IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 December 2023, accepted 3 January 2024, date of publication 8 January 2024, date of current version 16 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3350777

RESEARCH ARTICLE

Federated Learning-Aided Prognostics in the Shipping 4.0: Principles, Workflow, and Use Cases

ANGELOS ANGELOPOULOS[1], ANASTASIOS GIANNOPOULOS[2], (Member, IEEE), NIKOLAOS NOMIKOS[2], (Senior Member, IEEE), ALEXANDROS KALAFATELIS[2], (Graduate Student Member, IEEE), ANTONIOS HATZIEFREMIDIS[3], AND PANAGIOTIS TRAKADAS[2]

[1]General Department, National and Kapodistrian University of Athens, 34400 Euboea, Greece
[2]Department of Ports Management and Shipping, National and Kapodistrian University of Athens, 34400 Euboea, Greece
[3]Department of Aerospace Science and Technology, National and Kapodistrian University of Athens, 34400 Euboea, Greece

Corresponding author: Nikolaos Nomikos (nomikosn@pms.uoa.gr)

This work was supported in part by the ICOS project ("Towards a functional continuum operating system") funded from the European Union's HORIZON Research and Innovation Programme, under grant agreement No 101070177 (https://www.icos-project.eu/).

ABSTRACT The next generation of shipping industry, namely Shipping 4.0 will integrate advanced automation and digitization technologies towards revolutionizing the maritime industry. As conventional maintenance practices are often inefficient, costly, and unable to cope with unexpected failures, leading to operational disruptions and safety risks, the need for efficient predictive maintenance (PdM), relying on machine learning (ML) algorithms is of paramount importance. Still, the exchange of training data might raise privacy concerns of the involved stakeholders. Towards this end, federated learning (FL), a decentralized ML approach, enables collaborative model training across multiple distributed edge devices, such as on-board sensors and unmanned vessels and vehicles. In this work, we explore the integration of FL into PdM to support Shipping 4.0 applications, by using real datasets from the maritime sector. More specifically, we present the main FL principles, the proposed workflow and then, we evaluate and compare various FL algorithms in three maritime use cases, i.e. regression to predict the naval propulsion gas turbine (GT) measures, classification to predict the ship engine condition, and time-series regression to predict ship fuel consumption. The efficiency of the proposed FL-based PdM highlights its ability to improve maintenance decision-making, reduce downtime in the shipping industry, and enhance the operational efficiency of shipping fleets. The findings of this study support the advancement of PdM methodologies in Shipping 4.0, providing valuable insights for maritime stakeholders to adopt FL, as a viable and privacy-preserving solution, facilitating model sharing in the shipping industry and fostering collaboration opportunities among them.

INDEX TERMS Federated learning, machine learning, maritime applications, predictive maintenance, Shipping 4.0.

I. INTRODUCTION

The maritime sector with its diverse services is a major driving force towards global economic growth. In this context, maritime transportation supports international trade activities and logistics and its efficient operation relies on intelligent fleet management, minimal downtime for maintenance tasks and reduced fuel consumption with low-carbon footprint. The integration of wireless communications and sensing technologies, following the Internet of Things (IoT) paradigm, has started to revolutionize the shipping sector by leveraging the plethora of collected,

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang.

stored, and processed data [1]. Consequently, future maritime transportation systems are envisioned to employ real-time monitoring and automated optimization methods to achieve ship path optimization, energy consumption reduction, and speed optimization [2], [3]. Still, conventional optimization approaches may not be effective in this area due to the formulation of non-convex problems, necessitating the use of alternative optimization techniques, such as machine learning (ML).

Over the past few years, ML algorithms have emerged as a promising solution for handling large datasets and optimizing processes. In such applications, neural networks (NNs) are trained to accurately map input variables to output vectors. However, in situations where the output variables have distinct values, a clustering strategy may be more suitable, as it significantly reduces training time compared to NNs. Generally, ML training can be accomplished through supervised learning, unsupervised learning, and reinforcement learning (RL). In supervised learning, data associations are known in advance, and the algorithm is trained using labeled data. On the contrary, unsupervised learning does not rely on predefined patterns for associating input variables with output metrics. Instead, unsupervised learning algorithms analyze a large dataset to identify any underlying patterns or associations between inputs and outputs. Lastly, in RL, an agent interacts with the operating environment and aims to determine the optimal set of policies based on rewards or penalties associated with specific actions. By accumulating a sufficient number of training samples, it becomes possible to derive the best actions for all possible states.

