Federated Learning at the Edge for Wind Turbine Predictive Maintenance

Charis Michailidis*, Alexandros Kalafatelis†, Georgios Alexandridis‡, Andreas Oikonomakis§, Achileas Economopoulos§, Andrea Carolina Fontalvo Echavez¶, Daniel Iglesias Canelo ¶, Michail-Alexandros Kourtis§ and Panagiotis Trakadas†
*School of Applied Mathematical & Physical Sciences
National Technical University of Athens, Zografou, 157 80, Greece
Email: mihailidis.charis@gmail.com
†Ports Management & Shipping Department
School of Science, National & Kapodistrian University of Athens, Psachna, 344 00, Greece
Email: {alexkalafat, ptrakadas}@pms.uoa.gr
‡Digital Industry Technologies Department
School of Science, National & Kapodistrian University of Athens, Psachna, 344 00, Greece
Email: gealexandri@uoa.gr
§Institute of Informatics and Telecommunications
National Centre for Scientific Research "DEMOKRITOS" (NCSRD), Aghia Paraskevi, 153 10, Greece
Email: {a.oikonomakis, aeconomopoulos, akis.kourtis}@iit.demokritos.gr
¶Capgemini, Spain
Email: {andreacarolina.fontalvoechavez, daniel.iglesias-canelo}@capgemini.com

***Abstract*—Wind energy plays a pivotal role in the global shift toward sustainable energy systems. However, the maintenance of wind turbines remains a significant challenge due to their distributed nature, harsh environmental exposure, and the high cost of unplanned downtime. In this work, a novel architecture for predictive maintenance of wind turbines based on continuous acoustic monitoring is presented, based upon OASEES—a decentralized, intelligent, and programmable edge framework that spans the full computing continuum. The proposed system leverages low-cost recording equipment to capture turbine-generated sound data, which are processed locally at the edge using Federated Learning, thus preserving data privacy and reducing communication overhead. A pre-trained deep learning model based on wav2vec is fine-tuned to classify turbine operational states, using labeled acoustic datasets. The effectiveness of the architecture, which, to the best of the authors' knowledge, is among the first to utilize the said distributed learning paradigm for acoustic-based wind turbine predictive maintenance, is validated in a proof-of-concept experimental setting using a publicly available relevant dataset, where both centralized and federated training methods are evaluated. The results demonstrate promising classification accuracy, with the federated model achieving over 78% accuracy, closely matching the centralized baseline.**

***Index Terms*—wind turbines, acoustic data, predictive maintenance, federated learning, decentralized intelligent edge framework, wav2vec, oasees**

I. Introduction

Wind energy has become a cornerstone of modern electrical grids, playing an increasingly vital role in the global transition toward sustainable and low-carbon energy systems. As concerns about climate change and energy security intensify, wind energy offers a renewable, cost-effective, and scalable solution to reduce the dependence on fossil fuels. Technological advances and supportive policy frameworks have enabled wind energy to grow significantly in the past two decades. In 2024, wind power supplied approximately 8.1% of global electricity, reaching more than $2,494$ TWh, with cumulative installed capacity exceeding $1,136$ GW [1]. Integrating wind-produced energy into electrical grids presents both opportunities and challenges, especially w.r.t. intermittency, grid stability, as well as the need for enhanced storage and forecasting systems. Understanding the role and impact of wind energy within current power systems is essential for designing resilient and future-proof electricity infrastructures.

Despite their benefits, the maintenance of wind turbines presents significant challenges due to their remote locations, high operational loads, and exposure to harsh environmental conditions. Traditional maintenance strategies, such as scheduled or reactive maintenance, often lead to increased downtime and higher operational costs. In response, predictive maintenance strategies that rely on continuous monitoring and advanced analytics to anticipate failures before they occur, have started to be adopted instead. Among them, acoustic measurements are gaining traction, as through the analysis of the subtle sounds and acoustic emissions generated by turbine components, such as gearboxes, bearings, and even blades, researchers and operators can detect early signs of wear, fatigue, or damage. This non-contact and non-intrusive method provides valuable insights into the health of critical components, enabling timely interventions and optimizing maintenance schedules, thereby reducing operational costs and

European Union's Horizon Europe Framework Programme for Research and Innovation (Grant Agreement No. 101092702)

maximizing turbine availability. Recent research has demonstrated the effectiveness of acoustic-based condition monitoring systems in identifying gear and bearing defects under variable load conditions [2], [3].

In principle, wind turbines generate vast amounts of sensor data (acoustic, vibration, temperature, etc), often located in geographically dispersed and bandwidth-limited environments. Therefore, transmitting raw data to a centralized server and then training machine learning-based (ML) or deep learning-based (DL) anomaly detection models becomes impractical and also raises security issues. In this respect, decentralized computing solutions at the intelligent edge provide a viable alternative, as they address privacy and cybersecurity concerns by keeping sensitive operational data on-site. One such promising methodology is Federated Learning (FL) which, instead of transmitting raw data to a centralized server, allows local edge devices (e.g., embedded controllers or nearby gateways) to train machine learning models on-site and share only model updates with a central aggregator. In this way, individual wind turbines or local wind farm edge devices train predictive maintenance models using their own proprietary data, thereby significantly reducing data transmission needs, lowering latency for real-time anomaly detection and enhancing data security and compliance. Furthermore, by collaboratively training on diverse data originating from various turbines operating under different environmental conditions, the global FL model can enhance its generalization ability in predicting complex failure patterns, ultimately optimizing maintenance schedules and extending turbine lifespan across an entire fleet [4].

