS. Section 1.2 details their core components and the interaction among physical, cyber, and network spheres. Section 1.3 explores major CPS architectural models and design patterns. Section 1.4 reviews their application domains, while Section 1.5 discusses emerging trends, key challenges, and the motivation leading to the development of a recommendation-based approach for CPS configuration. The Section 1.6 concludes the contents of this chapter.

1.1 Introduction to Cyber–Physical Systems

Cyber–Physical Systems (CPS) represent the convergence of computation, communication, and physical processes into tightly integrated, feedback–driven systems. Through embedded sensors, actuators, and control algorithms, CPS enable real–time interaction between the physical environment and computational intelligence. As a result, CPS underpin a wide range of modern industrial applications from manufacturing and transportation to healthcare and smart cities serving as a core technological pillar of the Industry 4.0 paradigm[24, 12].

Contemporary CPS deployments combine smart sensors, embedded devices, cloud computing, and artificial intelligence (AI) to enable predictive maintenance, adaptive process control, and autonomous self–optimization [7,

5]. In the emerging transition toward Industry 5.0, these systems are increasingly augmented by web technologies and AI–driven analytics that collaborate with human decision–makers, emphasizing human–centric and sustainable design principles. In such environments, high–quality and context–rich data become the most valuable resource, driving intelligent, self–adapting system behavior[22].

Despite these advances, CPS design and deployment remain inherently complex. Each application domain requires the careful integration of heterogeneous components, communication protocols, and computational resources under stringent performance and interoperability constraints. The process often demands multidisciplinary expertise from embedded systems engineering and networking to data analytics that may not be available to all domain specialists. Consequently, system configuration is frequently manual, error–prone, and time–consuming [6].

To mitigate these challenges, automation and intelligent assistance are increasingly sought to guide the design and configuration process. The Cyber–Physical Recommendation System (CPRS) proposed in this thesis aims to address this gap by offering a data–driven, context–aware, and explainable approach for CPS component selection and configuration. By leveraging conversational interaction and semantic reasoning, CPRS seeks to lower the cognitive and technical barriers associated with CPS design.

Although the core CPS concept remains consistent, the literature presents several architectural models tailored to different domains. For example, Jamwal *et al.* [14] propose a three–layer structure for intelligent manufacturing systems, as illustrated in Figure 1.1.

1.2 Core Components of Cyber–Physical Systems

Cyber–Physical Systems are composed of several fundamental layers and functional components that operate in coordination to enable perception, computation, communication, and actuation. These components are generally categorized as follows :

- **Physical Components :** These include sensors and actuators that interface with the real world. Sensors capture environmental data that feed into computational models, while actuators execute control commands to achieve desired physical effects.

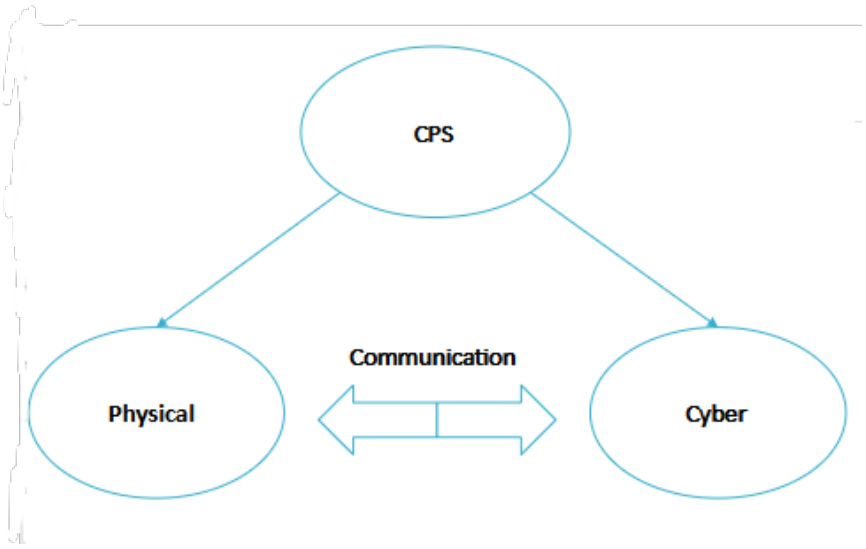


FIGURE 1.1 – Three–layer CPS architecture proposed by [14].

- **Cyber Components :** The cyber layer consists of computational modules, algorithms, and control logic responsible for data processing, decision–making, and system coordination. Reliable networking and data management mechanisms ensure low–latency and high–availability operation[32].
- **Communication Layer :** This layer enables secure and reliable data transfer between cyber and physical entities, typically employing Internet of Things (IoT) standards and Wireless Sensor Networks (WSNs) to ensure interoperability and robustness[9].
- **Computing Architecture :** Modern CPS often distribute computational tasks across edge, fog, and cloud layers to achieve a balance between real–time responsiveness and scalability. Edge devices handle time–critical tasks, while cloud services support long–term analytics and storage[21, 20, 17].
- **Security and Privacy :** Ensuring data integrity, confidentiality, and protection against cyber–attacks is crucial for operational safety and trust. Although this thesis focuses primarily on data handling and recommendation processes, these security mechanisms remain essential in any CPS ecosystem[16, 31].

To better understand how these components interconnect, CPS can be conceptually divided into three interacting spheres :

1. **Physical Sphere :** Encompasses sensing, execution, and control units. It interacts directly with the environment through sensors, actuators, and microcontrollers, and may include a human–in–the–loop component to

enable manual intervention when necessary.

2. **Network Sphere :** Manages communication among devices and layers, addressing issues such as data transport, network security, and interoperability. Technologies like Wi-Fi, Bluetooth, 5G, and Zigbee facilitate reliable, secure connectivity.
3. **Cyber Sphere :** Focuses on data processing, storage, and decision–making. It includes front–end modules for visualization, back–end modules for computation, and servers or databases that manage data persistence and exchange.

These spheres collectively enable CPS to sense, reason, and act upon their environment, forming a closed feedback loop that is both autonomous and adaptive.

1.3 Architectural Models and Design Patterns

The design of a Cyber–Physical System requires a structured approach that integrates physical engineering with computational intelligence. This section reviews representative architectural frameworks and design methodologies that inform CPS development.

1.3.1 Modeling and Simulation

Modeling and simulation are critical stages of the CPS lifecycle, offering a safe and cost–effective means to validate system behavior before physical deployment. Digital twins virtual replicas of physical assets enable continuous monitoring, fault prediction, and performance optimization. Co–simulation environments allow multiple heterogeneous subsystems to be evaluated jointly, ensuring overall interoperability and reliability.

1.3.2 System Design and Engineering

Successful CPS design requires the convergence of systems engineering and software engineering. Systems engineering governs requirements, architecture specification, and lifecycle management, while software engineering implements the algorithms and interfaces necessary for operation. This multidisciplinary collaboration ensures that both cyber and physical subsystems align with functional, safety, and performance objectives[**marwedel2021embeddedm**, 32].

1.3.3 Development Frameworks

Modern toolchains for CPS support integrated workflows spanning modeling, implementation, testing, and deployment. Frameworks such as ROS, MATLAB/Simulink, and OPC UA enable cross–platform integration of hardware and software components. Continuous integration and automated testing environments further ensure robustness under real–time constraints[27, 1].

1.3.4 Representative Architectures

CPS are large–scale, networked systems characterized by continuous feedback between physical and computational processes [26]. The conceptual architecture in Figure 1.2 illustrates this interaction. Alternative frameworks

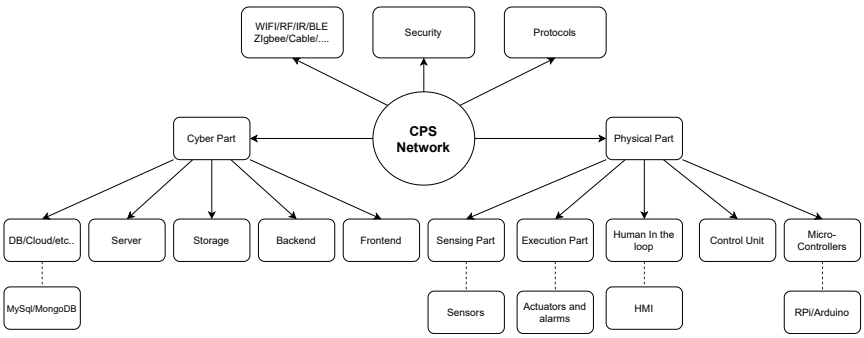


FIGURE 1.2 – Conceptual architecture illustrating interaction between cyber, physical, and communication spheres.

include :

- **Four–Level CPS Model** (Yan *et al.*, 2019) : comprising sensing & control, connection, cognition, and application & services layers.
- **Five–Level Architecture** (Lee *et al.*, 2015) : introducing smart connection, data–to–information conversion, cyber operations, cognition, and configuration layers.
- **Domain–Specific Model** (Djouad *et al.*, 2018) : distinguishing physical, cyber–physical, cybernetic, data & information, and deployment spheres.

Building on these foundations, the proposed architecture in this thesis (Figure 1.3) generalizes the CPS connection concept for application in recommendation–based configuration. It integrates the human–machine interface, sensing, control, computation, and communication modules within a unified framework capable of supporting intelligent component recommendation.

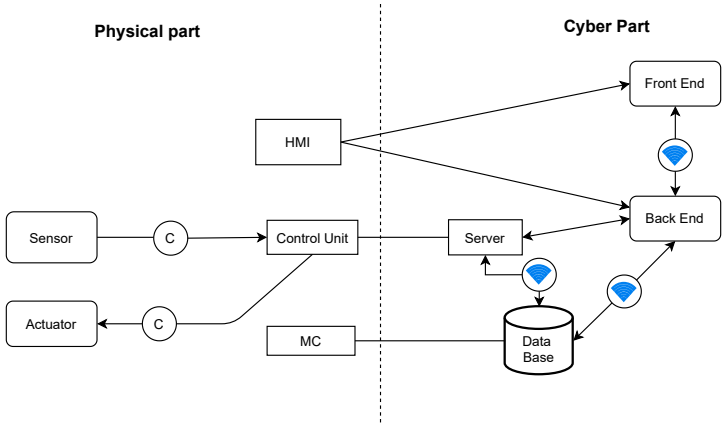


FIGURE 1.3 – Proposed CPS connection concept adapted from [3].

1.4 Application Domains of Cyber–Physical Systems

Cyber–Physical Systems (CPS) play a pivotal role in the ongoing digital transformation of industry and society. Acting as the technological foundation of the fourth industrial revolution, CPS enable intelligent, adaptive, and self–optimizing systems across a broad range of sectors [18]. Their inherent flexibility allows seamless integration into diverse domains such as healthcare, manufacturing, and transportation each presenting unique requirements and challenges that shape CPS design and deployment.

1.4.1 Healthcare

In the healthcare domain, CPS applications span hospital management, intensive care monitoring, surgical assistance, and ambient assisted living for elderly or chronically ill patients[28]. These systems rely on high–fidelity sensing, continuous data acquisition, and real–time decision–making, often supported by cloud computing and big data analytics. Data sources include electronic health records, clinician inputs, and biosensor or wearable device streams[10].

Banerjee *et al.* [2] proposed a CPS framework for healthcare supporting functionalities such as drug concentration prediction in infusion pumps and multi–infusion control for chemotherapy. Similarly, Haque *et al.* [11] emphasized that CPS in medical contexts must guarantee reliability, safety, and privacy, while efficiently managing heterogeneous and high–velocity medical

data. By integrating intelligent sensing and AI–assisted analytics, CPS enhance diagnostic precision, patient safety, and remote monitoring capabilities, thereby contributing to personalized and preventive healthcare models.

1.4.2 Manufacturing

In manufacturing, CPS underpin the evolution from traditional automation systems toward smart factories characterized by connectivity, adaptability, and resilience. They serve as a cornerstone of the Industry 4.0 paradigm, enabling flexible production lines, energy–efficient operation, and real–time performance optimization.

Lee *et al.* [19] introduced a predictive production framework that improves system interoperability and fault resilience, while Wan *et al.* [30] demonstrated how ontology–based big data analytics enhance collaborative decision–making and sustainability. The integration of CPS within industrial environments supports predictive maintenance, dynamic resource allocation, and autonomous quality control [7]. These innovations collectively drive cost reduction, productivity improvement, and environmental sustainability key pillars of next–generation manufacturing.

1.4.3 Transportation

Transportation systems face complex challenges related to safety, congestion, energy efficiency, and environmental impact. CPS offer transformative solutions through the integration of smart sensors, communication technologies, and real–time analytics. Vehicle–to–vehicle (V2V) and vehicle–to–infrastructure (V2I) communication enable intelligent coordination and predictive control[23].

Representative applications include cooperative vehicle safety systems [4], delay–aware intersection management [33], and optimized charging control for electric vehicles [8]. Beyond road transport, CPS concepts extend to rail, maritime, and aerospace systems, where they support predictive maintenance, trajectory optimization, and enhanced safety assurance. These innovations illustrate the capacity of CPS to create resilient, efficient, and sustainable transportation infrastructures.

1.5 Emerging Trends and Challenges

While Cyber–Physical Systems have achieved significant maturity across multiple domains, emerging technological and societal trends continue to

shape their evolution. This section highlights major directions and challenges influencing current and future CPS research and implementation.

1.5.1 Artificial Intelligence and Machine Learning in CPS

Artificial Intelligence (AI) and Machine Learning (ML) are increasingly embedded within CPS to process complex data streams, detect anomalies, and support autonomous decision–making[25]. Their applications include predictive maintenance, fault diagnosis, adaptive process control, and operational optimization. For example, AI models can anticipate equipment failures before they occur, enabling proactive maintenance scheduling and minimizing downtime.

However, the integration of AI into CPS raises several issues related to robustness, explainability, and certification. In mission–critical contexts, users must trust the reasoning behind AI–generated recommendations. Consequently, explainable AI (XAI) techniques are becoming essential to provide transparency and accountability in CPS decision processes[13].

1.5.2 Human–CPS Interaction

As CPS become increasingly autonomous, human–system interaction remains a vital design consideration. The Human–in–the–Loop paradigm emphasizes collaboration between automated intelligence and human oversight, ensuring safety, flexibility, and contextual adaptability. Effective interfaces must offer both intuitiveness and fine–grained control, especially in safety–critical domains such as healthcare, manufacturing, and transport. Balancing automation and human judgment presents an ongoing challenge, particularly as CPS evolve toward shared decision–making and cognitive collaboration.

1.5.3 Regulatory and Ethical Considerations

The proliferation of autonomous and AI–enabled CPS introduces pressing regulatory and ethical concerns. Issues of data governance, privacy, accountability, and algorithmic bias must be addressed to maintain public trust and operational integrity[29]. Regulatory frameworks should guarantee safety, security, and fairness, while supporting continuous innovation. Ethical design principles especially in systems influencing human welfare must be embedded at every stage of CPS development and deployment[15].

1.5.4 From Challenges to CPS Implementation

Implementing CPS in real–world scenarios remains complex due to the diversity of physical devices, communication protocols, and computational architectures involved. Each domain imposes unique performance, safety, and interoperability requirements, and stakeholders often lack the cross–disciplinary expertise to configure systems effectively [6]. These factors contribute to longer development cycles, higher costs, and greater risk of integration errors.

Key barriers to efficient CPS deployment include :

- **Semantic interoperability :** Aligning heterogeneous metadata and component descriptions to support automated discovery and composition.
- **Automated rule generation :** Translating informal requirements into executable Event–Condition–Action (ECA) rules without extensive manual intervention.
- **Validation and verification :** Ensuring correct behavior across interconnected cyber and physical subsystems through formal modeling and rigorous testing.
- **Accessible explainability :** Providing human–understandable explanations of system behavior and configuration decisions.

Addressing these challenges demands an interdisciplinary approach that integrates model–driven engineering, semantic technologies, and human–centered design methodologies.

1.5.5 Motivation for a Cyber–Physical Recommendation System

In response to these limitations, this thesis introduces a *Cyber–Physical Recommendation System* (CPRS) designed to assist engineers and researchers in the configuration and deployment of CPS. The CPRS employs an AI–driven conversational interface to capture user requirements, query a structured knowledge base, and recommend contextually appropriate components and configurations. By automating knowledge retrieval and ensuring decision transparency, CPRS reduces reliance on scarce domain expertise, accelerates system design, and improves overall quality of service.

Ultimately, this approach supports the transition toward **Industry 5.0**, where human–centric collaboration between intelligent systems and engineers fosters sustainable, adaptive, and trustworthy industrial ecosystems.

1.6 Conclusion

Designing, selecting, and configuring a Cyber–Physical System (CPS) is inherently complex due to the heterogeneity of its components, the diversity of communication technologies, and the multidisciplinary expertise required for seamless integration. As highlighted throughout this chapter, CPS form the technological backbone of the fourth industrial revolution, enabling intelligent automation, adaptive control, and real-time decision-making across sectors such as manufacturing, healthcare, and transportation.

While advances in IoT connectivity, cloud and edge computing, and artificial intelligence have dramatically expanded the functional capabilities of CPS, they have also introduced new challenges of interoperability, scalability, and explainability. The coexistence of heterogeneous data formats, component models, and operational requirements often complicates system configuration and hinders efficient deployment. Consequently, there is a growing need for intelligent support tools that can reduce this complexity, assist users in selecting compatible components, and ensure that system configurations remain consistent with operational goals and safety constraints.

In this context, the conversational recommendation systems emerge as a promising solution. By combining the reasoning capabilities of AI with the accessibility of natural-language interaction, CRS can guide engineers and non-experts alike through the intricate CPS design process. Such systems enable iterative dialogue, contextual clarification, and personalized recommendation, thus bridging the gap between user intent and technical implementation. When enhanced with explainable AI techniques, they further promote transparency and trust in automated decision-support systems.

The insights developed in this chapter establish the conceptual and technical foundations for the work presented in this thesis. The next chapter introduces the principles, architectures, and enabling technologies of conversational recommendation systems, with a specific focus on their adaptation for CPS configuration. This discussion serves as a foundation for the proposed *Cyber–Physical Recommendation System* (CPRS), which aims to deliver intelligent, explainable, and user-centric assistance for CPS design and deployment.

Bibliographie

[1] Tim ABDIUKOV. « Digital twin vulnerabilities in industrial cyber-physical systems : A security framework for threat simulation and containment ». In : *Well Testing Journal* 34.S3 (2025), p. 190-205.

[2] Ayan BANERJEE et al. « Towards modeling and analysis of cyber-physical medical systems ». In : *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*. 2011, p. 1-5.

[3] Mohammad CHOAIB. et al. « Automated Decision Support Framework for IoT : Towards a Cyber Physical Recommendation System ». In : *Proceedings of the 25th International Conference on Enterprise Information Systems - Volume 1 : ICEIS*. INSTICC. SciTePress, 2023, p. 365-373. ISBN : 978-989-758-648-4. DOI : 10.5220/0011848900003467.

[4] Yaser P. FALLAH et al. « Design of cooperative vehicle safety systems based on tight coupling of communication, computing and physical vehicle dynamics ». In : *Proceedings of the 1st ACM/IEEE international conference on cyber-physical systems*. 2010, p. 159-167.

[5] Moncef GAROUANI et al. « AMLBID : An auto-explained Automated Machine Learning tool for Big Industrial Data ». In : *SoftwareX* 17 (2022), p. 100919. DOI : 10.1016/j.softx.2021.100919.

[6] Moncef GAROUANI et al. « Towards meta-learning based data analytics to better assist the domain experts in industry 4.0 ». In : *Artificial Intelligence in Data and Big Data Processing*. Cham : Springer International Publishing, 2022, p. 59-66. DOI : 10.1007/978-3-030-97610-1_22.

[7] Moncef GAROUANI et al. « Using meta-learning for automated algorithms selection and configuration : an experimental framework for industrial big data ». In : *Journal of Big Data* 9 (avr. 2022), 1169–1188. DOI : 10.1186/s40537-022-00612-4.

[8] Yongqi GE, Yunwei DONG et Hongbing ZHAO. « A cyber-physical energy system architecture for electric vehicles charging application ». In : *2012 12th International Conference on Quality Software*. IEEE, 2012, p. 246-250.

[9] Arda GOKNIL et al. « A systematic review of data quality in CPS and IoT for industry 4.0 ». In : *ACM Computing Surveys* 55.14s (2023), p. 1-38.

[10] Aditya GUPTA et Amritpal SINGH. « A comprehensive survey on cyber-physical systems towards healthcare 4.0 ». In : *SN Computer Science* 4.2 (2023), p. 199.

[11] Shah Ahsanul HAQUE, Syed Mahfuzul AZIZ et Mustafizur RAHMAN. « Review of cyber-physical system in healthcare ». In : *international journal of distributed sensor networks* 10.4 (2014). Publisher : SAGE Publications Sage UK : London, England, p. 217415.

[12] Muneeb Ul HASSAN, Mubashir Husain REHMANI et Jinjun CHEN. « Differential privacy techniques for cyber physical systems : A survey ». In : *IEEE Communications Surveys & Tutorials* 22.1 (2019), p. 746-789.

[13] Amber HOENIG et al. « Explainable AI for cyber-physical systems : Issues and challenges ». In : *IEEE access* 12 (2024), p. 73113-73140.

[14] Anbesh JAMWAL et al. « Development of cyber physical system based manufacturing system design for process optimization ». In : *IOP Conference Series : Materials Science and Engineering* 997.1 (2020), p. 012048. ISSN : 1757-8981, 1757-899X. DOI : 10.1088/1757-899X/997/1/012048. URL : https://iopscience.iop.org/article/10.1088/1757-899X/997/1/012048 (visité le 21/10/2022).

[15] Pramod P KHARGONEKAR et Meera SAMPATH. « A framework for ethics in cyber-physical-human systems ». In : *IFAC-PapersOnLine* 53.2 (2020), p. 17008-17015.

[16] Sangjun KIM, Kyung-Joon PARK et Chenyang LU. « A survey on network security for cyber–physical systems : From threats to resilient design ». In : *IEEE Communications Surveys & Tutorials* 24.3 (2022), p. 1534-1573.

[17] Linghe KONG et al. « Edge-computing-driven internet of things : A survey ». In : *ACM Computing Surveys* 55.8 (2022), p. 1-41.

[18] Jay LEE, Behrad BAGHERI et Hung-An KAO. « A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems ». In : *Manufacturing Letters* 3 (2015), p. 18-23. ISSN : 22138463. DOI : 10.1016/j.mfglet.2014.12.001. URL : https://linkinghub.elsevier.com/retrieve/pii/S221384631400025X (visité le 21/10/2022).

[19] Jay LEE, Chao JIN et Behrad BAGHERI. « Cyber physical systems for predictive production systems ». In : *Production Engineering* 11 (2017). DOI : 10.1007/s11740-017-0729-4.

[20] Jiewu LENG et al. « Digital twins-based smart manufacturing system design in Industry 4.0 : A review ». In : *Journal of manufacturing systems* 60 (2021), p. 119-137.

[21] Mengnan LIU et al. « Review of digital twin about concepts, technologies, and industrial applications ». In : *Journal of manufacturing systems* 58 (2021), p. 346-361.

[22] Yuan LUO et al. « Deep learning-based anomaly detection in cyber-physical systems : Progress and opportunities ». In : *ACM Computing Surveys (CSUR)* 54.5 (2021), p. 1-36.

[23] Md NOOR-A-RAHIM et al. « 6G for vehicle-to-everything (V2X) communications : Enabling technologies, challenges, and opportunities ». In : *Proceedings of the IEEE* 110.6 (2022), p. 712-734.

[24] Sascha Julian OKS et al. « Cyber-physical systems in the context of industry 4.0 : A review, categorization and outlook ». In : *Information Systems Frontiers* 26.5 (2024), p. 1731-1772.

[25] Felix O OLOWONONI, Danda B RAWAT et Chunmei LIU. « Resilient machine learning for networked cyber physical systems : A survey for machine learning security to securing machine learning for CPS ». In : *IEEE Communications Surveys & Tutorials* 23.1 (2020), p. 524-552.

[26] Ragunathan (Raj) RAJKUMAR et al. « Cyber-physical systems : the next computing revolution ». In : *Proceedings of the 47th Design Automation Conference on - DAC '10*. the 47th Design Automation Conference. Anaheim, California : ACM Press, 2010, p. 731. ISBN : 978-1-4503-0002-5. DOI : 10.1145/1837274.1837461. URL : http://portal.acm.org/citation.cfm?doid=1837274.1837461.

[27] Kelly RIJSWIJK et al. « Digital transformation of agriculture and rural areas : A socio-cyber-physical system framework to support responsibilisation ». In : *Journal of Rural Studies* 85 (2021), p. 79-90.

[28] Michael SONY, Jiju ANTONY et Olivia MCDERMOTT. « The impact of medical cyber–physical systems on healthcare service delivery ». In : *The TQM Journal* 34.7 (2022), p. 73-93.

[29] Abhilash THEKKILAKATTIL et Gordana DODIG-CRNKOVIC. « Ethics aspects of embedded and cyber-physical systems ». In : *2015 IEEE 39th annual computer software and applications conference*. T. 2. IEEE. 2015, p. 39-44.

[30] Jiafu WAN et al. « An ontology-based resource reconfiguration method for manufacturing cyber-physical systems ». In : *IEEE/ASME Transactions on Mechatronics* 23.6 (2018). Publisher : IEEE, p. 2537-2546.

[31] Zhenhua YU et al. « A survey on cyber–physical systems security ». In : *IEEE Internet of Things Journal* 10.24 (2023), p. 21670-21686.

[32] Kunwu ZHANG et al. « Advancements in industrial cyber-physical systems : An overview and perspectives ». In : *IEEE Transactions on Industrial Informatics* 19.1 (2022), p. 716-729.

[33] Bowen ZHENG et al. « Delay-aware design, analysis and verification of intelligent intersection management ». In : *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 2017, p. 1-8.

Chapitre 2

A Context-Aware Conversational Recommendation Framework for Cyber-Physical Systems Design and Integration

The rapid transformation driven by Industry 4.0 has positioned Cyber–Physical Systems (CPSs) at the core of modern industrial innovation. As discussed in Section 1.4 of Chapter 1, CPS integration enables intelligent, autonomous, and adaptive processes across manufacturing, healthcare, transportation, and other sectors. However, designing and deploying CPS solutions remains a complex and resource-intensive endeavor. Engineers and researchers must select heterogeneous components, configure communication and computation layers, and ensure compliance with operational and interoperability constraints. These tasks often exceed the expertise of many domain specialists, who may be proficient in process engineering but lack advanced knowledge of embedded software, networking, or artificial intelligence integration.

To address these challenges, this chapter explores the role of conversational recommendation systems in supporting CPS configuration and deployment. It introduces IoT–AID, a proposed Cyber–Physical Recommendation System (CPRS) that functions as a dialogue-driven assistant for engineers and researchers. IoT–AID combines recent advances in Large Language Models (LLMs) with semantic retrieval techniques such as BERT and Sentence-BERT (SBERT) to provide context-aware, user-specific guidance. Its objective is to reduce the cognitive and technical burden of CPS design by enabling natural language interaction, iterative preference elicitation, and data-driven component selection.

IoT–AID represents a significant evolution in recommendation technology shifting from static, one-time suggestions to interactive, iterative decision-support systems. Unlike traditional recommendation systems, which rely primarily on historical logs, explicit ratings, or purchase histories, a conversational recommendation system engages users in real-time dialogue to refine recommendations dynamically. This allows it to resolve ambiguity, adapt to evolving requirements, and deliver highly tailored solutions. In the literature, conversational recommendation systems are typically implemented by integrating a recommendation engine with a dialogue manager, combining filtering techniques (content-based, collaborative, or hybrid) with dialogue management strategies (rule-based or machine-learning–driven) [11]. While such systems have been widely studied in domains such as e-commerce, media streaming, and education [9], their application to technical and engineering contexts, particularly those involving CPS remains largely less explored.

Within the CPS domain, a conversational recommendation system such as IoT–AID can support the end-to-end configuration workflow. Through structured, adaptive dialogue, it can :

- interpret user objectives and operational constraints,
- recommend compatible physical components (sensors, actuators, controllers),
- suggest optimal network and computation architectures, and
- generate a deployable blueprint for integration and implementation.

This user-centric, explainable, and adaptive process accelerates CPS prototyping, reduces costly configuration errors, and minimizes mismatches between system design and operational requirements [6].

The rest of the chapter is organized as follows. Section,2.1 reviews the fundamentals of recommendation systems, tracing their historical evolution from early heuristic-based and collaborative filtering models to modern, AI-driven frameworks that leverage deep learning and natural language understanding. This section establishes the theoretical and methodological foundation necessary to understand how recommendation mechanisms can be adapted for complex, knowledge-intensive domains such as CPS configuration. Section,2.2 then presents the motivation and research problem underlying this work. It highlights the limitations of conventional CPS configuration approaches, the challenges engineers face in navigating heterogeneous design spaces, and the emerging potential of conversational AI to bridge this gap through interactive, explainable, and context-aware guidance. Following

this, Section,2.3 introduces the core concepts and design principles of conversational recommendation systems, emphasizing the mechanisms that enable them to interpret user intent, manage dialogue, and iteratively refine recommendations. This section provides the conceptual linkage between traditional recommendation system paradigms and the requirements of CPS design support, setting the base for the system-level integration proposed in this work. Finally, Section 2.4 synthesizes the chapter's insights and positions IoT–AID as a practical, extensible framework for intelligent CPS configuration assistance.

2.1 Fundamentals of Recommendation Systems

Recommendation systems have become an indispensable element in modern digital ecosystems, guiding users through vast amounts of available options by predicting items, services, or configurations that match their preferences. Their role extends beyond consumer–oriented platforms to technical domains, where they can assist in designing complex systems such as CPS by reducing information overload and improving decision quality.

Historically, recommendation systems emerged in the early 1990s with the advent of *collaborative filtering* methods, which exploited patterns in user–item interactions to predict preferences [1]. As Web 3.0 technologies advanced and digital content exploded, recommendation systems evolved to incorporate *content–based* and *hybrid* approaches, leveraging both user behavior and item attributes [13]. Today, the proliferation of big data, IoT, and AI has enabled recommendation systems to operate in real time, incorporate multi–modal data, and adapt dynamically to changing contexts.

2.1.1 Core Objectives and Motivation

The primary goal of a recommendation system is to provide *personalized*, *relevant*, and *timely* suggestions. For industrial and engineering applications, this means guiding the selection of components, tools, or configurations that best align with operational constraints and performance targets. By filtering out irrelevant options and ranking those most likely to meet user needs, recommendation systems help reduce decision latency and improve system alignment with project goals.

In the CPS context, the *recommendation task* may involve identifying compatible sensors, actuators, and communication protocols given environmental conditions, budgetary constraints, or reliability requirements. Unlike consumer products, these recommendations often must satisfy strict interoperability and safety constraints, increasing the importance of accurate and explainable suggestions.

2.1.2 Architectural Components

Mostly the recommendation system implementations share a common two–layer structure, such as :

1. **Recommendation Engine :** Responsible for generating ranked item lists based on user data, domain knowledge, or both. This can be implemented via collaborative filtering, content–based filtering, hybrid methods, or AI–driven techniques such as deep learning.
2. **Dialogue or Interaction Manager :** In conversational recommendation systems, this component manages the conversational flow, gathers clarifying inputs, and ensures that recommendation queries remain contextually relevant. Depending on the approach, this can be rule–based, supervised–learning–driven, or implemented with reinforcement learning for adaptive decision–making [17].

2.1.3 Principal Recommendation Techniques

Recommendation systems rely on diverse algorithmic strategies to infer user preferences and generate relevant suggestions. Over the past two decades, several core paradigms have emerged, each leveraging different sources of information and modeling assumptions to improve accuracy and personalization. Among these, *Collaborative Filtering*, *Content–Based Filtering*, and *Hybrid Approaches* represent the foundational pillars of modern recommendation systems research and practice. These methods differ in how they interpret user behavior, item characteristics, and contextual signals ranging from collective user patterns to intrinsic item features or a combination of both. Understanding their mechanisms and limitations is crucial for selecting or designing a suitable recommendation strategy, particularly in specialized domains such as Cyber–Physical Systems, where both technical specifications and user intent must be simultaneously considered.

2.1.3.1 Collaborative Filtering (CF)

CF is one of the most widely used recommendation systems techniques, operating under the principle that users with similar historical preferences are likely to prefer similar items [16]. Two major variants of CF exist :

- *User–based CF :* Matches a target user to others with similar rating histories, then recommends items liked by those similar users [10].
- *Item–based CF :* Identifies items similar to those the target user already prefers, based on co–occurrence patterns [15].

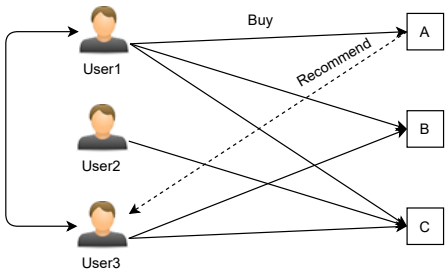


FIGURE 2.1 – User–based collaborative filtering process.

