IoT-AID: Leveraging XAI for Conversational Recommendations in Cyber-Physical Systems

Mohammad Choaib[1,2], Moncef Garouani[3], Mourad Bouneffa[1] and Adeel Ahmad[1]
[1] *Univ. Littoral Côte d'Opale, UR 4491, LISIC, Laboratoire d'Informatique Signal et Image de la Côte d'Opale, F-62100 Calais, France*
[2] *Lebanese University, Lebanon*
[3] *IRIT, UMR 5505 CNRS, Université Toulouse Capitole, Toulouse, France*

Keywords: Cyber Physical System, Conversational Recommender Systems, Machine Learning, Industry 4.0, Decision Support Systems, NLP, Large Language Models, Explainable AI.

Abstract: The rapid evolution of Industry 4.0 has introduced transformative technologies such as the Internet of Things (IoT), Artificial Intelligence (AI), and big data, facilitating real-time data collection, processing, and decision-making. At the heart of this revolution lies Cyber-Physical Systems (CPS), which integrate computational algorithms with physical components to create intelligent, resilient, and adaptive systems. However, CPS deployment remains complex due to the need for extensive domain expertise. This paper introduces IoT-AID, a novel Explainable AI (XAI)-driven Cyber-Physical Recommendation System (CPRS) that enhances transparency, trust, and efficiency in CPS design. IoT-AID integrates traditional machine learning models, deep learning architectures, and fine-tuned transformer-based models with XAI techniques to automate and improve CPS configuration. Our approach ensures that AI-driven recommendations are interpretable, thereby increasing adoption across industries.

1 INTRODUCTION

Industry 4.0 marks a pivotal era where the fusion of IoT, AI, and big data transforms industries, enabling real-time monitoring, predictive maintenance, and decentralized decision-making (Choaib et al., 2024). Cyber-Physical Systems (CPS) form the backbone of this transformation, integrating physical components with computational intelligence to optimize industrial processes. Despite their potential, CPS deployment faces key challenges. One of the major issues is the complexity of configuring such systems, as engineers must integrate multiple components, requiring deep domain expertise in hardware and software (Garouani et al., 2022a). Additionally, the availability of high-quality domain-specific datasets, remain a bottleneck for training AI models effectively (Whang et al., 2023). Many CPS recommendation systems also lack transparency, often functioning as black-box models, making it difficult for users to trust their outputs and interpret how recommendations are generated (Garouani et al., 2022c).

To address these challenges, IoT-AID has been developed as a novel solution that integrates Explainable AI (XAI) techniques with advanced AI models, ensuring transparent and user-centric CPS design recommendations (Moosavi et al., 2024). By implementing a multi-faceted approach combining machine learning, deep learning, and transformer-based models, IoT-AID enhances the accuracy and interpretability of recommendations while simplifying the configuration process for engineers and decision-makers.

Building effective CPS applications necessitates a deep understanding of user needs and the accurate identification of necessary physical components, such as sensors. While Large Language Models (LLMs) have revolutionized various aspects of Industry 4.0, their potential in smart manufacturing and Cyber-Physical Systems (CPS) remains largely unexplored (Choaib et al., 2024). LLMs, which are trained on extensive, generalized knowledge, often lack the specialized insights required to navigate the complex challenges of these domains (Zaheer et al., 2020). To bridge this gap, we propose fine-tuning LLMs with datasets specifically built for entity recognition, enhancing their ability to generate customized recom-

Choaib, M., Garouani, M., Bouneffa, M. and Ahmad, A.
IoT-AID: Leveraging XAI for Conversational Recommendations in Cyber-Physical Systems.
DOI: 10.5220/0013497100003929

In *Proceedings of the 27th International Conference on Enterprise Information Systems (ICEIS 2025) - Volume 1*, pages 671-679
ISBN: 978-989-758-749-8; ISSN: 2184-4992

mendations for CPS configuration. However, we also leveraged traditional machine learning models such as Decision Trees, Support Vector Machines (SVM), and Naive Bayes, as well as deep learning models like Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN). This multi-faced approach, combined with fine-tuning a pre-trained BERT model for text classification, ensured a comprehensive understanding of user needs and facilitated the recommendation of essential components. By integrating these techniques into the Cyber-Physical Recommender System (CPRS), we automated and enhanced the recommendation process, leading to more efficient CPS design and implementation in the realm of Industry 4.0.