A robust way for the efficient deployment of FL algorithms, particularly in complex and distributed environments like wind farms, smart grids, or industrial IoT systems, is through decentralized, intelligent, and programmable edge frameworks designed for swarm architectures. These frameworks leverage the collaborative and self-organizing nature of swarm intelligence—where multiple edge nodes operate autonomously yet cooperatively—to support dynamic workload distribution, fault tolerance, and real-time decision-making. By embedding intelligence into edge nodes, such architectures allow local processing and learning while maintaining seamless coordination for global model convergence. Programmability at the edge enables flexible orchestration of FL tasks, such as adaptive client selection, privacy-preserving aggregation, and context-aware optimization.

In the current work, the *Open autonomous programmable cloud apps & smart sensors* (OASEES) framework [5] has been selected as the decentralized, edge framework of choice, as it embraces the entire computing continuum, including central infrastructures (public clouds and networks), as well as smart devices. From the data scientist and engineer viewpoint, OASEES provides user-friendly abstractions (notebooks, simplified administration interfaces, graphical workflow designers, etc.) to data experts, so that the latter can concentrate on the management of the data and the selection and optimization of the ML/AI algorithms, rather than on the management of the physical and virtual resources which are needed and committed.

Upon OASEES & exploiting its SDK, we build and evaluate a DL model for the predictive maintenance of wind turbines. Edge devices record acoustic data on-site (near the wind turbines) and process them locally (at the edge), through a FL training procedure, which pushes gradients to a centralized aggregator. Model performance is then compared to centralized training and the corresponding conclusions are drawn. The rest of the paper is organized as follows; Section II overviews related works w.r.t. the objectives of the current contribution. Section III outlines the proposed methodology, while Section IV presents the obtained experimental results, as well as discussing their implications. Finally, Section V concludes the paper.

II. Related Work

Fault detection in wind turbine operation has emerged as a highly active and evolving area of research. Most existing approaches are grounded in the analysis of condition monitoring (CM) data and operational data, particularly those collected via Supervisory Control and Data Acquisition (SCADA) systems. These datasets are typically derived from a variety of monitoring signals directly obtained from the turbine, such as vibration measurements, temperature readings, rotational speed, power output, and other performance indicators. Leveraging such data enables the early identification of abnormal patterns that may indicate developing faults.

In this context, the literature presents a broad range of computational models designed to process and interpret these high-dimensional, heterogeneous signals. Among the most notable are deep autoencoders [6], which can learn compact representations of normal turbine behavior and detect deviations indicative of faults; Gaussian process models [7], which offer probabilistic predictions and uncertainty quantification; and ensemble learning techniques [8], which combine multiple models to enhance robustness and accuracy. Furthermore, hybrid architectures have been explored, such as the integration of long short-term memory (LSTM) networks—capable of capturing temporal dependencies in sequential data—with autoencoder structures for improved anomaly detection performance [9].

Lately, approaches utilizing acoustic signals captured by audio equipment in the vicinity of the wind turbines have also emerged. By converting raw audio recordings from turbine operation into spectrograms, these methods transform time-series data into rich, two-dimensional visual representations of frequency and amplitude variations over time. This enables the application of well-established image processing and computer vision techniques, such as convolutional neural networks (CNNs) [10], to identify characteristic patterns associated with mechanical faults or component degradation.

Beyond conventional CNNs, more sophisticated deep learning architectures have been investigated to improve detection accuracy and generalization across different turbines and operating conditions. Models such as MobileNet, ResNet, and VGG have been adapted for spectrogram-based fault classification, taking advantage of their ability to learn hierarchical and high-level feature representations while balancing computational efficiency [11].

In addition to these, hybrid and attention-based architectures have emerged as powerful alternatives. For example, an attention-enhanced convolutional recurrent neural network (ACRNN) has been proposed to classify known blade failure modes, effectively combining CNN layers for spatial feature extraction with recurrent layers to capture temporal dependencies in the acoustic signal [12]. The attention mechanism further allows the model to focus on the most informative segments of the input, improving interpretability and performance. To address scenarios where abnormal or faulty operation data is scarce or unavailable, this approach has been complemented with a normal-encoder network, enabling semi-supervised learning and anomaly detection through modeling of normal operational behavior.

A persistent challenge in the wind energy sector lies in the restricted exchange of operational and performance data between different stakeholders—such as energy companies, wind farm operators, and original equipment manufacturers (OEMs). These data include mechanical and aerodynamic properties of turbines, as well as their performance metrics under varying weather and environmental conditions. Such restrictions are often due to commercial confidentiality, intellectual property protection, and regulatory constraints, which hinder the development of collaborative, data-driven solutions.

In the domain of fault detection, FL has already demonstrated potential. For example, it has been applied to blade icing detection using SCADA data, enabling real-time, online detection methods that achieve a high rate of success without requiring centralized datasets citeCHENG2022124441. Similarly, FL has been used to train long short-term memory (LSTM) models capable of learning a wind turbine's normal operating patterns from historical SCADA data, which can then be leveraged for anomaly detection [14].

Beyond failure detection, FL has also been explored in short-term wind power forecasting, allowing multiple wind farms to collaboratively improve prediction accuracy for energy production while maintaining data privacy [15]. However, despite these advances, no studies to date have investigated the application of FL to anomaly detection or blade failure classification using acoustic data – to the best of our knowledge at least. In this respect, the current work is among the first to explore the potential of FL techniques in acoustic-based wind turbine predictive maintenance.

III. Methodology

This section presents in more detail the OASEES framework [5]; the decentralized, edge solution used for DL-based wind turbine predictive maintenance (Section III-A). Following, Section III-B outlines the actual use of the aforementioned framework for a DL-based predictive maintenance task on wind turbines.