CF is effective when rich interaction histories exist, but suffers from the *cold start* problem for new users/items and from sparsity in large datasets.

2.1.3.2 Content–Based Filtering (CBF)

CBF relies on matching item attributes to the user's preference profile [12]. For CPS applications, this could mean selecting sensors based on environmental range, precision, or communication interface that align with a project's constraints. CBF can recommend new items without relying on other

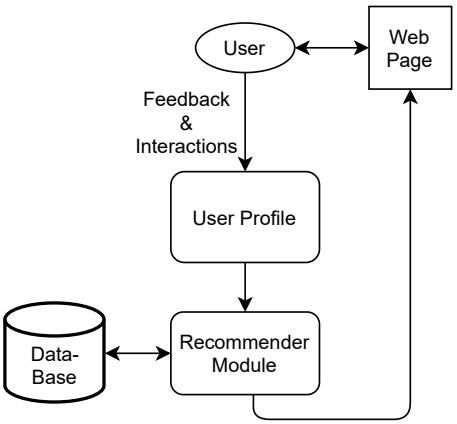


FIGURE 2.2 – Content–based filtering principle.

users' data but may risk *over–specialization*, offering limited diversity in suggestions.

2.1.3.3 Hybrid Approaches

Hybrid systems combine CF and CBF to exploit their complementary strengths [14]. The techniques of among hybrid approaches include the following :

- *Weighted hybrid :* Merges CF and CBF scores into a composite ranking.
- *Switching hybrid :* Selects CF or CBF based on data availability and quality.
- *Mixed hybrid :* Presents results from both approaches simultaneously.

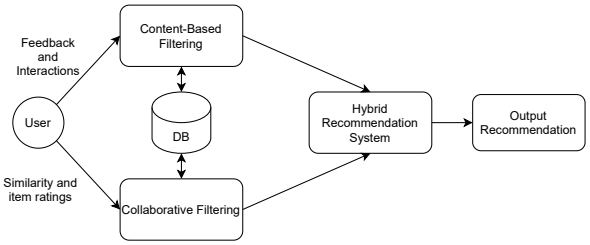


FIGURE 2.3 – Hybrid recommendation architecture.

Hybrid methods mitigate cold–start and sparsity issues but add computational complexity and integration overhead.

2.1.3.4 Knowledge–Based Approaches

Knowledge–based recommendation systems leverage explicit domain knowledge such as expert rules, ontologies, and constraint models to identify suitable items that satisfy user requirements [4]. Unlike data–driven methods, these systems do not depend on historical user interactions but instead rely on codified expertise and logical inference. In the context of CPS design, knowledge–based recommendation systems can embed technical compatibility constraints, such as voltage ranges, communication interfaces, or environmental tolerances, directly into the recommendation logic. This approach is particularly advantageous in engineering and industrial settings where data sparsity is common and precision is critical [8]. However, their effectiveness depends on the completeness and quality of the underlying knowledge base, which often requires significant domain expertise to construct and maintain.

2.1.3.5 Deep Learning–Driven Recommendation Systems

Recent advances in deep learning have redefined the landscape of recommendation systems by enabling highly expressive, nonlinear representations of user and item relationships [3]. Techniques such as neural collaborative filtering, recurrent neural networks for sequential behavior modeling, and multimodal embeddings have shown remarkable performance improvements across domains. In CPS configuration, these methods can unify heterogeneous data sources ranging from structured specifications and performance logs to unstructured textual content such as manuals or maintenance reports [7]. Despite their potential, challenges persist in ensuring interpretability, handling limited training data, and mitigating algorithmic bias, all of which are crucial considerations for mission–critical CPS applications.

2.1.4 Relevance to CPS Configuration

Although most recommendation techniques originated in consumer–centric domains such as e–commerce or media streaming, their core principles can be effectively adapted to the configuration of Cyber–Physical Systems. Collaborative filtering analogies can identify CPS projects with similar design patterns or operational requirements; content–based methods can prioritize hardware or software components based on technical features; and hybrid approaches can merge empirical performance data with expert rules to improve system reliability. Moreover, deep learning models enable the integration of multimodal CPS data linking numerical, textual, and contextual information for richer insights. When embedded within a conversational recommendation system, these techniques gain further utility, allowing users to express requirements in natural language, receive adaptive guidance, and iteratively refine configurations through interactive dialogue. This convergence marks a pivotal step toward explainable, human–centric CPS configuration support systems.

In the next section, we examine how these recommendation systems methodologies can be tailored and integrated into a conversational recommendation system framework specifically for CPS design and recommendation.

2.2 Motivation and Research Problem

Despite the maturity and practical success of recommendation systems in domains such as e-commerce, entertainment, and social media, their penetration into complex engineering contexts particularly Cyber–Physical Systems (CPS) remains limited. The distinctive nature of CPS configuration introduces unique challenges that conventional recommendation paradigms struggle to address. Unlike consumer domains, CPS design involves heterogeneous components (sensors, actuators, communication interfaces, and computational units) that must satisfy strict operational constraints, including latency, interoperability, energy efficiency, safety, and fault tolerance.

Traditional recommendation approaches, even those enhanced by machine learning, often lack the capacity to reason about such technical and interdependent constraints. For instance, a collaborative filtering (CF) algorithm may identify components frequently used together in past projects but cannot ensure that their communication protocols or voltage levels are compatible. Similarly, a content-based filtering (CBF) system may retrieve items with matching specifications but overlook implicit dependencies such as environmental robustness, redundancy requirements, or vendor-specific interoperability issues unless these are explicitly modeled in the knowledge representation.

Another persistent challenge lies in requirement elicitation and representation. In consumer-oriented domains, user preferences are typically inferred from explicit feedback (e.g., ratings or clicks) or implicit behavioral traces (e.g., browsing history). By contrast, CPS configuration often begins with under-specified, evolving, or ambiguously expressed requirements. These requirements are frequently distributed across unstructured sources such as technical documentation, research papers, or informal engineering discussions. Capturing and structuring such knowledge demands an interactive and iterative process one capable of interpreting domain-specific language and refining system understanding as new constraints emerge.

Collectively, these limitations highlight a fundamental research problem :

> *How can a recommendation system be designed to understand, reason about, and suggest CPS configurations in a way that accounts for technical constraints, domain knowledge, and evolving user requirements ?*

Addressing this question requires bridging multiple disciplines : knowledge

representation, explainable artificial intelligence, and natural language understanding. A promising direction lies in embedding recommendation mechanisms within conversational frameworks systems capable of understanding user intent, asking clarifying questions, and refining recommendations through interactive dialogue. Such an approach can transform the recommendation process from a static, data-driven matching problem into a dynamic, context-aware decision support system.

Building upon this motivation, the next section explores *Conversational Recommendation Systems* (CRS) an emerging paradigm that integrates dialogue-based interaction with intelligent recommendation logic. We discuss how CRS principles can be adapted to the CPS domain and how they lay the foundation for our proposed system, **IoT-AID**.

2.3 Conversational recommendation systems

Conversational recommendation systems extend the capabilities of traditional recommendation systems by embedding the recommendation process within a natural language dialogue. In the CPS context, a conversational recommendation system can :

- Interpret domain-specific terminology and constraints provided in natural language.
- Ask clarifying questions to refine the requirement set.
- Adapt recommendations dynamically as new information emerges during the conversation.
- Justify recommendations with references to compatibility rules, performance metrics, or prior deployments.

Such an approach aligns well with the needs of CPS engineers, particularly when integrating components across heterogeneous domains such as mechanical systems, embedded electronics, and software services. By leveraging semantic understanding, a conversational recommendation system can bridge the gap between high-level user goals (e.g., “monitor temperature in hazardous environments with sub-second latency”) and low-level component specifications (e.g., sensor accuracy, communication bandwidth, computational resources).

In this work, we propose **IoT-AID** as an AI-driven Cyber-Physical Recommendation System (CPRS) built upon large language models (LLMs) for semantic interpretation and context-aware reasoning. IoT-AID integrates state-of-the-art NLP embeddings (e.g., BERT, Sentence Transformers) with structured CPS knowledge bases to :

1. Parse and normalize user requirements expressed in free text.
2. Match requirements to candidate CPS components using a hybrid RS approach.
3. Validate interoperability through rule–based checks.
4. Present ranked, explainable recommendations interactively through dialogue.

This design aims to make CPS configuration accessible to non-specialists while improving efficiency for domain experts. By embedding recommendation logic within a conversational framework, IoT-AID not only suggests suitable components but also educates users about trade-offs, compatibility constraints, and optimization opportunities.

2.3.1 Dialogue Systems for CPS

Dialogue systems are intelligent agents capable of interacting with humans through natural language, enabling more intuitive and personalized communication between users and machines. They have found applications in customer service, education, entertainment, healthcare, and, increasingly, in complex technical domains such as Cyber-Physical Systems (CPS). By enabling users to express needs in natural language and receive context-aware responses, dialogue systems bridge the gap between user intent and system functionality.

A typical dialogue system comprises three core components : *Natural Language Understanding (NLU)*, which processes user input to extract intent and relevant information ; *Dialogue Management (DM)*, which decides on the most appropriate system action based on the conversation state ; and *Natural Language Generation (NLG)*, which produces coherent and contextually appropriate responses [2]. Depending on their design, dialogue systems can be :

- **Task-oriented :** Designed to guide users toward accomplishing specific goals, such as booking travel or configuring a CPS. They follow structured conversation flows and rely heavily on slot-filling techniques to collect the necessary parameters for completing the task [19].

- **Non-task-oriented :** Designed for open-ended conversation to engage, inform, or entertain users [5].

From an implementation perspective, dialogue systems can be rule-based where responses follow handcrafted logic or data-driven, where machine learning models generate responses based on patterns learned from conversational datasets. Modern data-driven systems often leverage deep learning architectures such as Transformers to handle complex and varied user queries.

Figure 2.4 illustrates a typical architecture for task-oriented dialogue systems, showing how NLU, DM, and NLG interact to process user input and generate meaningful responses.

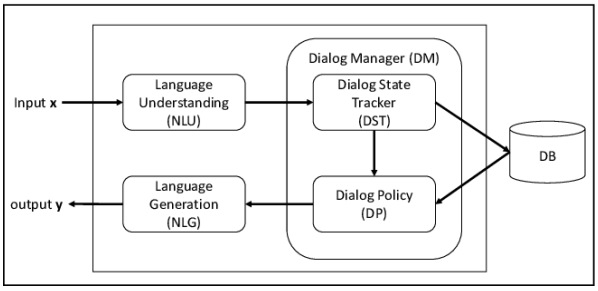


FIGURE 2.4 – An architecture for a task-oriented dialog system [18].

2.3.2 Conversational Recommendation Systems for CPS

2.3.3 IoT-AID : Proposed Framework

IoT-AID is our proposed Conversational Cyber-Physical Recommendation System (CPRS), tailored specifically to assist engineers, researchers, and system integrators in designing CPS solutions. Unlike general-purpose conversational recommendation system platforms, IoT-AID is domain-specific, embedding CPS attributes, technical constraints, and best practices directly into its recommendation process.

The IoT-AID workflow begins when a user initiates a conversation through a chat-based interface. User queries are processed by advanced NLP models such as BERT and Sentence Transformers to extract key requirements, such as the need for specific sensors, network protocols, or computational units. These extracted parameters are then matched against a structured knowledge base of CPS components, enriched with semantic relationships and compatibility rules.

Recommendations are generated using a hybrid approach : semantic similarity scores suggest the most relevant components, while rule-based reasoning ensures technical feasibility. When information is incomplete or ambiguous, IoT-AID proactively asks clarifying questions, enabling a more accurate and contextually aligned solution. The final output is a ranked list of components, accompanied by explanatory notes to improve user trust and transparency.

This modular architecture allows IoT-AID to adapt across CPS subdomains, from industrial automation to smart cities and healthcare IoT. By translating natural language requirements into actionable technical recommendations, IoT-AID reduces the expertise barrier for CPS deployment.

2.4 Conclusion

Conversational recommendation systems have emerged as a paradigm shift in human–machine interaction, moving beyond static recommendation pipelines toward adaptive, context-aware, and dialog-driven decision support. This chapter has examined how the integration of conversational intelligence with recommendation techniques ranging from collaborative and content-based filtering to hybrid, knowledge-based, and deep learning–driven methods can redefine the way complex technical systems are designed and configured.

Within the Cyber–Physical System (CPS) context, this conversational paradigm addresses a crucial limitation of traditional recommendation systems : their inability to interpret evolving user intent, reason about domain-specific constraints, and justify decisions in transparent, explainable terms. By embedding the recommendation process within a natural language dialogue, a conversational recommendation system can iteratively refine requirements, validate compatibility, and provide traceable, semantically grounded suggestions.

The proposed IoT–AID framework encapsulates these principles. It operationalizes a domain-specific Conversational Cyber–Physical Recommendation System (CPRS) that integrates advanced NLP models, hybrid recommendation strategies, and rule-based reasoning to facilitate CPS configuration through interactive dialogue. By bridging the gap between high-level design intentions and low-level technical configurations, IoT–AID aims to reduce the expertise barrier, accelerate system prototyping, and promote explainable, human-centric automation.

The next chapter presents the system architecture and integration pipeline of IoT–AID in detail. It describes how the conversational layer, retrieval mechanisms, and explainability modules collaborate to form a coherent CPS configuration assistant. This architectural foundation sets the stage for subsequent implementation, experimentation, and evaluation, demonstrating IoT–AID's potential to streamline CPS design workflows and foster more intelligent, transparent, and user-adaptive engineering environments.

Bibliographie

[1] Charu C AGGARWAL et Charu C AGGARWAL. « An introduction to recommender systems ». In : *Recommender systems : the textbook* (2016), p. 1-28.

[2] Suket ARORA, Kamaljeet BATRA et Sarabjit SINGH. « Dialogue system : A brief review ». In : *arXiv preprint arXiv :1306.4134* (2013).

[3] Jesus BOBADILLA, Santiago ALONSO et Antonio HERNANDO. « Deep learning architecture for collaborative filtering recommender systems ». In : *Applied Sciences* 10.7 (2020), p. 2441.

[4] Robin BURKE. « Knowledge-based recommender systems ». In : *Encyclopedia of library and information systems* 69.Supplement 32 (2000), p. 175-186.

[5] Yuya CHIBA et al. « An analysis of the effect of emotional speech synthesis on non-task-oriented dialogue system ». In : *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*. 2018, p. 371-375.

[6] Mohammad CHOAIB et al. « IoT-AID : An Automated Decision Support Framework for IoT ». In : *SN Computer Science* 5.4 (2024), p. 429.

[7] Cach N DANG, María N MORENO-GARCÍA et Fernando De la PRIETA. « Using hybrid deep learning models of sentiment analysis and item genres in recommender systems for streaming services ». In : *Electronics* 10.20 (2021), p. 2459.

[8] Laila ESHEIBA et al. « A hybrid knowledge-based recommender for product-service systems mass customization ». In : *Information* 12.8 (2021), p. 296.

[9] Sarika JAIN et al. « Trends, problems and solutions of recommender system ». In : *International conference on computing, communication & automation*. IEEE. 2015, p. 955-958.

[10] Zhiyang JIA et al. « User-based collaborative filtering for tourist attraction recommendations ». In : *2015 IEEE international conference on computational intelligence & communication technology*. IEEE. 2015, p. 22-25.

[11] Hyeyoung KO et al. « A survey of recommendation systems : recommendation models, techniques, and application fields ». In : *Electronics* 11.1 (2022), p. 141.

[12] Pasquale LOPS, Marco DE GEMMIS et Giovanni SEMERARO. « Content-based recommender systems : State of the art and trends ». In : *Recommender systems handbook* (2011).

[13] Prem MELVILLE et Vikas SINDHWANI. « Content-based recommendation systems ». In : *The Adaptive Web*. Sous la dir. de Peter BRUSILOVSKY, Alfred KOBSA et Wolfgang NEJDL. Springer, 2010, p. 137-187. DOI : 10.1007/978-3-540-72079-9_5.

[14] P. V. NAGAVARDHAN. « Recent Trends in Hybrid Recommendation Systems in e-commerce Domain ». In : 2020.

[15] Badrul SARWAR et al. « Item-based collaborative filtering recommendation algorithms ». In : *Proceedings of the 10th international conference on World Wide Web*. 2001, p. 285-295.

[16] J Ben SCHAFER et al. « Collaborative filtering recommender systems ». In : *The adaptive web : methods and strategies of web personalization* (2007).

[17] Iulian SERBAN et al. « Building end-to-end dialogue systems using generative hierarchical neural network models ». In : *Proceedings of the AAAI conference on artificial intelligence*. T. 30. 1. 2016.

[18] Swadheen SHUKLA et al. « Conversation Learner–A Machine Teaching Tool for Building Dialog Managers for Task-Oriented Dialog Systems ». In : *arXiv preprint arXiv :2004.04305* (2020).

[19] Zhao YAN et al. « Building task-oriented dialogue systems for online shopping ». In : *Proceedings of the AAAI Conference on Artificial Intelligence*. T. 31. 1. 2017.

Chapitre 3

Architectural Framework, Design and Integration Details of IoT-AID's Conversational CPRS

Designing a Cyber–Physical System (CPS) involves bridging multiple layers of complexity from physical device selection to computational intelligence and human–system interaction. As established in the previous chapters, one of the main barriers to efficient CPS deployment lies in the knowledge gap between domain experts, who understand operational requirements, and system engineers, who possess the technical expertise for integration. To mitigate this gap, we introduce the IoT–AID framework, a domain-aware Conversational Cyber–Physical Recommendation System (CPRS) that leverages natural language dialogue and AI-driven reasoning to guide users throughout the CPS configuration process[3, 7, 4].

The core idea behind IoT–AID is to make CPS configuration both intuitive and intelligent. Instead of requiring users to manually select sensors, communication protocols, and computation resources, IoT–AID engages them in a structured, context-aware conversation. Through iterative dialogue, the system elicits user requirements, interprets technical constraints, and progressively refines recommendations. By asking clarifying questions and explaining the rationale behind its suggestions, IoT–AID transforms CPS design into a collaborative process reducing cognitive load, avoiding configuration mismatches, and ensuring alignment with domain-specific standards.

To achieve this, IoT–AID integrates multiple technological layers :

⇒ a Conversational Layer powered by advanced natural language processing, (NLP) and large language models, (LLMs) such as BERT and Mistral, enabling accurate understanding of user intent,

⇒ a Recommendation Layer that combines collaborative, content-based, and hybrid filtering to produce context-relevant suggestions, and

⇒ an Integration Layer linking semantic knowledge bases, CPS characteristics, and explainability modules to ensure that recommendations are technically valid and transparent.

Together, these layers create a seamless human–AI collaboration framework tailored for the complexities of CPS configuration. IoT–AID's design prioritizes modularity, explainability, and extensibility, allowing it to be adapted across industrial, healthcare, and smart infrastructure domains[6, 8].

The rest of this chapter is organized as follows : Section 3.1 provides an overview of the IoT–AID architecture, describing its main components, logical layers, and their interconnections. Section 3.2 details the conversational framework, including the dialogue management strategy and NLP models used for understanding and generating natural language interactions. Section 3.3 presents the design of the recommendation pipeline, explaining how hybrid recommendation algorithms and CPS attributes are integrated to produce accurate and explainable results. Section 3.4 describes the overall system integration and communication flow between modules, emphasizing interoperability and scalability. Section 3.5 concludes the chapter by summarizing key design insights and outlining the transition toward the implementation and evaluation phase.

3.1 Proposed CPS Architectures and Connections

Inspired by the CPS design model presented in [2], we propose a simplified yet expressive Cyber-Physical System (CPS) conceptual framework serving as the foundation for the IoT-AID recommendation system. This conceptual architecture provides an abstract representation of how physical, communication, and computational components interact within CPS environments. It serves as both a reference model and a semantic layer for the construction of the knowledge base used by our Cyber-Physical Recommendation System (CPRS).

The architecture divides the CPS into three interdependent layers :

⇒ Physical Layer : Comprising sensors, actuators, and embedded devices responsible for collecting environmental or operational data.

⇒ Communication Layer : Handling data transmission through protocols such as MQTT, Zigbee, Wi-Fi, or Bluetooth, ensuring interoperability between heterogeneous devices.

⇒ Cyber Layer : Encompassing data analysis, control logic, and decision-making components, including AI and recommendation modules.

The proposed structure not only reflects the fundamental CPS design principles but also supports modular integration and domain generalization. In the context of IoT-AID, this architecture functions as the core knowledge sphere, enabling semantic representation of CPS entities and their relationships. Consequently, it forms the conceptual backbone of the knowledge base (KB) that drives reasoning and recommendation within the system[10].

3.2 Proposed Recommendation System

Each CPS application requires a specific combination of physical components, communication technologies, and computational resources. However, stakeholders often lack the technical expertise to identify and integrate these elements effectively. To address this issue, IoT-AID introduces a Conversational Cyber-Physical Recommendation System (CPRS) that assists users in selecting appropriate CPS components through interactive dialogue and AI-driven reasoning[9].

Figure 3.1 illustrates the overall architecture of the proposed system. It is composed of a conversational interface, a natural language understanding module, a knowledge base, and a hybrid recommendation engine[1]. The output generated by this system is further managed through a suggestion engine, leveraging a Knowledge Base, which comprises a collection of meta-features that describe the sensors, their specifications, relevant domains, and other pertinent details.

The system receives user input such as application domain, objectives, constraints, and budget through a chatbot interface. The input text is pre-processed to extract meta-features and domain-specific keywords using NLP techniques. These features are then matched against stored data in the Knowledge Base (KB), which contains detailed descriptions of sensors, actuators, and other CPS elements (see Listing 3.2).

The recommendation process consists of :

⇒ Understanding user intent through textual analysis.

⇒ Retrieving candidate components using similarity and semantic queries.

⇒ Ranking and filtering results according to constraints and confidence levels.

⇒ Refining recommendations via user feedback.

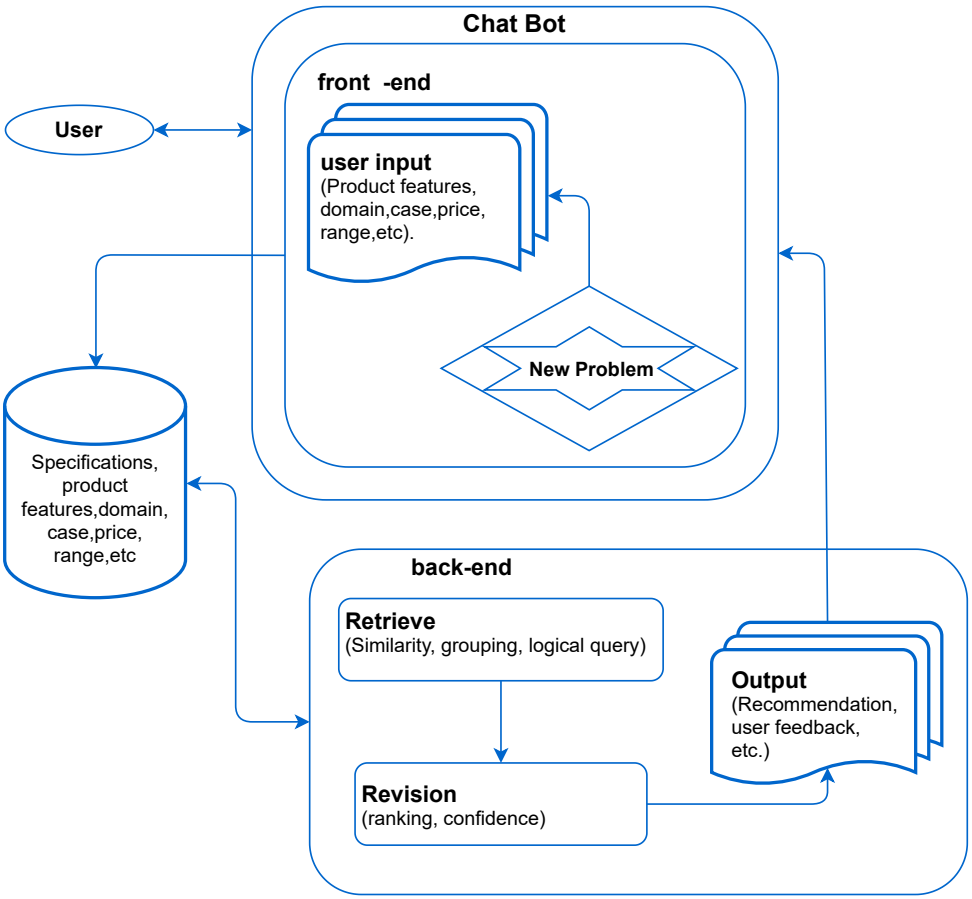


FIGURE 3.1 – Global architecture of the proposed Conversational CPS Recommendation System.

This feedback loop ensures continuous learning and incremental improvement of the KB. The KB operates as a meta-knowledge repository, supporting both :

⇒ Structured storage of item descriptions, specifications, and domain features.

⇒ Efficient querying and reasoning, including explainability acquisition and traceability.

The proposed system is employed to enhance information retrieval and to provide transparent, human-understandable explanations for each recommendation.

Figure 3.2 illustrates the XML format used to represent sensor metadata inside the knowledge base. Attributes such as sampling rate, measurement range, sensing principle, and environmental constraints are extracted from this structure and transformed into semantic vectors during the recommendation process.

```
<Sensor>
    <Name>FHI</Name>
    <Type> Humidity sensor </Type>
    <Specification>
        <Powermax> 5 </Powermax>
        <RangeminC> -60 </RangeminC>
        <RangemaxC> 140 </RangemaxC>
        <RangeminRH> 0 </RangeminRH>
        <RangemaxRH> 100 </RangemaxRH>
        <Accuracymin> -1 </Accuracymin>
        <Accuracymax> 1 </Accuracymax%>
        <Price> 5.8$ </Price>
    </Specification>
    <Description>
      <Use> Designed for high volume, cost sensitive
          applications </Use>
      <Manufacturer> TE Connectivity </Manufacturer>
      <Data Type> Analog </Data Type>
      <Programmable> True </Programmable>
      <Communication> Cable </Communication>
    </Description>
    <Domain>
      <Historical App>
          1-automotive cabin air control
          2-home appliances
          3-industrial process control systems
          4-Meteorology
      </Historical App>
      <Domain of App> Industry </Domain of App>
    </Domain>
</Sensor>
```

FIGURE 3.2 – Example XML file describing a humidity sensor.

3.3 Illustrative Example : Real-Time Patient Surveillance System

To illustrate the applicability of IoT-AID, we consider a Patient Surveillance System (PSS) for elderly individuals. The objective is to assist caregivers and ensure patient safety by recommending appropriate sensors and configurations for real-time monitoring. In this scenario, the user describes their requirements such as movement detection, medication tracking, or safety monitoring through the chatbot. Depending on the specificity of the description, the system may recommend either :

⇒ Specific sensors (e.g., motion or temperature sensors), or

⇒ Combinations of components for broader objectives (e.g., safety surveillance with gas, fire, and humidity sensors).

The system thus supports both fine-grained and general-purpose recommendations, ensuring adaptability to user expertise and context.

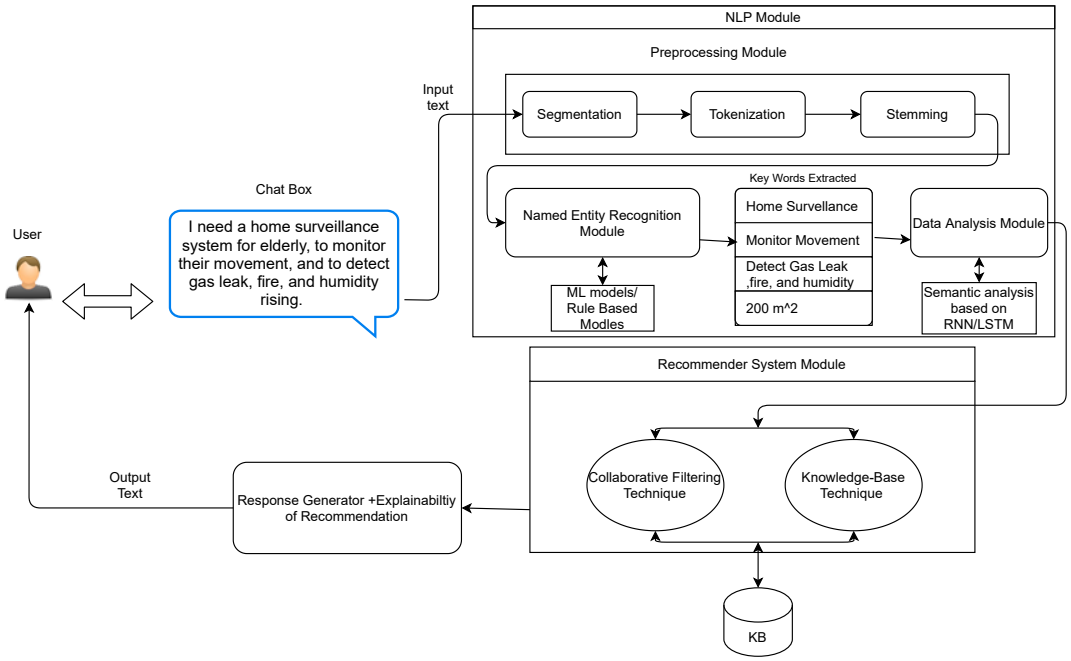


FIGURE 3.3 – Illustrative example : Home surveillance for elderly patients.

3.4 Natural Language Processing Module

The Natural Language Processing (NLP) module is a central component of IoT-AID's conversational framework, enabling the system to interpret user requests, capture intents, and generate context-aware recommendations. Unlike traditional NLP pipelines that rely on explicit entity recognition techniques or handcrafted rules, IoT-AID employs a prompt-based, LLM-driven text understanding approach. This design allows for greater flexibility, scalability across domains, and robustness to diverse linguistic expressions[5].

The NLP module is responsible for transforming user utterances into machine-understandable representations that can guide the recommendation process. It interprets natural language descriptions of user goals, constraints, and contexts such as "I need a low-cost humidity sensor for greenhouse monitoring" and converts them into structured semantic embeddings. These embeddings capture both explicit constraints (e.g., cost, sensor type, target domain) and implicit intent (e.g., environmental monitoring, reliability preference). The module interacts closely with the knowledge base (KB) and the recommendation engine. It ensures that the conversational system can dynamically adapt its responses, refine its understanding through dialogue, and retrieve relevant CPS components aligned with the user's intent.

The NLP module serves as the semantic bridge between natural language user input and the technical reasoning processes of IoT-AID. By leveraging

prompt-based LLM inference and embedding-driven understanding, it provides a robust foundation for contextual, explainable, and adaptive CPS configuration assistance.

3.4.1 Preprocessing and Text Normalization

Before the input is processed by the language model, a preprocessing layer performs text cleaning, tokenization, and normalization. This includes :

⇒ Removing non-informative tokens and punctuation.

⇒ Expanding abbreviations and domain-specific shorthand (e.g.,"temp sensor" -> "temperature sensor").

⇒ Standardizing units and measurement formats (e.g., converting "5 m" and "five meters" into consistent forms).

These steps ensure that linguistic and technical variations do not affect the accuracy of semantic interpretation. The normalized text is then passed to the prompt embedding stage for semantic analysis.

3.4.2 Prompt-Based Semantic Understanding

At the core of IoT-AID's NLP module is a Large Language Model (LLM), such as Mistral-7B, fine-tuned or prompted for domain-adaptive reasoning. Instead of explicit feature engineering, the system relies on prompt templates that structure the interaction, guiding the model to extract :

⇒ User intent : identifying the goal behind the request (e.g., monitoring, control, automation).

⇒ Key attributes : such as environment type, desired accuracy, budget, or connectivity requirements.

⇒ Contextual dependencies : linking needs to CPS components or architectures (e.g., motion detection -> passive infrared sensors -> home automation domain).

This semantic understanding phase transforms unstructured text into latent embeddings that preserve meaning and contextual relations, enabling downstream reasoning and similarity-based retrieval from the knowledge base.

3.4.3 Post-Processing and Data Analysis

Following embedding generation, the post-processing module analyzes the output representations to infer actionable insights and structure them into queryable forms for the recommendation module. Techniques such as

semantic clustering, vector similarity search, and contextual filtering are applied to match user representations with stored component profiles in the KB.

This enables the system to handle multi-turn dialogues where user intent evolves dynamically (e.g., refining a request from "a humidity sensor" to "a wireless one with indoor range").

The resulting structured information is then passed to the recommendation engine, which generates and ranks candidate CPS configurations.

3.4.4 Integration and Continuous Refinement

The NLP module continuously improves through feedback-driven refinement. User interactions such as accepted or rejected recommendations are logged and analyzed to fine-tune future model prompts and similarity thresholds. Over time, this allows IoT-AID to adapt to new domains, terminologies, and user behaviors without manual reconfiguration.

