MEPI. Deep Learning-based System for Maintenance Event Prediction in Industry 4.0

José Ángel Noguera-Arnaldos[1], Juan Antonio Sánchez-Gil[1], Tomás Bernal-Beltrán[2], Ronghao Pan[2], José Antonio García-Díaz[2] and Rafael Valencia-García[2]

[1]*NEORADIX SOLUTIONS S.L., Calle Paraguay, bloque 11 C., Polígono Oeste, 30002 Murcia, España*
[2]*Facultad de Informática, Universidad de Murcia, Campus de Espinardo, 30100 Murcia, España*

Abstract

Predictive maintenance requires operational data to be collected, processed and analysed in order to predict failures and optimise asset management. However, the diversity of industrial environments and the variety of data sources make it challenging to implement scalable, flexible solutions. In this project, we present MEPI (Maintenance Environment for Predictive Intelligence), a predictive maintenance dashboard designed to monitor and manage industrial assets using deep learning, survival analysis, natural language processing, and computer vision techniques. The platform enables users to configure assets using meta-models, collect real-time data from sensors and devices, and generate predictive models on demand. Maintenance tasks can be scheduled manually or automatically by integrating textual and visual information. All of this functionality is accessible via a web-based dashboard consisting of configurable KPIs and a REST API.

Keywords

Predictive Maintenance, Industrial IoT, Computer Vision, Natural Language Processing

1. Introduction

Predictive maintenance (PdM) is a key Industry 4.0 strategy, enabling failures to be anticipated and asset management to be optimised through the analysis of operational data. In this context, deep learning techniques have demonstrated significant potential, offering accurate and scalable solutions for industrial applications.

Although deep learning models have been shown to be effective at predicting equipment failures, most existing platforms focus solely on numerical time-series data, overlooking the potential of unstructured sources such as technician reports or maintenance logs. Recent studies have demonstrated the effectiveness of Natural Language Processing (NLP) techniques in extracting valuable insights from such textual data to improve maintenance strategies [1]. However, there is a lack of practical, integrated solutions that combine predictive modeling with NLP to leverage both sensor data and human-generated documentation. This gap limits the ability to develop fully informed maintenance strategies that utilize the full range of available data.

This paper presents the MEPI project: a novel PdM platform designed within the context of Industry 4.0. The proposed system uses a combination of deep learning models, survival analysis techniques, NLP and computer vision to provide a comprehensive solution for monitoring, analysing and predicting maintenance needs in diverse industrial settings. The TECNOMOD research group at the University of Murcia is developing this work in collaboration with Neoradix Solutions S.L., as part of the Spanish National Call for Public-Private Collaborative R&D Projects 2021 (reference CPP2021-008465).

The platform's architecture is built around a set of configurable meta-models that allow flexible adaptation to indoor and outdoor environments (see 3.1), incorporating data from Internet of Things

SEPLN 2025: 41st International Conference of the Spanish Society for Natural Language Processing, Zaragoza, Spain, 23-26 September 2025.
joseangel.noguera@neoradixsolutions.com (J. : Noguera-Arnaldos); juanantonio.sanchezg@neoradixsolutions.com (J. A. Sánchez-Gil); tomas.bernalb@um.es (T. Bernal-Beltrán); ronghao.pan@um.es (R. Pan); joseantonio.garcia8@um.es (J. A. García-Díaz); valencia@um.es (R. Valencia-García)
0009-0006-6971-1435 (T. Bernal-Beltrán); 0009-0008-7317-7145 (R. Pan); 0000-0002-3651-2660 (J. A. García-Díaz); 0000-0003-2457-1791 (R. Valencia-García)

(IoT) sensors, engineering reports, images and Global Positioning System (GPS) signals that can be applied to different domains such as farming [2], health [3]. These data sources are automatically processed to train customised predictive models (see 3.2) that estimate the Remaining Useful Life (RUL) of assets and trigger maintenance alerts accordingly. In addition to, the platform includes intelligent task scheduling mechanisms (see 3.3) and a dashboard (see 3.4) composed of configurable Key Performance Indicators (KPIs), allowing human operators to interact with the system, monitor tasks and analyse the status of the assets in real time. The deployment is cloud-ready and follows a microservices architecture, facilitating scalability and integration into real-world production environments.

The platform is currently in its final stages of development and is being validated through two case studies in the manufacturing and agriculture sectors. The aim of these pilots is to assess the system's effectiveness in improving operational efficiency and promoting more sustainable maintenance practices by reducing unnecessary interventions and optimising the use of resources.