A. The OASEES Framework

The OASEES framework [5] proposes a decentralized, intelligent, and programmable edge architecture for swarm-based applications, built upon the Decentralized Autonomous Organization (DAO) paradigm and integrating Human-in-the-Loop (HITL) mechanisms for effective decision-making. It emphasizes open, secure tools for swarm orchestration across diverse fields, while addressing critical challenges like portable, privacy-preserving identity federation aligned with GAIA-X [16] and International Data Spaces Association (IDSA) [17] standards. Unlike existing platforms that focus narrowly on edge infrastructure management, OASEES envisions a holistic approach spanning the full compute continuum—from cloud to edge to smart devices—while supporting heterogeneous accelerators (e.g., Graphical Processing Units, Neural Processing Units, Spiking Neural Networks, Quantum). It also targets usability gaps by introducing intuitive interfaces for data scientists and engineers, akin to those in public cloud environments. Furthermore, OASEES prioritizes dynamic security and integrity verification in multi-actor, rapidly evolving edge contexts, ensuring privacy and reliability throughout the infrastructure.

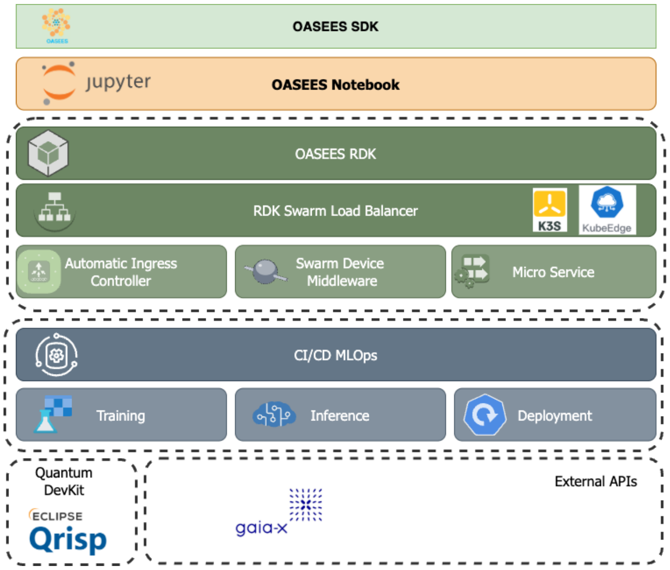

Fig. 1. OASEES SDK Programmability Stack

The OASEES framework is accessed through its SDK (Figure 1), aiming to provide a user-friendly environment for developers and engineers to access and program swarm clusters of devices for different applications and services. The OASEES SDK is primarily accessed via a graphical user interface based on Jupyter [18], offering a Python notebook-like experience and facilitating MLOps capabilities. The operational basis of the SDK is split among three base programming components, tailored and aimed for swarm-based deployments and architectures; i) the Rapid Development Kit (RDK), ii) the CI/CD MLOps and iii) the Quantum DevKit — Eclipse Qrisp [19]. The SDK also offers external APIs and interfaces to publish OASEES developed data assets to the European Open Science Cloud (EOSC) [20] and GAIA-X [16].

The OASEES RDK is basically, a suite of tools that enables developers to create edge applications tailored to their needs and the targeted edge device. It implements a cloud native approach, supporting different technologies (i.e., K8s, K3s and KubeEdge) depending on the needs and specifications of the scenario, and the limitations of the swarm devices. The CI/CD MLOps component can support the full lifecycle of AI/ML operations and proceeds one step further to package accordingly the model and algorithm and place them in the proper device to be executed upon. Qrisp [19] is an open-source python framework for high-level programming of Quantum computers, and in the context of the OASEES framework, specific swarm related functions are bridged and executed through the OASEES SDK environment for particular scenarios. Last but not least, the external APIs offer the developer the option to publish their asset, mode, algorithm to EOSC and/or GAIA-X in a seamless manner, with the SDK undertaking all the underlying necessary steps to establish a connection and execute the required actions.

B. Predictive Maintenance

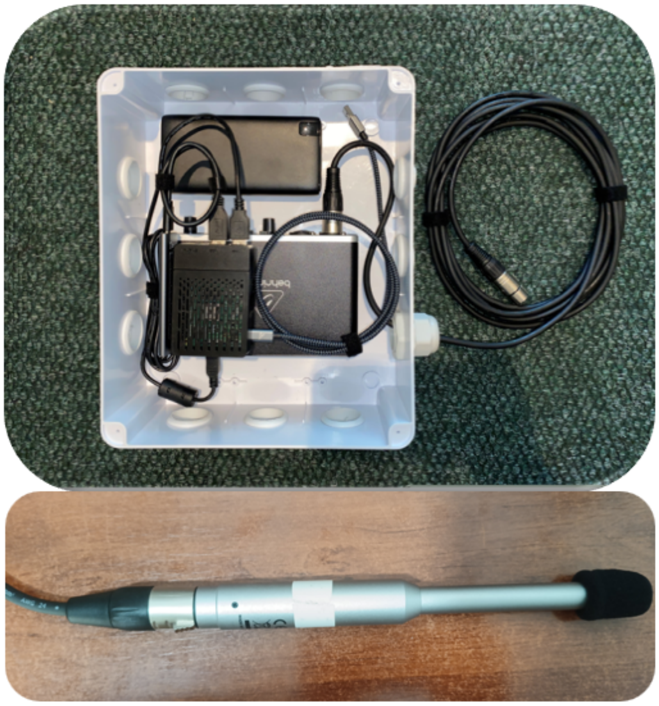

Fig. 2. Generic Acoustic Monitoring System

The predictive maintenance task that is going to be implemented is the Blade Acoustic Monitoring System (BAMS), which is portable, non-intrusive and manufacturer-independent. Using the recording equipment of Figure 2 to capture acoustic data (sounds), BAMS can listen to the wind turbine, acquire the acoustic signals produced when the wind interacts with the blades, and detect and identify faults in them, thus improving the performance and operational reliability of the wind turbine. This early detection also allows the optimization of energy production, extending the useful life of the blades. BAMS can detect abnormal operations in the blades such as structural failures, wear, ice, corrosion, or dirt.