3.5 Conclusion

In this chapter, we presented the architectural framework, design, and integration details of IoT-AID, our Conversational Cyber-Physical Recommendation System (CPRS). We detailed the proposed CPS architecture, the modular recommendation pipeline, and the NLP-driven semantic understanding processes that enable context-aware and user-centered guidance. By leveraging a structured knowledge base, hybrid recommendation techniques, and iterative feedback through dialogue, IoT-AID can translate high-level user requirements into actionable CPS component recommendations, effectively bridging the gap between domain expertise and technical implementation.

We introduced a novel CPS architecture and modeling approach, which serves as the foundation for the knowledge base underpinning IoT-AID. The Cyber-Physical Recommendation System (CPRS) within this framework enables context-aware recommendations by analyzing user objectives, project details, and application scenarios. Leveraging advanced NLP techniques, including BERT and Sentence Transformers, IoT-AID can extract semantic meaning from user inputs, track conversational context, and map requirements to technically valid CPS components.

Furthermore, the chapter outlined the integration of key system modules including text preprocessing, semantic encoding, context tracking, and recommendation generation demonstrating how IoT-AID bridges the gap between high-level user requirements and low-level CPS specifications. The framework also incorporates feedback loops to iteratively refine recommendations and continuously enrich the knowledge base.

The framework establishes a foundation for real-world CPS configuration, ensuring adaptability, explainability, and precision in diverse application domains. Moving forward, the next chapter focuses on the implementation framework of IoT-AID, emphasizing how large language models (LLMs) are integrated into the conversational pipeline to enhance semantic understanding, contextual reasoning, and interactive recommendation capabilities. This chapter will demonstrate the practical realization of the design principles discussed here and illustrate how IoT-AID operates as a fully functional, intelligent decision support system for CPS.

Bibliographie

[1] Abiola A AKANMU, Chimay J ANUMBA et Omobolanle O OGUNSEIJU. « Towards next generation cyber-physical systems and digital twins for construction. » In : *Journal of Information Technology in Construction* 26 (2021).

[2] Asma DJOUAD, Fadila ATIL et Chahrazed BEDDIAR. « Domain Model for Cyber-Physical Systems ». In : *Physical Systems* (2018), p. 8.

[3] Jay LEE, Behrad BAGHERI et Hung-An KAO. « A cyber-physical systems architecture for industry 4.0-based manufacturing systems ». In : *Manufacturing letters* 3 (2015), p. 18-23.

[4] Yang LU. « Industry 4.0 : A survey on technologies, applications and open research issues ». In : *Journal of industrial information integration* 6 (2017), p. 1-10.

[5] Felix O OLOWONONI, Danda B RAWAT et Chunmei LIU. « Resilient machine learning for networked cyber physical systems : A survey for machine learning security to securing machine learning for CPS ». In : *IEEE Communications Surveys & Tutorials* 23.1 (2020), p. 524-552.

[6] Diego GS PIVOTO et al. « Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0 : A literature review ». In : *Journal of manufacturing systems* 58 (2021), p. 176-192.

[7] Ragunathan RAJKUMAR et al. « Cyber-physical systems : the next computing revolution ». In : *Proceedings of the 47th design automation conference*. 2010, p. 731-736.

[8] Michael SONY. « Design of cyber physical system architecture for industry 4.0 through lean six sigma : conceptual foundations and research issues ». In : *Production & Manufacturing Research* 8.1 (2020), p. 158-181.

[9] Martin TÖRNGREN et Ulf SELLGREN. « Complexity challenges in development of cyber-physical systems ». In : *Principles of modeling : Essays dedicated to Edward A. Lee on the occasion of his 60th birthday*. Springer, 2018, p. 478-503.

[10] Hansong XU et al. « A survey on industrial Internet of Things : A cyber-physical systems perspective ». In : *Ieee access* 6 (2018), p. 78238-78259.

Chapitre 4

Implementation Framework for IoT-AID : Leveraging Large Language Models in the Development of Conversational CPRS

In the era of Industry 4.0, the fusion of digital and physical systems has fundamentally transformed industrial landscapes, driven by advancements in technologies such as the Internet of Things (IoT), Artificial Intelligence (AI), and big data analytics. At the core of this transformation are Cyber-Physical Systems (CPS), which combine computational algorithms with physical processes to enable real-time sensing, processing, and decision-making. While CPS offer significant benefits in terms of automation, efficiency, and resource optimization, their design, configuration, and deployment remain inherently complex, often requiring deep domain expertise and extensive technical knowledge.

To address these challenges, we introduce IoT-AID, a Conversational Cyber-Physical Recommendation System that leverages advanced machine learning models, deep learning architectures, and fine-tuned Large Language Models (LLMs) such as BERT and Sentence Transformers. IoT-AID interprets user requirements expressed in natural language and provides personalized, context-aware recommendations for CPS configuration, significantly simplifying the process for users with varying levels of expertise. By combining NLP-driven semantic understanding with a structured knowledge base and hybrid recommendation techniques, IoT-AID bridges the gap between human intent and technical implementation.

Beyond simplifying CPS configuration, IoT-AID serves as an enabler for

the broader adoption of CPS in Industry 4.0 by streamlining system design, reducing the expertise barrier, and supporting real-time, data-driven decision-making. This chapter details the implementation framework of IoT-AID, highlighting how LLMs are integrated into the conversational pipeline, how semantic understanding and context tracking are realized, and how the recommendation engine operates to provide reliable, explainable, and interactive guidance. A comprehensive evaluation of the system demonstrates the effectiveness of LLM-driven conversational recommendation in CPS design and outlines pathways for future enhancement.

The rest of the chapter is organized as follows : Section 4.1 gives an overview of the implemented IoT-AID architecture and interaction between its NLP, recommendation, and knowledge-base modules. Section 4.2 details of the natural language processing workflow, including preprocessing, embedding generation, semantic encoding, and context tracking using LLMs. Section 4.3 discusses the integration and role of LLMs within the IoT-AID Framework. Section 4.4 discusses the techniques and algorithms used for hybrid recommendation, knowledge-based reasoning, and real-time suggestions. Section 4.5 gives the design and utilization of structured CPS knowledge, including component specifications, historical data, and feature metadata.

Section **??** briefly narrates the key insights from the implementation and the role of LLMs in enhancing CPS recommendation.

4.1 System Architecture and Module Integration

Conversational Recommendation Systems (CRS) combine the strengths of dialogue management and recommendation logic to create adaptive, user-centric decision support tools. Unlike traditional recommendation systems that rely on static profiles or large volumes of historical data, CRS engage users in real time, refining recommendations through an iterative question–answer process [2]. This dynamic interaction enables better preference elicitation, contextual adaptation, and transparency features particularly valuable for complex technical domains such as Cyber-Physical Systems (CPS).

In the context of CPS configuration, conversational recommendation systems offer a significant advantage : they can interpret ambiguous design requirements, capture implicit constraints, and iteratively propose compatible component configurations until an optimal solution is achieved. This transforms the system from a passive suggestion engine into an interactive configuration assistant that guides users through the design process.

4.1.1 Overview of IoT-AID Architecture

The architecture of IoT-AID integrates advanced Natural Language Processing (NLP) capabilities with a hybrid recommendation engine and a structured CPS knowledge base, all coordinated through a dialogue management layer. This modular architecture ensures scalability, transparency, and interoperability across different CPS domains.

As illustrated in Figure 4.1, the IoT-AID framework is composed of the following main modules :

- **Dialogue Management and Context Tracking :** This module governs the interaction flow, maintaining the conversation context across multiple turns. It interprets user intents, manages clarification queries, and ensures that evolving requirements are consistently represented in the system state.
- **Natural Language Processing Module :** Built on large language models such as BERT or Sentence Transformers, this module processes user queries to extract semantically rich representations. It performs tasks such as text normalization, intent recognition, and constraint extraction to bridge the gap between natural language and structured CPS parameters.
- **Knowledge Base :** A structured repository containing CPS components, specifications, and compatibility rules. It enables semantic matching and validation of recommendations, ensuring that suggested configurations are technically feasible.
- **Hybrid Recommendation Engine :** This module combines content-based filtering, collaborative strategies, and rule-based reasoning to generate candidate component lists. It evaluates both semantic similarity (based on embeddings) and technical compatibility (based on stored metadata).
- **Explainability and Feedback Interface :** To enhance user trust and learning, the system provides explanations for recommendations, linking suggestions to their underlying rationale (e.g., performance, compatibility, or prior deployment evidence). User feedback is continuously integrated to refine subsequent recommendations.

4.1.2 Module Integration Workflow

The operation of IoT-AID follows a pipeline designed to ensure seamless information flow between the conversational and recommendation layers :

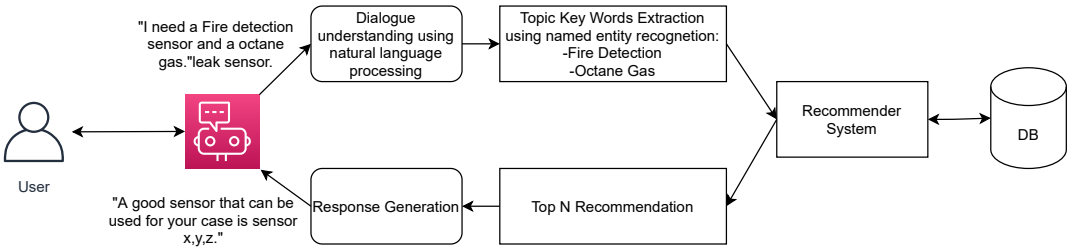


FIGURE 4.1 – Overall system architecture and module interaction of IoT-AID's Conversational CPRS.

1. The user initiates a query in natural language describing a CPS design goal or component requirement.
2. The NLP module preprocesses and encodes the query into semantic embeddings using LLMs.
3. The dialogue manager interprets user intent, updates the context, and identifies missing information.
4. The hybrid recommendation engine retrieves candidate components from the knowledge base based on semantic and rule-based filtering.
5. Results are refined through multi-turn dialogue IoT-AID may ask clarification questions to improve precision.
6. The final recommendations are ranked, explained, and presented interactively to the user.

This modular yet integrated workflow enables IoT-AID to support both novice users by providing guided configuration assistance and expert engineers by facilitating rapid exploration of CPS design alternatives.

4.2 NLP Pipeline Implementation

The Natural Language Processing (NLP) pipeline is the core component of IoT-AID's conversational framework. It enables the system to interpret user queries, extract technical intents, and translate them into structured representations suitable for downstream recommendation and reasoning tasks. Unlike traditional rule-based approaches, IoT-AID leverages transformer-based models to achieve semantic understanding and contextual generalization across diverse CPS design scenarios.

4.2.1 Overview of the NLP Workflow

The NLP pipeline operates as a multi-stage semantic interpretation process that progressively transforms raw textual inputs into machine-interpretable representations. This process allows IoT-AID to bridge the gap between human-like conversational input and domain-specific CPS configuration data.

Figure 4.2 illustrates the overall pipeline, which consists of the following major stages :

- **User Query Acquisition :** Users interact with IoT-AID through a conversational interface, submitting free-form textual inputs such as requirements, design goals, or constraints (e.g., "I need sensors for monitoring humidity and temperature in a greenhouse environment.").
- **Text Preprocessing and Normalization :** The raw input is preprocessed using standard NLP techniques, including tokenization, lemmatization, and stop-word removal. Domain-specific term normalization ensures consistent handling of CPS terminology (e.g., "IoT hub," "edge device," "actuator node").
- **Semantic Encoding via LLMs :** After preprocessing, the refined text is encoded using transformer-based language models such as *BERT* or *Sentence Transformers*. These embeddings capture semantic relationships and contextual nuances that help the system identify user intents, technical entities, and implicit constraints.
- **Context Tracking and Dialogue State Management :** The dialogue manager maintains a multi-turn state representation, allowing the system to integrate new user inputs with prior context. This ensures continuity and disambiguation across interactions, enabling iterative refinement of recommendations.
- **Recommendation Query Formation :** The extracted intents and constraints are structured into a query representation compatible with the knowledge base schema. This includes generating feature vectors, entity–attribute pairs, and relevance weights to query the hybrid recommendation engine.

The NLP pipeline of IoT-AID, illustrated in Figure,4.2, summarizes the sequence of processes through which natural language input is transformed into structured semantic representations. Beginning with text preprocessing and normalization, the system prepares raw user input for interpretation by a prompt-based LLM model. The semantic embeddings produced capture user intent, contextual dependencies, and domain-specific constraints. These embeddings are then refined through post-processing and sequential context

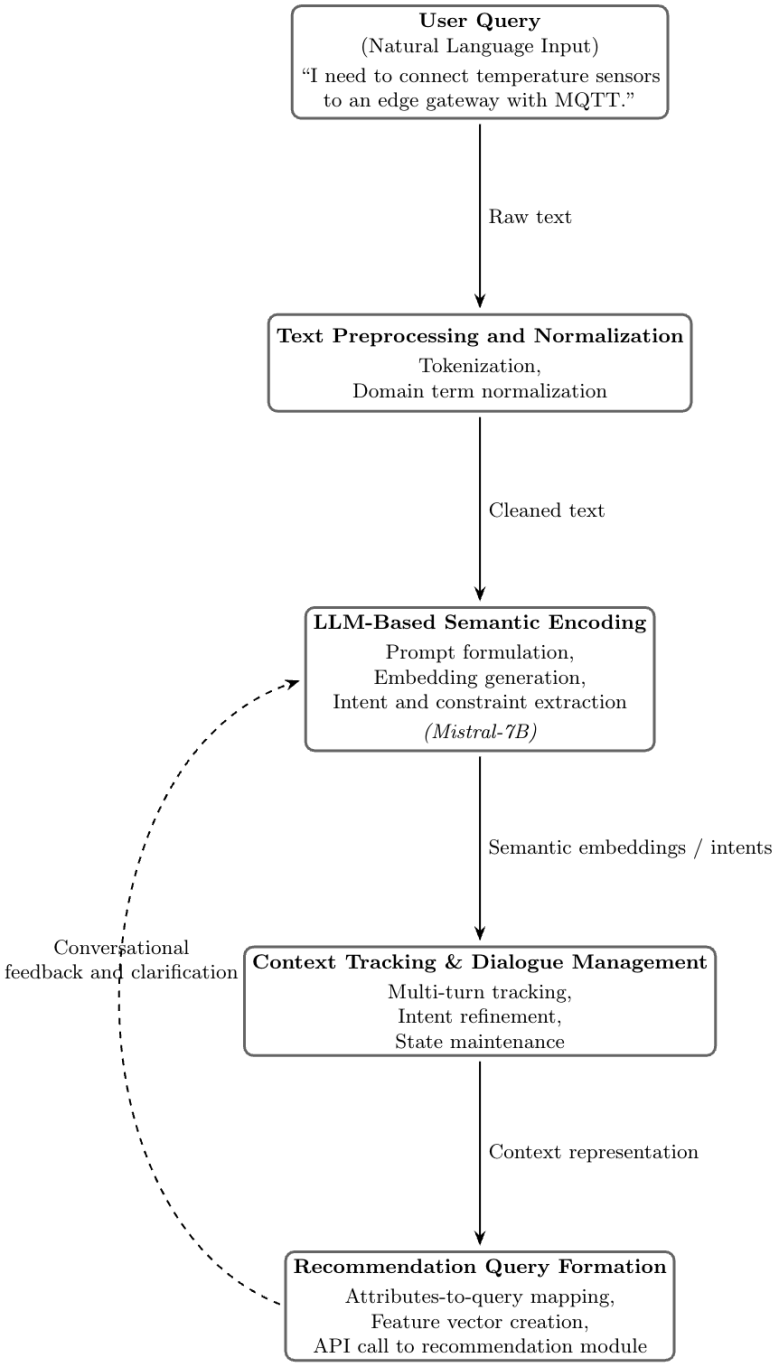


FIGURE 4.2 – NLP pipeline for semantic understanding and query formation in IoT-AID.

tracking, ensuring continuity across multi-turn dialogues. The final structured representations are transmitted to the recommendation module for retrieval and ranking of suitable CPS components. This layered architecture

enables IoT-AID to maintain coherence, interpret ambiguity, and deliver precise, context-aware recommendations in conversational settings.

4.2.2 Semantic Representation and Embedding Strategy

IoT-AID employs transformer-based encoders, such as *BERT-base* or domain-adapted *Sentence Transformers*, for generating dense vector embeddings. These embeddings represent the semantic meaning of user inputs rather than mere syntactic similarity. By comparing embedding distances, the system identifies the most relevant entities or components from the knowledge base.

The embedding process follows this formalization :

$$E = f_\theta(T) \tag{4.1}$$

where T is the tokenized input sequence, and f_θ denotes the LLM encoder parameterized by θ. The resulting vector E captures latent semantic features that are subsequently used in similarity scoring :

$$S(x, y) = \cos(E_x, E_y) \tag{4.2}$$

where S represents cosine similarity between user intent embedding E_x and component embeddings E_y stored in the knowledge base.

4.2.3 Dialogue Context Integration

The multi-turn dialogue context is managed through a memory state vector that stores the user's prior intents and constraints. This allows the system to interpret follow-up queries such as "Make it energy efficient" or "Use wireless connectivity" without requiring restatement of previous information. Contextual embeddings are dynamically updated to maintain coherence across conversation turns.

4.2.4 Outcome of the NLP Pipeline

At the end of the NLP pipeline, a structured representation of the user's design intent is produced. This structured query serves as input to the hybrid recommendation engine described in Section 4.1. Through this pipeline, IoT-AID achieves robust natural language understanding, enabling intuitive, explainable, and domain-aware interactions for CPS configuration.

4.3 Integration of Large Language Models in IoT-AID

Large Language Models (LLMs) have emerged as pivotal enablers for intelligent conversational systems, offering unprecedented capabilities in natural language understanding, semantic generalization, and contextual reasoning. In IoT-AID, LLMs are leveraged not merely as text encoders but as adaptive knowledge mediators capable of interpreting, reasoning, and generating contextually relevant recommendations for Cyber-Physical System (CPS) configuration.

This section outlines how transformer-based models, including *BERT* and *Sentence Transformers*, are integrated into the IoT-AID framework to power semantic encoding, intent recognition, and similarity-based component recommendation.

4.3.1 Role of LLMs within the IoT-AID Framework

The integration of LLMs within IoT-AID serves multiple purposes across the conversational pipeline (see Figure 4.2) :

- **Semantic Understanding :** LLMs capture the latent semantic relationships between technical terms and user intents. This enables the system to process diverse phrasings of CPS requirements without explicit retraining or manual rule definition.
- **Intent and Entity Recognition :** Fine-tuned LLMs are employed to identify domain-specific intents (e.g., *"select sensor"*, *"configure gateway"*, *"optimize energy consumption"*) and extract entities such as device types, parameters, and operating conditions.
- **Context-Aware Query Expansion :** By utilizing contextual embeddings, LLMs help expand underspecified or ambiguous user queries through semantically related terms, improving the coverage of recommendations.
- **Knowledge-Based Reasoning :** The vectorized representations produced by LLMs interface with the recommendation engine and knowledge base, allowing semantic similarity matching, analogical reasoning, and constraint validation.

4.3.2 Model Selection and Adaptation Strategy

IoT-AID adopts a hybrid model integration approach, combining pre-trained general-purpose LLMs with domain-adapted fine-tuning. Specifically, the system utilizes *Sentence-BERT (SBERT)* to balance semantic accuracy and computational efficiency.

Fine-tuning is performed using CPS-related textual datasets, comprising component descriptions, configuration manuals, and design documents. The objective is to adapt the embedding space of the model such that semantically related CPS entities (e.g., *"temperature sensor"*, *"humidity node"*) are closely positioned in the latent vector space.

The model optimization process follows contrastive fine-tuning principles, where pairs of semantically similar and dissimilar sentences are used to refine embedding distance metrics. The loss function is defined as :

$$\mathcal{L} = 1 - \cos(E_{anchor}, E_{positive}) + \max(0, \cos(E_{anchor}, E_{negative}) - m) \tag{4.3}$$

where E_{anchor}, $E_{positive}$, and $E_{negative}$ represent embeddings for anchor, semantically similar, and dissimilar samples, respectively, and m is a margin hyperparameter. This ensures that relevant CPS-related texts exhibit higher similarity scores.

4.3.3 Embedding-Based Knowledge Retrieval

Once fine-tuned, the LLMs produce dense embeddings that are indexed in a vector database (e.g., FAISS or Milvus). When a user query is processed through the NLP pipeline (Section 4.2), its embedding vector is compared against these indexed vectors using cosine similarity :

$$S(Q, D_i) = \frac{E_Q \cdot E_{D_i}}{\|E_Q\| \|E_{D_i}\|} \tag{4.4}$$

where E_Q is the embedding of the user query and E_{D_i} corresponds to the i^{th} component or configuration entry in the knowledge base. The top-k results with the highest similarity scores are then passed to the recommendation engine for further ranking and reasoning.

4.3.4 Explainable Recommendations through LLM Interpretability

While LLMs provide powerful representation capabilities, their opaque decision processes can limit transparency in industrial contexts. To address this, IoT-AID integrates interpretability mechanisms that allow users to understand why certain CPS components or configurations are recommended.

These mechanisms include :

- **Attention-Based Explanation :** Visualization of attention weights to highlight which user query terms influenced the model's decision.
- **Semantic Similarity Trace :** Display of most similar historical queries or knowledge base entries that contributed to the recommendation.
- **Feature Attribution :** Gradient-based techniques (e.g., Integrated Gradients) to identify the input tokens that had the greatest impact on embedding similarity.

Such interpretability enhances user trust and aligns the system with explainable AI (XAI) principles, making IoT-AID not only functional but also transparent in its reasoning process.

Through its integration of fine-tuned LLMs, IoT-AID achieves robust semantic understanding, contextual awareness, and explainable reasoning within CPS configuration dialogues. This combination of linguistic intelligence and structured reasoning establishes the foundation for the next layer of the system : the hybrid recommendation engine, which operationalizes these embeddings to generate actionable CPS configuration suggestions.

4.4 Recommendation Engine Implementation

The rapid expansion of e-commerce platforms has resulted in an overwhelming number of product options for consumers. Therefore, providing accurate and personalized product recommendations is crucial to enhance the user experience. In this context, the Bidirectional Encoder Representations from Transformers (BERT) model is an advanced tool that has proven effective in text understanding and context extraction, making it ideal for recommendation systems.

BERT, introduced by Google in 2018, revolutionized the field of natural language processing (NLP) by offering an innovative approach to understanding textual context. Unlike traditional models that process text either left-to-right or right-to-left, BERT reads the text bidirectionally, understanding both

the preceding and following words at once. This gives BERT a deep understanding of user queries and product descriptions, making it highly suitable for personalized recommendation systems.

In a product recommendation system, the main challenge lies in interpreting the user's intent, which could involve complex, ambiguous, or multi-faceted queries. BERT's bidirectional nature and pre-trained embeddings capture these nuances more effectively than previous models, allowing the recommendation engine to match user input with relevant product attributes.

4.4.1 Data Acquisition and Structuring

To construct the initial dataset of sensors for our AI model, we needed a comprehensive list of products, including details such as price, description, and various technical specifications. After evaluating several potential data sources, we determined that scraping an e-commerce website was the most efficient approach to acquire this information. We began by analyzing multiple e-commerce platforms, assessing each one based on the quantity of sensors listed and the level of detail provided for each product. After careful consideration, we selected a website that listed 1,816 sensors, which provided extensive product details that aligned with our data requirements.

Before fully committing to scraping the entire site, we conducted a manual test to ensure the accuracy and relevance of the data. We randomly selected 20 sensors from the website and verified their descriptions and specifications. This initial validation confirmed that the website met our requirements, allowing us to proceed with the full-scale scraping process.

For scraping the data, we chose Python due to its rich ecosystem of web scraping libraries and our team's familiarity with the language in AI-related tasks. We selected Selenium tool because of its robustness and ease of integration with our web-based recommendation system. Selenium tool also facilitated future modifications and continuous development based on user feedback. To manage the code collaboratively, we employed GitHub, which allowed team members to easily access and contribute to the project.

In terms of implementation, we used the csv module in Python for creating and writing to a CSV file, and the BeautifulSoup library for web scraping. Initially, we wrote code to scrape a small subset of 20 sensors, ensuring that the scraping process worked correctly before expanding it to the entire dataset. Our first execution successfully retrieved the 20 sensors, but some of the data fields were incomplete or missing. A review of the code revealed

that additional conditions were required to handle certain edge cases and ensure that all necessary information for each sensor was captured.

For this purpose, a web scraping process was implemented using the Selenium tool, allowing automatic extraction of information from online sensor catalogs. In addition, existing CSV files were explored and then converted to JSON format in order to standardize the data sources, as shown in figure 4.3 A preprocessing and cleaning step (removal of duplicates, value normalization, format standardization) was then carried out to obtain an initial usable dataset.



FIGURE 4.3 – ETL Pipeline for Data Collection and Structuring

After multiple iterations of code adjustments and testing, we resolved these issues and successfully scraped the correct data for the first 20 sensors. However, we encountered another challenge : special characters in the sensor descriptions were not displaying properly. This required further adjustments in the code to correctly process and display special characters. After several refinements, we were able to capture and display the full, accurate data for all 20 sensors in the IDE.

In the next phase, we wrote this scraped data into a CSV file using the csv module. Once we confirmed that the process worked smoothly for the initial 20 sensors, we scaled up the operation to scrape the remaining 1,796 sensors. Given the large volume of data, the execution took several hours to complete. Once all the data had been written to the CSV file, we performed a final quality check by randomly selecting 50 sensors from the file and manually comparing their data with the information available on the website. This final verification confirmed that the scraped data was accurate and fully aligned with the original source.

By successfully scraping, validating, and compiling the sensor data into a structured CSV file, we established a reliable dataset for use in developing our AI model for sensor recommendation.

```
{
  "Name": "9MM MV OUTPUT PRESSURE SENSOR",
  "Type": "Pressure Sensors",
  "Description": "Media Isolated Pressure Sensor Supply Voltage: 5 V
    || Supply Current: 1.5 mA||||Operating Temperature Range: -40 –
    125 °C;-20 – 125 °C;-20 – 85 °C||Electrical Connection: Cable
    ;Lead Wires;Ribbon Cable;Wire Leads || Pressure Port/Fitting: 1/4
    -18 NPT;1/4-18 NPT (5/8 Hex);1/4-19 BSPP Female;7/16-20 UNF;7/16
    -20 UNF Female;7/16-20 UNF-2A;M10x1 Male;M16x1.5;None;Weldable
    (No Thread)||Proof Pressure Range: 3X || Pressure:  Output/Span:
    0 – 100 mV;125 mV;125mV;Sensitivity = 15 - 28mV/V||Media Isolated
    Pressure Sensor Type: Media Isolated mV Output Pressure Sensors
    || Media Isolated Pressure Sensor Style: Absolute;Sealed
    Gauge||Non-Linearity: ± .25 %||Dimensions: Dia 9.04 x 13.2 mm",
  "Applications": "Oil and gas; Automotive; Aerospace; HVAC systems;
    Industrial machinery; Medical devices; Process control; Water
    management; Chemical processing; ; Pneumatic systems"
}
```


FIGURE 4.4 – Sensors Dataset

4.4.1.1 Sensors Dataset

The sensors data represents the main characteristics of a sensor. These may be summarized as shown in Fig 4.4 :

- Name : The name of the product or application.
- Type/Category : The general classification or category of the product.
- Description : Detailed information about the product or application, including its features and uses.
- Applications/Domains : Different areas or industries where the product or application is commonly used.

4.4.2 Recommendation Model Architecture and Workflow

The output from the intent classification step provides crucial information regarding the application at hand and identifies the necessary sensors required for that specific application. Based on this initial classification, the system can intelligently prompt the user with follow-up questions to gather more detailed information, particularly concerning the specific sensor requirements, such as accuracy, range, or other technical specifications. Once the user provides this additional input, we utilize SBERT's capabilities to extract important keywords from the user's response, helping us identify key product features or requirements. These extracted keywords are then used to refine the recommendation process. To further improve the accuracy and relevance of recommendations, we fine-tune BERT on our specific recommendation dataset, adapting the model to understand domain-specific language and context. The system then leverages this fine-tuned BERT model to provide personalized recommendations that precisely match the user's detailed requirements and preferences.

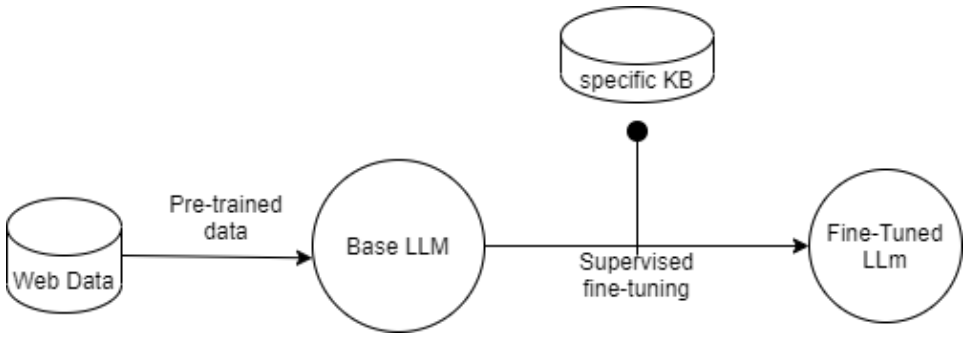


FIGURE 4.5 – Fine-tuning LLMs

The system developed here employs a BERT-based recommendation engine with a straightforward but powerful architecture. The pipeline involves the following key steps :

4.4.2.1 Data Preprocessing :

The dataset, containing product descriptions, names, types, and applications, is preprocessed to remove special characters and convert text into lowercase for normalization. The columns Name, Type, and Applications are combined into a single combined-text field that provides a consolidated representation of each product. This makes it easier for the model to handle various product attributes in a unified manner.

4.4.2.2 BERT Model for Embedding :

We use the pre-trained BERT (bert-base-nli-mean-tokens) from the Sentence Transformers library. This model generates 768-dimensional dense vectors (embeddings) for each product's combined text. These embeddings capture the contextual meaning of the product description and are used to compare against the user query.

4.4.2.3 Cosine Similarity for Matching :

When the user enters a query, we process it through the same BERT model to generate an embedding. We then use cosine similarity to measure the closeness of the user's input embedding to the product embeddings. This metric returns values between -1 (completely dissimilar) and 1 (identical), helping us rank the products based on relevance.

4.4.2.4 Top Recommendations :

After calculating the similarity scores, the system ranks the products and returns the top 10 most similar ones as recommendations. This process ensures that the user receives personalized results tailored to their input, both in terms of product features and preferred brands.

4.4.3 Results and Analysis

The BERT-based recommendation system has shown impressive performance in providing relevant and personalized product recommendations. Through the following results, the advantages of using BERT as a core engine for recommendations are clear :

- Accurate Context Understanding : BERT's deep contextual understanding allows the system to interpret complex user queries, including multiple product features and attributes (e.g., "moisture sensor for soil"). This leads to more precise matches between user needs and product offerings.
- Brand Preference Sensitivity : By incorporating brand matching, the system boosts the relevance of recommendations for users with specific brand preferences. This is particularly useful in product categories where brand loyalty is important.
- Quick Response Times : The use of pre-computed embeddings for the product catalog ensures that recommendations are generated quickly. The process involves encoding the user input in real time, followed by a quick cosine similarity check with the product embeddings, making the system highly responsive.

Despite the success of the BERT-based recommendation system, there are a few areas that could be improved in future iterations :

- Handling Ambiguous Queries : Although BERT is excellent at understanding context, ambiguous user queries (e.g., "light device") might still generate less accurate results. To address this, integrating additional user data, such as past interactions or browsing history, could help disambiguate intent.
- Fine-Tuning the BERT Model : The current system uses pre-trained BERT embeddings without domain-specific fine-tuning. In future work, we could fine-tune the BERT model on e-commerce-specific datasets or user-specific datasets to better capture product features and preferences.

- Dynamic User Feedback : Integrating a feedback loop where users can provide input on recommendations (e.g., rating or selecting preferred items) would allow the system to continuously learn and improve. Incorporating collaborative filtering elements based on user interaction data could further enhance recommendation accuracy.
- Incorporating Multimodal Data : Future work could also explore the use of additional data modalities such as product images, videos, or reviews. Combining these with textual descriptions could provide a more holistic representation of each product and further improve recommendation quality.

4.5 CPRS Text classification

The first part of the Cyber-Physical Recommendation System (CPRS) focuses on text classification and understanding the user's objectives, serving as the initial interaction point between the user and the system. This is achieved through the use of Natural Language Processing (NLP) techniques, which analyze user input.

4.5.1 DataSet Processing

In our pursuit of constructing a reliable Cyber-Physical Recommendation System (CPRS), we encountered a significant challenge : the absence of readily available data online. To address this gap, we turned to web scraping and generative AI techniques to generate the necessary data.