In maritime services, effective ML training involves collecting and processing data from numerous distributed sources, covering a vast geographical area with diverse communication technologies and maritime nodes [4]. This poses challenges for conventional centralized ML approaches, particularly for applications relying on fast and reliable data transmissions, where efficient data gathering over large propagation distances is crucial. Additionally, proper data manipulation techniques are necessary to mitigate the effects of varying propagation conditions. Furthermore, data privacy concerns may arise as in many instances, training data may contain sensitive information specific to a particular maritime component, making it preferable to store it locally rather than transmit it via the wireless interface. To address these issues and ensure privacy preservation, the federated learning (FL) paradigm has emerged as an alternative ML architectural approach in recent years. FL aims to accelerate ML execution times while keeping data localized [5], [6]. In FL, maritime nodes utilize shared models trained on extensive amounts of data without the need for central data storage [7].

A. SHIPPING 4.0

Industry 4.0, also known as the fourth industrial revolution, aims to create an environment relying on real-time, intelligent, interoperable, and highly autonomous industrial systems. This vision is based on innovative information and communication technologies, such as cyber-physical systems (CPS), the IoT, and cloud computing (CC) [8]. CPS connects physical elements, i.e. sensors, operator panels, and computers, with cyber-elements. The physical elements collaborate and communicate to collect and provide data to the cyber-elements, where management, processing, and decision-making procedures occur. Moreover, IoT enables real-time interconnection of various objects, such as sensors, actuators, machines, and robots, in a secure and reliable manner. IoT relies on diverse communication networks, such as fifth- and sixth-generation (5G & 6G) networks, Wi-Fi, machine-to-machine (M2M) deployments, and cloud-edge technologies.

Meanwhile, Industry 4.0 does not ignore the role of humans in the manufacturing processes. In these smart industrial environments, humans are equipped with smart devices and have access to augmented and virtual reality (AR/VR) technologies. In addition, they remain actively involved in the manufacturing process by leveraging AI/ML-based decision-making capabilities. By harnessing these advanced technologies, Industry 4.0 has the potential to revolutionize current industrial production processes, benefiting stakeholders, personnel, and consumers, while also promoting environmental sustainability.

In the Industry 4.0 ecosystem, a wide range of applications is envisioned, offering flexibility, real-time self-optimization, automation, and the ability to handle complex tasks and meet high quality standards. To enhance these applications, the integration of advanced fault detection, prediction, and prevention technologies are crucial. The main enabler of predictive maintenance (PdM) in industrial settings is the wealth of data from industrial processes, allowing for the precise prediction of machine conditions, remaining useful life, and faults, enabling an appropriate and cost-effective maintenance schedule. As a result, ML-aided PdM algorithms play a significant role towards early and accurate fault detection, leading to minimum downtime of machinery by identifying in real-time, damaged or defective products and parts [9].

The maritime industry ecosystem hosts various activities such as fishing, shipbuilding, shipping, ports, offshore energy, equipment manufacturing, tourism, financial services, and logistics and it is anticipated to undergo significant changes due to Industry 4.0 advancements. According to the "Ocean Economy in 2030" report published by the Organization for Economic Cooperation and Development (OECD), the maritime industries have the potential to double their contribution to global value creation by 2030, driven by increased demand for shipping, shipbuilding, marine equipment, and related services. However, these forecasts are heavily based on the efficient integration of technological innovations introduced by the Industry 4.0 paradigm.

In the future maritime ecosystem, known as Shipping 4.0, technology-based innovations will play a fundamental role. More specifically, the main CPSs in the maritime sector, referred to as "smart ships," have the potential to replace conventional vessels in a fully interconnected maritime environment. Smart ships will adhere to new design criteria and operational requirements, leading to enhanced efficiency and sustainability. The development and implementation of the Industry 4.0 paradigm will form the foundation of this future maritime ecosystem, being characterized by ML-aided systems for autonomous navigation, PdM and fleet management.

As the shipping industry experiences increasing technological advancements and digitization, it is undergoing a process of digital transformation, being driven by the growing demand for improved data collection, processing, and networking capabilities [10]. It is evident that the future of the shipping industry will be increasingly relying on advanced IIoT systems for real-time acquisition, transmission, storage, and analysis of large volumes of relevant data. In this sense, Shipping 4.0 is expected to offer significant benefits to the shipping industry, including reduced operational costs, increased overall revenue, and extended machine service life.