The recording equipment (Figure 2) is a general-purpose portable monitoring system, assembled using low-cost components, since expensive calibrated sound pressure level meters are not required for the task at hand. It is comprised of a microphone, a USB audio interface, a solar-powered battery, USB data storage and a Single Board Computer which processes the audio data. This device is installed near the base of the wind turbine, as illustrated in Figure 3.

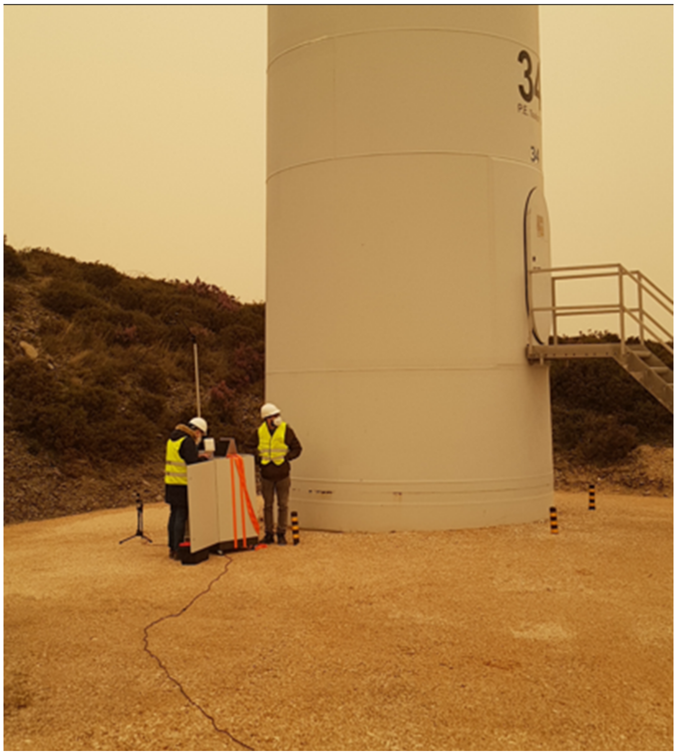

Fig. 3. Acoustic Monitoring System under a Wind Turbine

BAMS functions as an IoT device in swarm mode, allowing the simultaneous capture and processing of data from a considerable number of wind turbines, resulting in the creation of a network. This is illustrated in Figure 4, which depicts the architectural overview of the edge services deployed in the context of the wind farm to enable the distributed MLOps. The key component of the architecture is the IoT device (agent) residing at the network edge (wind turbine location), as depicted on the block-diagram on the left hand-side of Figure 4. The sound recorded from the USB audio interface is sent to a message bus enabled by the MQTT protocol, which makes audio chunks available for subscription in real time from different applications. Those chunks are stored locally on the agent and they can be directly accessed by the FL client, which performs local model training, sending gradients and model parameter updates to the FL server, residing at the wind farm station. Data are exchanged via a secure communication channel (SSL-enabled).

The MQTT client also feeds data to subscribed analytics services at the edge device, such as the anomaly detection module, which performs anomaly detection and failure classification in real-time. Furthermore, feature extraction is performed on the audio data in order to obtain relevant features for the training of additional ML algorithms. Data generated from the audio chunks and the extracted features can be saved in local storage for further offline analysis and