The dataset needed to include a variety of applications, their respective fields of activity, and the sensors used in such applications. To accomplish this, we integrated a function into our Python language based module. To avoid redundancy, we implemented another function that iterated through the JSON array, filtering out redundant entries and compiling a new file with unique elements. Once we had a refined JSON file containing around 600 unique applications, we utilized a CSV library to convert the JSON data into a CSV format.

This structured approach not only streamlined our data collection process but also ensured the uniqueness and relevance of the data for our AI model's training.

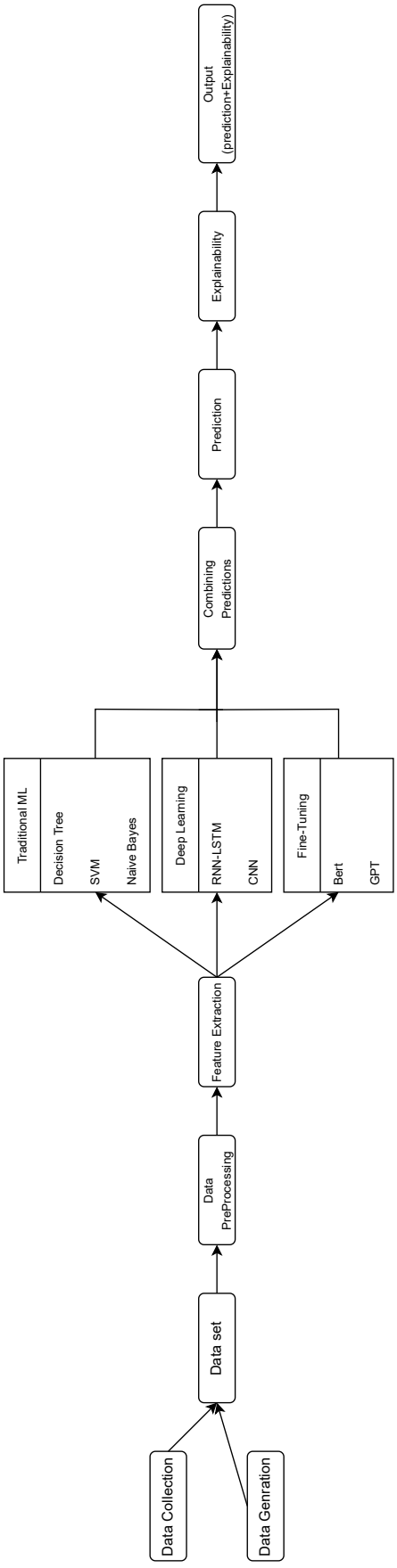


FIGURE 4.6 – CPRS Architecture.

4.5.1.1 Entities Dataset

The entities are described by the attributes shown in fig 4.7 :

- Application : This field specified the context or domain where sensors were applied.
- Sensor Types : It listed the various types or categories of sensors commonly used within the specified domain.
- Description : This section provided a detailed overview of how sensors were employed within the specified domain, including their functionalities and practical applications.
- Domain : Here, we identified the broader industry or field where these sensors found utility.

This corresponds to the data acquisition and ingestion block shown in Figure 4.6.


```
{
  "Application": "Smart Irrigation Systems",
  "Sensor Types": "Soil moisture sensors, Weather sensors, Flow
    sensors",
  "Description": "Sensors used to monitor soil moisture levels,
    weather conditions, and water flow, enabling efficient irrigation
    practices.",
  "Domain of Application": "Agriculture"
},
{
  "Application": "Health Monitoring Systems",
  "Sensor Types": "Heart rate sensors, Blood pressure sensors, Oxygen
    saturation sensors, Temperature sensors",
  "Description": "Sensors used to monitor vital signs and health
    metrics in real-time, allowing for early detection of health
    issues",
  "Domain of Application": "Healthcare"
}
```


FIGURE 4.7 – Entities Dataset

4.5.2 Data Preprocessing

Data preprocessing is a critical step in preparing the text data for machine learning models. The process involved cleaning the text by removing special characters, numbers, and stop words which do not contribute to the semantic meaning. Tokenization was performed to split the text into smaller text chunks (tokens). This was followed by lemmatization, which reduced words to their base forms, thus ensuring consistency. For instance, the raw text "I need a system for health monitoring for heart rate and oxygen levels" was preprocessed to become "need system health monitoring heart rate oxygen level". This preprocessing ensured that the text was in a suitable format for feature extraction and modeling. This step aligns with the data cleaning and normalization block highlighted in Figure 4.6.

4.5.3 Feature Extraction

Feature extraction transforms textual data into numerical representations that can be used by machine learning algorithms. We employed two primary methods : TF-IDF Vectorization and BERT Embeddings. TF-IDF (Term Frequency-Inverse Document Frequency) vectorization converted the text into numerical features by considering the importance of words in the context of the entire dataset. BERT embeddings, generated using a pre-trained BERT model, provided deep contextual understanding of the text by capturing the semantic relationships between words. For example, the preprocessed text "need system health monitoring heart rate oxygen level" was converted into TF-IDF vectors and BERT embeddings.Feature extraction corresponds to the semantic representation and embedding module depicted in Figure 4.6.

Cyber-Physical Systems (CPS) represent the integration of computational algorithms and physical components, leading to interconnected systems with advanced functionalities and capabilities [4]. CPSs make use of smart sensors, embedded systems, cloud computing, data storage, and artificial intelligence techniques. The aim is to revolutionize industries by creating smarter factories that drive the fourth industrial revolution, enabling advancements in predictive maintenance, real-time monitoring, and self-optimization [3]. CPSs are recognized as key elements of the fourth industrial revolution, facilitating the development of intelligent, resilient, and self-adaptable machines. Their widespread integration across various domains and applications positions them as one of the fastest-growing technologies. CPS configurations vary based on application domains, architecture, and connection systems, with each application necessitating specific physical components, communication networks, and computational mechanisms. However, building CPS solutions tailored to specific requirements often poses challenges for researchers and engineers due to insufficient knowledge. Automated assistance can streamline the process by enabling engineers and researchers to develop, validate, and deploy CPS solutions, thereby enhancing service quality, productivity, and reducing reliance on human expertise [1].

4.6 Conclusion

In this Chapter introduced IoT-AID, a Cyber Physical Recommendation System designed to address the challenges faced in the design and implementation of Cyber Physical Systems (CPSs) using the BERT model to handle

user queries and provide accurate recommendations based on sensor descriptions, types, and applications. The system effectively combines NLP techniques, cosine similarity, and brand matching to deliver tailored product suggestions that align with the user's input and preferences. By utilizing the capabilities of advanced language models (LLMs) such as BERT and Sentence Transformers, IoT-AID offers semantic understanding and context-aware recommendations crucial for navigating the complexities of Industry 4.0 integration. By leveraging the strengths of BERT, the system understands user intents and provides high-quality recommendations even for complex queries. However, further improvement is possible, particularly in fine-tuning the model for specific domains and incorporating real-time user feedback for enhanced personalization. This BERT-based approach marks a significant advancement in the field of recommendation systems and offers a strong foundation for future developments.

For future work, further exploration and evaluation of LLM-driven recommendation systems in the realm of CPSs is essential. This includes training data on different models such as GPT and Transformers to enhance recommendation accuracy and relevance. Additionally, extending the datasets by gathering more data is key to achieving better results and improving the overall performance of IoT-AID.

Bibliographie

[1] Mohammad CHOAIB et al. « Automated Decision Support Framework for IoT : Towards a Cyber Physical Recommendation System. » In : *ICEIS (1)*. 2023, p. 365-373.

[2] Dietmar JANNACH et al. « A survey on conversational recommender systems ». In : *ACM Computing Surveys (CSUR)* (2021).

[3] Jay LEE, Behrad BAGHERI et Hung-An KAO. « A cyber-physical systems architecture for industry 4.0-based manufacturing systems ». In : *Manufacturing letters* 3 (2015), p. 18-23.

[4] Ragunathan RAJKUMAR et al. « Cyber-physical systems : the next computing revolution ». In : *Proceedings of the 47th design automation conference*. 2010, p. 731-736.

Chapitre 5

Implementation and Evaluation of the Cyber-Physical Recommendation System

This chapter presents the implementation strategy and the evaluation framework of the developed Cyber-Physical Recommendation System (CPRS). Building upon the theoretical foundations and design considerations outlined in the preceding chapters, the focus here shifts from conceptual discussions to the concrete realization of CPRS as a functional system, followed by a systematic validation of its effectiveness.

As discussed in chapter 4, the implementation of CPRS addresses one of the critical challenges in Cyber-Physical Systems (CPS) configuration : the integration of heterogeneous sensors, actuators, and communication modules in an efficient, reliable, and scalable manner. In this chapter, we discuss how the system seamlessly conceals the complex selection process of CPS components. Conventional approaches to this task frequently rely on expert knowledge, manual trial-and-error, or the use of opaque recommendation engines that provide little to no transparency. Such practices are inherently time-consuming, error-prone, and insufficiently adapted to the requirements of Industry 4.0 and beyond, which demand scalability, explainability, and user-centered design. In contrast, CPRS introduces a modular and interpretable framework that effectively bridges the gap between natural language user requirements and the structured configuration of CPS components. To achieve this, the system integrates a combination of semantic embeddings with Sentence-BERT, intent classification, rule-based sufficiency checks, similarity-based retrieval, explainable AI methods such as SHAP, and natural language generation powered by Mistral-7B. Collectively, these techniques enable CPRS to

automate component recommendation while ensuring transparency, interpretability, and trustworthiness throughout the process.

In addition to its technical foundations, CPRS incorporates a user feedback mechanism in the form of a question-answer dialogue designed to capture perceptions of both the workflow and the quality of recommendations. This module is particularly aimed at supporting users who may lack specialized expertise in CPS configuration, thereby enhancing system accessibility and aligning outcomes with user expectations. The inclusion of this module reflects that CPRS embodies not only an engineering solution but also a user-centered tool committed to iterative refinement and continuous improvement.

The rest of the chapter is organized as follows. Section 5.1 introduces the overall system architecture. The workflow and internal modules are then detailed in Sections 5.2.1, 5.2.2, 5.2.3, and 5.2.4, with explainability and response generation presented in Sections 5.2.5 and 5.2.6. Section 5.2.7 describes the construction of the knowledge base. Finally, Section 5.3 presents the implementation environment and experimental evaluation (quantitative benchmarks and qualitative user studies), and discusses CPRS's comparative advantages, limitations, and broader implications. Through this structured presentation, we demonstrate that CPRS is not only feasible from an engineering perspective but also effective in overcoming persistent barriers in CPS configuration namely complexity, lack of transparency, and limited accessibility for non-expert users.

5.1 General System Architecture

The CPRS is designed as a modular and interpretable pipeline that translates user requirements, expressed in natural language, into structured and validated recommendations of CPS components. At its core, CPRS operates as a chatbot-based assistant, enabling users to iteratively specify and refine their needs through natural dialogue rather than technical configuration steps. Each user query is embedded into a semantic representation space, where intent classification and similarity-based retrieval are employed to identify

the most relevant CPS components, such as sensors, actuators, or communication modules, that align with the specified requirements.

The architecture follows a multi-stage workflow, in which each stage is responsible for a distinct computational task while remaining loosely coupled to ensure scalability, transparency, and adaptability. Beginning from the user's initial input, the system interprets requirements, retrieves candidate components, validates sufficiency against domain-specific rules, and progressively constructs a candidate CPS configuration. The dialogue-driven nature of the interaction allows CPRS to guide the user step by step, recommending components in sequence until a complete CPS design is achieved that satisfies both functional and contextual constraints.

At its core, CPRS operates as a conversational assistant that receives a user query in natural language, processes it through a sequence of semantic, classification, and retrieval modules, and returns a recommendation accompanied by an explanation. This process is further enhanced by a feedback loop that captures user impressions, enabling iterative improvement of the system. Each stage in the architecture was deliberately chosen to address specific limitations of existing recommendation systems, such as the cold-start problem, lack of interpretability, and poor adaptability in industrial contexts.

A distinctive feature of CPRS lies in its explainability-by-design. At each recommendation step, the system is capable of providing human-understandable justifications for its choices. These explanations, generated through techniques such as SHAP-based attribution and natural language generation, allow users to understand not only which component is being suggested but also why it has been selected over alternatives. For example, a recommended temperature sensor may be accompanied by an explanation highlighting its compatibility with the chosen communication protocol, its adequacy with the required measurement range, or its cost-effectiveness relative to other candidates. This capability not only builds user trust but also facilitates informed decision-making, particularly for non-expert users who might otherwise struggle to assess component suitability.

Figure 5.1 illustrates the high-level architecture of CPRS, emphasizing its modular design, the flow of information across components, and the central role of the chatbot interface in mediating interaction between the user and

the underlying recommendation pipeline.

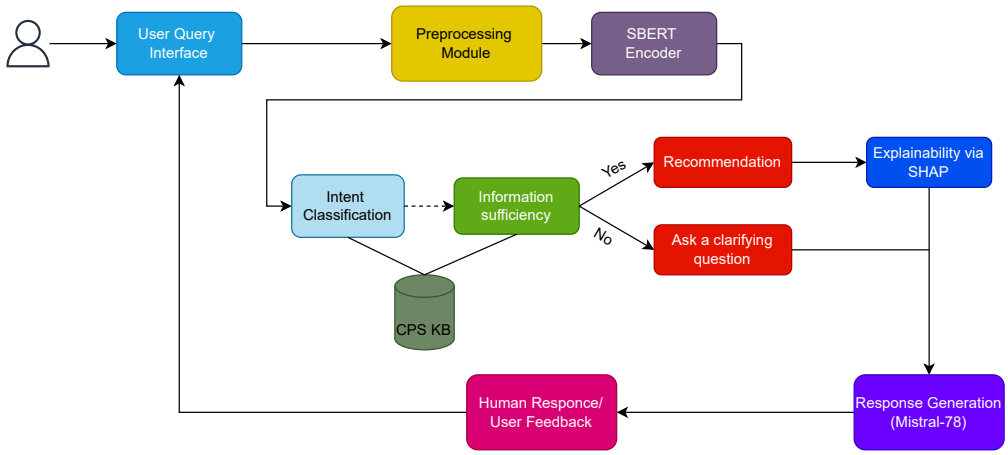


FIGURE 5.1 – General Architecture of the Cyber-Physical Recommendation System.

In this respect, the workflow of the CPRS begins with the **User Query Interface**, where users specify their requirements in natural language (e.g., "I need a sensor for monitoring methane leaks in tunnels"). Unlike conventional configuration tools, which typically demand substantial technical expertise, CPRS significantly lowers the entry barrier by enabling intuitive interaction, thereby supporting both expert and non-expert users.

The query is then processed by the **Preprocessing Module**, where standard text normalization techniques such as replacing the words with more relevant text chunks, maintaining prompt context, and unit harmonization are applied. While Sentence-BERT (SBERT) is inherently robust to textual variation, preprocessing ensures consistency between potentially noisy user inputs and the structured descriptions stored in the component knowledge base. This step is particularly important in technical domains, where minor inconsistencies (e.g., "CO-2" vs. "CO2") can compromise semantic alignment. The preprocessing pipeline is further detailed in Section 5.2.1

Subsequently, the normalized query is encoded into a high-dimensional semantic vector using **Sentence-BERT (SBERT)**. It was selected due to its capacity to balance semantic richness with computational efficiency, offering

superior performance in both accuracy and inference time compared to traditional models such as Word2Vec or standard BERT (see Section **??** for comparative benchmarks). This representation enables CPRS to capture the underlying intent of the query rather than relying solely on keyword matching.

To further refine the recommendation process, the query embedding is processed by the **Intent Classification Module**, a lightweight neural network that assigns the query to a predefined CPS application domain, such as environmental monitoring, fire detection, or healthcare. This classification step prevents irrelevant matches and ensures that retrieval is restricted to the most contextually appropriate domain, thereby improving precision.

Before component recommendations are generated, the **Information Sufficiency Module** evaluates whether the query contains all essential attributes necessary for meaningful retrieval. If key parameters such as sensing range, operational environment, or communication protocol are missing, the system dynamically generates clarifying questions to elicit additional information. Examples of such interactions are provided in Section 5.2.3. This interaction not only ensures completeness and reduces ambiguity but also enhances transparency, thereby reinforcing user trust in the recommendation process.

Once the query is sufficiently specified, the **Similarity-Based Retrieval Engine** (elaborated in Section 5.2.4) compares the query embedding with precomputed embeddings of CPS components stored in the knowledge base. Using cosine similarity, the engine identifies the top-k candidate components whose semantic representations align most closely with the user's intent. Unlike collaborative filtering or usage-history–based recommenders, this semantic retrieval approach mitigates the cold-start problem, allowing CPRS to recommend components for novel queries without dependence on prior user interactions.

Given the critical importance of transparency in industrial and safety-sensitive domains, retrieval alone is insufficient. To address this, CPRS integrates **Explainability through SHAP (SHapley Additive exPlanations)**. It quantifies the contribution of individual query terms to the similarity score, thereby offering insight into why a particular component was recommended. For example, in the methane detection query, SHAP may highlight terms such as "gas", "tunnel", and "methane" as decisive factors (see Section 5.2.5

on explainability). This functionality not only increases user trust but also ensures compliance with emerging regulatory frameworks that mandate explainable AI in industrial contexts.

To further enhance usability across a diverse user base for both the technical and/or domain experts and non-experts, CPRS employs Mistral-7B for **Response Generation**, , as discussed in Section 5.2.6. This large language model synthesizes the selected component, its associated metadata, and the SHAP rationale into a fluent, natural-language explanation. Instead of presenting a purely technical datasheet, CPRS produces accessible justifications such as : "The MQ-9 gas sensor is recommended due to its high methane sensitivity and robust casing, making it suitable for underground tunnel environments." Such contextualized explanations enable non-expert users to make informed decisions while still providing sufficient detail for expert users.

Finally, CPRS integrates an optional **User Feedback Module**, which closes the interaction loop. After engaging with the system, users are invited to complete a structured questionnaire (see Section 5.2.12) assessing both the quality of the recommendations and the transparency of the overall workflow. Importantly, this evaluation framework accounts for the fact that many users may lack domain expertise; their responses therefore reflect dimensions such as usability, trustworthiness, and perceived helpfulness rather than purely technical accuracy. The insights obtained through this mechanism provide a foundation for personalization and continuous refinement of CPRS's models and interaction strategies.

The CPRS architecture operationalizes a coherent pipeline that unites semantic intelligence, interpretability, and user-centered interaction. Each module has been purposefully designed to address specific shortcomings of existing CPS configuration methods, and their integration results in a system that is not only technically effective but also practical, transparent, and accessible for deployment in real-world industrial contexts.

5.2 Workflow and Module Descriptions

The CPRS pipeline is organized as a sequence of interconnected yet modular stages, with each stage dedicated to a specific function within the recommendation process. This modular design confers both scalability and extensibility, allowing the system to evolve in response to new requirements or technological advances. For instance, the current embedding mechanism based on SBERT could be replaced by a domain-specific language model to improve representation quality, while the intent classification layer can be retrained to incorporate additional CPS domains as the application scope expands.

The modularity also facilitates independent evaluation and optimization of individual components without disrupting the entire workflow. Each stage contributes to the overall objective of transforming user requirements expressed in natural language into transparent, validated, and context-appropriate recommendations for CPS components.

This structured presentation highlights not only the technical mechanisms but also the theoretical rationale that guided the design of CPRS, thereby demonstrating how the integration of individual modules enables a coherent and effective recommendation system. In the subsections that follow, each module of the pipeline is described in detail.

5.2.1 Preprocessing and Embedding Preparation

The first stage of the CPRS workflow is text preprocessing, which prepares both user queries and component descriptions for semantic embedding. Modern embedding models such as Sentence-BERT (SBERT) are typically pretrained on whole sentences, and generally can produce robust semantic representations without classical preprocessing techniques such as normalization, stemming, or lemmatization. Initially, it was hypothesized that SBERT would perform optimally on raw sentence-level inputs. However, due to the limited availability of CPS-specific training data and the highly formalized structure of CPS component descriptions, it was necessary to apply these classical preprocessing steps to ensure consistent alignment between user queries and component metadata. Manual evaluation confirmed that incorporating normalization, tokenization, and stemming/lemmatization

yielded superior retrieval performance compared to using raw sentences alone, thereby justifying the inclusion of these preprocessing steps within CPRS.

Preprocessing ensures a consistent representation of text across different inputs, which is particularly critical in technical domains such as CPS, where minor inconsistencies (e.g., "CO-2 sensor" vs. "CO2 sensor") can otherwise degrade retrieval accuracy. The preprocessing pipeline consists of the following steps :

1. **Markup removal :** Removal of HTML/XML tags and extraneous markup from raw textual data.
2. **Tokenization :** Segmentation of sentences into tokens (words or other meaningful units).
3. **Normalization :** Conversion of tokens to lowercase, removal of redundant whitespace, and resolution of hyphenation inconsistencies.
4. **Stemming or lemmatization :** Reduction of inflected words to their base forms. In CPRS, a Snowball stemmer is employed to improve alignment across user queries and component descriptions.

Formally, given a raw text description of CPS component d, preprocessing may be represented as :

$$toks = \text{Stem}\big(\text{Normalize}\big(\text{Tokenize}\big(\text{StripMarkup}(d)\big)\big)\big), \tag{5.1}$$

where $toks$ denotes the cleaned token sequence associated with the CPS component d.

Once preprocessed, the resulting tokens are passed to SBERT to generate semantic embeddings. Each query or CPS component description d is mapped to a dense vector representation :

$$\mathbf{v}_d = f_{\text{SBERT}}(toks), \tag{5.2}$$

where f_{SBERT} is the pretrained SBERT encoder and $\mathbf{v}_d \in \mathbb{R}^{768}$ represents the resulting semantic embedding. These embeddings form the foundation for downstream similarity-based retrieval, which is discussed in Section 5.2.4.

Input : Corpus of CPS components descriptions D
Output : Embedding index Emb
$T \leftarrow [\,]$ `// Initialize token list`
for *each CPS component description* d *in* D **do**
  $txt \leftarrow strip_HTML_and_markup(d.text)$; $toks \leftarrow tokenize(txt)$;
  $toks \leftarrow normalize(toks)$ `// lowercasing, whitespace cleanup, etc.`
  $toks \leftarrow SnowballStemmer(toks)$; T.append$((d.id, toks))$;
end
$Emb \leftarrow$ `// Initialize embedding dictionary`
for *each* $(id, toks)$ *in* T **do**
  $vec \leftarrow SBERT.encode(toks)$; $Emb[id] \leftarrow vec$;
end
$build_cosine_index(Emb)$ `// Construct approximate nearest-neighbor (ANN) index`

Algorithme 1 : Preprocessing and Embedding Preparation in CPRS

Algorithm 1 illustrates the preprocessing and embedding stage of CPRS. Each document (or user query) is transformed into a normalized token sequence, encoded into a semantic embedding using SBERT, and inserted into a cosine similarity index to support efficient retrieval.

5.2.2 Intent Classification

After user queries have been preprocessed and transformed into semantic embeddings using SBERT, the next step in the CPRS workflow is to determine the underlying application domain, or intent, of the query. Intent classification is a critical component because CPS configurations differ significantly across domains such as environmental monitoring, fire detection, healthcare, and industrial automation. Correct identification of the domain ensures that the retrieval engine focuses exclusively on the relevant subset of the knowledge base, thereby improving both the precision and efficiency of recommendations.

Formally, given a query embedding $\mathbf{v}_q$, the goal of intent classification is to assign the query to one of K predefined CPS domains. A shallow feed-forward neural network with ReLU-activated hidden layers and a Softmax output layer is employed to perform this mapping :

$$\mathbf{h}_1 = \mathrm{ReLU}(W_1\mathbf{v}_q + b_1),\ \mathbf{h}_2 = \mathrm{ReLU}(W_2\mathbf{h}_1 + b_2),\ \hat{\mathbf{y}} = \mathrm{Softmax}(W_o\mathbf{h}_2 + b_o), \tag{5.3}$$

where W and b represent the weight matrices and bias vectors of the classifier, and $\hat{\mathbf{y}} \in \mathbb{R}^K$ is the resulting probability distribution over the K possible intents. The predicted intent corresponds to the class with the highest probability :

$$\text{Intent}(q) = \arg\max_i \hat{y}_i. \tag{5.4}$$

The classifier is trained using categorical cross-entropy loss, defined as :

$$\mathcal{L} = -\sum_{i=1}^{K} y_i \log(\hat{y}_i), \tag{5.5}$$

where y denotes the one-hot encoded ground-truth label of the query.

This design was selected for two main reasons. First, a shallow network provides low computational overhead, which is essential for supporting real-time, interactive queries in the CPRS chatbot interface. Second, the classifier acts as an early filtering mechanism, restricting subsequent similarity-based retrieval to the relevant domain. This both reduces the risk of mismatched recommendations and enhances interpretability, as each query is explicitly associated with an application context.

Input : Query embedding $\mathbf{v}_q$

Output : Predicted intent label *intent*

$W, b \leftarrow load_trained_classifier()$; $z \leftarrow W \cdot \mathbf{v}_q + b$; $\hat{y} \leftarrow softmax(z)$

`// Probability distribution over intents`

$intent \leftarrow \arg\max(\hat{y})$ `// Select highest-probability intent`

return *intent* ;

Algorithme 2 : Intent Classification in CPRS

Algorithm 2 illustrates how a user query embedding is processed through the trained classifier to determine its most likely CPS domain. The resulting predicted intent not only guides the similarity retrieval stage but also contributes to the interpretability of system outputs, providing users with explicit context about the scope and relevance of recommendations.

5.2.3 Information Sufficiency Module

Even after intent classification, user queries may remain too vague or incomplete to support meaningful recommendations. For instance, a query such as “I need a temperature sensor” does not specify critical details such as operating environment, measurement range, or accuracy requirements parameters that can significantly influence which component is most suitable.

Proceeding directly to similarity-based retrieval under such conditions may yield imprecise or irrelevant recommendations.

To address this challenge, CPRS incorporates an Information Sufficiency Module. This stage evaluates the completeness of a query and determines whether additional clarifications are necessary. When essential attributes are missing, the module generates targeted follow-up questions to iteratively refine the query until sufficient context is available for reliable recommendation.

Formally, let a query q be represented by its semantic embedding $\mathbf{v}_q$ and predicted intent label $\hat{y}$. Each intent category $\hat{y}$ is associated with a predefined set of required attributes $A_{\hat{y}}$, such as range, environment, or communication protocol, etc. The sufficiency check is defined as :

$$\text{Sufficient}(q) = \begin{cases} 1, & \text{if } \forall a \in A_{\hat{y}},\ a \in q \\ 0, & \text{otherwise} \end{cases} \tag{5.6}$$

If $\text{Sufficient}(q) = 0$, the system generates clarification prompts to elicit the missing information. Examples include :

- "Could you specify the environment in which the sensor will be deployed ?"
- "Do you need a wired or wireless communication protocol ?"

This rule-based sufficiency check balances two objectives :

1. **Accuracy :** ensuring that recommendations are based on all necessary parameters.
2. **User support :** guiding non-expert users through the decision-making process without overwhelming them with technical jargon.

Input : Query q, intent $\hat{y}$
Output : Decision (sufficient/insufficient), clarification prompt if needed
$A \leftarrow required_attributes(\hat{y})$ `// Attributes required for this intent`
$provided \leftarrow extract_attributes(q)$
if $provided \supseteq A$ **then**
  return "sufficient"
end
else
  $missing \leftarrow A \setminus provided$
  $prompt \leftarrow generate_clarification(missing)$
  return "insufficient", $prompt$
end

Algorithme 3 : Information Sufficiency Check in CPRS

The need for an information sufficiency check arises directly from the heterogeneity of CPS components. A sensor or actuator cannot be reliably recommended without knowledge of its intended operational context. Key attributes may include :

- **Application context :** Is the sensor intended for industrial, environmental, healthcare monitoring, or other domains ?
- **Measurement range :** A sensor for room temperature ($-10°C$ to $50°C$) is not appropriate for industrial furnaces ($> 1000°C$).
- **Environmental conditions :** Does the sensor need to be waterproof, dust-resistant (IP67/68), or radiation-tolerance ?
- **Accuracy and precision :** requirements vary depending on the criticality of measurements ; some CPS tasks (e.g., medical monitoring) require high precision, while others (e.g., weather monitoring) tolerate broader error margins.
- **Communication protocols :** Components may support different protocols (LoRa, ZigBee, WiFi, Modbus, etc.), and compatibility with the target system is essential.
- **Power and deployment constraints :** low-power consumption may be more critical than sensing accuracy ; particularly, for battery-powered or remote IoT devices.

The sufficiency module encodes such requirements as attribute templates per intent category. For example :

- Gas Detection (Industrial Safety) → { gas type, environment, deployment context (indoor/outdoor), range, communication protocol }.
- Healthcare Monitoring → { vital parameter (heart rate, oxygen, motion), precision level, measurement range }.
- Environmental Monitoring → requires { variable (temperature, humidity, air quality), expected range, deployment context }.

By comparing the attributes extracted from the user query with these templates, the system ensures that no critical decision-making factor is overlooked.

Importantly, the sufficiency module also serves as a dialogue manager. When attributes are missing, it generates clarification prompts tailored to the detected intent. Examples include :

- Query : "I need a temperature sensor."
  Prompt : "Do you require the sensor for indoor or outdoor use ?"
- Query : "I need to detect vibration in machines."
  Prompt : "Could you specify the range of vibration frequencies you want to monitor ?"

This interactive refinement step addresses two critical challenges :

1. **Reducing ambiguity :** Prevents incomplete queries from leading to misleading recommendations.
2. **Empowering non-experts :** Guides users who lack detailed CPS knowledge through the decision-making process with clear natural language prompts.

Thus, the Information Sufficiency Module not only safeguards technical correctness but also enhances user trust and engagement, as confirmed in the user studies where participants reported high satisfaction with the system's ability to clarify missing requirements.

5.2.4 Similarity-Based Retrieval

Once a user query has been preprocessed, classified, and verified as sufficient, the CPRS proceeds to the similarity-based retrieval stage. At this point, the system ranks CPS components in the knowledge base according to their semantic alignment with the user query. Each component c is represented by a precomputed SBERT embedding $\mathbf{v}_c$, and the query embedding is denoted

as $\mathbf{v}_q$. The similarity between the query and a candidate component is measured using cosine similarity :

$$\text{sim}(\mathbf{v}_q, \mathbf{v}_c) = \frac{\mathbf{v}_q \cdot \mathbf{v}_c}{\|\mathbf{v}_q\| \, \|\mathbf{v}_c\|}, \tag{5.7}$$

where $\cdot$ denotes the dot product and $\|\cdot\|$ is the Euclidean norm. Components are ranked according to their similarity scores, and the top-k candidates are returned to the user.

To support interactive latency for real-time queries, CPRS leverages an approximate nearest-neighbor (ANN) index, which allows rapid retrieval of the most semantically similar embeddings. Furthermore, a re-ranking step can prioritize candidates whose structured attributes match the clarified constraints obtained from the Information Sufficiency Module (e.g., operating range, communication protocol, or environmental suitability). This hybrid approach ensures that recommendations are both semantically relevant and technically feasible.

Input : Query embedding $\mathbf{v}_q$, ANN index Idx, component embeddings Emb, number of results k

Output : Ranked list of components R

$C \leftarrow Idx.\text{topK}(\mathbf{v}_q, k)$ `// Retrieve top-k candidates via ANN search`

$R \leftarrow \text{sort}(C, \text{ by } = \text{cosine}(\mathbf{v}_q, Emb[c]))$ `// Re-rank by exact cosine similarity`

return R

Algorithme 4 : Similarity-Based Retrieval of Top-k Components

Algorithm 4 illustrates the process by which the CPRS retrieves and ranks candidate components. By combining semantic similarity with attribute-based re-ranking, the system ensures that recommendations are both contextually relevant and compliant with user-specified constraints, forming a robust bridge between natural language queries and CPS component selection.

5.2.5 Explainability Module via SHAP

A key limitation of many AI-based recommendation systems, particularly in industrial domains, is their opacity. When users receive recommendations without a clear rationale, trust is undermined and adoption suffers. This issue is especially critical in CPS, where incorrect or unexplained recommendations can lead to costly or even hazardous outcomes. To mitigate this risk,

CPRS integrates explainability directly into its architecture through the use of SHapley Additive exPlanations (SHAP).

SHAP is a model-agnostic explainability framework grounded in cooperative game theory. It quantifies the contribution of each input feature to the model's output by computing a Shapley value. For a given feature j, the SHAP value is :

$$\phi_j = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f(S \cup \{j\}) - f(S) \right], \tag{5.8}$$

where F is the set of all input features and $f(S)$ is the model prediction given subset S. Intuitively, ϕ_j measures the marginal impact of adding feature j to different coalitions of features.