B. MACHINE LEARNING

The growth in the field of ML-aided systems, driven by research in utilizing machines to process large volumes of data, whether from previous tasks or simulated scenarios, has been remarkable [11]. Within the maritime sector, a sheer number of vessels, containers, personnel and machines are connected under the IoT paradigm to support the next generation of shipping, smart ports and logistics. The exponential rise in wireless data related to maritime services necessitates the use of advanced ML solutions to meet diverse service requirements. To address these challenges, the adoption of online ML solutions in such dynamic and complex environments aims at the accurate prediction of parameters for accurate PdM, as well as time-of-arrival and fuel consumption estimation, amongst others, thus enabling optimal decision making.

ML is mainly classified into supervised learning, unsupervised learning, and reinforcement learning (RL). Other categories include semi-supervised learning, deep learning and in recent years, transfer learning [12] and federated learning (FL) [13]. The following section provides details for each ML category.

- **Supervised learning:** Supervised learning involves algorithms, relying on datasets consisting of both input and corresponding output. While supervised learning offers enhanced decision-making capabilities, the requirement for labeled data can often be impractical. Classification and regression analysis are examples of supervised learning algorithms, which can aid in the prediction of vessel condition and fault detection.
- **Unsupervised learning:** Unsupervised learning operates on training data that lacks labeled output. Clustering is a widely used technique within unsupervised learning, enabling the identification of patterns within datasets. In the context of PdM, unsupervised learning algorithms can be trained to learn the normal operating conditions of equipment, using operational data and identify possible anomalies, indicating faults.
- **Semi-supervised learning:** Semi-supervised learning follows an intermediate approach, regarding the nature of the available training data. Algorithms in this ML category exploit both labeled and unlabeled data during the training process.
- **Reinforcement learning:** RL involves determining an agent's strategy autonomously by considering the costs and rewards associated with each action. This type of learning significantly differs from the supervised/unsupervised learning categories, relying on the use of historical data. RL algorithms are trained by receiving feedback on previously taken actions and adapt their behavior based on the accumulated reward. Unlike supervised learning, where the model is trained with correct answers, RL does not have explicit answers. Instead, the RL agent makes decisions on how to perform a given task. If a training dataset is not available, RL learns from its own experiences and focuses on selecting suitable actions to maximize rewards (e.g., optimal behavior or path) in specific situations.
- **Deep learning:** DL is closely intertwined with the aforementioned ML categories and relies on multiple layers to construct artificial neural network architectures, enabling accurate decision-making. In this hierarchical architecture, lower-level features define higher-level ones, while feature extraction is performed autonomously. In PdM for Shipping 4.0, DL exploits the increasing amount of data from industrial processes for monitoring assets and optimizing maintenance tasks. In such cases, observations of vessels and machinery give rise to specific states that serve as input to the deep neural network (DNN), which determines the action to be taken by the agent. Each action yields specific rewards, which, in the long term, dictate the efficiency of the adopted DL policy.
- **Transfer learning:** In maritime environments where edge nodes' capabilities, e.g. ships, unmanned aerial/surface/underwater vehicles (UAVs/USVs/UUVs), and buoys might differ, the energy and resource requirements for training models might be prohibitive, particularly when constrained devices are involved. In such cases, knowledge transfer techniques can improve the learning performance without requiring extensive data labeling procedures. Transfer learning operates by first training a base network, often referred to as the "teacher" network. The learned features from this teacher network are subsequently transferred to a target "student" network, allowing it to benefit from the knowledge already acquired. This approach minimizes the need for extensive training on the target network, thus reducing the energy and resource demands in maritime environments. So, the acquired knowledge from a general source problem is

exploited to solve a related specific problem. Considering the maritime nodes as "students", transfer learning can provide resource savings, as long as the relation among the source and target problems is high.

- **Federated learning:** The FL category separates the process of model training from the requirement for direct access to raw training data. In FL, users collaborate by utilizing shared models, trained from extensive amounts of data, eliminating the need for centralized data storage and significantly easing the stress on communication links. In FL, nodes participate as clients in a federation with the goal of collectively solving the learning task, coordinated by a central server. Each client maintains a local training dataset that remains on the device and is not uploaded to the server. Instead, the client computes and communicates an update to the current global model, residing at the server. FL offers several advantages, particularly in maritime applications where training can be based on pre-existing data available at each client. Furthermore, FL ensures high levels of privacy and security since attacks can only impact individual nodes instead of compromising the entire cloud infrastructure. In maritime environments, FL-based model integration facilitates the creation of global popularity prediction models by leveraging local models [14], [15].