The rest of the paper is organized as follows, a small background on text classification and its different models, in addition to the state of the art for cyber physical systems and utilizing XAI. Section 3 discusses the main components of the proposed system and we also discuss how these components collaborate to achieve the pursued goals, and how to utilize different models in this system, Finally in section 4 we will discuss the obtained results and what are the faced challenges and concludes the paper and outlines future perspectives in section 5.

2 BACKGROUND

2.1 Text Classification

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 (Mardiansyah and Surya, 2024), analyzing social media sentiment for hate speech (Zampieri et al., 2023), and categorizing news articles and videos by topic (Zaheer et al., 2020), highlighting its versatile and practical applications.

2.1.1 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 (McCallum and Nigam, 1998).
- **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 (Joachims, 1998).
- **Decision 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 (Quinlan, 1986).

2.1.2 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 (Goodfellow et al., 2016).
- **Convolutional Neural Networks (CNN):** Utilize convolutional layers to capture local patterns in text. They are particularly effective for sentence classification and sentiment analysis (Zhao and Wu, 2016; Garouani et al., 2023).
- **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 (Hochreiter and Schmidhuber, 1997).

2.1.3 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 (Devlin et al., 2019).
- **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 (Brown et al., 2020).
- **XLNet:** Combines the strengths of autoregressive and autoencoding models, outperforming BERT

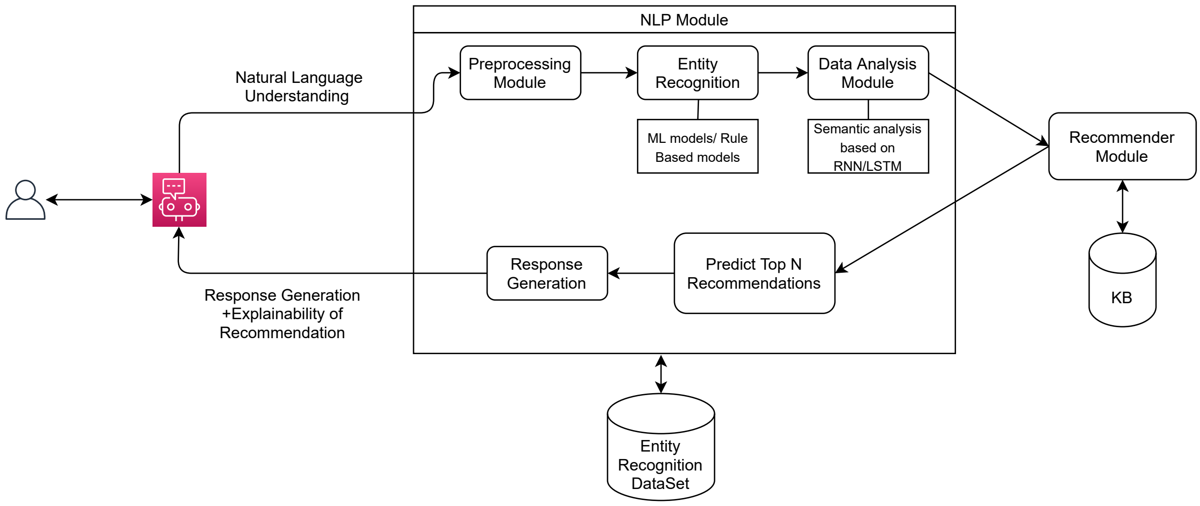

Figure 1: CPRS Architecture.

on various NLP benchmarks by capturing bidirectional contexts without masking (Yang et al., 2019).