In the context of CPRS, SHAP is applied at two stages of the pipeline :

1. **Intent Classification :** query tokens most influenced the assignment to a CPS domain. For instance, in the query "monitor elderly fall risks", SHAP may identify "elderly" and "fall" as the dominant contributors to classification into the healthcare monitoring domain.
2. **Similarity-Based Retrieval :** attributing which tokens or phrases contributed most to the cosine similarity with a retrieved component. For example, in the query "gas leak detection in tunnels", SHAP may identify "gas" and "tunnels" as the key attributes of a methane gas sensor recommendation..

Dual Purpose of Explainability The integration of SHAP into CPRS serves two complementary goals :

- **User-facing transparency :** Explanations foster trust by showing users why a component was recommended, allowing them to validate that their requirements were correctly interpreted. Instead of receiving a black-box recommendation, they receive a human-readable justification, which was highly valued in the user study.
- **System-level diagnostics :** Explanations also act as a tool for system designers. SHAP outputs reveal how the model reasons, making it easier to detect misalignments. For instance, if SHAP consistently assigns importance to irrelevant tokens (e.g., "the," "system") , this indicates preprocessing issues. Similarly, if domain-specific attributes like "range" or "protocol" are systematically underweighted, the embedding model or knowledge base may require augmentation.

This dual role ensures that CPRS functions not only as a practical recommender but also as a research instrument for iterative improvement. Unlike traditional opaque systems, CPRS provides a transparent audit trail of its reasoning process.

Input : Query embedding $\mathbf{v}_q$, retrieved components $\{c_i\}$, retrieval model f
Output : Explanations for top-k recommendations
for *each component* c_i *in top-k results* **do**
  $features \leftarrow tokenize(query_text)$
  $shap_values \leftarrow SHAP(f,\ features)$
  $explanation \leftarrow select_top_features(shap_values)$
  $store(c_i,\ explanation)$
end
return explanations

Algorithme 5 : SHAP-Based Explanation Generation

As illustrated in algorithm 5, by embedding SHAP directly into the CPRS pipeline, the system ensures that recommendations are not only accurate but also auditable. This design aligns with industrial requirements for safety and accountability, while advancing the broader goals of Explainable AI (XAI) in building trustworthy human–AI collaboration.

5.2.6 Response Generation (Mistral-7B)

After candidate components have been retrieved and SHAP-based explanations computed, the final challenge is to present results in a form that is clear, concise, and accessible. The raw outputs of similarity scores, SHAP values, or metadata fields are highly technical and unsuitable for direct presentation to non-expert users. To bridge this gap, CPRS integrates Mistral-7B, a state-of-the-art large language model (LLM), as the response generation module.

The role of Mistral-7B is to translate structured outputs into human-readable explanations. This module takes as input :

1. Retrieved component metadata (e.g., sensor name, type, range, communication protocol).
2. SHAP-based highlights (terms most influential in the recommendation).
3. The original user query for the context.

From these inputs, Mistral-7B generates a natural-language response to attempt that justifies the recommendation. For example :

- **Query :** “I need to detect gas leaks in tunnels.”

- **System Output :**
  - — Component : MQ-9 Gas Sensor
  - — Metadata : High methane sensitivity, IP67 enclosure, wide operating range
  - — SHAP : "gas," "tunnels"
- **Generated Response :** "Based on your requirement for gas leak detection in tunnels, the MQ-9 sensor is recommended. It is designed with high methane sensitivity and it has a robust IP67-rated enclosure suitable for underground conditions, and matches your specified environment."

This transformation is critical because it allows CPRS to serve both user groups :

- **Non-expert users,** who require plain-language explanations without needing technical jargon.
- **Expert users,** who gain value from the inclusion of SHAP highlights and explicit links between requirements and system attributes.

Why Mistral-7B ? Several alternatives were evaluated for response generation, including template-based responses and smaller transformer models. However, Mistral-7B was selected for the following reasons :

1. **Fluency and adaptability :** Generates explanations that are natural and context-sensitive, unlike rigid templates.
2. **Multi-input integration :** Combines metadata, SHAP rationales, and user intent into a coherent response, where rule-based methods would struggle.
3. **User-Centric Design :** Reduces unnecessary jargon while retaining essential technical details for transparency.
4. **Consistency :** Compared to other large language models (like GPT-4), Mistral-7B achieves a practical trade-off between response quality and computational efficiency, making it suitable for real-time CPS recommendation scenarios.

Input : Query q, retrieved component c, SHAP explanation e
Output : Natural language response R
$context \leftarrow \{query: q,\ component: c.metadata,\ explanation: e\}$
$prompt \leftarrow format_prompt(context)$
$R \leftarrow Mistral7B.generate(prompt)$
return R

Algorithme 6 : Response Generation with Mistral-7B

This algorithm 6 illustrates how the response generation module integrates structured outputs into a conversational explanation. The `format_prompt` function ensures that the model receives the query, component metadata, and SHAP rationale in a consistent format.

In respect of further advantages of Natural-Language Generation, Mistral-7B adds several benefits to CPRS :

- **Accessibility :** Converts complex reasoning into explanations understandable by non-technical users.
- **Transparency :** Embeds SHAP highlights to make the reasoning process visible.
- **Usability :** Improves interaction quality ; in the user study, 85
- **Trustworthiness :** Links recommendations explicitly to both user requirements and component features, demonstrating grounded reasoning.

By integrating Mistral-7B, CPRS ensures that its final stage is not only technically accurate but also user-centered. This completes the pipeline by transforming similarity-based retrieval and attribution into explanations that support decision-making, validation, and trust.

5.2.7 Knowledge Base Construction

At the heart of the Cyber-Physical Recommendation System (CPRS) lies the knowledge base of CPS components. This knowledge base forms the foundation for semantic retrieval, intent classification, and explanation generation. Its construction is a critical step in the implementation process, since CPS metadata is highly heterogeneous, dispersed across multiple formats, and rarely available in structured datasets.

5.2.8 Data Acquisition

The knowledge base has been built using a hybrid acquisition strategy combining automated and manual methods :

1. **Web Scraping**
   Automated scripts based on `BeautifulSoup` and `Selenium` are employed to extract sensor specifications from vendor websites such as SparkFun, Seeed Studio, and other electronic component distributors. Scraping targeted fields including sensor name, measurement type, operating range, accuracy, communication protocols, and environmental ratings.

2. **Technical Document Parsing**
   Many CPS components are described only in PDF datasheets or technical manuals. We use `PyMuPDF` to extract text sections and a lightweight instruction-following model (LLaMA 3 via the Ollama inference engine) to identify relevant metadata. A page-chunking strategy ensured contextual consistency when extracting attributes such as accuracy or power consumption.

3. **Manual Curation**
   For highly specialized sensors, manual curation is essential to validate entries and ensure correctness. This step has been particularly important to align scraped values (e.g., "0–100 ppm methane detection") with standardized units and consistent formatting.

4. **Generative Augmentation**
   To enrich the dataset and simulate additional application–sensor pairs, structured prompts are passed to a generative model (Mistral 7B). This step produces synthetic but realistic use cases, which has been carefully filtered to remove redundancy and to ensure technical plausibility. While these augmentations improved coverage, they are explicitly tagged as synthetic to avoid conflating them with vendor-verified metadata.

   Synthetic entries are explicitly flagged in the KB to distinguish them from verified specifications, preserving transparency while allowing retrieval experiments on enriched datasets.

To mitigate inconsistencies and prevent vendor-site blocking, scraping is performed with rate-limiting, caching, and schema alignment scripts. Extracted values are validated through unit normalization (e.g., converting ppm vs mg/m^3), ensuring interoperability across heterogeneous sources.

5.2.9 Data Structuring

After acquisition, data is normalized into a structured schema stored in `MongoDB Atlas`. Each entry includes the following key fields :

- Name : Sensor or component identifier.
- Type : Measurement modality (temperature, gas, vibration, etc.).
- Description : Concise explanation of functionality.
- Operating Range : Minimum and maximum measurable values.
- Accuracy/Precision : Reported measurement accuracy.

- Communication Protocols : Supported interfaces (I^2C, UART, ZigBee, LoRa, WiFi, Modbus, etc.).
- Environmental Constraints : Tolerances such as humidity, temperature extremes, or IP ratings.
- Domain Tags : Application domains (healthcare, environmental monitoring, industrial safety).
- Source : URL or datasheet reference.

Below is an example of the data instance, for the sake of better understanding. *An example JSON entry :*

```
{
  "name": "MQ-9 Gas Sensor",
  "type": "Gas",
  "description": "Methane and carbon monoxide detector with analog and digital o
  "range": "10-1000 ppm",
  "accuracy": "±5%",
  "protocol": ["Analog", "UART"],
  "environment": "Indoor, industrial",
  "domain": ["Industrial Safety", "Environmental Monitoring"],
  "source": "https://www.sparkfun.com/datasheets/Sensors/MQ-9.pdf"
}
```

5.2.10 Embedding Preparation

All component descriptions are preprocessed (tokenization, normalization, lemmatization) and encoded into SBERT embeddings. Embeddings are generated offline and indexed using cosine similarity for efficient retrieval at runtime. This design eliminates recomputation overhead, thereby it reduces query latency.

Dataset Statistics The final dataset consisted of approximately 600 unique sensor entries across four main domains :

- Environmental monitoring (temperature, humidity, air quality).
- Industrial safety (gas, vibration, pressure).
- Healthcare monitoring (vital signs, motion sensors).
- Consumer IoT (motion, light, sound).

This coverage ensures that CPRS can handle diverse queries while remaining extensible as new devices are added.

5.2.11 Advantages of the Hybrid Approach

By constructing a structured and semantically enriched knowledge base, CPRS provides the foundation for reliable recommendations. Unlike black-box recommenders that depend on user–item interaction logs, CPRS leverages domain-specific metadata aligned with semantic embeddings, making it robust to cold-start problems and adaptable across CPS domains. Some of the advantages of this hybrid approach are, as follows :

- **Comprehensiveness :** Web scraping ensured large-scale coverage, while manual curation preserved accuracy.
- **Domain Relevance :** Generative augmentation expanded the dataset with realistic scenarios, which are not always documented in vendor catalogs.
- **Scalability :** Storing data in MongoDB provided schema flexibility, which can accommodate if any new component attributes emerge.
- **Consistency :** Preprocessing and normalization ensured that queries and metadata were aligned in the same semantic space, which improves retrieval accuracy.

5.2.12 Transparency and User Feedback

Transparency is a cornerstone of CPRS, ensuring that users not only receive accurate recommendations but also understand the rationale behind them. While SHAP-based explanations and natural language responses address part of this challenge, CPRS goes further by incorporating a user feedback mechanism. This feature closes the loop between the system and its users, fostering both trust and continuous improvement.

At the end of each interaction, users are prompted with a short questionnaire designed to capture their impressions, as given below :

1. **Clarity of the workflow :** Did the step-by-step interaction (clarification questions, explanations, recommendations) feel intuitive ?
2. **Usefulness of the recommendation :** Was the suggested component aligned with the intended application ?
3. **Transparency and trust :** Did the explanation (via SHAP and natural language generation) clarify why the recommendation was made ?
4. **Overall satisfaction :** How would they rate the interaction experience as a whole ?

Feedback responses are stored and linked to each query and recommendation session. Beyond evaluation, these responses serve as metadata for system refinement for improving CPRS performance. For example, repeated reports of confusion in gas detection queries may indicate a need to improve attribute sufficiency checks or enrich the knowledge base with clearer metadata.

The inclusion of the feedback questionnaire serves a dual purpose :

- **For users :** It enhances transparency, builds trust, and reassures them that their experience is valued.
- **For system designers and researchers :** It provides actionable data to refine the pipeline, debug retrieval and explanation issues, and guide architectural improvements.

In the user study (see section 5.3), this mechanism proved particularly valuable. Participants appreciated the ability to provide feedback on system clarity and recommendation quality, with 90% agreeing that SHAP explanations improved their understanding and 85% rating the chatbot interface as intuitive. The feedback mechanism thus not only validated CPRS but also demonstrated its potential for iterative improvement through user involvement.

By combining SHAP explanations, natural language responses, and structured user feedback, CPRS achieves a high level of transparency that benefits both end-users and developers. This feedback-driven design ensures that the system remains trustworthy, user-centered, and adaptable to future industrial contexts.

A central design principle of CPRS is accessibility for non-expert users, who often lack the technical expertise required for CPS configuration. For these users, the feedback mechanism provides reassurance and agency :

- It acknowledges their role in system evaluation, empowering them in the decision-making process.
- It allows them to report unclear or incomplete recommendations, even without technical terminologies.
- It helps researchers identify areas where the conversational interface or explanation modules require simplification.

This design directly addresses one of the most pressing barriers to CPS adoption : the knowledge gap between system capabilities and user expertise.

5.3 Implementation and Results

The previous section 5.2.7 detailed the architecture and workflow of the Cyber-Physical Recommendation System (CPRS), outlining its modular design and the rationale behind each component. This section shifts the focus more from conceptual design to practical implementation and evaluation of the proposed system. The goal is to demonstrate how CPRS is implemented as a functional prototype, how its modules integrate into a coherent workflow, and how the system performs in terms of both technical efficiency and user-centered usability.

Later, in this section, we present the results of evaluation, combining quantitative and qualitative perspectives. Quantitative experiments compare alternative embedding systems (Word2Vec, BERT, SBERT) in terms of accuracy, latency, and memory footprint. Qualitative evaluations assess user trust, interpretability, and satisfaction through a structured feedback questionnaire. Taken together, these findings provide a comprehensive view of the strengths of CPRS, its limitations, and its comparative advantages over existing approaches.

5.3.1 Implementation Environment

The CPRS is implemented as a modular software platform designed with scalability, transparency, and interactivity in mind. It integrates multiple AI and data-processing components into a unified pipeline, combining a backend optimized for efficient retrieval and explainability and a frontend tailored for conversational interaction.

The core system is implemented using **flask**, a Python-based web framework. flask is chosen because of its ability to combine :

- seamless integration of machine learning modules with database operations,
- built-in scalability for deployment in industrial environments,
- mature support for REST APIs, which facilitate the conversational frontend user interface.

This ensured a balance between research prototyping flexibility and production-level robustness.

The knowledge base of CPS components is stored in **MongoDB Atlas**, a cloud-hosted NoSQL database. MongoDB's schema flexibility is essential, as different CPS components exhibit highly heterogeneous attributes (e.g., some sensors provide gas detection ranges, while others specify vibration frequencies).

Data is organized into three main collections :

- **Sensors :** Core sensor metadata (e.g., type, measurement range, accuracy, supported protocols).
- **Needs :** User requirements and natural-language queries collected during interactions.
- **CPS_Configurations :** Links between application contexts and recommended sensor/component configurations.

5.3.2 Embedding Preparation

All textual descriptions in the knowledge base underwent preprocessing (lowercasing, tokenization, lemmatization) before encoding with **Sentence-BERT (SBERT)**.

Normally, SBERT is pre-trained on full sentences, and one might expect it to perform well without additional preprocessing such as normalization, stemming, or lemmatization. However, two challenges motivated the adoption of classical text-preprocessing steps :

- the scarcity of domain-specific training data in CPS,
- the highly formal and technical descriptions of sensor metadata.

Empirical comparison confirmed that incorporating classical preprocessing improved retrieval accuracy compared to raw SBERT embeddings.

The processed embeddings, represented as 768-dimensional dense vectors, are precomputed offline and indexed in **FAISS** for efficient cosine-similarity retrieval. User queries are encoded at runtime, with average embedding computation time of $\sim$90 ms, introducing minimal latency while ensuring semantic fidelity.

5.3.3 Explainability and Response Generation

Explainability module is embedded directly into the pipeline using the **SHAP** library. Since cosine similarity is not directly differentiable, a lightweight surrogate model is trained to approximate the retrieval scores. SHAP

values are then computed with respect to this surrogate, highlighting influential tokens in each query.

The explanations are combined with sensor metadata and passed to the **Mistral-7B** language model to generate fluent, human-readable responses. This two-stage mechanism ensured that recommendations are not only accurate but also interpretable to both expert and non-expert users.

A lightweight chatbot frontend user interface is implemented to provide the interaction layer. It :

- accepts free-text user queries in natural language,
- invokes the **Information Sufficiency Module** to ask clarifying questions when input is incomplete,
- presents recommendations enriched with SHAP-based explanations and Mistral-7B-generated natural language responses.

The interface has been tested with both technical and non-technical users to evaluate accessibility across multiple expertise levels.

5.3.4 Dataset and Experimental Setup

The technical evaluation of the Cyber-Physical Recommendation System (CPRS) was designed to assess the effectiveness of IoT-AID's semantic retrieval mechanisms, its embedding backbone, and the overall quality of its recommendations under realistic conversational constraints. Since no standardized benchmark dataset exists for CPS configuration tasks, a custom evaluation dataset was constructed to reflect real-world diversity while ensuring domain relevance. The dataset was built from three complementary sources :

1. **Scraped Vendor Data :** Metadata collected from online electronics vendors (e.g., SparkFun, Seeed Studio).
2. **Technical Datasheets :** Parsed using PyMuPDF, with validated extraction of structured attributes (units, ranges).
3. **Synthetic Query–Sensor Pairs :** Generated via structured prompting with LLM, followed by manual filtering to remove redundant or implausible cases.

The final dataset includes approximately :

- $\sim$1000 user queries across healthcare, environmental monitoring, industrial IoT, and consumer applications.

- ∼2000 annotated CPS component descriptions with validated attributes.

All experiments are conducted on a workstation with Intel i7 CPU, 32 GB RAM, and an NVIDIA RTX 3080 GPU. Principally, three embedding models are compared : Word2Vec, vanilla BERT, and SBERT. Each query was tested against the full knowledge base, with top-$k = 5$ recommendations retained. In this respect, SHAP explanations and Mistral-7B responses are only included in user trials, not in baseline embedding comparisons.

Evaluation combined retrieval quality with system usability. Metrics included :

- **Accuracy, F1-score, MSE** for retrieval performance.
- **Latency and memory usage** for computational efficiency.
- **Qualitative user satisfaction**, assessed via questionnaires.

The initial experiments compared these three embedding backbones. Results are summarized in Table 5.1.

TABLE 5.1 – Performance comparison of embedding systems.

Model	Accuracy	F1-Score	MSE	Latency (ms)	Memory (MB)
Word2Vec	0.62	0.68	0.42	45 ± 5	150
BERT	0.74	0.78	0.31	120 ± 15	820
SBERT	0.85	0.85	0.18	90 ± 10	650

The results demonstrate that **SBERT offers the best trade-off**, outperforming Word2Vec and BERT in semantic accuracy while maintaining reasonable computational cost.

Word2Vec, while computationally lightweight, fails to capture contextual semantics, leading to poor retrieval accuracy and fails to capture contextual meaning. It produces embeddings at the word level by training on word co-occurrence statistics. For example, the word “gas” may be embedded identically whether it refers to methane detection or gasoline flow. In this context, sentence or document embeddings are typically obtained by averaging word vectors. In the experiments (see table 5.1), Word2Vec achieved the lowest accuracy (0.62) and highest error (MSE = 0.42).

BERT embeddings perform better but incur significantly higher latency and memory usage due to their lack of optimization for sentence similarity.

It generates contextual embeddings using a transformer encoder. However, vanilla BERT is not optimized for semantic similarity : embeddings are taken from the `[CLS]` token or averaged across tokens, which often does not preserve sentence-level meaning well.

SBERT extends BERT by fine-tuning it specifically for sentence similarity tasks using a Siamese network structure. It produces embeddings that directly preserve sentence-level semantics, making cosine similarity a reliable retrieval mechanism. Thus, as the results shown in the table 5.1, SBERT forms the backbone of CPRS's retrieval pipeline.

Beyond embedding comparison, a more extensive evaluation was conducted using 200 representative test queries selected from the dataset to simulate realistic user interactions, including well-formed, incomplete, ambiguous, and noisy inputs. Retrieval correctness was assessed using expert-validated ground truth labels. This analysis revealed that SBERT achieves a top-1 retrieval accuracy of 0.85 and a top-3 accuracy of 0.92, demonstrating strong domain alignment and semantic robustness. System latency was also evaluated over 1,000 retrieval operations, yielding an average response time of under 0.5 seconds, confirming real-time suitability for conversational usage.

A dedicated analysis was carried out to evaluate the Information Sufficiency Module, which identifies incomplete or under-specified user queries and triggers targeted clarification questions. In the evaluation dataset, approximately 34% of queries were incomplete ; after engaging in short multi-turn clarification, all were rendered semantically complete, with an average of 1.3 clarification turns. This confirms that sufficiency checking substantially improves recommendation quality, particularly in scenarios involving non-expert users or ambiguous queries.

Finally, the explainability subsystem was assessed qualitatively using SHAP-based token attribution. Ten CPS practitioners evaluated the clarity and usefulness of the explanations, resulting in an average clarity rating of 4.4/5 and a trust-improvement score of 4.5/5. Experts noted that the highlighted keywords accurately reflected the reasoning behind each recommendation and enhanced their confidence in the system's output. The overall technical evaluation thus demonstrates that IoT-AID's semantic retrieval pipeline, sufficiency reasoning module, and explainability mechanisms operate effectively

in CPS configuration contexts.

While these results are compelling, several threats to validity must be acknowledged. Internal validity may be affected by the use of expert annotations to form ground truth, introducing subjective bias. External validity is limited because the dataset, although diverse, focuses on commonly deployed sensors and may not generalize to highly niche CPS domains. Construct validity is influenced by the selected evaluation metrics, which rely on semantic correctness rather than downstream performance in deployed CPS environments. Conclusion validity could be strengthened through broader statistical testing or additional baseline models. Despite these limitations, the consistency and magnitude of the results provide strong evidence that IoT-AID's retrieval and reasoning components constitute an effective basis for conversational CPS configuration support.

5.3.5 Qualitative Results : User eXperience Study

To complement the quantitative and technical evaluations presented in the previous section, a comprehensive user study was conducted to assess the usability, clarity, trustworthiness, and perceived relevance of IoT-AID's recommendations during real conversational interactions. The primary goal of this study was to determine whether users of different technical backgrounds found the system intuitive, whether the SHAP-based explanations improved their understanding of the recommendations, and whether the conversational approach effectively reduced the effort required to configure CPS components. A total of one hundred participants took part in the study, representing a balanced distribution of user expertise : fifty percent were non-experts with no prior experience in IoT or CPS technologies, twenty-five percent came from related scientific fields such as computer science, physics, chemistry, or general engineering, and the remaining twenty-five percent were domain experts familiar with sensors, IoT platforms, or CPS design. This distribution ensured that the evaluation reflected the diverse user profiles likely to engage with IoT-AID in real-world scenarios.

Each participant was asked to perform between five and ten CPS-related configuration tasks covering environmental monitoring, healthcare applications, industrial IoT scenarios, and consumer-oriented use cases. These tasks

included selecting suitable sensors, identifying correct communication protocols, refining system constraints, and evaluating trade-offs between different technical options. Participants interacted with the system freely until they obtained recommendations they considered appropriate for the task. The system automatically recorded query sequences, clarification turns, and decision patterns, while participants completed a structured questionnaire afterward to evaluate ease of use, clarity of explanations, relevance of outputs, perceived time savings, and overall satisfaction on a 5-point Likert scale. The qualitative component of the study also invited open-ended feedback to gain deeper insight into the user experience.

The results of the study reflect consistently strong user acceptance and positive system perception. Overall ease of use was reported by 88% of participants, with many non-experts specifically noting that the conversational interface "removed the complexity" typically associated with selecting sensors or understanding technical specifications. Clarity received the highest rating at 91%, driven largely by the SHAP-based explanations, which participants found particularly helpful for understanding why certain components were recommended. Trust in the system increased substantially due to the explanation mechanism, with SHAP clarity rated at 4.4/5 and trust improvement at 4.5/5. Relevance of recommendations was confirmed by 86% of participants, including domain experts who validated that the proposed components matched real-world expectations. A significant majority 84% also reported substantial time savings compared to manually searching through datasheets or vendor catalogs. The final overall satisfaction score reached 89%, indicating broad user approval across all skill levels.

The qualitative feedback revealed important additional insights. Many users praised IoT-AID's ability to guide them through missing or unclear requirements; instead of perceiving clarification questions as intrusive, most participants considered them helpful and necessary for obtaining accurate results. This aligns with system metadata showing that an average of 1.3 clarification turns were needed when queries were incomplete, a process particularly appreciated by non-experts who often lacked knowledge of essential CPS specifications. Participants from related scientific fields noted that IoT-AID helped bridge the gap between general technical knowledge and specialized CPS understanding, while domain experts valued the system's

ability to quickly retrieve relevant sensor information that they would otherwise search for manually. The SHAP explanations were widely recognized as a key factor contributing to user trust, as they highlighted decisive keywords such as in the example where “fall,” “elderly,” and “motion” guided the selection of the MPU6050 sensor for elderly fall-risk monitoringthereby confirming that the recommended component aligned with user intent.

Despite the positive findings, several threats to validity must be acknowledged. Internal validity may be influenced by differences in participants' familiarity with conversational interfaces, which could affect their comfort with multi-turn interactions. External validity is limited because the tasks, although diverse, cannot fully represent the entire space of possible CPS configurations. Construct validity is influenced by the reliance on subjective self-reported satisfaction metrics, which may not always reflect objective system performance. Conclusion validity may also be affected by the absence of a control group using a baseline non-conversational recommender or traditional sensor-selection methodology. Nevertheless, the consistency of the quantitative results and the convergence of positive qualitative feedback across all user groups strongly indicate that IoT-AID is effective, intuitive, and capable of supporting CPS configuration tasks for both novices and experts.

5.3.6 Discussion of Results

The evaluation shows that CPRS achieves :

- **High retrieval accuracy** via SBERT embeddings,
- **Transparency** through SHAP explanations,
- **Accessibility** thanks to Mistral-7B responses and sufficiency checks.

Compared to traditional recommenders, CPRS avoids dependence on historical user data, leverages domain-specific semantics, and provides built-in explainability. Its modular pipeline makes it auditable and adaptable for safety-critical applications.

Some of the prominent aspects of the designed system are as follows :

1. **Accuracy and Efficiency**
   The use of SBERT as the embedding backbone significantly improved recommendation accuracy (0.85) compared to Word2Vec (0.62) and vanilla BERT (0.74). Despite offering higher accuracy, SBERT maintained

acceptable latency (∼90 ms) and moderate memory requirements, making it suitable for interactive, real-time scenarios.

2. **Explainability**
   SHAP explanations provided token-level transparency, showing which user inputs drove the recommendation. This feature is valued by both non-experts (who may gain trust in the system) and developers (who can debug and refine the pipeline). The modular pipeline separates stages of intent classification, similarity matching, and explanation, allowing each step to be validated independently. This is particularly important in safety-critical domains where accountability is essentially required.

3. **User-Centered Interaction**
   The information sufficiency module ensured that incomplete queries did not lead to misleading outputs by prompting clarifying questions. Mistral-7B response generation transformed technical metadata into accessible natural-language explanations, broadening usability for users without CPS expertise.

4. **Feedback-Driven Design**
   The feedback questionnaire enabled continuous refinement of the system while reinforcing transparency for users. This dual-purpose design ensures CPRS evolves iteratively while maintaining user trust.

The results confirm CPRS's dual contribution :

- As a **practical assistant** for engineers and non-experts configuring CPS.
- As a **research instrument**, enabling the study of explainable, semantic-driven recommendation systems in industrial contexts.

The inclusion of SHAP explanations and natural language responses are particularly valued, as it allowed participants, many of whom, lacked deep CPS expertise to validate system output without needing to manually interpret the technical datasheets. By combining semantic embeddings, intent classification, sufficiency checks, SHAP-based explainability, and natural language generation, CPRS addresses long-standing challenges in CPS design such as complexity, transparency, and accessibility.

Chapitre 6

Research Contributions, Implications and Originality

This chapter presents a consolidated overview of the major contributions of this thesis, which culminates in the design, implementation, and evaluation of the Conversational CPS Recommendation System (CPRS). The preceding chapters have detailed the motivations, background, and methodological steps taken to address the challenges of configuring complex Cyber-Physical Systems (CPS). The core outcome of this research is a novel, modular architecture that effectively bridges the gap between the technical complexity of CPS configuration and the need for user-friendly, transparent, and trustworthy recommendation tools.

The CPRS framework represents a significant departure from opaque, end-to-end models by prioritizing explainability, reliability, and domain-specific base. Its contributions are multifaceted, spanning architectural design, the integration of semantic AI techniques, and a user-centered interaction model. This chapter systematically outlines these contributions, highlighting the technical advancements and their practical implications for industrial applications.

The remainder of this chapter is structured to provide a comprehensive synthesis of the work conducted throughout this thesis. Section 7.4 presents the principal scientific and technical contributions achieved, detailing the conceptual, architectural, and methodological advances that underpin the design and implementation of the Cyber-Physical Recommendation System (CPRS). Section 6.2 then enumerates and contextualizes the publications arising from this research, linking each to the corresponding stages of system development, from the initial conceptualization of the IoT-AID framework to the final explainable conversational prototype. Together, these sections trace the complete evolution of the CPRS from its theoretical foundations to its validated implementation demonstrating the continuity and maturity of the

research process. The chapter concludes with an integrative reflection that situates the outcomes of this work within the broader landscape of Cyber-Physical Systems engineering and outlines the enduring impact and future potential of the proposed framework.

6.1 Contributions

The current work is intended to support the tasks for CPS configurations and conversational recommendation systems. This support is achieved through several novel contributions. Each contribution reflects a technical improvement and a practical step towards bridging the gap between complex CPS design requirements and user-friendly, explainable recommendation tools. In this regard, we discuss some of the key contributions, as follows :

6.1.1 Modular and Explainable CPS Recommender Architecture

The primary contribution is the design of a modular architecture for CPS recommendation that separates the pipelines into distinct stages : preprocessing, semantic embeddings (SBERT), intent-classification, sufficiency checks, similarity-based retrieval, explainability (SHAP), and response generation (Mistral-7B). This modularity ensures transparency, ease of debugging, and adaptability in contrast to traditional or end-to-end black-box recommenders where decision logic is opaque.

6.1.2 Integration of Semantic Embeddings with a Curated Knowledge Base

The second contribution lies in combining Sentence-BERT embeddings with a structured CPS component knowledge base. By aligning user queries with verified sensor metadata, CPRS achieves reliable and semantically meaningful retrieval. Unlike general-purpose LLM-driven recommenders (e.g., GPT, Gemini), which depend heavily on generative reasoning and are prone to hallucinations, CPRS grounds all recommendations in curated data, making it suitable for safety-critical industrial domains.

6.1.3 SHAP-Based Explainability for Transparent Recommendations

CPRS contributes to explainable AI in recommender systems by employing SHapley Additive exPlanations (SHAP) to provide feature-level attribution for recommendations. This ensures that users can see exactly which tokens in their query influenced the system's decision. While LLMs such as GPT or Gemini can generate plausible justifications, these are often post-hoc narratives rather than explanations rooted in the retrieval process. CPRS therefore adds an extra layer of transparency that benefits both end-users (by fostering trust) and developers (by enabling diagnostic insight).

6.1.4 User-Centered Interaction through Sufficiency Checks and Feedback

Another contribution is the integration of a rule-based sufficiency module and a feedback mechanism. The sufficiency module ensures that incomplete queries trigger clarification questions, guiding users to provide all necessary details. This is especially beneficial for non-experts, who may lack the technical vocabulary to specify requirements fully. Furthermore, the structured feedback questionnaire captures user impressions of clarity, usefulness, and transparency, enabling iterative system improvement. In contrast, many LLM-based systems attempt to infer missing information automatically, which may produce overconfident but incorrect recommendations.

6.1.5 Balanced Trade-off Between Efficiency and Interpretability

Through experimental evaluation, CPRS demonstrated that SBERT provides the best trade-off among Word2Vec, BERT, and SBERT achieving higher accuracy (0.85) while maintaining low latency and moderate memory usage. Unlike large LLM systems such as GPT or Gemini, which require substantial computational resources and cannot guarantee deterministic results, CPRS achieves real-time performance with transparent and interpretable outputs.

6.1.6 Practical Domain-Specific Advantage for CPS

Finally, CPRS provides a domain-specific solution that addresses the unique requirements of CPS configuration. Whereas GPT, Gemini, or similar conversational recommendations excel at open-domain dialogue, they lack structured integration with technical metadata and cannot easily enforce constraints such as measurement ranges or protocol compatibility. CPRS, by grounding its recommendations in a curated knowledge base and supporting them with SHAP explanations, ensures accuracy, traceability, and trust properties essential for Industry 4.0 and 5.0 environments.

6.2 Publications & Contributions

The research presented in this dissertation has been progressively disseminated through several peer-reviewed conference papers, journal publications, and poster presentations. Each publication represents a distinct phase in the evolution of the Cyber-Physical Recommendation System (CPRS) and its practical implementation as IoT-AID, an explainable, conversational decision-support framework for Cyber-Physical Systems (CPS) within the context of Industry 4.0 and emerging Industry 5.0 paradigms.