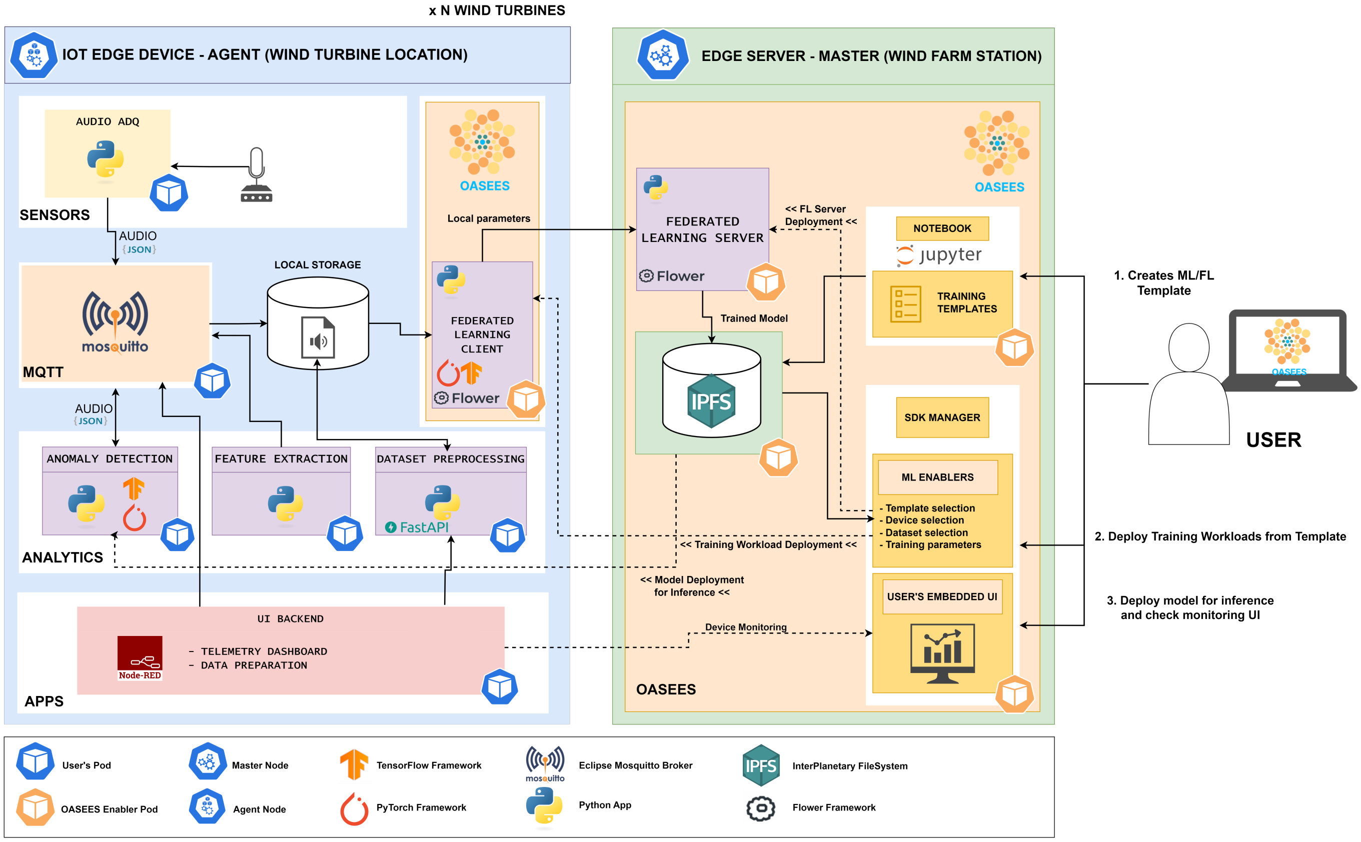

Fig. 4. Wind Farm distributed monitoring architecture within OASEES

dataset generation. Finally, the platforms offer the possibility to telemetry applications, to directly communicate with the edge device to inspect data and check the wind turbine blade health in real time.

The Edge Server (right hand-side of Figure 4) resides at the main premises of the wind farm station and communicates, through the OASEES framework, with all edge devices near the turbines. Apart from hosting the FL server, which aggregates model updates from all clients, the server provides a connection to the Interplanetary File System (IPFS). IPFS can hold trained models, which may be monetized and made available to other users of the framework, as well as data. The end user (engineer and/or data scientist) accesses the framework through this master node. S/he can deploy ML and FL templates using Jupyter notebooks or the OASEES SDK CLI utilities. The researcher may deploy training workloads after selecting predefined templates, devices, datasets and training parameters. Additionally, s/he may write and deploy his/her own models from scratch. Telemetry and UI capabilities permit the overall monitoring of the training procedure.

The whole architecture is fully modular and can be adapted to work in any wind farm or an array of wind farms. There is also the possibility of directly storing the collected data on the IPFS (via the master node) to facilitate data analysis when network or storage limitations are present.

IV. Experimental Procedure

Non-disclosure agreements do not permit us to report results of the deployed architecture on actual wind farms, so we will perform a slightly different but fully relevant experiment. We will store a compatible dataset (Section IV-A) on the IPFS and we will subsequently train a DL model (Section IV-B), simplifying the architecture of Figure 4 (Section IV-C). Finally, we will discuss the obtained results (Section IV-D).

A. Dataset

The *WEA-Acceptance Data (version 1.0)* [21], is a dataset comprised of acoustic, meteorological and operational wind turbine measurements. It has been compiled by researchers at the Leibniz University Hannover, who performed 5 measurements campaigns at an undisclosed location in northern Germany, characterized by homogenous, flat land of agricultural use. Out of the aforementioned campaigns, only data from the 5th campaign were publicly released, comprised of measurements spanning the month of April (again, not disclosing the year). Even though measurements contained acoustic, as well as meteorological data, in order to be aligned with the architecture presented in Section III-B, only the former were considered.

Acoustic measurements were captured by 3 different microphones, placed at the vicinity of three different wind turbines at the aforementioned agricultural location in northern Germany, surrounded by fields & separated by ditches, which can contain water. The dataset was split in 30 distinct files, one for each day of April. In our experiments, we used the first 24 days as the training set (80% of the dataset) and the last 6 as the test set (20% of the dataset).

Recordings for each day are stored in a single archive, composed of 3 subfolders (one for each location). Every subfolder includes 48 audio files in FLAC format, containing half-an-hour recordings for that day and for the specific microphone. The only exception to this rule are the recordings for the 22nd of April, where a number of recordings are missing. Finally, a CSV file accommodates information about the operational status of the wind turbines (target variable) in 10-minute, non-overlapping intervals. Therefore, the audio was segmented into 10-minute segments, aligned with the target variable. No other augmentation or filtering technique has been employed.

The operational status can take five distinct labels, turning the predictive maintenance task into a classification problem in this case. These are i) the STOP label, when the power output of the farm is equal to zero, ii) the NORMAL label, when the farm is operating smoothly and produces power at the desired level, iii) the PARTIAL STOP, which denotes the state in-between STOP and NORMAL in which the turbine starts running, or, in reverse, in which the turbine stops running, but momentum has it still moving, iv) the CURTAILMENT label, when the turbines produces energy below the desired threshold and v) the PARTIAL CURTAILMENT label, which, similar to the PARTIAL STOP; this state describes the change between CURTAILMENT and NORMAL operation, during which the blade pitch is regulated and the power output values change to the values expected for the given wind speed. Finally, the OUTLIER label is for those data points that were not classified in one of the preceding categories and have a “power output” value that is much smaller than the value of the manufacturer’s power curve for the specific wind speed.