2. Background information

The development of PdM systems in Industry 4.0 is based on the convergence of several technological advances. Deep learning techniques such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs) and survival models such as the Cox Proportional Hazards Model or Random Survival Forests have shown strong potential to predict equipment failures based on time series data and historical records [4].

In parallel, the growing adoption of Industrial Internet of Things (IIoT) devices enables real-time monitoring of assets through distributed sensors that collect environmental and operational data such as temperature, humidity, pressure and usage patterns. This sensor data is essential for feeding predictive models and triggering timely interventions [5].

Several commercial and research-oriented platforms have addressed PdM using AI techniques. Cloud-based services like Amazon Lookout for Equipment analyse industrial sensor data to detect anomalies and forecast failures at scale. IBM Maximo Predict [6] integrates predictive capabilities within broader asset management systems, offering insights into asset health and maintenance planning. Benchmark initiatives such as those promoted by the PHM Society, particularly using the C-MAPSS dataset [7], have become standard references for evaluating RUL models.

While deep learning and sensor-based approaches have dominated the PdM landscape, the application of NLP in this domain is gaining attention. Maintenance logs, technician reports and work orders contain valuable unstructured information that can improve failure diagnosis and maintenance planning when combined with time series data [8]. Recent efforts, such as the MaintIE dataset [9], explore the use of NLP to extract actionable insights from maintenance reports, highlighting the potential of integrating textual data into predictive workflows. Other works [1] explore techniques like Named Entity Recognition (NER), relation extraction, and topic modeling to identify fault patterns and contextual factors from written records. However, compared to sensor-based modeling, NLP applications in industrial maintenance remain relatively unexplored, and comprehensive frameworks or standardized benchmarks are still lacking.

3. System architecture

Figure 1 shows the overall architecture of the system. The platform is organised into four main modules. The first module is responsible for configuration and data acquisition (see 3.1), allowing companies to define metamodels and collect sensor data from assets and industrial environments, including geolocation and multimedia inputs. The second module focuses on predictive analytics (see 3.2), where time series forecasting models and demand-driven survival analysis are trained using historical and contextual data. The third module focuses on intelligent task scheduling (see 3.3), which uses NLP and employee activity tracking mechanisms to effectively manage maintenance activities. The fourth module is the dashboard (see 3.4), which provides a web-based interface for users to monitor KPIs, configure alerts, visualise asset status and manage maintenance workflows. The following subsections describe these modules in more detail.