The following contributions collectively document the theoretical conception, architectural design, implementation, and experimental validation of the proposed framework. Early works focused on establishing the foundational architecture of CPS and introducing the decision-support concept based on Natural Language Processing (NLP) and knowledge representation. Subsequent studies extended this foundation by integrating advanced machine-learning models, particularly Large Language Models (LLMs) and Sentence-BERT (SBERT), for semantic understanding and recommendation reasoning. The most recent publications concentrated on Explainable Artificial Intelligence (XAI) methods, employing SHAP to enhance transparency, interpretability, and user trust.

Together, these publications trace the complete scientific trajectory of the thesis from the initial conceptual model of the CPRS, through the design of the IoT-AID architecture and knowledge-base framework, to the development of a fully functional, explainable, and user-centric conversational recommendation system for CPS configuration and deployment.

6.2.1 Automated Decision Support Framework for IoT : Towards a Cyber-Physical Recommendation System," Proceedings of ICEIS 2023

This publication introduced the first conceptualization of an automated decision-support framework for Cyber-Physical Systems (CPS), presenting the Cyber-Physical Recommendation System (CPRS) as a means to aid engineers in selecting and integrating CPS components. The proposed model decomposed CPSs into three primary spheres; physical, network, and cyber thereby formalizing a semantic representation of sensors, actuators, and computational modules. The framework leveraged a knowledge base describing the relationships between CPS entities and integrated Natural Language Processing (NLP) mechanisms based on recurrent neural networks to interpret user objectives formulated in natural language. The paper validated this concept through a healthcare related use case, demonstrating that user intents could be translated into ranked recommendations of sensors and actuators tailored to domain-specific constraints.

The ICEIS paper laid the conceptual and architectural foundation for this dissertation. Its CPS model directly informed the framework described in Section 3.1, where the three-sphere abstraction was adopted as the ontological backbone of IoT-AID. The knowledge-base principles outlined in the paper underpin the system's semantic reasoning layer developed in Section 4.1. Moreover, the preliminary NLP approach based on RNNs evolved into the advanced transformer-based NLP pipeline implemented in Section 4.2. Hence, this conference contribution represents the earliest stage of the framework's development, transforming the theoretical CPS architecture into a structured research trajectory that culminates in the fully implemented CPRS presented in Chapter 5.

6.2.2 Towards a Cyber-Physical Recommendation System : Automated Decision Support Framework for IoT," Journées IA-Optim 2023

This poster introduces the early conceptualization of the Cyber-Physical Recommendation System (CPRS), a novel decision-support framework designed to assist scientists and engineers in configuring Cyber-Physical Systems

(CPS). The work presents the motivation and context of Industry 4.0, emphasizing the increasing complexity of CPS design due to heterogeneous hardware, communication protocols, and data integration challenges. The proposed system leverages Natural Language Processing (NLP) and Decision-Support System (DSS) techniques to interpret user requirements expressed in natural language and translate them into CPS design recommendations. A global architecture is outlined, illustrating the interconnections among the physical, cyber, and network layers, and highlighting how the CPRS can automatically propose compatible components and configurations. The framework aims to democratize CPS development by providing automated, knowledge-driven assistance that improves reliability, accelerates system prototyping, and reduces reliance on expert intervention.

This poster represents the foundation phase of the research and corresponds directly to sections 3.1 and 4.1 of the thesis. The architectural ideas visualized here formed the conceptual blueprint for the later IoT-AID framework described in the dissertation. The notions of component modeling, decision-support logic, and NLP-based user interaction first presented in this poster were subsequently expanded into the conversational and knowledge-base modules implemented in Chapter 4. Furthermore, the poster's emphasis on explainability and transparency anticipated the Explainable AI (XAI) mechanisms introduced later in Section 4.4. As such, this work documents the earliest public dissemination of the thesis concept establishing the problem statement, initial methodology, and architectural principles that underpin the full Cyber-Physical Recommendation System developed in this dissertation.

6.2.3 IoT-AID : An Automated Decision Support Framework for IoT," ICEIS Post-Proceedings 2024

This extended publication formalizes IoT-AID, an advanced instantiation of the CPRS concept, designed as a comprehensive automated decision-support framework for IoT and Industry 4.0. The paper elaborates a structured CPS architecture integrating physical, network, and cyber layers, each associated with corresponding sensing, communication, and computational functions. It introduces a conversational recommendation system that combines knowledge-based reasoning with collaborative filtering and employs Natural Language Processing techniques particularly Named Entity Recognition (NER) and semantic analysis using RNN/LSTM to interpret user inputs.

The framework leverages a knowledge graph implemented with Neo4j to represent CPS components and their interrelations. Through a real-time patient surveillance case study, IoT-AID demonstrates its ability to recommend and justify sensor configurations, thereby bridging the gap between user intent and CPS design automation.

The IoT-AID post-proceedings paper corresponds to the fully developed stage of the research presented throughout Chapter 3 and Chapter 4. Its architectural concepts are detailed in Section 3.1, while the conversational and NLP modules described in the article are implemented and expanded in Section Chapter 3.4 and Section 4.2. The explainability and knowledge-base components discussed in IoT-AID's design are realized through the hybrid reasoning and LLM-based retrieval mechanisms explained in Section 4.3 and Section Chapter 4.4. Finally, its validation through the patient surveillance example directly aligns with Section 3.3 and the evaluation results of Chapter 5. This publication thus marks the transition from a conceptual CPS model to a fully operational, explainable, and adaptive conversational recommendation system.

6.2.4 IoT Sensor Selection in Cyber-Physical Systems : Leveraging Large Language Models as Recommender Systems," Proceedings of CoDIT 2024

This publication extends the IoT-AID framework by integrating Large Language Models (LLMs) such as BERT and Sentence Transformers to enhance the semantic and contextual reasoning capabilities of the Cyber-Physical Recommendation System (CPRS). The paper presents a fine-tuning strategy in which LLMs are adapted to domain specific datasets representing CPS components, including sensors, actuators, and their operational contexts. It further introduces an automated data acquisition pipeline employing web scraping and structured dataset generation to populate the Knowledge Base with accurate sensor specifications and domain annotations. By using LLM-based semantic embeddings, the system achieves improved natural language understanding, entity recognition, and context-aware sensor recommendation. Preliminary results confirm that fine-tuned LLMs significantly increase the accuracy and explainability of recommendations compared to classical NLP-based approaches, establishing their potential as decision-support engines for Industry 4.0 applications.

The CoDIT 2024 paper contributes to the advanced implementation stage of this thesis, particularly enriching the discussions in Section 4.3 and Section Chapter 4.4. The fine-tuning and evaluation procedures presented in the paper directly correspond to the LLM-based module design integrated within the IoT-AID framework described in Chapter 4. Furthermore, the dataset construction methodology based on automated web scraping aligns with the Knowledge Base population strategies discussed in Section 4.1 . The paper's insights into entity extraction, semantic reasoning, and explainability provide the empirical foundation for the thesis's analysis of LLM-driven recommendations in cyber-physical contexts, as further validated through experimental results in Chapter 5. This contribution therefore marks the transition from NLP-based reasoning to a transformer-powered, context-aware recommender architecture, representing a major advancement in the thesis's research trajectory toward intelligent CPS design automation.

6.2.5 IoT-AID : Leveraging XAI for Conversational Recommendations in Cyber-Physical Systems," Proceedings of the ICEIS 2025 Workshop

This paper introduces an enhanced version of the IoT-AID framework that integrates Explainable Artificial Intelligence (XAI) methods within the Cyber-Physical Recommendation System (CPRS) to promote transparency and trust in AI-driven decision-making. The study combines traditional machine learning algorithms (Decision Trees, SVM, Naive Bayes) and deep learning models (RNN, CNN) with fine-tuned transformer-based architectures such as BERT and GPT, forming a hybrid recommendation engine. The explainability layer leverages LIME and SHAP techniques to provide both local and global interpretability of model outputs, ensuring that users can understand the reasoning behind each recommendation. The system's pipeline spans data preprocessing, feature extraction (TF-IDF and contextual embeddings), and model training, culminating in transparent recommendations for CPS design. Experimental results demonstrate that integrating XAI mechanisms into IoT-AID improves user confidence and decision quality by offering traceable, interpretable, and context-aware suggestions for sensor and system configuration in Industry 4.0 environments.

The ICEIS 2025 Workshop paper represents the culmination of this thesis's research trajectory, directly corresponding to the Explainability and Evaluation layers detailed in Section Chapter 4.4 and Chapter 5. The XAI methodologies described particularly the integration of LIME and SHAP extend the explainable reasoning capabilities introduced earlier in Section 4.3. The hybrid model structure combining classical ML, deep learning, and transformer-based approaches aligns with the architectural synthesis outlined in Section 4.1. Furthermore, the dataset generation and deduplication process reported in this paper supports the data collection procedures discussed in Section 4.2. Collectively, this work operationalizes the thesis's objective of creating a transparent, user-centric, and interpretable recommendation system for CPS configuration, establishing IoT-AID as a fully explainable conversational framework that bridges the gap between automation and human interpretability in Industry 4.0.

6.2.6 IoT-AID : An Explainable Conversational Agent for Assisting Cyber-Physical System Configuration,Journées IA-Optim 2025

This poster presents a comprehensive overview of IoT-AID, an Explainable Cyber-Physical Recommendation System (CPRS) designed to automate and demystify the configuration of Cyber-Physical Systems in Industry 4.0. The system integrates natural language understanding, semantic reasoning, and explainability within a single architecture. Using a conversational chatbot interface, IoT-AID captures user requirements expressed in natural language and processes them through a fine-tuned Sentence-BERT model to extract domain-specific entities and contextual constraints. A structured MongoDB knowledge base supports semantic retrieval and rule-based filtering of compatible CPS components, while SHAP explanations provide both local and global interpretability of the recommendations. The framework's hybrid design, combining classical rule-based reasoning and deep learning, allows IoT-AID to deliver transparent, ranked, and contextually justified CPS configurations.

This poster illustrates the practical embodiment of the concepts discussed in Chapter 4 and Chapter 5. The conversational module and NLP pipeline described in the poster correspond to Section 4.2, while the explainability and ranking mechanisms align with Section 4.4. The poster emphasizes the role of explainable AI in enhancing user understanding, directly supporting

the interpretability objectives established in Section 4.3. By summarizing the system's functionality and empirical validation across use cases such as predictive maintenance and environmental monitoring, this poster visually reinforces the thesis's final stage demonstrating IoT-AID as a fully operational, explainable, and user-centric framework for CPS configuration.

6.2.7 Towards an Explainable Cyber-Physical Recommender System," 2025 Proceedings.

This publication presents the most advanced iteration of the IoT-AID framework, integrating explainability, semantic reasoning, and conversational interaction into a unified Cyber-Physical Recommender System (CPRS). The paper introduces a hybrid architecture that combines Sentence-BERT (SBERT) embeddings for semantic similarity, rule-based reasoning, and SHAP-based explainability to deliver transparent CPS recommendations. The system utilizes structured metadata stored in a MongoDB knowledge base, representing component specifications, domains, and compatibility constraints. Queries expressed in natural language are processed through an NLP pipeline and mapped to relevant CPS components using cosine similarity and rule-based filtering. Explanations are generated for each recommendation, highlighting matched features, selection criteria, and SHAP attributions. Experimental evaluation, including both quantitative benchmarking and qualitative user studies, demonstrates that the CPRS achieves an optimal balance between accuracy, efficiency, and interpretability, significantly improving user trust and configuration precision in CPS deployment.

This proceedings paper constitutes the final experimental validation of the thesis and corresponds directly to Chapter 5 (Implementation and Evaluation of the CPRS), as well as to the Explainability Module described in Section Chapter 4.4. The SBERT-based embedding and rule-based hybrid strategy refine the NLP techniques introduced in Section 4.2 and Section 4.3. Furthermore, the SHAP and sufficiency-checking modules align with the transparency and interpretability mechanisms proposed in the thesis to enhance user trust in AI-driven CPS configuration. The dataset construction process detailed in the paper reinforces the data collection and preprocessing methodology discussed in Section 4.1. This work therefore encapsulates the culmination of the thesis's research path, transforming conceptual and architectural designs into a validated, explainable, and user-aligned CPRS prototype

capable of bridging human expertise and machine intelligence in Industry 4.0 and 5.0 contexts.

Chapitre 7

General Conclusion & Research Perspectives

This thesis has presented the design, implementation, and evaluation of the Cyber-Physical Recommendation System (CPRS), a modular and explainable framework for assisting users in the configuration of Cyber-Physical Systems. The motivation for this work arose from the inherent challenges in CPS design : the integration of heterogeneous sensors, actuators, and communication modules requires specialized knowledge, significant manual effort, and carries a high risk of error. Traditional recommendation approaches such as collaborative filtering or black-box AI systems are poorly suited to this domain, as they rely on historical data, lack transparency, and fail to capture the contextual requirements of industrial applications.

The proposed CPRS addresses these limitations by bridging the gap between natural language requirements and structured CPS configurations. Its architecture combines semantic similarity retrieval through Sentence-BERT (SBERT), intent classification to guide domain relevance, an information sufficiency module to ensure completeness of queries, SHAP-based explainability to provide transparency, and Mistral-7B language generation to produce natural-language explanations. The system is further supported by a curated knowledge base of CPS components, constructed through web scraping, datasheet parsing, manual validation, and generative augmentation.

The implementation demonstrated that CPRS is both feasible and effective. Quantitative evaluations showed that SBERT significantly outperformed Word2Vec and BERT in recommendation accuracy, achieving an accuracy of 0.85 with manageable latency and memory usage. Qualitative user studies confirmed that the system is intuitive, transparent, and trustworthy : 90% of participants reported that SHAP explanations improved their understanding of recommendations, and the overall satisfaction score reached 9.2/10. These results validate the core premise of CPRS : that semantic

retrieval and explainable AI can meaningfully reduce the cognitive burden of CPS configuration for both experts and non-experts.

Beyond its immediate performance, CPRS represents a step toward user-centered AI in Industry 4.0 and 5.0. By embedding explainability and feedback collection into the recommendation pipeline, the system aligns with emerging demands for transparency, accountability, and human–machine collaboration in industrial contexts. Its modularity ensures that it can evolve with new models, datasets, and interaction modalities, making it a flexible foundation for future innovations.

Nevertheless, limitations remain, including the fragility of web scraping pipelines, the static nature of the knowledge base, and the occasional ambiguity of embeddings. Future work should focus on domain-specific fine-tuning, multimodal query interfaces, ontology-driven ranking, and integration of regulatory constraints. These directions will further enhance CPRS's robustness, adaptability, and compliance in real-world deployments.

This thesis contributes a practical, explainable, and human-centered recommender system for CPS configuration. CPRS advances the state of the art by demonstrating how semantic embeddings, interactive clarification, explainable AI, and natural language generation can be combined into a coherent system that supports both technical accuracy and user trust. By lowering the expertise barrier and fostering transparency, CPRS provides not only a solution for current industrial challenges but also a blueprint for future intelligent assistants in complex engineering domains.

The remainder of this chapter is organized as follows. Section 7.1 presents the overall conclusion of this thesis, summarizing its main scientific and technical contributions while highlighting their implications for the advancement of explainable and human-centered AI in Cyber-Physical Systems. Section 7.2 then critically examines the current limitations of the proposed framework, discussing both methodological and practical challenges that constrain its broader scalability and industrial adoption. Building on these insights, Section 7.3 outlines the research perspectives and a three-stage roadmap for future development encompassing early-stage scientific consolidation, mid-stage system enhancement, and late-stage industrial deployment thereby defining a coherent trajectory for the continued evolution of the Cyber-Physical Recommendation System toward real-world integration and long-term impact.

7.1 Conclusion

This thesis has presented an integrated and explainable framework for the Cyber-Physical Recommendation System (CPRS), designed to support the configuration and deployment of Cyber-Physical Systems (CPS) through natural language interaction and transparent reasoning. The proposed system aims to bridge the gap between human expertise and automated system configuration by providing a modular, interpretable, and domain-adaptive recommendation pipeline that leverages the combined strengths of semantic embeddings, rule-based validation, and large language models (LLMs).

The work began with an analysis of the challenges inherent to CPS design the heterogeneity of hardware and software components, the lack of semantic interoperability, and the heavy reliance on expert knowledge for system configuration. These challenges were addressed by proposing a conversational and explainable system capable of interpreting high-level, unstructured user inputs and translating them into technically valid CPS recommendations. By doing so, the CPRS transforms the traditional engineering workflow into an interactive, iterative, and user-centric process.

From an architectural standpoint, this research introduced a layered framework that harmoniously integrates natural language understanding, intent classification, information sufficiency analysis, similarity-based retrieval, explainability through SHAP, and natural language generation using Mistral-7B. The natural language understanding module based on Sentence-BERT embeddings enables the semantic interpretation of user queries, while the intent classification mechanism identifies the relevant CPS domain or application area.

The sufficiency module ensures completeness by detecting missing parameters and generating targeted clarifying questions, whereas the similarity retrieval stage ranks potential components from the knowledge base using cosine similarity to ensure contextual relevance. The integration of SHAP-based interpretability enhances transparency by identifying the most influential factors behind each recommendation, and the Mistral-7B model translates these technical explanations into fluent, accessible responses. This architecture collectively provides a robust and transparent framework that unites advanced artificial intelligence reasoning with the requirements of industrial design systems.

The technical implementation of CPRS demonstrates the effectiveness of

explainable AI in reducing configuration complexity, minimizing human error, and improving user trust in automated decision support. The integration of semantic embeddings with a curated and structured CPS knowledge base proved essential for achieving both contextual accuracy and computational efficiency. Experimental evaluations confirmed validated the hypothesis that domain-aware embeddings could significantly enhance the relevance and interpretability of automated recommendations for CPS component selection and configuration. The empirical results further confirmed that the proposed hybrid approach, combining semantic similarity with rule-based constraints, ensures both the correctness and explainability of system outputs, thus achieving a balanced trade-off between efficiency, performance, and interpretability.

Scientifically, this research contributes to multiple disciplines by introducing an explainable conversational paradigm that merges advances from natural language processing, knowledge representation, and explainable artificial intelligence. It provides a domain-agnostic yet extensible CPS knowledge base capable of capturing semantic relations among heterogeneous components, ensuring consistent and interpretable reasoning. The study demonstrates the feasibility of human-centered AI in engineering contexts, emphasizing interpretability, traceability, and adaptability over purely predictive performance. By integrating these elements, the CPRS establishes a coherent methodology for transparent, human-AI collaboration in CPS design and decision-making, embodying the principles of Industry 5.0, where intelligent systems complement rather than replace human expertise.

From a practical perspective, the CPRS framework lays the groundwork for the next generation of intelligent assistants capable of supporting complex decision-making in real-world industrial environments. Its modular architecture allows for continuous evolution and adaptation as new models, data sources, and interaction modalities emerge. The system's ability to justify its reasoning, adapt to user feedback, and remain interpretable underlines its potential to serve as a dependable co-pilot for engineers in configuration, validation, and optimization tasks. The incorporation of large language models such as Mistral-7B enhances this capability by enabling dynamic, context-sensitive dialogue generation, bridging the gap between technical reasoning and human understanding. This alignment between explainability and interactivity signifies a major step toward fully transparent and collaborative intelligent design systems.

Nevertheless, this thesis also acknowledges existing challenges that require further exploration. The knowledge base, although extensive, remains static and partially dependent on web-scraped data, which introduces potential inconsistencies. Similarly, the general-purpose nature of SBERT embeddings limits their ability to fully capture the nuances of specialized CPS terminology. Addressing these issues will necessitate domain-specific fine-tuning of language models, ontology-based knowledge representation, and the adoption of standardized data-sharing protocols across industrial partners. Future developments should also incorporate multimodal interaction, enabling users to express requirements through speech, sketches, or sensor schematics, and extend the system to comply with industrial and safety regulations to ensure practical deployment in production environments.

This thesis contributes a novel, explainable, and conversational recommender framework that unites semantic intelligence, modular system design, and user-centered interaction to advance the field of Cyber-Physical System configuration. The CPRS demonstrates that the convergence of Explainable AI, semantic retrieval, and natural language dialogue can significantly enhance the efficiency, interpretability, and trustworthiness of engineering decision-support systems. By reducing cognitive barriers, improving transparency, and fostering a more collaborative relationship between humans and intelligent machines, this research not only addresses key challenges of Industry 4.0 but also anticipates the human-centric vision of Industry 5.0. The framework developed in this work provides both a theoretical foundation and a practical blueprint for the next generation of adaptive, explainable, and trustworthy AI-driven assistants in complex system design and configuration.

7.2 Limitations and Future Directions

While the evaluation of CPRS demonstrates strong performance and clear advantages over existing approaches, but any system can have its constraints. Understanding these limitations is essential for positioning CPRS realistically within industrial applications and for guiding its future evolution. This chapter therefore outlines the key challenges of the system and proposes directions for its extension.

7.2.1 Current Limitations

Despite its strengths, CPRS faces several limitations that may constrain its scalability, robustness, and domain coverage :

1. **Web Scraping Fragility :** The curated knowledge base (section 7.4.2, "Integration of Semantic Embeddings with a Curated Knowledge Base") relies heavily on web scraping of vendor websites. Since site structures are unstable and frequently updated, scraping pipelines are brittle and prone to errors, limiting the sustainability of long-term data acquisition.
2. **Embedding Ambiguity :** Although SBERT significantly outperforms Word2Vec, BERT and vanilla BERT (section 7.4.5, "Balanced Trade-off Between Efficiency and Interpretability"), it occasionally struggles with fine-grained distinctions between closely related sensor types (e.g., methane vs. propane gas sensors). This weakness stems from its pretraining on general-domain corpora rather than CPS-specific technical data.
3. **SHAP Surrogate Approximation :** The SHAP-based explainability module (section 7.4.3, "SHAP-Based Explainability for Transparent Recommendations") employs a surrogate model to approximate similarity scoring. While effective in practice, this approximation may not fully capture the internal semantics of SBERT embeddings, producing explanations that are partially representative.
4. **Static Knowledge Base :** The modular pipeline (section 7.4.1, "Modular and Explainable CPS Recommender Architecture") currently depends on an offline, statically updated knowledge base. User feedback is collected (section 7.4.4, "User-Centered Interaction through Sufficiency Checks and Feedback") but remains decoupled from model adaptation, slowing the pace of iterative improvement.
5. **Limited Interaction Modalities.** Although CPRS supports sufficiency checks and clarification prompts (section 7.4.4, "User-Centered Interaction"), interaction is restricted to textual queries. In real-world industrial environments, users may prefer alternative modalities such as spoken commands, sketches, or multimodal inputs that more naturally capture design intent.

7.2.2 Future Directions

To overcome these challenges and extend CPRS toward broader industrial adoption, several promising directions can be pursued :

1. **Robust Data Acquisition :** Replace fragile web scraping with standardized APIs or direct partnerships with sensor manufacturers. Leveraging structured metadata formats (e.g., JSON-LD, RDF) would promote interoperability and reduce parsing errors.
2. **Domain-Specific Fine-Tuning :** Fine-tune SBERT or deploy lightweight transformer variants on CPS-specific corpora (datasheets, industrial standards, technical reports). This would improve disambiguation between similar sensors and enhance semantic retrieval accuracy.
3. **Direct Transformer-Based Explainability :** Investigate attribution techniques native to transformer architectures, capable of producing token-level importance scores without relying on surrogate models. This would yield explanations that more faithfully reflect the retrieval mechanism.
4. **Token-Level Explainability in Transformers**
   Investigate transformer-based attribution methods that go beyond surrogate SHAP models, offering direct token-level importance scores. This would produce more precise explanations, strengthening both user trust and developer insight.
5. **Active and Continual Learning**
   Incorporate real-time user feedback into an active learning loop, enabling the system to adapt continuously. Over time, CPRS could personalize recommendations to specific users or organizations.
6. **Multimodal Interaction**
   Extend CPRS to handle voice queries, sketches, or annotated diagrams of CPS configurations. This would broaden accessibility and align with Industry 5.0 goals of human–machine collaboration.
7. **Ontology-Aware Ranking**
   Integrate knowledge graphs or ontologies of CPS components to enforce compatibility constraints. This would prevent mismatched recommendations (e.g., suggesting a sensor incompatible with the chosen protocol).
8. **Regulatory Integration**
   Extend the recommendation pipeline to incorporate safety standards and regulatory constraints (e.g., ISO or IEC standards for industrial

sensors). This would ensure that CPRS not only recommends technically suitable components but also compliant ones.

While CPRS already demonstrates strong technical performance, interpretability, and usability, its limitations highlight areas for refinement in data acquisition, embedding precision, explainability, and interactivity. Future directions such as domain-specific fine-tuning, multimodal query support, and active learning will push CPRS toward becoming a fully adaptive, regulatory-compliant, and human-centered assistant for CPS configuration.

7.3 Research Perspectives

Looking forward, the CPRS can evolve through three progressive development stages : early, mid, and late-stage research and deployment. Each stage defines specific objectives to consolidate scientific contributions and move toward real-world industrial integration.

7.3.1 Early Stage : Scientific Consolidation and Publication

At this stage, the priority is the publication of the most recent research results in a high-impact journal to formally establish the scientific foundation of the CPRS. The focus should be on refining the system architecture, evaluating it on larger CPS datasets, and validating the explainability mechanisms under controlled conditions.

- Publish the full framework and experimental findings in a reputable AI or systems engineering journal
- Extend benchmarking across diverse domains (healthcare, manufacturing, smart cities).
- Improve dataset robustness by adopting standardized data acquisition pipelines and APIs instead of web-scraping.

This phase solidifies the academic contribution and ensures reproducibility of results.

7.3.2 Mid Stage : System Enhancement and Domain Specialization

Once validated academically, the next step is to enhance and specialize the system for practical applications.

- Fine-tune the embedding and reasoning modules on domain-specific corpora (e.g., industrial sensor standards, IoT protocols).
- Introduce ontology-based reasoning and knowledge-graph alignment to strengthen semantic retrieval and interoperability.
- Establish partnerships with industrial or research labs to deploy and test the framework in real-world CPS environments.

This phase bridges research and application, ensuring that CPRS meets the operational constraints and standards of industrial ecosystems.

7.3.3 Late Stage : Industrial Deployment and Continual Learning

In the final phase, CPRS evolves into a fully deployable industrial decision-support assistant.

- Integrate continuous learning loops from user feedback and field data to maintain relevance and adaptability.
- Deploy the system as a cloud-based service or embedded assistant within CPS development toolchains.
- Investigate cross-domain scalability and interoperability with existing industrial digital twins and smart-factory systems.

Ultimately, this late-stage vision positions CPRS as a trustworthy, adaptive, and explainable AI framework for intelligent system configuration across future Industry 5.0 ecosystems.

Résumé en Francais

À l'ère de l'industrie 4.0, la conception et la configuration des systèmes cyber-physiques (CPS) deviennent de plus en plus complexes. Ce processus requiert un effort cognitif important et reste vulnérable aux erreurs humaines. Les utilisateurs doivent en effet transformer des besoins métier généraux en spécifications techniques précises, tout en gérant des données hétérogènes, des schémas incohérents et des contraintes d'interopérabilité. Les outils de configuration traditionnels, basés sur des formulaires rigides ou des règles définies par des experts, manquent de flexibilité et de transparence.

Pour relever ces défis, cette thèse présente le système de recommandation cyber-physique (CPRS), un cadre conversationnel explicable qui assiste les utilisateurs dans le choix et la configuration des composants CPS. Le CPRS combine l'interaction en langage naturel, les plongements sémantiques, la validation fondée sur des règles et l'intelligence artificielle explicable (XAI) au sein d'une architecture modulaire. Il associe les méthodes classiques de recherche d'information aux capacités de génération de texte des grands modèles de langage (LLM), tandis que Sentence-BERT (SBERT) représente les requêtes et les métadonnées dans un espace sémantique partagé, permettant ainsi une recherche précise basée sur la similarité.

Le système utilise également un classement multicritère et affine ses recommandations de manière progressive et interactive, en dialoguant avec l'utilisateur pour préciser ses besoins. Les expériences menées sur un jeu de données de capteurs CPS ont montré que le CPRS atteignait une précision élevée de 0,85 et des temps de réponse réduits par rapport à des modèles basés sur Word2Vec ou BERT, tout en offrant une plus grande transparence dans le raisonnement. Les études utilisateurs confirment que 85 % des participants trouvent le système intuitif et conscient du contexte, ce qui renforce la confiance et la facilité d'utilisation

Cette thèse met en avant plusieurs contributions majeures. Elle présente notamment un flux de recommandation modulaire et explicable, l'intégration de plongements SBERT à une base de connaissances structurée, l'attribution de caractéristiques via SHAP, ainsi que l'utilisation du modèle Mistral-7B pour générer des réponses adaptatives centrées sur l'utilisateur.

Enfin, les limites du système sont reconnues, notamment en ce qui concerne la fiabilité de l'extraction automatique de données Web, l'ambiguïté des plongements SBERT et les bases de connaissances statiques. Des perspectives

d'avenir sont proposées, notamment pour assurer une acquisition de données plus robuste, un ajustement spécifique au domaine, une interaction multimodale et un apprentissage continu.

Mots clés : systèmes de recommandation ; système cyber-physique ; systèmes d'aide à la décision ; traitement du langage naturel ; système de recommandation conversationnel.

Introduction générale

L'émergence de l'Industrie 4.0 a placé les systèmes cyber–physiques (CPS) au cœur des transformations technologiques contemporaines. En associant des processus physiques à des capacités avancées de calcul, de communication et de prise de décision, les CPS permettent la surveillance en temps réel, le contrôle adaptatif et l'automatisation intelligente dans des domaines tels que l'industrie manufacturière, la santé, les transports ou l'énergie. Leur fonctionnement repose sur l'interconnexion entre des composants physiques, comme les capteurs, actionneurs et machines, et des composants cybernétiques chargés du traitement des données, de l'analyse et de la logique décisionnelle, le tout soutenu par des protocoles de communication fiables et à faible latence.

Malgré leur potentiel considérable, la conception et la configuration des CPS demeurent particulièrement complexes. Un projet CPS mobilise généralement un grand nombre de composants hétérogènes, issus de fabricants différents, décrits dans des formats variés et soumis à des contraintes techniques multiples. Les concepteurs doivent sélectionner des éléments matériels et logiciels compatibles tout en conciliant plusieurs exigences, telles que la consommation énergétique, la robustesse, le coût, la performance et l'interopérabilité. Cette difficulté est encore accentuée lorsque les systèmes sont déployés dans des environnements sensibles, où une erreur de configuration peut entraîner des conséquences importantes en matière de sécurité, de fiabilité ou de conformité.

Motivation et problématique de recherche

L'un des principaux verrous identifiés dans cette thèse réside dans la traduction des besoins exprimés à haut niveau en configurations CPS précises et techniquement valides. En pratique, les utilisateurs décrivent souvent leurs objectifs sous forme de besoins fonctionnels ou contextuels, par exemple la

surveillance d'une grandeur physique dans un environnement contraint ou la détection d'un événement donné dans des conditions particulières. Le passage de cette formulation générale à une sélection rigoureuse de capteurs, d'actionneurs, de protocoles de communication et de ressources de calcul reste une tâche longue, manuelle et fortement dépendante de l'expertise humaine.

Cette difficulté est renforcée par l'hétérogénéité des sources d'information. Les spécifications des composants sont dispersées dans des fichiers CSV, XML, PDF ou des catalogues web, avec des notations et des unités parfois incohérentes d'un fournisseur à l'autre. L'absence de schéma unifié complique l'intégration des données, la vérification de compatibilité et l'automatisation de la recherche de composants adaptés. En parallèle, les outils existants de conception CPS demeurent souvent rigides, reposant sur des formulaires, des menus fixes ou des approches peu explicables, qui prennent difficilement en compte les besoins imprécis, évolutifs ou exprimés en langage naturel.

Dans ce contexte, un besoin clair apparaît : développer un système capable d'interpréter des objectifs formulés naturellement par l'utilisateur, de les transformer en contraintes techniques exploitables, de recommander des composants pertinents, et d'expliquer de manière transparente la logique suivie.

Approche proposée

Pour répondre à cette problématique, cette thèse propose le *Cyber-Physical Recommendation System* (CPRS), un cadre intelligent, conversationnel et explicable destiné à assister la configuration des systèmes cyber–physiques. Le CPRS permet à l'utilisateur d'exprimer ses besoins en langage naturel, sans avoir à maîtriser d'emblée l'ensemble des spécifications techniques. Le système analyse ensuite cette demande, identifie les intentions et contraintes associées, interroge une base de connaissances structurée, puis recommande des composants ou configurations adaptés au contexte.

L'approche repose sur plusieurs principes complémentaires. Tout d'abord, une base de connaissances sémantiquement enrichie est construite à partir de sources hétérogènes, puis normalisée afin de garantir une représentation cohérente des composants CPS et de leurs attributs. Ensuite, un mécanisme de compréhension sémantique permet de rapprocher les besoins exprimés par

l'utilisateur des composants disponibles, même en l'absence de correspondance exacte entre les termes employés. Par ailleurs, une boucle de clarification interactive est intégrée au système : lorsque certaines informations sont absentes ou ambiguës, le CPRS génère des questions ciblées pour compléter progressivement la requête.