2.2 Cyber-Physical Systems

Cyber-Physical Systems (CPS) integrate computational algorithms with physical elements to create interconnected systems that offer advanced functionalities and capabilities. CPSs leverage smart sensors, embedded systems, cloud computing, data storage, and artificial intelligence techniques to transform industries, paving the way for smarter factories that are at the forefront of the fourth industrial revolution. These advancements enable predictive maintenance, real-time monitoring, and self-optimization(Garouani et al., 2022b). As critical components of this revolution, CPSs contribute to the development of intelligent, resilient, and adaptive machines, facilitating their widespread adoption across various sectors and applications. Despite their potential, configuring CPS solutions to meet specific needs remains challenging for researchers and engineers due to knowledge gaps. Automated assistance can help by enabling engineers and researchers to efficiently develop, validate, and deploy CPS solutions, thereby enhancing service quality, productivity, and reducing dependence on human expertise .

The proposed Cyber-Physical Recommender System (CPRS), illustrated in Figure 1, comprises a Knowledge Base (KB) and a recommendation engine. The KB contains comprehensive information about sensors, including their specifications and application domains. Users interact with the system through a chatbot, providing details such as product information and budget constraints. To address new challenges, the system employs Natural Language Processing (NLP) to extract keywords and match them with data stored in the KB . Recommendations are refined based on rankings and user feedback. Furthermore, a Meta-knowledge base stores knowledge acquired during offline training and uses ontologies to enhance information retrieval and query understanding (Choaib et al., 2024).

To improve decision-making, the CPRS incorporates LLMs such as BERT and some other deep learning models such as CNN and LSTM. Although LLMs are proficient in general language understanding, they may lack the domain-specific knowledge needed for cyber-physical systems (Devlin et al., 2019). To mitigate this, fine-tuning is performed, training LLMs on domain-specific datasets to understand the terminology and context of cyber-physical systems (Choaib et al., 2024). The CPRS uses LLMs to enhance Natural Language Understanding (NLU), extract entities from user inputs, generate recommendations, and continuously optimize based on user feedback. Evaluation metrics include the accuracy of entity extraction and the relevance of recommendations, ultimately facilitating the development of smart cyber-physical systems.

2.3 XAI

Explainable AI (XAI) is crucial in addressing the lack of transparency in AI-driven CPS recommendation systems. XAI aims to make AI decisions comprehensible to humans by providing clear insights into how models generate predictions (Ribeiro et al., 2016b). In the IoT-AID system, post-hoc interpretability methods such as Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP) have been incorporated to enhance transparency (Ribeiro et al., 2016a; Garouani and Bouneffa, 2023). LIME generates localized explanations by approximating decision boundaries around individual predictions, enabling users to understand the key factors influencing each recommendation. On the other hand, SHAP provides a global interpretability framework by calculating the contribution of each input feature across multiple predictions. These techniques empower users to validate the CPS recommendations, fostering greater trust and usability.

3 A NOVEL CYBER PHYSICAL RECOMMENDER SYSTEM

3.1 System Architecture

IoT-AID is designed as a modular system comprising several key components that work in tandem to streamline CPS design and configuration. The first component, the Preprocessing Module, is responsible for cleaning and tokenizing text input using natural language processing tools such as spaCy and NLTK. This ensures that the input is standardized and free from noise. Next, the NLP module utilizes fine-tuned BERT embeddings to extract relevant entities and contextual information from the input. The extracted entities are then processed in the data analysis module, which leverages a combination of traditional machine learning models, deep learning architectures, and transformer-based models to analyze user requirements comprehensively. Based on this analysis, the Recommendation Engine generates ranked CPS configurations tailored to the user's specific application needs. Finally, the Explainability Layer integrates XAI techniques such as LIME and SHAP to provide transparent insights into the recommendations, allowing users to understand the reasoning behind each decision.

3.2 Dataset Generation and Feature Extraction

Data Acquisition Challenges: In our pursuit of constructing a reliable Cyber-Physical Recommender System (CPRS) as shown in figure 2, 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.