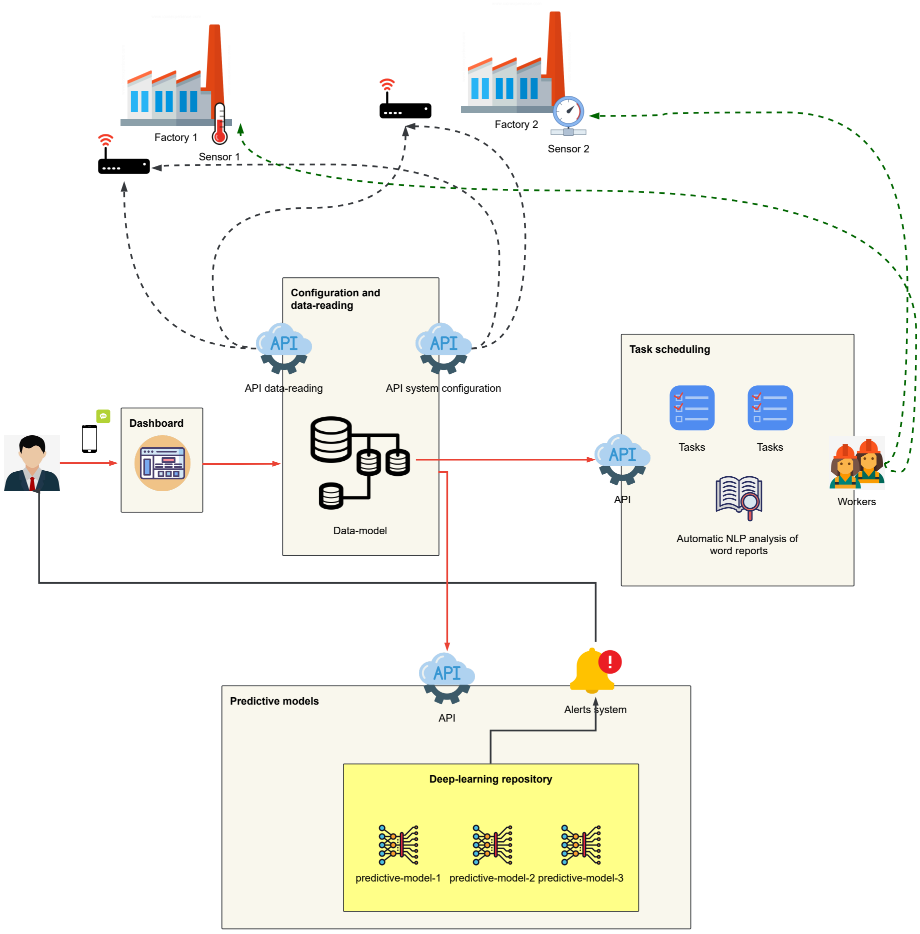

Figure 1: Overview of the modules that conform the MEPI system architecture

The platform will be offered on a Software-as-a-Service (SaaS) model, providing basic functionality free of charge with certain usage restrictions. For example, standard users will be able to configure a limited number of assets and build predictive models with predefined settings. Advanced features such as full dashboard customisation, advanced historical data access or integration with external enterprise systems are reserved for premium users.

All services and software components are deployed using Docker containers. This strategy enables horizontal scalability and flexible orchestration of the platform according to the performance requirements and deployment constraints of each industrial customer.

3.1. Configuration and Data Reading Module

The first core component of the platform is the configuration and data collection module, which is based on a metamodel-driven approach. A dedicated API allows each provider, also known as tenant, to define its own configuration by specifying different types of assets, associated attributes and the events to predict. This flexibility enables the system to adapt to a wide range of industrial scenarios by modeling the specific characteristics of each environment, such as operational context, location or asset category.

Once configured, the Data Reader module collects real-time information from multiple sources, including IoT sensors, mobile devices with GPS, and multimedia inputs such as images, video streams, maintenance report or work orders. All incoming data is structured according to the defined metamodels and transferred to the backend via secure REST APIs. In addition, a Historical Data API is available for querying stored information, supporting both model training and retrospective analysis. This modular and extensible architecture ensures seamless integration with existing monitoring systems and industry protocols.

In terms of data preprocessing, the numerical data are subjected to missing value imputation using the average of the two most recent sliding windows. To enable merging with numerical data, multimedia data in textual form is converted into a fixed-length array using one of the following approaches: (1) obtaining the document embedding for the entire text; (2) splitting the text into chunks, obtaining an embedding for each chunk, and concatenating them; or (3) applying a binary representation based on a configurable bag-of-words approach, where each element indicates the presence (1) or absence (0) of predefined terms in the text.

3.2. Predictive Models Generator

This module is responsible for generating PdM models tailored to each asset. The system monitors the flow of event data using scheduled CRON jobs and automatically triggers training processes when sufficient historical and contextual information is available. Model training is fully automated and orchestrated through configurable pipelines that run periodically.

To carry out this training, the data received from the Data Reader module is transformed into a regression problem. This is necessary because the information on the events to be predicted is initially presented as a binary signal, which makes it necessary to transform it into a numerical interpretation that facilitates the training process. Therefore, this transformation is considered a hyperparameter of the training process. The system takes into account various numerical approximations, such as linear, exponential, logistic and logarithmic.

The training pipeline performs multiple train-test splits and explores different configurations, treating the model architecture itself (e.g. recurrent networks, convolutional networks or survival analysis models) as a tunable parameter. This allows the system to identify the most appropriate prediction strategy for each use case. The evaluated models includes Facebook Prophet, Random Forest, Support Vector Regression, Multi-layer Perceptron and Convolutional Neural Networks. Furthermore, when using time series models, such as Facebook Prophet, the training process incorporates as a hyperparameter different approaches to the number of predictor variables used in the prediction of events. On the one hand, the univariate approach, which uses a single input variable and, on the another hand, the multivariate approach, which integrates multiple input variables to perform the prediction.

On the other hand, when using regression models, two strategies for the construction of the training dataset are evaluated. The first one consists of prediction without incorporating delays (no lags), while the second one integrates lags, using previous values to predict the current value. In addition, the training process adjusts the size of the time window (number of lags) considered, making it, in turn, a hyperparameter of the training process.

Trained models are stored in a central versioned repository, enabling comparison, retraining or rollback as required. The orchestration of the entire machine learning lifecycle follows best practices inspired by platforms such as MLflow [10], ensuring reproducibility, traceability and efficient deployment. In addition, an alerting component generates alerts based on model output when failure probabilities exceed configured thresholds, and an API is provided to expose predictions and model metadata for integration with external applications or decision support tools.