B. Model

Since, to the best of the authors’ knowledge, no publicly available DL-based model trained on wind turbine acoustic data exists, the decision was made to pre-train an existing model than train a new one from scratch. This decision was bolstered by the fact that, in wind turbine predictive maintenance, collecting labeled fault data is expensive and rare (wind turbines spend most of their time in a healthy state). Therefore, by starting from pretrained representations, only a small amount of task-specific labeled data is needed to fine-tune a downstream fault detection model, significantly lowering data annotation costs.

For this reason, the popular *wave2vec* speech recognition model [22] has been selected, which was further fine-tuned on the dataset described in Section IV-A. Wav2vec transforms raw audio into compact representations (codes) that can be used with existing audio recognition models. It is trained directly on large amounts of unlabeled audio, learning to map raw waveforms into compact, information-rich vector representations. In this respect, the model learns how to extract fundamental audio patterns—such as frequency structures and noise characteristics—that are present in many acoustic domains (and not just speech). In addition, wav2vec’s pretraining involves augmentation with noise and masking, making its features more resilient to environmental variations—important for wind turbines that operate in noisy, changing outdoor conditions where wind, rain, and background noise can mask fault-related sounds.

One key challenge was handling the continuous nature of audio, which wav2vec addresses using a self-supervised training approach inspired by word2vec. The architecture includes two stacked convolutional neural networks; an encoder that processes 30 ms segments of audio and a context network that builds longer-range representations. During training, wav2vec introduces distractor samples—short segments of audio swapped with others—and tasks the model with identifying the correct version. This challenge is repeated across each 10-second clip, with increasing difficulty as the model also predicts upcoming audio changes in 10 ms intervals.

C. Experimental Architecture

Figure 5 illustrates the experimental, swarm architecture used in the experiments. It is directly derived from Figure 4, with the main difference being that no node-clients exist. Data are stored at the IPFS, with the three different folders designating different wind turbine locations and therefore using them as separate data sources in the FL training scenario. Additionally, in this proof-of-concept (POC) implementation the swarm coordination abilities and HITL features of the OASEES SDK are not fully exploited.

Otherwise, the model training procedure remains the same; the developers select the ML project to deploy training workloads for, also selecting which node(s) and specific compatible dataset they want to assign the workload to. Additionally, they also configure the training parameters (Steps 1 & 2 of Figure 5)

A call is made to the node that the SDK Manager’s backend resides in, to initiate the training procedure, which deploys the training workloads on the specified nodes via the Kubernetes API. Each node retrieves the project folder and the datasets from IPFS (Step 3 of Figure 5) and the client nodes execute their local model’s training (Step 4 of Figure 5). Local model parameters are sent to the server node and are aggregated into the full model (Step 5 of Figure 5). Upon finishing the training process, the server node stores the final model on IPFS (Step 6 of Figure 5).

D. Results

Table I presents the key training parameters and the corresponding performance results for the deep learning model described in Section IV-B, trained in a centralized setting outside of the OASEES framework. In this setup, all training data were consolidated on a single node, allowing conventional training without the constraints of distributed learning. The hyperparameters listed, optimized through extensive tuning, include the number of training epochs, the learning rate and the choice of the optimizer. In particular, the model achieved an accuracy of 85%, which is a promising result given that it was originally pre-trained on a completely unrelated dataset designed for speech recognition tasks. This outcome highlights

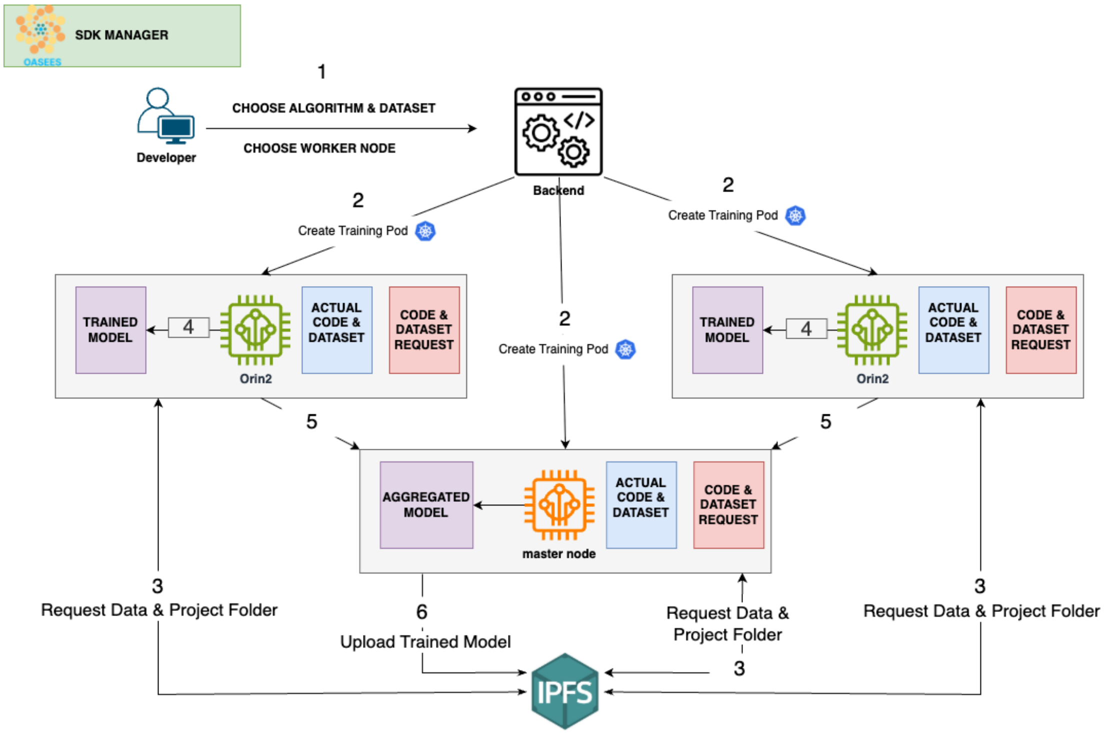

Fig. 5. OASEES SDK AI/ML operational diagram

the adaptability and potential of transfer learning approaches, even when applied to domain-shifted problems such as wind turbine acoustic signal classification.