Dataset Composition: 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 Django project. This function leverages the OpenAI library and an OpenAI API key to make requests to the ChatGPT API, allowing us to retrieve the data we required.

Data Generation Process: We initiated the data generation by passing a specific phrase as a parameter in the request, instructing ChatGPT to provide us with the needed data. After executing the request, we successfully obtained several lines of data. However, we faced a limitation: ChatGPT could only provide limited lines per request, and we needed hundreds. To overcome this, we ran the request in a loop 100 of times, storing all the data in JSON format within a JSON file.

Data Redundancy Issue: This process took several hours, and we eventually collected all required JSON elements. Yet, we encountered an issue with data redundancy—too many repeated lines. Despite instructing ChatGPT not to provide previously given applications, the data still contained duplicates.

Redundancy Resolution: To resolve this, we implemented a 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 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.

3.2.1 Entities Dataset

The entities are described by the attributes shown in figure 3 :

- 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.

3.3 Data Preprocessing

Data preprocessing is a critical step in preparing the text data for machine learning models. We utilized

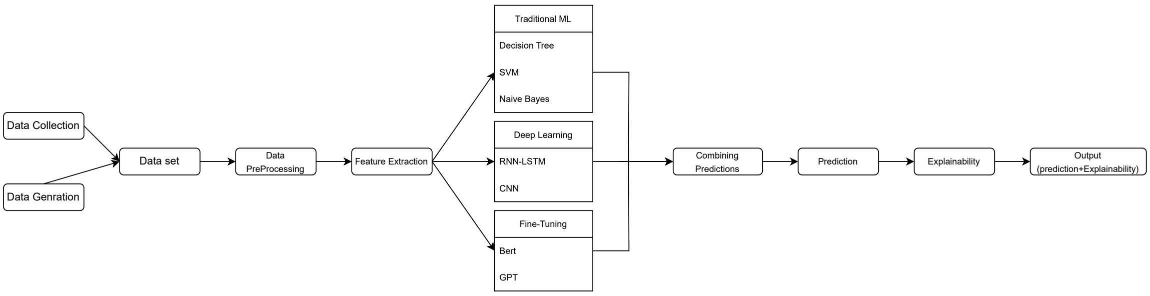

Figure 2: CPRS Pipeline.


```
{
  "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 3: Entities Dataset.

both NLTK and spaCy for this purpose. 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 individual words (tokens). This was followed by lemmatization, which reduced words to their base forms, thus ensuring consistency (Garouani and Kharroubi, 2022). For instance, the raw text ”I need a system for health monitoring for heart rate and oxygen levels” was preprocessed using spaCy 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.

3.4 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 (Term Frequency-Inverse Document Frequency) Vectorization and BERT Embeddings. TF-IDF 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, which served as inputs to the various models.

3.5 Model Training

3.5.1 Machine Learning Models

- Decision Tree classifier (DT): The Decision Tree classifier is a simple yet powerful model that splits the data based on feature importance to make predictions. We trained a single decision tree on TF-IDF vectors using sklearn’s Decision Tree classifier. The model was capable of capturing complex decision boundaries but was prone to overfitting. Despite this, it provided a clear and interpretable decision-making process, which we visualized through a generated decision tree diagram.
- Support Vector Machine (SVM): The SVM model with a linear kernel is well-suited for text classification due to its robustness and ability to generalize well. We implemented an SVM classifier within a pipeline that included TF-IDF vectorization, using sklearn’s SVC with probability estimates. This model transformed the preprocessed descriptions into numerical vectors and then classified them. The SVM showed high accuracy on validation data and was less prone to overfitting compared to the Decision Tree.
- Naive Bayes: The Naive Bayes classifier, particularly the Multinomial Naive Bayes, is efficient and effective for text data. It makes simple assumptions about the data distribution but performs well. Using sklearn’s MultinomialNB, we trained the model on TF-IDF vectors. The NB model was fast to train and predict, and it achieved good performance on the text classification task.

3.5.2 Deep Learning Models

- Recurrent Neural Network (RNN): RNNs are designed to handle sequential data and are particularly effective for tasks involving time series or text sequences. We built an RNN using a sequential model with an Embedding layer, an LSTM layer, and a Dense output layer using TensorFlow. The model captured sequential dependencies in the text, allowing it to understand context over long sequences. We tokenized and padded the input sequences before feeding them into the RNN. This model achieved good accuracy on the validation data, demonstrating its ability to handle complex dependencies in the text.