The table 1 shows the results obtained in terms of MSE and RMSE for the different regression models evaluated by the system during the training process. In particular, the results of each model are detailed for each possible event transformation, without taking into account the lags in the prediction. As can be seen, the Random Forest model with the exponential transformation of the events is the one with the best performance, achieving both a lower MSE and a lower RMSE than the other models. On the other hand, the CNN and MLP based models perform significantly worse. This is because, given the large amount of data for training, relatively simple architectures were chosen for these models to allow reasonable training times, which limited their ability to generalize effectively during the training process.

Table 1
MSE and RMSE of different regression models across different transformations without incorporating delays (no lags).

Model	Event transformation	Prediction Approach	MSE	RMSE
RF	Linear	no lags	0.708	0.842
	Exponential		**0.701**	**0.837**
	Logistic		0.924	0.961
	Logarithmic		0.934	0.967
SVR	Linear	no lags	167.890	12.957
	Exponential	no lags	167.287	12.934
	Logistic	no lags	275.478	16.598
	Logarithmic	no lags	165.533	12.866
CNN	Linear	no lags	2953.117	54.343
	Exponential	no lags	2479.461	49.794
	Logistic	no lags	3708.795	60.900
	Logarithmic	no lags	1735.816	41.663
MLP	Linear	no lags	2953.117	54.342
	Exponential	no lags	2479.461	49.794
	Logistic	no lags	3708.795	60.900
	Logarithmic	no lags	1735.816	41.663

3.3. Task Scheduling Module

The Task Scheduling module co-ordinates the execution of maintenance activities based on the output of the predictive models. It supports both manual and automated scheduling, taking into account asset criticality, recent maintenance reports, technician availability, geolocation and time-to-failure estimates.

A key component of this module is the NLP subsystem, which analyses textual maintenance reports, logs and work orders generated by technicians. Based on the Stanza NLP library [11], this subsystem provides a robust pipeline for NER. It is currently used to extract key entities, such as technician names, task types and component references, with a particular focus on identifying the personnel involved in each intervention. Furthermore, we are training domain-specific models using Transformer encoder-only architectures, which have been fine-tuned using a translated version of the MaintIE dataset [9]. This dataset provides fine-grained annotations for maintenance-related texts. These models aim to improve the recognition of specialised entities in industrial contexts. The extracted entities are integrated into the scheduling process to support technician assignment based on task history and detected fault types. Future iterations will incorporate semantic enrichment using ontologies to improve entity disambiguation and task recommendation.

The NLP pipeline automatically classifies reports to estimate their priority level as critical, moderate or low. This is achieved by combining MarIA sentence embeddings [12] from the reports with historical asset data. Furthermore, the system uses relation extraction techniques based on Stanza's dependency parsing to link relevant entities within each document. These relationships support the creation of structured records that feed the task assignment algorithm, contributing to a more informed, context-aware planning process. Future iterations will incorporate semantic enrichment using ontologies to improve disambiguation and expand recommendation capabilities.

Based on the insights extracted, the system can recommend the most suitable technician for a given task by taking into account their expertise, proximity and availability. Additionally, a GPS-based tracking subsystem enables real-time supervision of interventions, and embedded computer vision tools provide visual diagnostics and augmented reality overlays to enhance field support.

To improve task scheduling, computer vision techniques are employed. Augmented reality methods are used to assist workers during maintenance activities. For instance, a web service has been developed that enables workers to scan QR codes on machines using mobile devices to access critical information. This enables them to view the machine's maintenance manual, failure history and maintenance records. They can also view predictions generated by trained models that estimate the likelihood of the machine needing maintenance.

3.4. Dashboard

The dashboard module offers a centralised interface for PdM, managing maintenance tasks and visualising predictive insights. It is highly configurable, enabling users to define KPIs relevant to their operational context, such as estimated RUL, alert frequency, asset health or intervention history.

Through the interface, end users can access real-time alerts, track scheduled and ongoing maintenance tasks, and interact with data collected from the field. The dashboard includes a Kanban-style task manager, interactive charts, geolocation maps and sensor data visualisation tools. It also supports role-based access, providing different views and functionality for technicians, supervisors or administrators.

This module serves as the primary access point for end users and is closely integrated with all the platform's other components, providing a seamless and comprehensive overview of the maintenance ecosystem.

4. Further work

The current version of the platform incorporates essential modules for asset configuration, data acquisition, model training, task scheduling, and dashboard interaction. Ongoing work includes a pilot deployment in water treatment plants, with a focus on the predictive cleaning of filtration systems using sensor data.