Even though, FL has shown numerous benefits in enabling privacy preservation, it also presents significant challenges in resource management, robustness, security, and incentive mechanisms [16]. These challenges become even more pronounced and difficult to address in the context of industrial IoT (IIoT), where much higher levels of security, safety, and reliability are demanded in applications,such as shipping, smart ports, smart factories, smart manufacturing and smart transportation. In the area of PdM there have been various studies presenting FL-based anomaly detection with minimized computation cost for industrial control systems [17], autoencoder-based FL, using vibration sensor data from rotating machines, enabling distributed training on edge devices [18], split-learning-based PdM, facilitating FL clients to maximize available resources within their local network, and addressing orchestration, device heterogeneity and scalability issues [19], blockchain-based FL, relying on a hierarchical aggregator network, punishing and rewarding clients according to their local model quality updates [20]. Still, the integration of FL-aided prognostics in Shipping 4.0 applications has been largely missing from the literature and in this study, we aim to fill this gap by carefully tackling a variety of maritime use cases.

C. CONTRIBUTIONS

Shipping 4.0 relies on ML to facilitate the provision of novel services, including PdM to ensure continuous operation and minimized costs. To the best of the authors' knowledge the only work dealing with FL-based PdM is [21]. In that work, FL allowed multiple smart ship owners to collaboratively train a DL-based model to perform fault diagnosis. The proposed solution relied on a control algorithm which adaptively adjusted the model aggregation interval during training and Paillier-based communication scheme to protect the data resources of industrial stakeholders during training. Contrary to that work, here, we proposed an end-to-end and general-purpose FL scheme for supporting PdM on multiple maritime nodes and diverse use cases. Also, contrary to the majority of the existing works that uses the standard FedAvg policy for combining the local models, here we assessed different FL policies. By testing three types of PdM problems, namely regression, classification and time-series forecasting, we adopt different strategies for implementing the federation step (FedAvg, FedAvgM, FedSGD and FedProx), as an attempt to deal with the dataset-specific intrinsic properties (e.g. data distributions, unknown inter-feature relationships) and heterogeneity between agents. Specifically, FL-based predictive maintenance is presented and evaluated for three practical maritime use cases, i.e. regression to predict the naval propulsion gas turbine (GT) measures, classification to predict the ship engine condition, and time-series regression to predict ship fuel consumption. Our contributions are the following:

- This paper presents a comprehensive and end-to-end pipeline for supporting FL-based PdM capabilities in the Shipping 4.0 era. Although the emphasis is given in the maritime sector, the architectural principles and the workflow for achieving FL is general-purpose.
- Complete experimental testing of three types of PdM problems, namely regression, classification and time-series forecasting, was conducted to concretely evaluate the FL-based PdM performance in practical maritime use cases, exploiting both open and real datasets.
- Since FL model performance can be significantly be affected by the policy used for combining the local model weights, this paper compares different broadly-used FL strategies for model aggregation. By comparing four different aggregation policies (FedAvg, FedProx, FedSGD and FedAvgM), results provide evidence on the suitability of each policy under system heterogeneity (i.e. variability in terms of the characteristics of each FL agent) or statistical heterogeneity (non-identically distributed data across the FL agents).

D. STRUCTURE

The rest of this article is organized as follows. Section II discusses relevant works in the field of ML-aided predictive maintenance in Shipping 4.0. Next, Section III includes the adopted methodology and provides details on the three maritime use cases. In Section IV, the proposed FL-based predictive maintenance in Shipping 4.0 is evaluated for different data sets and FL alternatives. Finally, Section V includes the conclusions of our study and future research directions.

TABLE 1. List of acronyms.

5G	Fifth generation
6G	Sixth generation
AIS	Automatic identification system
AR	Augmented reality
BCE	Binary cross-entropy
CC	Cloud computing
CP	Coolant pressure
CPS	Cyber-physical system
DL	Deep learning
DNN	Deep neural network
DWS	Discretized Wind Speed
FCNN	Fully-connected neural network
FedAvg	Federated averaging
FedAvgM	Federated averaging with server momentum
FedProx	Federated proximal learning
FedSGD	Federated learning with stochastic gradient descent
FL	Federated learning
FP	Fuel pressure
GAN	Generative adversarial network
GHG	Greenhouse gas
GT	Gas turbine
IIoT	Industrial Internet-of-Things
IoT	Internet-of-Things
LSTM	Long-short term memory network
LOP	Lub oil pressure
LOT	Lub oil temperature
MAE	Mean absolute error
M2M	Machine-to-machine
MCN	Maritime communication network
ML	Machine learning
MSE	Mean squared error
NN	Neural network
OECD	Organization for economic cooperation and development
PCB	Printed Circuit Boards
PdM	Predictive maintenance
PEC	Primary Engine Condition
PEPP	Primary Engine Propulsion Power
QoS	Quality-of-Service
RL	Reinforcement learning
SGD	Stochastic gradient descent
SOG	Speed over Ground
STW	Speed through Water
TFF	TensorFlow federated
TIC	Turbine Injection Control
UAV	Unmanned aerial vehicle
USN	Underwater sensor node
USV	Unmanned surface vehicle
UUV	Unmanned underwater vehicle
VR	Virtual reality
WD	Wind Direction