- Convolutional Neural Network (CNN): CNNs, although traditionally used for image processing, are also effective for text classification by capturing local patterns. Our CNN model consisted of an Embedding layer, a Conv1D (convolutional) layer, a GlobalMaxPooling1D layer, and a Dense output layer. This architecture allowed the model to detect key phrases and patterns within the text. We tokenized and padded the sequences before feeding them into the CNN. The model showed high accuracy on the validation data, effectively capturing local features in the text.
- Fine-Tuning BERT:Fine-tuning a pre-trained BERT model involves adapting it to our specific text classification task. BERT is a transformer-based model that excels at understanding context in text. For our task, we used the pre-trained BERT model from the Hugging Face library and added a classification layer on top of it. This involved training the entire model, including the BERT layers, to adjust the pre-trained weights to better fit our dataset. We tokenized the input text using BERT's tokenizer, ensuring compatibility with the model's expectations. The tokenized inputs were then fed into the BERT model, followed by a dense layer for classification. Training was performed using a learning rate schedule and early stopping to prevent overfitting. The fine-tuning process leveraged the deep contextual embeddings of BERT, making it highly effective for our classification task.

3.6 Proposed Model for XAI in CPS

3.6.1 Role of XAI

Explainable AI (XAI) is fundamental to the IoT-AID system, ensuring that its recommendations are not only accurate but also comprehensible. The importance of XAI lies in its ability to make AI systems transparent by explaining the reasoning behind their outputs. For example, in a CPS design scenario, XAI can clarify why specific sensors or communication protocols were chosen for a given application (Garouani and Bouneffa, 2023). Such explanations enable engineers to validate the system's recommendations and align them with real-world constraints and requirements. Moreover, by building trust and fostering informed decision-making, XAI mitigates skepticism often associated with AI-driven systems.

3.6.2 Utilization of XAI

IoT-AID incorporates XAI techniques at various stages of its workflow. During preprocessing, Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP) help identify the most significant features in user inputs. LIME generates local approximations of the model's decision boundaries, allowing users to understand which factors contributed most to specific recommendations. SHAP, on the other hand, provides a global explanation by computing the average contribution of each input feature across multiple predictions. These techniques provide localized insights, making the system's behavior transparent for specific recommendations. Additionally, during entity recognition and recommendation generation, XAI techniques highlight the role of extracted features and contextual relationships in shaping system outputs. This layered approach to interpretability ensures that every stage of IoT-AID's process is comprehensible and user-centric.

3.6.3 Structure of the Model

The IoT-AID system integrates traditional machine learning and modern deep learning techniques, complemented by XAI tools. Traditional algorithms, such as Decision Trees and Support Vector Machines (SVMs), are employed for their inherent interpretability and simplicity. For more complex tasks, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are used to capture intricate relationships and temporal dependencies in the data. Fine-tuned transformer-based models, such as BERT, enable contextual understanding and domain-specific entity recognition. These models utilize structured CPS datasets and expert-generated synthetic data to improve recommendation accuracy and robustness. The architecture ensures that each layer processes data efficiently, applying XAI techniques to provide transparent insights at every stage.

3.6.4 Model Architecture and Workflow

IoT-AID's architecture consists of five key modules:

Preprocessing Module: This module cleans and tokenizes user inputs, removing noise and standardizing the data format. Tools like spaCy and NLTK are employed for efficient text preprocessing.

NLP Module: Using fine-tuned BERT embeddings, this module extracts entities and semantic context from user inputs, ensuring accurate interpretation of requirements.

Data Analysis Module: Outputs from various machine learning and deep learning models are integrated in this module to evaluate user requirements comprehensively.

Recommendation Engine: Based on the processed data, this engine generates ranked CPS configurations tailored to the user's application.