TABLE I
TRAINING PARAMETERS AND RESULTS FOR CENTRALIZED TRAINING

Epochs	10
Learning Rate	3×10^{-5}
Optimizer	Adam
Accuracy	85%

TABLE II
TRAINING PARAMETERS FOR FEDERATED LEARNING

Training Rounds	3
Learning Rate	3×10^{-5}
Optimizer	Adam
Clients	3
Aggregation Strategy	Averaging

Table II outlines the selected hyperparameters and configuration used for the FL scenario. To ensure a fair comparison with the centralized training approach, the same core hyperparameters, such as learning rate and optimizer, were retained in both sets. The number of FL clients participating was set to three, corresponding to the three distinct microphone locations in the dataset, reflecting a realistic edge deployment, as described in Section III-B. For this POC implementation, the basic Federated Averaging (FedAvg) algorithm [23] was used as the aggregation strategy. In this approach, each client independently trains the model on its local data and periodically sends updated model parameters to a central aggregator. The aggregator then computes the average of these parameters across all clients to form a new global model. Although FedAvg offers simplicity and ease of implementation, it does not account for variations in data distribution or client contribution, making it a useful baseline for evaluating the initial performance and convergence behavior of FL in this context.

Table III illustrates the progression of the loss function and the accuracy metric over the course of the three FL training rounds. Initially, the model exhibits relatively high loss and low accuracy, which is expected given the limited training time and the decentralized nature of the setup. However, as the training rounds advance, a clear trend of performance improvement emerges. By the third round, the loss has decreased substantially to 0.527, while the accuracy has risen to 78.57%. This steady improvement indicates that the model is successfully learning from the decentralized data sources and gradually converging towards a more accurate global representation. Notably, the final FL-based accuracy is quite close to the 85% achieved in the centralized training scenario, demonstrating the viability of FL for audio-based predictive maintenance, despite its inherent challenges, such as non-IID data and limited communication rounds. These results are particularly promising, as they were obtained using a basic averaging strategy for aggregation and without further fine-tuning of the model architecture specifically for FL.

V. CONCLUSIONS

In this work, a novel architecture for the predictive maintenance of wind turbines has been presented, based on acoustic signals, captured by microphones positioned in close proximity to the turbines. To the best of our knowledge, the proposed

TABLE III
LOSS AND ACCURACY VALUES FOR FL TRAINING ROUNDS

Training Round	Loss	Accuracy
1	0.627	61.90%
2	0.551	73.81%
3	0.527	78.57%

methodology is among the first to harness the potential of FL in acoustic-based wind turbine predictive maintenance.

The core elements of the described system are built upon the OASEES framework; a decentralized and programmable edge framework that spans the entire computing continuum, from central cloud infrastructures to smart edge devices. This framework has been shown to be particularly well-suited for the challenges of the task at hand, as it offers capabilities for continuous audio stream ingestion, real-time preprocessing, localized storage for historical data analysis and decentralized training of DL models directly at the edge, using the FL paradigm. To validate the effectiveness of the proposed system, a POC implementation was carried out using a reference dataset, demonstrating the architecture's efficiency, scalability and suitability for real-world deployment in wind turbine monitoring scenarios.

Of course, the methodology and results presented thus far are still preliminary and serve primarily as a foundational POC. The DL model integrated within the proposed architecture, which performs audio-based classification of wind turbine acoustical signals, remains at an early stage of development. While the current implementation demonstrates the technical feasibility of the approach, significant work is still required to refine the model architecture, optimize hyperparameters, and enhance its robustness in real-world conditions. Most importantly, the model must be pretrained on large and diverse datasets of labeled wind turbine acoustic signals, capturing a wide range of operational states and potential failure modes. This will enable the model to learn discriminative features that are essential for accurate classification and early fault detection. In future iterations, the inclusion of self-supervised pretraining techniques or transfer learning from related audio domains could be explored to address the challenge of limited labeled data. Additionally, further experiments and evaluations under realistic deployment scenarios will be necessary to assess generalization ability, latency, and energy efficiency at the edge.

Finally, to fully harness the potential of FL in the context of audio-based classification of wind turbine acoustical signals, further exploration of more complex aggregation strategies is essential. While the current implementation relies on the standard FedAvg approach, this method does not adequately address the inherent challenges of decentralized audio data, such as non-independent and identically distributed (non-IID) data distributions, varying data quality, and heterogeneous client resources. Therefore, more advanced and adaptive aggregation strategies should be considered, like (federated) Adagrad or Adam. The aforementioned techniques can help mitigate issues related to client drift and imbalanced data contributions, ultimately improving convergence stability and model performance. Overall, the exploration of diverse and intelligent aggregation strategies is a critical step toward realizing scalable, robust, and efficient FL for real-world predictive maintenance applications in the wind energy sector.