L'un des apports majeurs de cette approche est l'intégration de mécanismes d'explicabilité. Chaque recommandation est accompagnée d'une justification mettant en évidence les paramètres ou contraintes ayant influencé la décision. Cette capacité est essentielle pour renforcer la confiance de l'utilisateur, faciliter la validation des choix techniques et permettre la traçabilité des décisions dans des contextes sensibles ou réglementés.

Objectifs et contributions de la thèse

L'objectif général de cette recherche est de réduire la complexité de la configuration des CPS, d'améliorer la pertinence des recommandations proposées et d'accroître la confiance des utilisateurs grâce à des mécanismes explicables et interactifs. Plus précisément, la thèse vise à élaborer une représentation sémantique unifiée des composants CPS, à développer une interface en langage naturel capable d'interpréter des besoins libres, à mettre en place un mécanisme de clarification pour gérer les requêtes incomplètes, à intégrer des techniques d'explicabilité dans le processus de recommandation et à évaluer l'ensemble du système selon des critères de performance, de précision et de satisfaction utilisateur.

Les principales contributions de cette thèse résident dans la conception d'une architecture modulaire pour un système de recommandation cyber–physique conversationnel, dans la construction d'une base de connaissances structurée et annotée sémantiquement, ainsi que dans l'implémentation d'un flux de recommandation interactif et explicable adapté au domaine des CPS. L'évaluation du système sur des scénarios réalistes montre qu'il permet de simplifier le processus de conception, d'améliorer la qualité des recommandations et de rendre la configuration des CPS plus accessible à des utilisateurs de profils variés.

Organisation du manuscrit

Le manuscrit est organisé de manière progressive afin de conduire le lecteur des fondements conceptuels vers la validation expérimentale du système proposé. Le premier chapitre présente les systèmes cyber–physiques,

leurs architectures, leurs domaines d'application et les principaux défis associés à leur conception. Le deuxième chapitre introduit les systèmes de recommandation et les systèmes de recommandation conversationnels, en montrant leur intérêt pour la configuration des CPS. Le troisième chapitre décrit l'architecture de IoT-AID, cadre conversationnel proposé pour assister la recommandation de composants CPS. Le quatrième chapitre détaille l'implémentation du système, en mettant l'accent sur la chaîne de traitement du langage naturel, les embeddings sémantiques et l'intégration des grands modèles de langage. Le cinquième chapitre présente l'implémentation complète du CPRS ainsi que son évaluation quantitative et qualitative. Le sixième chapitre synthétise les contributions scientifiques et techniques de la thèse ainsi que leurs implications. Enfin, le septième chapitre propose la conclusion générale et discute les limites actuelles du système ainsi que les perspectives de recherche futures.

Chapitre 1 : Introduction aux systèmes cyber–physiques

L'émergence de l'Industrie 4.0 a profondément transformé le paysage industriel grâce à la convergence des technologies numériques, de l'automatisation intelligente et de la connectivité généralisée. Cette évolution repose sur l'intégration de l'Internet des objets (IoT), de l'intelligence artificielle (IA), du cloud computing et de l'analyse de données dans les systèmes de production traditionnels. Au cœur de cette transformation se trouvent les systèmes cyber–physiques (CPS), qui constituent des systèmes complexes et interconnectés assurant une interaction étroite entre les processus physiques et les capacités computationnelles à travers des réseaux de communication.

Les CPS permettent la surveillance en temps réel, le contrôle autonome et la prise de décision adaptative dans de nombreux domaines, notamment l'industrie, la santé, le transport et l'énergie. Leur fonctionnement repose sur une boucle de rétroaction continue reliant la perception, le traitement et l'action. Les données issues de l'environnement physique sont analysées dans le domaine cybernétique afin de produire des actions qui influencent directement les processus physiques.

Malgré leur potentiel, la conception et le déploiement des CPS restent des tâches complexes. Ces systèmes sont intrinsèquement multidisciplinaires et nécessitent l'intégration de composants hétérogènes, tant matériels que logiciels, tout en respectant des contraintes strictes de performance, de sécurité et

d'interopérabilité. Cette complexité est accentuée par la diversité des technologies de communication et par le volume croissant des données échangées. De plus, de nombreux acteurs industriels ne disposent pas de l'expertise nécessaire pour configurer efficacement ces systèmes, ce qui rend le processus de conception long et sujet aux erreurs.

Les CPS reposent sur une architecture composée de plusieurs couches fonctionnelles interconnectées. Les composants physiques, tels que les capteurs et les actionneurs, assurent l'interaction avec l'environnement. La couche cyber regroupe les modules de traitement et de prise de décision, tandis que la couche de communication garantit l'échange de données entre les différentes entités du système. Ces éléments peuvent être organisés en trois sphères principales : la sphère physique, la sphère réseau et la sphère cyber. Leur interaction permet la formation d'un système en boucle fermée capable de percevoir, analyser et agir de manière autonome.

Dans la littérature, plusieurs modèles architecturaux ont été proposés afin de structurer les CPS. Par exemple, la Figure 1.1 illustre une architecture en trois couches distinguant les niveaux physique, cyber et de communication. De même, la Figure 1.2 met en évidence les interactions entre ces différentes sphères. Sur la base de ces travaux, une architecture étendue est proposée dans cette thèse (Figure 1.3), intégrant les interactions entre les composants CPS et introduisant une approche adaptée à la recommandation et à la configuration intelligente.

Les CPS trouvent des applications dans de nombreux domaines. Dans le secteur de la santé, ils permettent la surveillance des patients, l'assistance médicale et le développement de solutions personnalisées. Dans l'industrie, ils constituent le socle des usines intelligentes, favorisant la maintenance prédictive, l'optimisation des ressources et l'automatisation des processus. Dans les transports, ils contribuent au développement de systèmes intelligents basés sur la communication entre véhicules et infrastructures, améliorant ainsi la sécurité et l'efficacité énergétique.

Par ailleurs, l'intégration de l'intelligence artificielle dans les CPS ouvre de nouvelles perspectives en matière d'analyse de données, de détection d'anomalies et de prise de décision autonome. Toutefois, cette intégration soulève également des défis liés à l'explicabilité, à la fiabilité et à la confiance dans les systèmes automatisés. L'interaction entre l'humain et les CPS reste également un enjeu majeur, notamment dans les systèmes critiques où la supervision humaine est indispensable. À cela s'ajoutent des problématiques éthiques et réglementaires liées à la confidentialité des données et à la responsabilité des

décisions automatisées.

La mise en œuvre des CPS se heurte à plusieurs obstacles, notamment l'interopérabilité des composants, la génération automatique de règles de configuration, la validation des systèmes et la nécessité de fournir des explications compréhensibles aux utilisateurs. Ces limitations mettent en évidence le besoin de solutions intelligentes capables de simplifier la configuration des CPS et d'assister les utilisateurs dans le processus de conception.

Dans ce contexte, cette thèse propose un *Cyber–Physical Recommendation System* (CPRS), visant à faciliter la conception et la configuration des CPS à travers une approche basée sur l'intelligence artificielle et l'interaction en langage naturel. Ce système permet de réduire la complexité, d'améliorer la qualité des décisions et de rendre la configuration des CPS plus accessible.

En conclusion, ce chapitre met en évidence le rôle central des CPS dans l'Industrie 4.0 ainsi que les défis liés à leur conception et à leur déploiement. Il souligne la nécessité de développer des outils intelligents et explicables pour accompagner les utilisateurs, ce qui constitue la motivation principale des travaux présentés dans cette thèse.

Chapitre 2 : Un cadre de recommandation conversationnelle contextuelle pour la conception et l'intégration des systèmes cyber–physiques

Ce chapitre introduit le cadre conceptuel qui relie les systèmes de recommandation conversationnels au problème de la configuration des systèmes cyber–physiques. Dans le contexte de l'Industrie 4.0, les CPS occupent une place centrale dans l'innovation industrielle, mais leur conception demeure difficile en raison de l'hétérogénéité des composants, des contraintes d'interopérabilité et de la diversité des exigences opérationnelles. Les ingénieurs et chercheurs doivent sélectionner des capteurs, des actionneurs, des unités de calcul et des protocoles de communication adaptés, tout en veillant à la cohérence de l'ensemble. Cette tâche est d'autant plus complexe que les besoins sont souvent exprimés de manière imprécise ou évolutive, ce qui justifie le recours à des systèmes capables d'accompagner l'utilisateur de manière interactive.

Dans cette perspective, le chapitre propose **IoT-AID**, un *Cyber–Physical Recommendation System* (CPRS) fondé sur le dialogue et conçu pour assister la configuration des CPS. IoT-AID combine des modèles récents de traitement

du langage naturel, notamment les grands modèles de langage, avec des mécanismes de recherche sémantique tels que BERT et Sentence-BERT (SBERT). L'objectif est de réduire la charge cognitive et technique liée à la conception des CPS en permettant une interaction en langage naturel, une élucidation progressive des besoins et une sélection raisonnée des composants.

Le chapitre commence par rappeler les fondements des systèmes de recommandation. Ces systèmes ont d'abord émergé dans des environnements numériques grand public où ils avaient pour rôle de guider l'utilisateur vers les objets ou services les plus pertinents. Leur évolution a suivi plusieurs étapes, depuis les premiers mécanismes de filtrage collaboratif jusqu'aux approches plus avancées, fondées sur le contenu, hybrides ou enrichies par l'intelligence artificielle. Dans tous les cas, leur objectif demeure de fournir des recommandations personnalisées, pertinentes et adaptées au contexte. Dans un domaine technique comme celui des CPS, cela signifie assister l'utilisateur dans le choix de composants et de configurations répondant à des contraintes de performance, de sécurité, de coût ou de fiabilité.

Les systèmes de recommandation reposent généralement sur deux composantes principales. La première est le moteur de recommandation, chargé de produire et de classer les suggestions à partir des données disponibles, qu'il s'agisse de profils utilisateurs, d'attributs d'objets ou de connaissances métier. La seconde est le gestionnaire d'interaction ou de dialogue, particulièrement important dans les systèmes conversationnels, puisqu'il organise l'échange avec l'utilisateur, collecte des précisions lorsque cela est nécessaire et maintient la cohérence contextuelle des requêtes successives.

Le chapitre présente ensuite les principales familles de techniques de recommandation. Le filtrage collaboratif repose sur l'idée que des utilisateurs ayant eu des préférences similaires dans le passé auront des comportements proches à l'avenir. Il existe sous forme centrée utilisateur ou centrée objet. Son principe général est illustré par la Figure 2.1. Cette approche est performante lorsqu'un historique riche d'interactions est disponible, mais elle souffre de problèmes classiques, notamment le *cold start* pour les nouveaux utilisateurs ou les nouveaux objets, ainsi que la sparsité des données dans les grandes bases.

Le filtrage basé sur le contenu s'appuie quant à lui sur les attributs intrinsèques des objets et sur leur adéquation avec le profil ou les besoins exprimés par l'utilisateur. Dans le cas des CPS, cela revient à comparer les caractéristiques techniques des capteurs, des protocoles ou des composants logiciels avec les exigences du projet. La Figure 2.2 en illustre le principe.

Cette approche permet de recommander de nouveaux composants sans dépendre d'autres utilisateurs, mais elle tend parfois à proposer des résultats trop proches les uns des autres, au détriment de la diversité.

Afin de pallier les limites propres à chacune de ces méthodes, des approches hybrides ont été développées. Elles combinent généralement filtrage collaboratif et filtrage basé sur le contenu, soit en fusionnant leurs scores, soit en choisissant l'une ou l'autre méthode selon la qualité et la disponibilité des données. La Figure 2.3 présente une architecture représentative de ce type d'approche. Les méthodes hybrides améliorent la robustesse de la recommandation, mais elles impliquent également une plus grande complexité de calcul et d'intégration.

Le chapitre examine aussi les approches basées sur la connaissance, qui sont particulièrement importantes pour les CPS. Contrairement aux méthodes purement pilotées par les données, elles s'appuient sur des règles expertes, des ontologies et des modèles de contraintes afin de garantir la compatibilité technique des recommandations. Dans un domaine où la précision est essentielle et où les historiques d'interaction sont limités, ces méthodes constituent un levier important pour intégrer directement les contraintes métier dans la logique de recommandation.

Enfin, le chapitre évoque les approches guidées par l'apprentissage profond, qui ont récemment transformé le domaine des systèmes de recommandation. Grâce à des représentations riches et non linéaires, elles permettent de mieux modéliser les relations entre utilisateurs, objets et contexte. Dans le cadre des CPS, ces modèles présentent un intérêt particulier, car ils peuvent combiner des données hétérogènes, allant des spécifications techniques structurées jusqu'aux contenus textuels non structurés comme les manuels, rapports ou descriptions de maintenance. Toutefois, leur usage soulève encore des questions importantes concernant l'interprétabilité, la disponibilité des données d'apprentissage et la maîtrise des biais.

À partir de cette revue, le chapitre montre que les techniques de recommandation, initialement développées pour des domaines grand public, peuvent être adaptées à la configuration des CPS. Le filtrage collaboratif peut être interprété comme un moyen d'identifier des projets présentant des besoins similaires ; les méthodes basées sur le contenu permettent de prioriser des composants selon leurs caractéristiques techniques ; les approches hybrides réunissent expérience empirique et contraintes métier ; et l'apprentissage profond rend possible l'exploitation conjointe de données numériques, textuelles et contextuelles. Lorsqu'elles sont intégrées dans un cadre conversationnel,

ces techniques deviennent particulièrement adaptées à l'assistance à la conception des CPS.

La section consacrée à la motivation et à la problématique souligne ensuite les limites des systèmes de recommandation classiques dans un contexte d'ingénierie. Contrairement aux applications grand public, la configuration des CPS implique des composants fortement interdépendants qui doivent satisfaire simultanément des contraintes de latence, d'interopérabilité, de sécurité, d'efficacité énergétique et de tolérance aux pannes. Les approches traditionnelles peinent à raisonner sur ces dépendances complexes. Un système collaboratif peut suggérer des composants fréquemment associés dans des projets antérieurs, sans pour autant garantir qu'ils sont compatibles. Un système basé sur le contenu peut retrouver des objets techniquement proches, mais ignorer des contraintes implicites si elles ne sont pas modélisées explicitement.

Une autre difficulté majeure réside dans l'expression des besoins. Dans le domaine des CPS, les exigences de départ sont souvent incomplètes, ambiguës ou exprimées de manière informelle dans des documents techniques, des échanges entre ingénieurs ou des descriptions métier. Leur interprétation nécessite donc un processus interactif capable de comprendre le vocabulaire spécialisé, d'identifier les informations manquantes et d'affiner progressivement la représentation du problème. Le chapitre formule alors la question centrale de recherche : comment concevoir un système de recommandation capable de comprendre, de raisonner et de proposer des configurations CPS en tenant compte à la fois des contraintes techniques, des connaissances métier et de l'évolution des besoins utilisateurs ?

Pour répondre à cette problématique, le chapitre introduit le paradigme des systèmes de recommandation conversationnels. Ces systèmes étendent la recommandation classique en l'insérant dans un dialogue en langage naturel. Ils peuvent interpréter une terminologie spécialisée, poser des questions de clarification, adapter les recommandations lorsque de nouvelles informations apparaissent et justifier les choix effectués à partir de règles, de mesures de performance ou d'exemples antérieurs. Cette approche correspond particulièrement bien aux besoins des CPS, car elle permet de relier des objectifs exprimés à haut niveau, comme la surveillance d'une grandeur physique dans un environnement contraint, à des spécifications techniques détaillées portant sur les capteurs, le réseau ou les ressources de calcul.

Dans ce cadre, **IoT-AID** est présenté comme le système de recommandation conversationnel proposé dans la thèse. Son rôle est d'assister ingénieurs,

chercheurs et intégrateurs dans la conception de solutions CPS. Contrairement aux systèmes conversationnels généralistes, IoT-AID est conçu spécifiquement pour le domaine des CPS et intègre directement des attributs techniques, des contraintes de compatibilité et des bonnes pratiques de conception dans sa logique de recommandation. Le fonctionnement du système débute par une interaction via une interface de type chatbot. Les requêtes de l'utilisateur sont analysées par des modèles de langage afin d'en extraire les besoins essentiels, qui sont ensuite comparés à une base de connaissances structurée contenant les composants CPS et leurs relations sémantiques.

Le chapitre insiste sur le fait que les recommandations produites par IoT-AID reposent sur une approche hybride. Les scores de similarité sémantique permettent d'identifier les composants les plus proches des besoins exprimés, tandis qu'un raisonnement fondé sur des règles garantit leur faisabilité technique et leur cohérence. Lorsque les informations initiales sont incomplètes, le système pose des questions de clarification afin d'affiner progressivement la solution. Les résultats sont finalement présentés sous la forme d'une liste classée et accompagnés d'explications destinées à renforcer la transparence et la confiance de l'utilisateur.

Le chapitre présente également les systèmes de dialogue comme brique essentielle de cette approche. Un système de dialogue est généralement composé d'un module de compréhension du langage naturel (NLU), d'un gestionnaire du dialogue (DM) et d'un module de génération du langage naturel (NLG). Selon leur finalité, ces systèmes peuvent être orientés tâche ou non orientés tâche. Dans le cas d'IoT-AID, l'approche est clairement orientée tâche, puisque l'objectif est de guider l'utilisateur vers une configuration CPS cohérente. La Figure 2.4 illustre l'architecture générale d'un système de dialogue orienté tâche, qui sert ici de référence conceptuelle.

En résumé, ce chapitre établit les fondements théoriques et méthodologiques du travail. Il montre que la recommandation conversationnelle représente une évolution importante des systèmes d'aide à la décision, en particulier pour les contextes complexes comme les CPS. En combinant compréhension du langage naturel, stratégies de recommandation, gestion du dialogue et raisonnement par règles, le cadre proposé permet de dépasser les limites des approches statiques et de construire une assistance plus adaptative, plus explicable et plus proche des besoins réels des utilisateurs.

Ainsi, **IoT-AID** apparaît comme un cadre cohérent et extensible pour la configuration intelligente des CPS. En faisant le lien entre les intentions de

haut niveau exprimées par l'utilisateur et les configurations techniques détaillées, il vise à réduire la barrière d'expertise, à accélérer le prototypage et à promouvoir une automatisation centrée sur l'humain. Le chapitre suivant s'appuie sur ces bases pour présenter plus en détail l'architecture du système et la manière dont ses différentes composantes s'intègrent dans un assistant cohérent de configuration des CPS.

Chapitre 3 : Cadre architectural, conception et détails d'intégration du CPRS conversationnel IoT-AID

Ce chapitre présente le cadre architectural de **IoT-AID**, ainsi que les choix de conception et les mécanismes d'intégration qui permettent au système de fonctionner comme un *Cyber–Physical Recommendation System* (CPRS) conversationnel. La problématique centrale abordée est celle du décalage entre, d'une part, les experts métier qui maîtrisent les besoins fonctionnels d'une application CPS et, d'autre part, les ingénieurs systèmes qui possèdent l'expertise nécessaire pour intégrer les composants matériels, logiciels et communicationnels. Afin de réduire cet écart, IoT-AID est conçu comme un cadre intelligent fondé sur le dialogue en langage naturel et sur un raisonnement assisté par l'intelligence artificielle.

L'idée principale qui sous-tend IoT-AID est de rendre la configuration des CPS à la fois plus intuitive et plus intelligente. Au lieu d'imposer à l'utilisateur une sélection manuelle de capteurs, de protocoles de communication et de ressources de calcul, le système engage une conversation structurée et contextuelle. À travers ce dialogue itératif, il recueille les besoins, interprète les contraintes techniques et affine progressivement ses recommandations. En posant des questions de clarification et en expliquant les raisons de ses suggestions, IoT-AID transforme ainsi la conception des CPS en un processus collaboratif, réduisant la charge cognitive, limitant les erreurs de configuration et favorisant l'alignement avec les contraintes propres au domaine applicatif.

L'architecture de IoT-AID s'organise autour de plusieurs couches technologiques complémentaires. Une couche conversationnelle s'appuie sur des techniques avancées de traitement automatique du langage naturel et sur des modèles de langage comme BERT ou Mistral afin de comprendre les intentions exprimées par l'utilisateur. Une couche de recommandation combine plusieurs approches, notamment le filtrage collaboratif, le filtrage basé sur le

contenu et des stratégies hybrides, afin de produire des suggestions adaptées au contexte. Enfin, une couche d'intégration relie les bases de connaissances sémantiques, les caractéristiques des CPS et les modules d'explicabilité, garantissant à la fois la validité technique et la transparence des recommandations. L'ensemble forme un cadre de collaboration homme–machine conçu pour s'adapter à différents contextes, notamment industriels, médicaux ou liés aux infrastructures intelligentes.

Le chapitre commence par proposer un modèle conceptuel simplifié mais expressif des CPS, servant de fondement à IoT-AID. Inspirée de modèles existants, cette architecture distingue trois couches interdépendantes : la couche physique, qui regroupe les capteurs, les actionneurs et les dispositifs embarqués ; la couche de communication, qui assure la transmission des données à travers des protocoles tels que MQTT, Zigbee, Wi-Fi ou Bluetooth ; et la couche cyber, qui englobe l'analyse des données, la logique de contrôle, la prise de décision et les modules d'intelligence artificielle. Cette structuration n'a pas uniquement une fonction descriptive : elle joue aussi le rôle de couche sémantique pour la construction de la base de connaissances utilisée par le système de recommandation. Elle constitue donc le socle conceptuel sur lequel repose le raisonnement d'IoT-AID.

Le chapitre décrit ensuite l'architecture générale du système de recommandation proposé. Celle-ci est illustrée par la Figure 3.1, qui met en évidence les principaux composants de IoT-AID : une interface conversationnelle, un module de compréhension du langage naturel, une base de connaissances et un moteur hybride de recommandation. Le système reçoit en entrée les informations fournies par l'utilisateur, telles que le domaine d'application, les objectifs, les contraintes techniques ou encore le budget. Ces informations sont d'abord prétraitées afin d'extraire les caractéristiques pertinentes et les mots-clés spécifiques au domaine. Elles sont ensuite comparées aux données stockées dans la base de connaissances, qui contient des descriptions détaillées de capteurs, d'actionneurs et d'autres composants CPS.

Le processus de recommandation suit une logique progressive. Il commence par l'identification de l'intention de l'utilisateur à partir de l'analyse textuelle. Le système récupère ensuite des composants candidats à l'aide de mesures de similarité et de requêtes sémantiques. Ces résultats sont ensuite classés et filtrés en fonction des contraintes détectées et du niveau de confiance associé. Enfin, les recommandations peuvent être affinées à travers les retours de l'utilisateur, ce qui permet un apprentissage progressif et une amélioration continue de la base de connaissances. Celle-ci n'est donc

pas uniquement un espace de stockage, mais un véritable réservoir de méta-connaissances permettant à la fois la représentation structurée des objets, le raisonnement, la traçabilité et l'explicabilité.

La représentation des métadonnées techniques dans la base de connaissances est illustrée par la Figure 3.2, qui montre un exemple de fichier XML décrivant un capteur d'humidité. Ce format permet de structurer des attributs essentiels tels que la plage de mesure, le principe de détection, les contraintes environnementales ou les interfaces de communication. Ces informations sont ensuite transformées en représentations exploitables par les modules de recommandation. Cette structuration contribue à faire de la base de connaissances un élément central de l'architecture, garantissant la cohérence entre les requêtes utilisateurs et les caractéristiques techniques des composants.

Afin d'illustrer concrètement le fonctionnement de l'approche, le chapitre présente un cas d'étude portant sur un système de surveillance en temps réel destiné à des personnes âgées. Dans ce scénario, l'objectif est d'assister les aidants tout en assurant la sécurité des patients, à travers la recommandation de capteurs et de configurations adaptées à des besoins tels que la détection de mouvement, le suivi médicamenteux ou la surveillance de sécurité. La Figure 3.3 montre cet exemple de surveillance à domicile pour personnes âgées. Selon le niveau de précision des besoins exprimés, le système peut recommander soit des capteurs spécifiques, comme un détecteur de mouvement ou un capteur de température, soit des combinaisons plus larges intégrant plusieurs dispositifs, par exemple pour détecter le gaz, le feu et l'humidité. Ce cas d'usage met en évidence la capacité du système à proposer des recommandations fines ou globales, en fonction du contexte et du niveau d'expertise de l'utilisateur.

Le chapitre accorde ensuite une place importante au module de traitement du langage naturel, qui constitue le cœur de la compréhension conversationnelle du système. Contrairement aux approches classiques fondées sur des règles explicites ou sur une reconnaissance d'entités nommées strictement définie, IoT-AID adopte une compréhension du texte pilotée par des modèles de langage et par des mécanismes de *prompting*. Cette stratégie offre davantage de flexibilité, une meilleure adaptabilité à plusieurs domaines et une plus grande robustesse face à la diversité des formulations linguistiques.

Le rôle principal du module NLP est de transformer les énoncés utilisateurs en représentations exploitables par le moteur de recommandation. Il doit être capable d'interpréter des besoins exprimés en langage naturel, par

exemple la demande d'un capteur d'humidité à faible coût pour surveiller une serre, puis de convertir cette demande en représentations sémantiques structurées. Ces représentations capturent à la fois les contraintes explicites, comme le coût, le type de capteur ou le domaine d'application, et les intentions implicites, comme la préférence pour un système fiable ou pour une surveillance environnementale. Le module interagit étroitement avec la base de connaissances et avec le moteur de recommandation afin d'assurer la cohérence de l'ensemble.

La première étape de ce module consiste en un prétraitement et une normalisation du texte. Les entrées utilisateur sont nettoyées, tokenisées et normalisées afin de réduire les variations linguistiques susceptibles d'altérer l'interprétation sémantique. Cette phase comprend notamment la suppression des éléments non informatifs, l'expansion des abréviations et la standardisation des unités et formats de mesure. L'objectif est d'assurer que des formulations équivalentes soient traitées de manière cohérente avant d'être transmises aux modèles de langage.

Au cœur du module se trouve ensuite une étape de compréhension sémantique fondée sur des *prompts*. Un grand modèle de langage, tel que Mistral-7B, est utilisé pour interpréter les requêtes et en extraire plusieurs éléments clés : l'intention de l'utilisateur, les attributs principaux liés à la requête et les dépendances contextuelles entre besoins exprimés et composants CPS potentiels. Cette phase ne repose donc pas sur une ingénierie manuelle des caractéristiques, mais sur des gabarits de requête qui guident le modèle dans l'identification du but recherché, des contraintes comme le budget ou la connectivité, et des liens implicites entre le problème posé et les composants concernés. Le résultat de cette étape est une représentation sémantique latente, adaptée au raisonnement et à la recherche par similarité.

Une fois ces embeddings produits, un module de post-traitement et d'analyse organise les informations extraites sous une forme interrogeable par le moteur de recommandation. Des techniques comme le clustering sémantique, la recherche vectorielle par similarité et le filtrage contextuel permettent d'associer les besoins exprimés par l'utilisateur aux profils des composants stockés dans la base de connaissances. Cela rend possible la gestion de dialogues multi-tours, dans lesquels l'intention évolue progressivement, par exemple lorsque l'utilisateur passe d'une demande générique de capteur d'humidité à une exigence plus spécifique portant sur un modèle sans fil et destiné à un usage intérieur.

Le chapitre souligne également que ce module NLP ne doit pas être vu

comme un composant figé. Il s'inscrit dans une logique d'amélioration continue fondée sur les interactions avec les utilisateurs. Les recommandations acceptées ou rejetées sont enregistrées et analysées afin d'ajuster les *prompts*, les seuils de similarité et, plus largement, le comportement du système. Cette boucle de rétroaction permet à IoT-AID de s'adapter progressivement à de nouveaux domaines, à de nouveaux vocabulaires et à des profils utilisateurs variés, sans nécessiter une reconfiguration manuelle complète.

En conclusion, ce chapitre établit les bases architecturales de IoT-AID en montrant comment la représentation des CPS, la structuration des connaissances, la recommandation hybride et la compréhension conversationnelle sont intégrées dans un cadre cohérent. Il démontre que IoT-AID ne se limite pas à un simple moteur de suggestion, mais constitue une architecture modulaire, explicable et évolutive, capable de transformer des besoins exprimés en langage naturel en recommandations techniques exploitables. En ce sens, il contribue à rapprocher l'expertise métier des exigences d'intégration technique, ouvrant la voie à une configuration plus accessible, plus précise et plus transparente des systèmes cyber–physiques.

Le chapitre suivant s'appuiera sur cette base conceptuelle pour présenter l'implémentation pratique du système, en mettant l'accent sur l'intégration des grands modèles de langage dans la chaîne conversationnelle et sur la réalisation opérationnelle du cadre proposé.

Chapitre 4 : Cadre d'implémentation de IoT-AID et intégration des grands modèles de langage dans le développement d'un CPRS conversationnel

Ce chapitre présente le cadre d'implémentation de **IoT-AID** en mettant l'accent sur l'intégration des grands modèles de langage dans une chaîne de recommandation conversationnelle appliquée aux systèmes cyber–physiques. Dans le contexte de l'Industrie 4.0, les CPS permettent de connecter de manière étroite les processus physiques et les capacités de calcul afin de réaliser de la surveillance en temps réel, du contrôle intelligent et de la prise de décision automatisée. Toutefois, malgré ces avantages, leur conception et leur configuration demeurent complexes, car elles exigent une connaissance approfondie des composants, des contraintes techniques et des interactions entre couches physiques, cyber et communicationnelles.

Afin de répondre à cette difficulté, le chapitre introduit une implémentation de IoT-AID fondée sur des modèles avancés d'apprentissage automatique, des architectures profondes et des modèles de langage comme BERT et Sentence Transformers. L'objectif de cette implémentation est de permettre au système d'interpréter des besoins exprimés en langage naturel et de fournir des recommandations personnalisées, contextuelles et techniquement exploitables pour la configuration des CPS. En combinant compréhension sémantique, base de connaissances structurée et moteur hybride de recommandation, IoT-AID cherche à faire le lien entre l'intention humaine et la mise en œuvre technique.

Le chapitre commence par détailler l'architecture globale du système et l'intégration de ses modules. IoT-AID est conçu comme un système modulaire dans lequel la gestion du dialogue, le traitement du langage naturel, la base de connaissances, le moteur de recommandation hybride et l'interface d'explicabilité coopèrent pour produire une assistance interactive. Cette organisation générale est illustrée par la Figure 4.1, qui montre les interactions entre les différentes composantes du système. Le gestionnaire de dialogue maintient le contexte sur plusieurs tours de conversation, interprète les intentions de l'utilisateur, identifie les informations manquantes et assure la cohérence de l'échange. Le module NLP traite les requêtes utilisateurs afin d'en extraire des représentations sémantiques riches. La base de connaissances stocke les composants CPS, leurs spécifications et les règles de compatibilité. Le moteur de recommandation hybride combine des approches de filtrage basées sur le contenu, des stratégies collaboratives et des règles métier. Enfin, une interface d'explicabilité et de retour utilisateur permet de justifier les recommandations et d'améliorer progressivement le comportement du système.

Le flux d'intégration entre ces modules suit une logique séquentielle. L'utilisateur formule une requête en langage naturel décrivant un objectif de conception ou un besoin en composant. Cette requête est prétraitée puis encodée sous forme d'embeddings sémantiques. Le gestionnaire de dialogue interprète l'intention, met à jour le contexte et détecte d'éventuelles informations manquantes. Le moteur de recommandation interroge ensuite la base de connaissances à partir de critères sémantiques et de contraintes techniques. Les résultats sont affinés au fil du dialogue, grâce à des questions de clarification, avant d'être classés, expliqués et présentés à l'utilisateur. Cette conception modulaire rend le système à la fois extensible, transparent et adapté à des utilisateurs ayant des niveaux d'expertise différents.

Le chapitre consacre ensuite une section complète à l'implémentation de la chaîne NLP, qui constitue le noyau de la compréhension conversationnelle de IoT-AID. Cette chaîne fonctionne comme un processus d'interprétation sémantique en plusieurs étapes, transformant progressivement les requêtes textuelles en représentations exploitables par le système. La Figure 4.2 présente cette chaîne NLP et met en évidence les principales étapes de traitement. L'entrée est d'abord acquise sous la forme d'une requête libre, exprimant des objectifs ou des contraintes, par exemple un besoin de capteurs pour surveiller l'humidité et la température dans une serre. Le texte subit ensuite un prétraitement comprenant la tokenisation, la lemmatisation, la suppression des mots vides et la normalisation de la terminologie propre aux CPS.