Future developments will enhance the model training (see Section 3.2) pipeline with automatic hyper-parameter tuning and extend the NLP capabilities to process technician reports during validation phases. First, we plan to explore the integration of emotion detection into the analysis of maintenance reports. Technicians often express implicit signals of urgency, stress or dissatisfaction in their written feedback, which could provide additional value for task prioritisation and risk assessment [13]. This will support more accurate fault detection and informed task planning based on unstructured textual data. Second, we will investigate the use of multi-task learning (MTL) strategies to train multiple predictive models at once. MTL has shown promise in improving generalization and learning efficiency by sharing representations across related tasks [14]. In our context, we will explore MTL approaches that focus on predicting multiple maintenance events for the same asset. For example, a single model could be trained to estimate different events by exploiting common patterns in the asset's operational behaviour and usage history. This approach could improve efficiency compared to training separate models for each type of event, while also capturing interdependencies between maintenance actions.

Acknowledgments

This work is being developed by the TECNOMOD research group at the University of Murcia, in collaboration with Neoradix Solutions S.L., within the framework of the Spanish National Call for Public-Private Collaborative R&D Projects 2021 (reference CPP2021-008465) funded by MICIU/AEI /10.13039/501100011033 and the European Union NextGenerationEU/PRTR.

Declaration on Generative AI

During the preparation of this work, the authors used DeepL for grammatical and spelling correction. After using this tool, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] K. Zhong, T. Jackson, A. West, G. Cosma, Natural language processing approaches in industrial maintenance: A systematic literature review, Procedia Computer Science 232 (2024) 2082–2097.

[2] L. O. Colombo-Mendoza, M. A. Paredes-Valverde, M. d. P. Salas-Zárate, R. Valencia-García, Internet of things-driven data mining for smart crop production prediction in the peasant farming domain, Applied Sciences 12 (2022) 1940.

[3] M. Uppal, D. Gupta, S. Juneja, G. Dhiman, S. Kautish, Cloud-based fault prediction using IoT in office automation for improvisation of health of employees, Journal of Healthcare Engineering 2021 (2021) 8106467.

[4] S. Wiegrebe, P. Kopper, R. Sonabend, B. Bischl, A. Bender, Deep learning for survival analysis: a review, Artificial Intelligence Review 57 (2024) 65.

[5] Y. Hu, Q. Jia, Y. Yao, Y. Lee, M. Lee, C. Wang, X. Zhou, R. Xie, F. R. Yu, Industrial internet of things intelligence empowering smart manufacturing: A literature review, IEEE Internet of Things Journal 11 (2024) 19143–19167.

[6] IBM Corporation, Ibm maximo predict documentation, 2025. URL: https://www.ibm.com/docs/es/masv-and-l/cd?topic=applications-maximo-predict.

[7] M. Artelt, M. Weiß, D. Dittler, Y. Goersch, N. Jazdi, M. Weyrich, Hybrid approaches and datasets for remaining useful life prediction: A review, Procedia CIRP 130 (2024) 294–300.

[8] J. Garcia, L. Rios-Colque, A. Peña, L. Rojas, Condition Monitoring and Predictive Maintenance in Industrial Equipment: An NLP-Assisted Review of Signal Processing, Hybrid Models, and Implementation Challenges, Applied Sciences 15 (2025) 5465.

[9] T. K. Bikaun, T. French, M. Stewart, W. Liu, M. Hodkiewicz, Maintie: A fine-grained annotation schema and benchmark for information extraction from maintenance short texts., in: 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, Association for Computational Linguistics (ACL), 2024, pp. 10939–10951.

[10] V. Spadari, F. Cerasuolo, G. Bovenzi, A. Pescapè, An MLOps Framework for Explainable Network Intrusion Detection with MLflow, in: 2024 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2024, pp. 1–6.

[11] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C. D. Manning, Stanza: A python natural language processing toolkit for many human languages, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2020, pp. 101–108.

[12] A. Gutiérrez-Fandiño, J. Armengol-Estapé, M. Pàmies, J. Llop-Palao, J. Silveira-Ocampo, C. P. Carrino, C. Armentano-Oller, C. Rodriguez-Penagos, A. Gonzalez-Agirre, M. Villegas, MarIA: Spanish Language Models, Procesamiento del Lenguaje Natural 68 (2022) 39–60.

[13] A. Salmerón-Ríos, J. A. García-Díaz, R. Pan, R. Valencia-García, Fine grain emotion analysis in Spanish using linguistic features and transformers, PeerJ Computer Science 10 (2024) e1992.

[14] Y. Zhang, Q. Yang, A survey on multi-task learning, IEEE transactions on knowledge and data engineering 34 (2021) 5586–5609.