ACKNOWLEDGMENT

This research initiative is supported by the European Union's Horizon Europe Framework Programme for Research and Innovation, under the OASEES project (Grant Agreement No. 101092702). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union, which cannot be held responsible for them.

REFERENCES

[1] G. W. E. Council, "Global wind report," Tech. Rep., 2025, (Accessed on July 11, 2025). [Online]. Available: https://26973329.fs1.hubspotusercontent-eu1.net/hubfs/26973329/2. Reports/Global Wind Report/GWEC Global Wind Report 2025.pdf

[2] D. J. Comboni and B. Fazenda, "Acoustic condition monitoring of wind turbines: Tip faults," *The Journal of the Acoustical Society of America*, vol. 132, no. 3_Supplement, pp. 1956–1956, 09 2012. [Online]. Available: https://doi.org/10.1121/1.4755203

[3] Y. Guo, A. Thomson, R. Bergua, O. Bankestrom, and J. Erskine, "Acoustic emission measurement of a wind turbine main bearing," National Renewable Energy Lab. (NREL), Golden, CO (United States), Tech. Rep., 2022. [Online]. Available: https://www.osti.gov/biblio/1893358

[4] R. Zhou, Y. Li, and X. Lin, "A clustered federated learning framework for collaborative fault diagnosis of wind turbines," *Applied Energy*, vol. 377, p. 124532, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261924019159

[5] "Open autonomous programmable cloud apps & smart edge sensors," 2025. [Online]. Available: https://cordis.europa.eu/project/id/101092702

[6] H. Zhao, H. Liu, W. Hu, and X. Yan, "Anomaly detection and fault analysis of wind turbine components based on deep learning network," *Renewable Energy*, vol. 127, pp. 825–834, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960148118305457

[7] R. K. Pandit and D. Infield, "Scada-based wind turbine anomaly detection using gaussian process models for wind turbine condition monitoring purposes," *IET Renewable Power Generation*, vol. 12, no. 11, pp. 1249–1255, 2018. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-rpg.2018.0156

[8] S. R. Moreno, L. d. S. Coelho, H. V. Ayala, and V. C. Mariani, "Wind turbines anomaly detection based on power curves and ensemble learning," *IET Renewable Power Generation*, vol. 14, no. 19, pp. 4086–4093, 2020. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-rpg.2020.0224

[9] H. Chen, H. Liu, X. Chu, Q. Liu, and D. Xue, "Anomaly detection and critical scada parameters identification for wind turbines based on lstm-ae neural network," *Renewable Energy*, vol. 172, pp. 829–840, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960148121004341

[10] H. Li and Z. Wang, "Anomaly identification of wind turbine blades based on mel-spectrogram difference feature of aerodynamic noise," *Measurement*, vol. 240, p. 115428, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224124013137

[11] Y. Zhu and X. Liu, "A lightweight cnn for wind turbine blade defect detection based on spectrograms," *Machines*, vol. 11, no. 1, 2023. [Online]. Available: https://www.mdpi.com/2075-1702/11/1/99

[12] H. Liu, W. Zhu, Y. Zhou, L. Shi, and L. Gan, "Nonintrusive wind blade fault detection using a deep learning approach by exploring acoustic informationa)," *The Journal of the Acoustical Society of America*, vol. 153, no. 1, pp. 538–547, 01 2023. [Online]. Available: https://doi.org/10.1121/10.0016998

[13] X. Cheng, F. Shi, Y. Liu, X. Liu, and L. Huang, "Wind turbine blade icing detection: a federated learning approach," *Energy*, vol. 254, p. 124441, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544222013445

[14] A. Grataloup, S. Jonas, and A. Meyer, “Wind turbine condition monitoring based on intra- and inter-farm federated learning,” 2024. [Online]. Available: https://arxiv.org/abs/2409.03672

[15] Y. Li, R. Wang, Y. Li, M. Zhang, and C. Long, “Wind power forecasting considering data privacy protection: A federated deep reinforcement learning approach,” *Applied Energy*, vol. 329, p. 120291, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261922015483

[16] “Gaia-x: A federated secure data infrastructure,” (Accessed on July 11, 2025). [Online]. Available: https://gaia-x.eu/

[17] “International data spaces,” (Accessed on July 11, 2025). [Online]. Available: https://internationaldataspaces.org/

[18] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, C. Willing, and J. development team, “Jupyter notebooks ? a publishing format for reproducible computational workflows,” in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Scmidt, Eds. IOS Press, 2016, pp. 87–90. [Online]. Available: https://eprints.soton.ac.uk/403913/

[19] R. Seidel, S. Bock, R. Zander, M. Petrič, N. Steinmann, N. Tcholtchev, and M. Hauswirth, “Qrisp: A framework for compilable high-level programming of gate-based quantum computers,” 2024. [Online]. Available: https://arxiv.org/abs/2406.14792

[20] “European open science cloud,” (Accessed on July 11, 2025). [Online]. Available: https://eosc.eu/

[21] D. Schössow, S. Preihs, and J. Peissig, “Wea-acceptance data—a dataset of acoustic, meteorological, and operational wind turbine measurements,” *Data*, vol. 9, no. 3, 2024. [Online]. Available: https://www.mdpi.com/2306-5729/9/3/46

[22] S. Schneider, A. Baevski, R. Collobert, and M. Auli, “wav2vec: Unsupervised pre-training for speech recognition,” *CoRR*, vol. abs/1904.05862, 2019. [Online]. Available: http://arxiv.org/abs/1904.05862

[23] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, “Federated learning of deep networks using model averaging,” *CoRR*, vol. abs/1602.05629, 2016. [Online]. Available: http://arxiv.org/abs/1602.05629