Explainability Layer: XAI techniques are applied here to provide transparent explanations for recommendations, fostering user trust and enabling informed decision-making. Figure 4 shows a snippet of how we used Lime highlighting key words that influence the model's decision, with important terms in red and less important terms in green.

I need a system for health monitoring for heart rate and oxygen level.	Important
Can you provide some general information about your services?	Not important

Figure 4: Lime Explanation.

To give an example of the work flow, the user begins by providing a query, such as "I need a system for health monitoring for heart rate and oxygen levels." This input is cleaned and tokenized in the Preprocessing Module before being passed to the NLP Module, where relevant entities like "health monitoring" and "heart rate" are extracted. These entities are analyzed in the Data Analysis Module using a combination of machine learning and deep learning models, which collectively generate a list of recommended configurations. The Recommendation Engine ranks these suggestions, while the Explainability Layer provides detailed insights into the reasoning behind each recommendation. The models use structured CPS datasets combined with expert-generated synthetic data to improve recommendation accuracy and robustness. The architecture also supports iterative feedback loops, allowing users to refine their requirements and improve the model's recommendations over time.

4 RESULTS AND DISCUSSION

Preliminary evaluations of IoT-AID have demonstrated its effectiveness in addressing key challenges in CPS design. The system achieved high accuracy in entity recognition and recommendation relevance, thanks to its integration of advanced language models and machine learning techniques. The incorporation of XAI methods significantly enhanced user trust and satisfaction, providing clear and concise explanations for recommendations. However, the system's performance remains constrained by the quality and scope of the training dataset, highlighting the need for further data expansion.

Also some challenges with data scarcity arise:

4.1 Data Collection Issues

CPS applications require domain-specific datasets, which are often unavailable or incomplete. IoT-AID addresses this challenge by leveraging web scraping techniques to collect data from publicly available sources. Additionally, generative AI models such as OpenAI's GPT are used to synthesize supplementary data, bridging gaps in the dataset.

4.2 Data Preprocessing and Deduplication

The data collection process is often plagued by redundancy, with duplicate entries diminishing the quality of the dataset. To counter this, IoT-AID employs automated scripts for deduplication and enriches the dataset with relevant attributes, such as sensor types and application domains. These steps ensure that the data is both comprehensive and unique.

4.3 Impact on Results

While these methods significantly enhance dataset quality, data scarcity remains a challenge in niche CPS domains. The system's performance is particularly affected in scenarios requiring highly specialized knowledge. Future work will explore collaborations with industry partners and the use of advanced data augmentation techniques to further expand and diversify the dataset.

4.4 Evaluating CPS Performance and Decision-Making with XAI

Evaluating CPS performance requires a problem-specific approach, as different applications necessitate distinct success metrics. A universal evaluation criterion is impractical, so future work will focus on defining tailored performance indicators or relying on domain expert validation. This approach ensures that reasoning strategies—whether pattern-based, logic-driven, or hybrid—are aligned with the intended function of the CPS.

In addition, it is important to recognize that explainability in AI-driven decision-making is not equivalent to correctness. XAI in IoT-AID is designed to enhance user understanding by providing interpretable recommendations rather than guaranteeing optimal solutions. By offering clear justifications for each recommendation, users can make more informed decisions, either accepting or rejecting suggestions based on their own expertise and contextual needs.

4.5 Addressing Bias Reinforcement in Recommendations

A significant challenge in AI-driven recommendations is the risk of bias reinforcement. If an XAI system prioritizes user trust and alignment over objective accuracy, it may reinforce predictable but suboptimal decisions. Future work will focus on developing mechanisms to detect and mitigate biases, such as adversarial testing, diverse training datasets, and alternative recommendation strategies. These measures aim to ensure that recommendations remain explainable while also being objectively beneficial for CPS configurations.