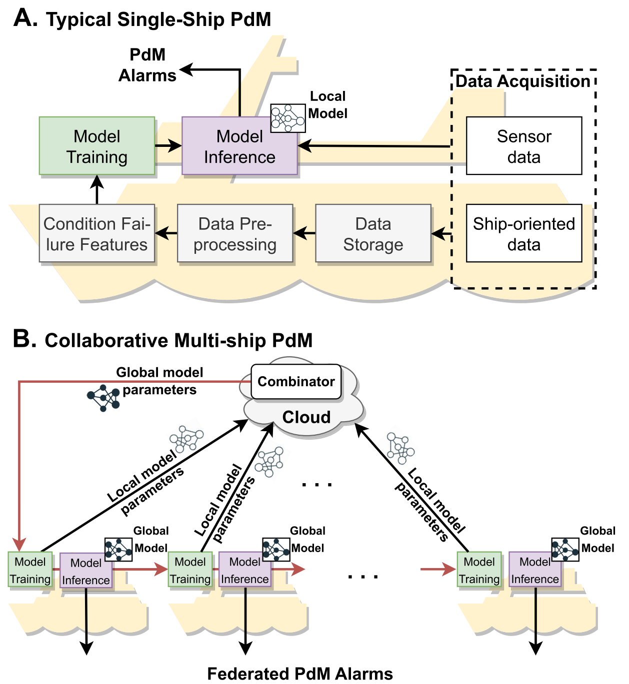


FIGURE 1. Predictive maintenance components in smart shipping considering single-ship (panel A) and multi-ship (panel B) learning.

II. PREDICTIVE MAINTENANCE IN SHIPPING 4.0

In the existing literature there have been various ML-aided approaches to conduct PdM in the context of the shipping industry. Fig. 1 depicts PdM components in the context of Shipping 4.0, considering single-ship (panel A) and multi-ship (panel B) ML-aided processes. More specifically, in the single-ship case, a local model is trained on the ship by exploiting data acquisition and processing, and the definition of condition failure features. On the contrary, for multi-ship ML-aided PdM, local model parameters are fed to a central cloud which derives the global model parameters that are transmitted to the ships towards improving their local models, thus achieving collaborative ML model training. Relevant studies have provided ML-aided PdM in Shipping 4.0 for both cases.

The study in [22] provided a short survey of PdM in the context of maritime systems, aiming at reducing maintenance and logistics costs while increasing asset availability. Specifically, the study discussed predictive methods based on the physics of failure, as well as various issues associated with the development and implementation of these models. Challenges in this field include critical part selection, the choice between data-driven or physics-based predictive modeling, monitoring/data collection, model validation, and formulating a business case. The paper examined two use cases, related to cylinder liners in a diesel engine and Printed Circuit Boards (PCBs) in a radar system. The findings indicated that companies aiming to transition to PdM solutions must invest in effectively measuring and documenting variations in operational profiles.

Next, the paper in [23] presented an ML-aided approach towards enabling maritime companies to benefit from advanced anomaly detection. Several contemporary methods in the field of fault detection while addressing crucial challenges such as interpretation, scale, accuracy, and complexity, which are inherent in many anomaly detection cases. In addition, the authors conduct a comparison of different approaches, including DNNs, SVM, Gradient Boosting, and statistical terms. For the comparisons a static window of 30 days was set, denoting the corrosion, prior to each defect. The combined model, comprising a fusion of these approaches, showed promising performance, in terms of F1 score and mean absolute error (MAE).