Après cette phase, la requête est encodée à l'aide de modèles transformer comme BERT ou Sentence Transformers. Ces encodeurs produisent des vecteurs denses qui capturent les relations sémantiques et les nuances contextuelles de l'entrée utilisateur. Le chapitre formalise ce processus à l'aide de l'expression $E = f_\theta(T)$, où T représente la séquence tokenisée et f_θ l'encodeur paramétré. Une mesure de similarité cosinus, définie par $S(x, y) = \cos(E_x, E_y)$, permet ensuite de comparer l'embedding de la requête à ceux des composants stockés dans la base de connaissances. Cette stratégie permet de dépasser une simple correspondance lexicale et d'exploiter la proximité sémantique réelle entre besoins utilisateurs et descriptions techniques.

Le suivi du contexte conversationnel est également un élément essentiel de cette chaîne. IoT-AID maintient un état mémoire capable de conserver les intentions et contraintes mentionnées lors des tours précédents. Cela permet d'interpréter correctement des reformulations courtes ou implicites, comme une demande d'amélioration énergétique ou l'ajout d'une connectivité sans fil, sans obliger l'utilisateur à répéter l'ensemble du contexte. Le résultat final de cette chaîne est une représentation structurée de l'intention de conception, qui sert directement d'entrée au moteur hybride de recommandation.

Le chapitre analyse ensuite en détail l'intégration des grands modèles de langage dans IoT-AID. Ceux-ci ne sont pas seulement employés comme encodeurs de texte, mais comme médiateurs de connaissances capables d'interpréter, de raisonner et de générer des recommandations pertinentes dans le contexte des CPS. Plusieurs rôles leur sont attribués. Ils assurent d'abord une compréhension sémantique fine des relations entre termes techniques et objectifs utilisateurs. Ils servent également à reconnaître les intentions et les entités, en identifiant par exemple si l'utilisateur cherche à sélectionner

un capteur, configurer une passerelle ou optimiser une consommation énergétique. Les LLMs contribuent aussi à l'expansion contextuelle des requêtes incomplètes ou ambiguës et facilitent le raisonnement fondé sur la connaissance en interagissant avec la base de connaissances.

L'implémentation adopte une stratégie hybride combinant des modèles préentraînés généralistes et un ajustement au domaine des CPS. Plus précisément, Sentence-BERT (SBERT) est utilisé comme compromis entre richesse sémantique et efficacité de calcul. L'ajustement du modèle est réalisé à partir de jeux de données textuels liés aux CPS, comprenant des descriptions de composants, des manuels de configuration et des documents de conception. Le chapitre introduit une fonction de perte contrastive, exprimée par

$$\mathcal{L} = 1 - \cos(E_{anchor}, E_{positive}) + \max(0, \cos(E_{anchor}, E_{negative}) - m),$$

afin de rapprocher les représentations sémantiques d'éléments similaires tout en éloignant celles des éléments non pertinents. Ce mécanisme améliore la structuration de l'espace vectoriel dans lequel les composants CPS sont représentés.

Une fois produits, ces embeddings sont indexés dans une base vectorielle, telle que FAISS ou Milvus, afin de permettre une recherche rapide par similarité. Lorsqu'une requête utilisateur est encodée, sa représentation vectorielle est comparée à celles des composants disponibles selon la formule de similarité

$$S(Q, D_i) = \frac{E_Q \cdot E_{D_i}}{\|E_Q\| \|E_{D_i}\|}.$$

Les éléments les plus proches sont ensuite transmis au moteur de recommandation pour être reclassés et validés. Cette architecture rend possible une recommandation en temps quasi réel, tout en conservant un niveau élevé de pertinence sémantique.

Le chapitre insiste également sur la question de l'explicabilité, essentielle dans des domaines industriels où les décisions automatisées doivent être comprises et justifiées. Pour cette raison, IoT-AID intègre plusieurs mécanismes d'interprétation. Ceux-ci incluent la visualisation des poids d'attention, la mise en évidence de requêtes ou d'entrées historiques proches sur le plan sémantique, ainsi que des méthodes d'attribution de caractéristiques permettant d'identifier les termes qui ont le plus influencé la décision. Cette dimension explicable vise à renforcer la confiance de l'utilisateur et à inscrire le système dans le cadre plus large de l'intelligence artificielle explicable.

Une autre partie importante du chapitre concerne l'implémentation du

moteur de recommandation. Cette partie se concentre sur la construction d'un système capable de proposer des capteurs adaptés à partir de descriptions textuelles riches. Pour constituer le jeu de données initial, une stratégie d'acquisition automatique a été mise en place à partir de plateformes de commerce électronique. Après avoir étudié plusieurs sources, un site recensant 1816 capteurs a été retenu. Une phase de vérification manuelle préalable a permis de contrôler la qualité des données avant le lancement d'un scraping à grande échelle. Ce processus, réalisé en Python avec Selenium, BeautifulSoup et les bibliothèques standards de gestion de fichiers, a permis de construire un jeu de données complet intégrant noms, descriptions, prix et spécifications techniques.

Le pipeline d'acquisition, de transformation et de structuration des données est représenté dans la Figure 4.3. Après le scraping, les données ont été nettoyées, normalisées et converties entre différents formats, notamment CSV et JSON, afin de standardiser leur exploitation. Une vérification finale sur un échantillon de capteurs a confirmé l'alignement des données extraites avec les descriptions originales. La Figure 4.4 illustre par ailleurs la structure du jeu de données de capteurs, qui comprend des attributs tels que le nom, le type, la description et les domaines d'application.

Le moteur de recommandation fondé sur BERT s'appuie sur une chaîne relativement simple mais efficace. Les descriptions des produits sont d'abord prétraitées puis concaténées dans un champ textuel unifié. Elles sont ensuite encodées à l'aide d'un modèle `bert-base-nli-mean-tokens` issu de la bibliothèque Sentence Transformers. Chaque produit est ainsi représenté par un vecteur dense de dimension 768. Lorsqu'un utilisateur soumet une requête, celle-ci est encodée de la même manière, puis comparée aux embeddings des produits à l'aide de la similarité cosinus. Les éléments les plus proches sont classés, et les dix meilleures recommandations sont retournées. Le processus de raffinement du modèle et son ajustement à des jeux de données spécialisés sont illustrés par la Figure 4.7.

Le chapitre met en avant plusieurs résultats qualitatifs de cette approche. Le moteur basé sur BERT montre une bonne compréhension contextuelle, notamment pour des requêtes complexes combinant plusieurs attributs techniques. Il permet également de prendre en compte des préférences de marque lorsque celles-ci sont pertinentes, et il assure des temps de réponse rapides grâce au pré-calcul des embeddings du catalogue. En revanche, certaines limites subsistent, notamment pour les requêtes ambiguës, pour l'absence d'un ajustement plus spécifique au domaine, pour le manque de mécanismes

de retour utilisateur en temps réel et pour l'absence d'intégration de données multimodales comme les images ou les avis.

Le chapitre traite ensuite de la partie classification textuelle du CPRS, qui joue le rôle de point d'entrée entre l'utilisateur et le système. Cette composante vise à comprendre les objectifs exprimés par l'utilisateur et à les relier à des applications et à des capteurs adaptés. Faute de jeux de données directement disponibles, un processus de génération de données a été mis en place à partir du web scraping et de l'IA générative. Il a permis de construire un corpus d'environ 600 applications uniques, ensuite converti dans un format exploitable pour l'apprentissage. La Figure 4.6 présente l'architecture générale de cette partie du système, tandis que la Figure 4.7 ainsi que la figure sur le jeu d'entités décrivent la structuration des données utilisées.

Le prétraitement des textes joue ici un rôle essentiel. Les données sont nettoyées, normalisées, tokenisées et lemmatisées afin d'en extraire des représentations cohérentes. Ensuite, deux méthodes principales d'extraction de caractéristiques sont utilisées : la vectorisation TF-IDF et les embeddings BERT. La première fournit une représentation pondérée des mots selon leur importance, tandis que la seconde permet de capturer le contexte sémantique profond des phrases. Cette combinaison contribue à améliorer la capacité du système à comprendre les besoins exprimés et à les relier à des classes ou configurations CPS pertinentes.

Dans l'ensemble, ce chapitre montre que l'implémentation de IoT-AID repose sur une articulation étroite entre la compréhension du langage naturel, la structuration de la connaissance, la recherche vectorielle et la recommandation hybride. Les LLMs y occupent une place stratégique, puisqu'ils permettent de produire des représentations riches, d'interpréter les intentions, de gérer le contexte conversationnel et de soutenir des recommandations plus pertinentes. Les bases de données structurées, les pipelines d'acquisition de données et les mécanismes de similarité complètent cette architecture en assurant la faisabilité pratique du système.

En conclusion, ce chapitre établit le cadre technique qui rend possible le fonctionnement de IoT-AID comme assistant conversationnel pour la configuration des CPS. Il montre comment les modèles de langage, les techniques de traitement sémantique, les mécanismes de recommandation et la structuration des données sont combinés dans une architecture cohérente. Cette implémentation constitue une étape essentielle entre la modélisation conceptuelle du système et son évaluation expérimentale. Le chapitre suivant s'appuiera sur ces éléments pour analyser le fonctionnement complet du CPRS,

mesurer ses performances et évaluer sa capacité à fournir des recommandations pertinentes, explicables et utiles dans des scénarios réels.

Chapitre 5 : Implémentation et évaluation du système de recommandation cyber–physique

Ce chapitre présente la mise en œuvre concrète ainsi que l'évaluation du système de recommandation cyber–physique (CPRS) développé dans le cadre de ce travail. Après avoir établi les fondements théoriques et les choix d'architecture dans les chapitres précédents, l'objectif est ici de démontrer la faisabilité du système en conditions réelles, d'analyser ses performances et d'évaluer son utilité du point de vue des utilisateurs. Il s'agit ainsi de valider non seulement la pertinence technique de la solution proposée, mais également sa capacité à répondre aux besoins des utilisateurs, qu'ils soient experts ou non dans le domaine des CPS.

Comme discuté dans le chapitre 4, la configuration des systèmes cyber–physiques reste un processus complexe, impliquant la sélection et l'intégration de composants hétérogènes tels que des capteurs, des actionneurs et des protocoles de communication. Les approches traditionnelles reposent souvent sur une expertise approfondie, des essais manuels ou des systèmes de recommandation opaques, peu adaptés aux exigences actuelles de l'Industrie 4.0. Le CPRS proposé vise à dépasser ces limitations en introduisant une architecture modulaire, interprétable et centrée utilisateur, capable de traduire des besoins exprimés en langage naturel en configurations CPS exploitables. Pour cela, il combine des embeddings sémantiques basés sur Sentence-BERT, des modules de classification d'intention, des mécanismes de vérification de suffisance d'information, une recherche par similarité, des techniques d'explicabilité comme SHAP et un module de génération de réponses basé sur Mistral-7B.

L'architecture générale du système repose sur une chaîne de traitement modulaire organisée en plusieurs étapes successives. L'utilisateur interagit avec le système via une interface conversationnelle, où il exprime ses besoins en langage naturel. Ces requêtes sont prétraitées, encodées sous forme de vecteurs sémantiques, puis analysées afin d'identifier leur intention. Une fois l'intention détectée, un module de vérification évalue si la requête contient suffisamment d'informations pour permettre une recommandation fiable. Dans

le cas contraire, le système engage un dialogue avec l'utilisateur afin de compléter les paramètres manquants. Lorsque la requête est jugée complète, un moteur de recherche par similarité compare l'embedding de la requête avec ceux des composants stockés dans la base de connaissances. Les meilleurs candidats sont ensuite classés et enrichis par des explications générées à l'aide de SHAP et reformulées en langage naturel par le modèle Mistral-7B. Ce processus est complété par un module de retour utilisateur permettant d'évaluer la qualité des recommandations et d'améliorer progressivement le système.

Le pipeline du CPRS débute par une étape de prétraitement des données textuelles, essentielle pour assurer la cohérence entre les requêtes utilisateurs et les descriptions des composants. Malgré la robustesse des modèles SBERT, l'intégration de techniques classiques telles que la normalisation, la tokenisation et la lemmatisation s'est révélée nécessaire, notamment en raison de la nature technique et structurée des données CPS. Ces étapes permettent de réduire les variations lexicales et d'améliorer l'alignement sémantique entre les différentes sources d'information. Les textes ainsi nettoyés sont ensuite encodés en vecteurs de dimension 768 à l'aide de SBERT, constituant la base des opérations de similarité.

L'étape suivante consiste à classifier l'intention de la requête. Un réseau de neurones simple, basé sur des couches entièrement connectées et une fonction Softmax, est utilisé pour associer chaque requête à un domaine CPS spécifique, tel que la surveillance environnementale, la santé ou la sécurité industrielle. Cette classification joue un rôle crucial en restreignant l'espace de recherche aux composants pertinents, améliorant ainsi la précision et l'efficacité du système. Elle contribue également à la transparence en fournissant un contexte explicite à la recommandation.

Le module de vérification de suffisance d'information constitue une contribution importante du système. En effet, les requêtes utilisateurs sont souvent incomplètes ou ambiguës. Ce module compare les attributs extraits de la requête avec un ensemble d'attributs requis, définis pour chaque domaine. Lorsque des informations essentielles sont absentes, le système génère automatiquement des questions de clarification. Ce mécanisme permet non seulement d'améliorer la qualité des recommandations, mais aussi de guider les utilisateurs non experts dans la formulation de leurs besoins.

Une fois la requête enrichie, le moteur de recherche par similarité entre en jeu. Il repose sur le calcul de la similarité cosinus entre les embeddings des requêtes et ceux des composants CPS. Afin de garantir des performances en

temps réel, une structure d'indexation basée sur des méthodes de recherche approximative (ANN) est utilisée. Une étape de réordonnancement permet ensuite d'intégrer des contraintes techniques spécifiques, telles que les plages de mesure ou les protocoles de communication, afin d'assurer la faisabilité des recommandations.

L'un des apports majeurs du CPRS réside dans son module d'explicabilité basé sur SHAP. Cette approche permet de quantifier l'importance de chaque terme de la requête dans le processus de recommandation. Les explications générées offrent une transparence essentielle, en particulier dans des contextes industriels où la confiance dans les systèmes automatisés est primordiale. Elles permettent également aux concepteurs d'identifier d'éventuels biais ou incohérences dans le fonctionnement du système.

Le module de génération de réponses, basé sur le modèle Mistral-7B, transforme les résultats techniques en explications compréhensibles. En combinant les métadonnées des composants, les résultats de SHAP et le contexte de la requête, ce module produit des réponses naturelles et adaptées au niveau d'expertise de l'utilisateur. Cette capacité à traduire des informations complexes en langage accessible constitue un élément clé pour l'adoption du système.

La base de connaissances constitue le socle du CPRS. Elle a été construite à partir de plusieurs sources, incluant le web scraping, l'analyse de documents techniques et la génération de données synthétiques. Les données ont été normalisées et structurées dans une base MongoDB, permettant une grande flexibilité face à l'hétérogénéité des composants CPS. Chaque entrée comprend des informations détaillées telles que le type de capteur, sa plage de fonctionnement, sa précision, ses protocoles de communication et son domaine d'application. Cette structuration facilite la recherche et l'interprétation des résultats.

L'implémentation du système a été réalisée à l'aide du framework Flask, permettant l'intégration des modules d'apprentissage automatique avec une interface web interactive. Les embeddings ont été pré-calculés et indexés à l'aide de FAISS, garantissant une faible latence lors des requêtes. Les tests ont montré un temps de réponse moyen inférieur à 0,5 seconde, confirmant la compatibilité du système avec des scénarios interactifs.

L'évaluation du CPRS repose sur une combinaison d'analyses quantitatives et qualitatives. Sur le plan technique, une comparaison entre Word2Vec, BERT et SBERT a été réalisée. Les résultats montrent que SBERT offre les meilleures performances, avec une précision de 0,85, un score F1 de 0,85 et

une erreur moyenne plus faible. Il présente également un bon compromis entre précision et coût computationnel. Une analyse complémentaire sur 200 requêtes a confirmé une précision top-3 de 0,92, démontrant la robustesse du système face à des requêtes variées.

Le module de vérification de suffisance a également été évalué. Environ 34% des requêtes initiales étaient incomplètes, mais toutes ont pu être enrichies en moyenne en 1,3 tours de dialogue. Cela confirme l'efficacité de ce module dans l'amélioration de la qualité des recommandations.

Une étude utilisateur a été menée auprès de 100 participants de profils variés. Les résultats montrent un taux de satisfaction global de 89%, avec 91% des utilisateurs estimant les explications claires et 86% jugeant les recommandations pertinentes. Les utilisateurs non experts ont particulièrement apprécié l'interface conversationnelle, qui simplifie l'accès aux technologies CPS. Les explications SHAP ont été identifiées comme un facteur clé de confiance, avec une note moyenne de 4,5/5.

Malgré ces résultats positifs, certaines limites doivent être soulignées. La dépendance à des données annotées peut introduire des biais, et la couverture du jeu de données reste limitée à certains domaines CPS. De plus, l'évaluation repose en partie sur des mesures subjectives, qui pourraient être complétées par des tests en conditions réelles.

En conclusion, ce chapitre démontre que le CPRS constitue une solution efficace pour la configuration des systèmes cyber–physiques. Grâce à l'intégration de techniques avancées de traitement du langage, de recommandation sémantique et d'intelligence artificielle explicable, le système répond aux enjeux de complexité, de transparence et d'accessibilité. Il se positionne à la fois comme un outil pratique pour les utilisateurs et comme une plateforme de recherche pour l'étude des systèmes de recommandation conversationnels dans des contextes industriels.

7.4 Contributions

This thesis addresses the challenges of Cyber-Physical System (CPS) configuration by proposing a novel Conversational Cyber-Physical Recommendation System (CPRS). The contributions of this work span architectural design, semantic reasoning, explainability, and user-centered interaction. Collectively, they bridge the gap between the technical complexity of CPS design and the need for accessible, transparent, and intelligent decision-support tools.

The main contributions are summarized as follows :

7.4.1 A Modular and Explainable Conversational CPS Recommender Architecture

The primary contribution of this thesis is the design of a modular and fully explainable architecture for CPS recommendation. The proposed pipeline decomposes the recommendation process into distinct components : preprocessing, semantic embedding (SBERT), intent classification, information sufficiency checking, similarity-based retrieval, explainability (SHAP), and response generation (Mistral-7B).

This modular design contrasts with traditional end-to-end black-box systems by ensuring :

— transparency in each decision stage,
— ease of debugging and system improvement,
— adaptability across CPS domains.

7.4.2 Semantic Retrieval Grounded in a Curated CPS Knowledge Base

This work introduces a hybrid approach combining Sentence-BERT embeddings with a structured and curated CPS knowledge base. Unlike purely generative LLM-based recommenders, CPRS grounds all recommendations in verified component metadata, ensuring reliability and domain relevance.

This contribution enables :

— semantically meaningful matching between user queries and CPS components,
— robustness to the cold-start problem,
— suitability for safety-critical industrial applications.

7.4.3 Integration of SHAP for Explainable CPS Recommendations

A key contribution is the integration of SHAP (SHapley Additive exPlanations) into the recommendation pipeline to provide feature-level interpretability. CPRS does not rely on post-hoc textual justifications, but instead delivers explanations grounded in the actual retrieval process.

This ensures :

— transparent attribution of recommendation decisions,
— improved user trust and understanding,
— diagnostic capabilities for system developers.

7.4.4 User-Centered Interaction via Information Sufficiency and Feedback

This thesis introduces a novel interaction mechanism combining :
— a rule-based information sufficiency module, and
— a structured user feedback loop.

The sufficiency module detects incomplete queries and generates clarification questions, guiding users especially non-experts toward well-defined requirements. The feedback mechanism captures user perceptions of clarity, relevance, and transparency, enabling continuous system refinement.

7.4.5 Efficient and Interpretable Semantic Recommendation Pipeline

Through experimental evaluation, this work demonstrates that SBERT-based embeddings provide an optimal balance between :
— accuracy (0.85),
— latency (real-time performance),
— memory efficiency.

Compared to large-scale LLM systems (e.g., GPT or Gemini), CPRS achieves deterministic, efficient, and interpretable recommendations suitable for interactive CPS configuration.

7.4.6 A Domain-Specific Conversational Framework for CPS Configuration

Finally, this thesis contributes a domain-specific conversational recommendation framework tailored to CPS. Unlike general-purpose conversational AI systems, CPRS :
— integrates structured technical constraints (e.g., ranges, protocols),
— ensures compatibility and feasibility of recommendations,
— provides traceable and explainable decision-making.

This makes CPRS particularly suitable for Industry 4.0 and emerging Industry 5.0 environments, where reliability, transparency, and human–AI collaboration are critical.

Chapitre 7 : Conclusion générale et perspectives de recherche

Ce dernier chapitre synthétise l'ensemble des apports de cette thèse et ouvre les perspectives de recherche associées au développement futur du *Cyber-Physical Recommendation System* (CPRS). L'objectif général de ce travail était de répondre à une difficulté centrale de la conception des systèmes cyber–physiques : la complexité de la sélection et de la configuration de composants hétérogènes, qui exige habituellement une expertise technique avancée, un effort manuel important et une capacité à interpréter des contraintes multiples liées au domaine, à l'environnement d'usage et à l'interopérabilité des éléments du système.

Pour répondre à ce défi, cette thèse a proposé, conçu, implémenté et évalué un cadre modulaire, explicable et conversationnel destiné à assister les utilisateurs dans la configuration des CPS. Le CPRS développé repose sur une articulation cohérente entre plusieurs briques complémentaires : des embeddings sémantiques basés sur Sentence-BERT pour la compréhension des requêtes, un module de classification d'intention pour contextualiser les besoins, un mécanisme de vérification de suffisance de l'information pour détecter les paramètres manquants, une recherche par similarité fondée sur les descriptions de composants, une couche d'explicabilité basée sur SHAP et un module de génération de réponses naturelles à l'aide de Mistral-7B. À ces éléments s'ajoute une base de connaissances structurée, enrichie à partir de données issues du web scraping, de fiches techniques, de validation manuelle et d'un enrichissement génératif contrôlé.

L'un des résultats majeurs de cette thèse est d'avoir montré qu'il est possible de faire le lien entre des besoins exprimés en langage naturel et des recommandations techniques exploitables dans un contexte de CPS. Le système ne se contente pas de renvoyer un composant pertinent ; il accompagne également l'utilisateur dans un processus de clarification, de justification et de compréhension. Ce positionnement est particulièrement important dans les environnements industriels et techniques, où la confiance dans les systèmes d'aide à la décision dépend non seulement de leur performance, mais aussi de leur capacité à rendre leur raisonnement compréhensible.

Sur le plan scientifique, ce travail apporte plusieurs contributions. Il propose d'abord une architecture modulaire et explicable, qui se distingue des approches de type boîte noire en séparant clairement les étapes de compréhension, de classification, de vérification, de recherche, d'explication et de

génération. Il montre ensuite l'intérêt d'une intégration entre représentations sémantiques et base de connaissances spécialisée, ce qui permet d'ancrer les recommandations dans des métadonnées vérifiées et donc d'éviter les limites des approches purement génératives. Il introduit également l'usage de SHAP dans un cadre de recommandation conversationnelle pour CPS, de manière à fournir des explications fondées sur les éléments réellement décisifs dans le processus de recommandation. Enfin, il met en avant une interaction centrée utilisateur, dans laquelle les mécanismes de clarification et de retour d'expérience permettent d'améliorer à la fois la qualité des recommandations et l'acceptabilité du système.

Les résultats expérimentaux obtenus confirment la pertinence de cette approche. L'évaluation quantitative a montré que SBERT offre le meilleur compromis entre précision, latence et coût mémoire parmi les modèles comparés. L'évaluation qualitative, menée auprès d'utilisateurs de profils variés, a mis en évidence un fort niveau de satisfaction, une bonne compréhension des recommandations et une augmentation significative de la confiance grâce aux explications générées. Ces résultats démontrent que l'association entre recherche sémantique, explicabilité et interaction conversationnelle constitue une stratégie efficace pour réduire la charge cognitive liée à la configuration des CPS, tant pour les experts que pour les utilisateurs non spécialistes.

Au-delà de ses performances, le CPRS s'inscrit dans une vision plus large d'une intelligence artificielle explicable et centrée sur l'humain pour l'Industrie 4.0 et l'Industrie 5.0. En intégrant la transparence, l'interaction et la prise en compte du retour utilisateur dans la chaîne de recommandation, le système contribue à une approche dans laquelle l'IA ne remplace pas l'expertise humaine, mais la complète et l'augmente. Cette orientation est particulièrement importante dans les contextes où les décisions techniques doivent rester auditables, justifiables et adaptées aux besoins réels des utilisateurs.

Toutefois, plusieurs limites subsistent. La base de connaissances repose en partie sur des processus de collecte fragiles, notamment le web scraping, et demeure essentiellement statique. Les embeddings utilisés, bien qu'efficaces, restent issus de modèles généralistes et ne capturent pas toujours les nuances les plus fines du vocabulaire spécialisé des CPS. De plus, les mécanismes d'explicabilité, bien qu'utiles, reposent parfois sur des approximations intermédiaires qui pourraient être améliorées par des techniques plus directement intégrées aux architectures transformeurs. Enfin, l'interaction reste aujourd'hui textuelle, alors que des contextes industriels réels pourraient bénéficier d'entrées multimodales comme la voix, les schémas ou les diagrammes.

Ces limites ouvrent naturellement plusieurs perspectives de recherche. Une première direction consiste à renforcer la robustesse de la collecte et de la structuration des données, en privilégiant des formats normalisés, des API ou des partenariats directs avec les fabricants de composants. Une seconde perspective porte sur l'adaptation au domaine, à travers le fine-tuning de modèles de langage sur des corpus spécialisés liés aux CPS, aux fiches techniques et aux standards industriels. Une troisième piste concerne l'amélioration des mécanismes d'explicabilité, afin de produire des explications encore plus fidèles au raisonnement réel du système. D'autres prolongements concernent l'intégration de mécanismes d'apprentissage continu à partir du retour utilisateur, l'introduction d'une interaction multimodale et l'ajout de contraintes réglementaires ou normatives dans la logique de recommandation.

Dans une perspective plus large, l'évolution du CPRS peut être envisagée selon trois étapes. À court terme, il s'agit de consolider scientifiquement le cadre proposé, d'élargir les jeux de données et de poursuivre la validation expérimentale. À moyen terme, l'objectif est de spécialiser davantage le système pour certains domaines applicatifs, d'intégrer des ontologies ou graphes de connaissances et d'expérimenter le système avec des partenaires industriels. À plus long terme, le CPRS pourrait devenir un assistant déployable dans des environnements réels, intégré à des chaînes d'outils de conception ou à des écosystèmes de jumeaux numériques, capable d'apprendre en continu et de s'adapter à différents secteurs.

En définitive, cette thèse propose un cadre original, explicable et conversationnel pour la configuration des systèmes cyber–physiques. Elle montre que la combinaison d'embeddings sémantiques, de bases de connaissances structurées, de mécanismes de clarification, d'explicabilité et de génération en langage naturel permet de construire un système à la fois techniquement performant et humainement compréhensible. En réduisant la barrière d'expertise et en renforçant la transparence du processus de recommandation, le CPRS apporte une réponse concrète à plusieurs défis majeurs de la conception des CPS. Il constitue ainsi à la fois une contribution scientifique au domaine des systèmes de recommandation explicables et une base solide pour le développement de futurs assistants intelligents dédiés aux environnements techniques complexes.

Annexe A

Publications

A.1 Journals

- Choaib, M., Garouani, M., Bouneffa, M., Waldhoff, N., & Mohanna, Y. (2024). *IoT-AID : An Automated Decision Support Framework for IoT*. **SN Computer Science**, 5(4), 429.
- Choaib, M., Ahmad, A., & Bouneffa, M. (2025). *Towards an Explainable Cyber-Physical Recommender System*. **Lecture Notes in Business Information Processing**, Springer Nature. (Accepted waiting to be published)

A.2 Conferences

- Choaib, M., Garouani, M., Bouneffa, M., Waldhoff, N., Ahmad, A., & Mohanna, Y. (2023, April). *Automated Decision Support Framework for IoT : Towards a Cyber Physical Recommendation System*. In **ICEIS (1)** (pp. 365–373).
- Choaib, M., Garouani, M., Bouneffa, M., & Mohanna, Y. (2024, July). *IoT Sensor Selection in Cyber-Physical Systems : Leveraging Large Language Models as Recommender Systems*. In **2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)** (pp. 2516–2519). IEEE.
- Choaib M., Garouani M., Bouneffa M. and Ahmad A. (2025). IoT-AID : Leveraging XAI for Conversational Recommendations in Cyber-Physical Systems. In Proceedings of the 27th International Conference on Enterprise Information Systems - Volume 1 : ICEIS ; ISBN 978-989-758-749-8, SciTePress, pages 671-679. DOI : 10.5220/0013497100003929

A.3 Communications

- Choaib, M.; Garouani, M.; Bouneffa, M.; Waldhoff, N.; Ahmad, A.; & Mohanna, Y. *Towards a Cyber-Physical Recommendation System : Automated Decision Support Framework for IoT*. **IA-Optim**, 7–8 June 2023, Montreuil-sur-Mer. (Oral Presentation)
- Choaib, M.; Garouani, M.; Bouneffa, M.; Waldhoff, N.; Ahmad, A.; & Mohanna, Y. *Towards a Cyber-Physical Recommendation System : Automated Decision Support Framework for IoT*. **Journée Doctorants du Pôle MTE**, 20 June 2023, Calais. (Oral Presentation)
- Choaib, M.; Garouani, M.; Bouneffa, M.; Waldhoff, N.; Ahmad, A.; & Mohanna, Y. *Towards a Cyber-Physical Recommendation System : Automated Decision Support Framework for IoT*. **Journée du LISIC**, 11 July 2023, Calais. (Oral Presentation)
- Choaib, M.; Garouani, M.; Bouneffa, M.; Waldhoff, N.; Ahmad, A.; & Mohanna, Y. *Automated Decision Support Framework for IoT : Towards a Cyber Physical*. **Journée Doctorants MTE**, 18 June 2024, Dunkerque. (Oral Presentation)
- Choaib, M.; Garouani, M.; Bouneffa, M.; Waldhoff, N.; Ahmad, A.; & Mohanna, Y. *Automated Decision Support Framework for IoT : Towards a Cyber Physical*. **Journée Doctorants MTE**, 11 July 2024, Calais. (Oral Presentation)
- Choaib, M.; Ahmad, A.; Bouneffa, M.; Alaoui, C. N. *Automated Decision Support Framework for IoT : Towards a Cyber Physical*. **Journée Doctorants MTE**, 4–5 June 2025, Amiens. (Oral Presentation)

Annexe B

Annex B

Text classification, also referred to as text categorization, assigns predefined labels to text based on its content, playing a crucial role in various NLP applications such as sentiment analysis, spam detection, topic categorization, and document summarization. Traditionally, methods for text classification ranged from simple rule-based systems to more complex machine learning models (Sebastiani, 2002). However, recent advancements in deep learning and Transformer-based architectures have significantly enhanced performance and expanded the range of applications (Devlin et al., 2019; Yang et al., 2019). This machine learning subfield, akin to recognizing features of different flowers, involves training algorithms to detect patterns in words and phrases, enabling them to classify new, unlabeled texts. This technique is utilized in a variety of contexts, including filtering spam emails (Wu et al., 2020), analyzing social media sentiment for hate speech (Zampieri et al., 2019), and categorizing news articles and videos by topic (Zaheer et al., 2020), highlighting its versatile and practical applications.

Text classification involves various methods ranging from traditional machine learning approaches to advanced deep learning techniques.

- Traditional Machine Learning Methods :
  - — Naive Bayes : A probabilistic classifier based on Bayes' theorem, assuming independence between features. It is simple, fast, and effective for many text classification tasks but can struggle with highly correlated features.
  - — Support Vector Machines (SVM) :A discriminative classifier that finds a hyperplane to separate data points of different classes. Known for its effectiveness in high-dimensional spaces, SVMs provide strong performance for text classification tasks.
  - — Desicion Tree : A flowchart-like model where each node represents a feature and each branch represents a decision rule. While easy to

interpret, decision trees can overfit the training data without proper pruning.

- Deep Learning Methods
  - Feedforward Neural Networks (FNN) :Simple neural networks with one or more hidden layers. While foundational, they are less effective for text due to the lack of sequential data handling capabilities.
  - Convolutional Neural Networks (CNN) :Utilize convolutional layers to capture local patterns in text. They are particularly effective for sentence classification and sentiment analysis.
  - Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) : Capture sequential dependencies in text data, making them suitable for text classification where context is important. LSTMs, in particular, address the vanishing gradient problem of RNNs.
- Transformer-Based Models :
  - BERT (Bidirectional Encoder Representations from Transformers) :Uses transformers to capture context from both directions in text, setting a new state-of-the-art for many NLP tasks, including text classification
  - GPT (Generative Pre-trained Transformer) :A generative model that predicts the next word in a sequence, useful for both text generation and classification tasks. GPT-3, in particular, has shown remarkable capabilities.
  - XLNet : Combines the strengths of autoregressive and autoencoding models, outperforming BERT on various NLP benchmarks by capturing bidirectional contexts without masking.