5 CONCLUSION AND FUTURE WORK

IoT-AID represents a significant advancement in the field of CPS design and implementation. By integrating XAI techniques into a comprehensive recommendation system, IoT-AID addresses critical challenges related to complexity, data scarcity, and transparency. Its ability to provide interpretable, accurate, and user-centric recommendations has the potential to democratize CPS adoption and accelerate the realization of Industry 4.0 objectives. Future iterations will refine the system's capabilities, ensuring its applicability across diverse industries and domains.

Future efforts will focus on several key areas to enhance IoT-AID's capabilities. First, expanding the dataset by collaborating with industry partners and employing advanced data augmentation techniques will improve model accuracy and generalizability. Second, the exploration of hybrid XAI techniques that combine intrinsic and post-hoc interpretability methods will further enhance transparency. Third, fine-tuning advanced transformer models, such as GPT variants, for domain-specific applications will enable more nuanced and accurate recommendations. Finally, real-world deployments of IoT-AID in industrial settings will provide valuable insights into its scalability, adaptability, and overall impact.

REFERENCES

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Choaib, M., Garouani, M., Bouneffa, M., Waldhoff, N., and Mohanna, Y. (2024). Iot-aid: An automated decision support framework for iot. *SN Computer Science*, 5(4):429.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Garouani, M., Ahmad, A., Bouneffa, M., and Hamlich, M. (2022a). Amlbid: An auto-explained automated machine learning tool for big industrial data. *SoftwareX*, 17:100919.

Garouani, M., Ahmad, A., Bouneffa, M., and Hamlich, M. (2023). Autoencoder-knn meta-model based data characterization approach for an automated selection of ai algorithms. *Journal of Big Data*, 10(1).

Garouani, M., Ahmad, A., Bouneffa, M., Hamlich, M., Bourguin, G., and Lewandowski, A. (2022b). Towards big industrial data mining through explainable automated machine learning. *The International Journal of Advanced Manufacturing Technology*, 120(1):1169–1188.

Garouani, M. and Bouneffa, M. (2023). Unlocking the black box: Towards interactive explainable automated machine learning. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 458–469. Springer.

Garouani, M., Hamlich, M., Ahmad, A., Bouneffa, M., Bourguin, G., and Lewandowski, A. (2022c). *Toward an Automatic Assistance Framework for the Selection and Configuration of Machine Learning Based Data Analytics Solutions in Industry 4.0*, page 3–15. Springer International Publishing.

Garouani, M. and Kharroubi, J. (2022). *Towards a New Lexicon-Based Features Vector for Sentiment Analysis: Application to Moroccan Arabic Tweets*, page 67–76. Springer International Publishing.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *European Conference on Machine Learning*, pages 137–142.

Mardiansyah, K. and Surya, W. (2024). Comparative analysis of chatgpt-4 and google gemini for spam detection on the spamassassin public mail corpus.

McCallum, A. and Nigam, K. (1998). A comparison of event models for naive bayes text classification. *AAAI workshop on learning for text categorization*, 752:41–48.

Moosavi, S., Farajzadeh-Zanjani, M., Razavi-Far, R., Palade, V., and Saif, M. (2024). Explainable ai in manufacturing and industrial cyber–physical systems: a survey. *Electronics*, 13(17):3497.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016a). Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016b). Why should i trust you?: Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Whang, S. E., Roh, Y., Song, H., and Lee, J.-G. (2023). Data collection and quality challenges in deep learning: A data-centric ai perspective. *The VLDB Journal*, 32(4):791–813.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. (2020). Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Zampieri, M., Rosenthal, S., Nakov, P., Dmonte, A., and Ranasinghe, T. (2023). Offenseval 2023: Offensive language identification in the age of large language models. *Natural Language Engineering*, 29(6):1416–1435.

Zhao, Z. and Wu, Y. (2016). Attention-based convolutional neural networks for sentence classification. In *Interspeech*, volume 8, pages 705–709.