In shipping, the ability to identify evolving faults that have a detrimental impact on ship system performance and hinder energy-efficient operations is of critical importance. As a result, the work in [24] presented a data-driven methodology for fault detection in shipboard systems by exploiting the availability of recorded voyage data for ML purposes. The proposed approach combined the advantages of expected behavior models, which involved selecting the optimal regression model, with the exponentially weighted moving average technique for fault detection, specifically tailored to ship applications. Results demonstrated that the multiple polynomial ridge regression model, achieving a testing R2 score of 0.96 is capable to detect developing faults in both the main engine cylinder exhaust gas temperature and the main engine scavenging air pressure. The early detection of these faults efficiently complements routine ship operations monitoring, facilitating proactive corrective measures.

Then, the paper in [25] developed a weakly supervised learning-based PdM, employing balanced random forest and multiple instance learning, relying on data from event logs from ships' electric propulsion systems. Objectives encompassed, predicting the likelihood of failure, predicting the time to failure, and providing explainability for the predictions. Towards addressing the limitations of current event-driven techniques, temporal random indexing was used to transform irregular textual logs into a consistent numerical array format, offering dimensionality reduction and accurate failure prediction. Additionally, event aggregation techniques were integrated to enable explainable PdM by tracking the sources of failures. Even without manually labeled data, the overall approach successfully discovered the unknown actual class labels of infected samples with high confidence. Most notably, the majority of actual failures were successfully predicted at least three days in advance, indicated by at least one witness sample. These results are of significance to the shipping industry as the proposed PdM can mitigate and minimize the likelihood of propulsion loss during critical maneuvers, promoting the safety of equipment and personnel.

The authors in [26] proposed a PdM solution, utilizing a computational AI model, based on real-time monitoring data from historical values corresponding to the health of the vessel's engines and compressors, with R software employed for data analysis. The results of the analysis highlighted the impact of key parameters on the overall condition of the vessel's components, indicating strong correlations among sensor data collected from the same equipment in the majority of cases. These findings highlighted the potential of utilizing such parameters as inputs for the development of a predictive model. However, it was noted that additional factors related to identifying failure modes, detecting potential failures, and assessing asset criticality must be addressed to increase the accuracy of the predictions.

Ship hull and propeller fouling have significant consequences for both fuel costs and greenhouse gas (GHG) emissions. In this context, the paper in [27] focused on developing a PdM solution to enhance energy efficiency and reduce emissions from ships. In greater detail, a two-step approach for assessing the actual propulsive ship performance by analyzing continuous onboard measurements was presented. In the first step, the onboard monitoring data underwent correction to account for the influence of wind and wave effects, utilizing fast and transparent empirical methods. Subsequently, the corrected data were filtered based on hydrodynamic criteria. Next, the processed data were subjected to mathematical analysis tools to derive an engine power-rpm curve, representing the ship's actual propulsive performance. It was shown that the proposed ML algorithms enable the macroscopic identification of abnormal operation data during engine operation, prior to conducting detailed data analysis.

Finally, the first work developing FL-based PdM for smart ships was [21]. The authors presented a solution, addressing the challenges of insufficient data by introducing FL, enabling multiple industrial smart ship owners to collaboratively train a DL-based model. To mitigate computation and communication costs, a control algorithm was designed, adaptively adjusting the model aggregation interval during training. Additionally, a Paillier-based communication scheme to safeguard the data resources of industrial stakeholders was employed during the training phase. The FL-based PdM was evaluated in terms of privacy, security and accuracy, demonstrating its effectiveness and potential to maintain high fault diagnosis accuracy while reducing computation and communication overheads through adaptive model aggregation intervals.

III. METHODOLOGY AND USE CASES

In this section, we present three PdM uses cases, along with the associated methodology, aiming at providing a practical overview on how Shipping 4.0 can exploit FL-based PdM schemes. The key motivation to promote the adoption of FL-aided PdM approach in the future smart shipping lies in the obstacles faced by traditional centralized architectures, which require: (i) centralization of the data collection (violating the privacy-sensitivity of the maritime data), (ii) frequent transmissions of data under coarse propagation conditions, and (iii) high-dimensionality of the central model to ensure generalization of the model on the massive multi-source data (posing concerns on whether the low-capacity edge servers of vessels can execute those large models). Furthermore, based on the performance variability of different FL approaches, we illustrate the application of separate widely used FL algorithms in the deployed use cases. Note that, given the intrinsic and heterogeneous patterns of a multi-source (e.g. containing more than one vessels) dataset, the suitability of a given FL algorithm to a specific PdM optimization task cannot be *a priori* known and can be found upon extensive experimentation.

For completeness, we have selected three different scenarios, each one requiring the adoption of a different ML-aided solution, i.e. regression, classification and time-series forecasting/regression. Specifically, the following use cases are considered:

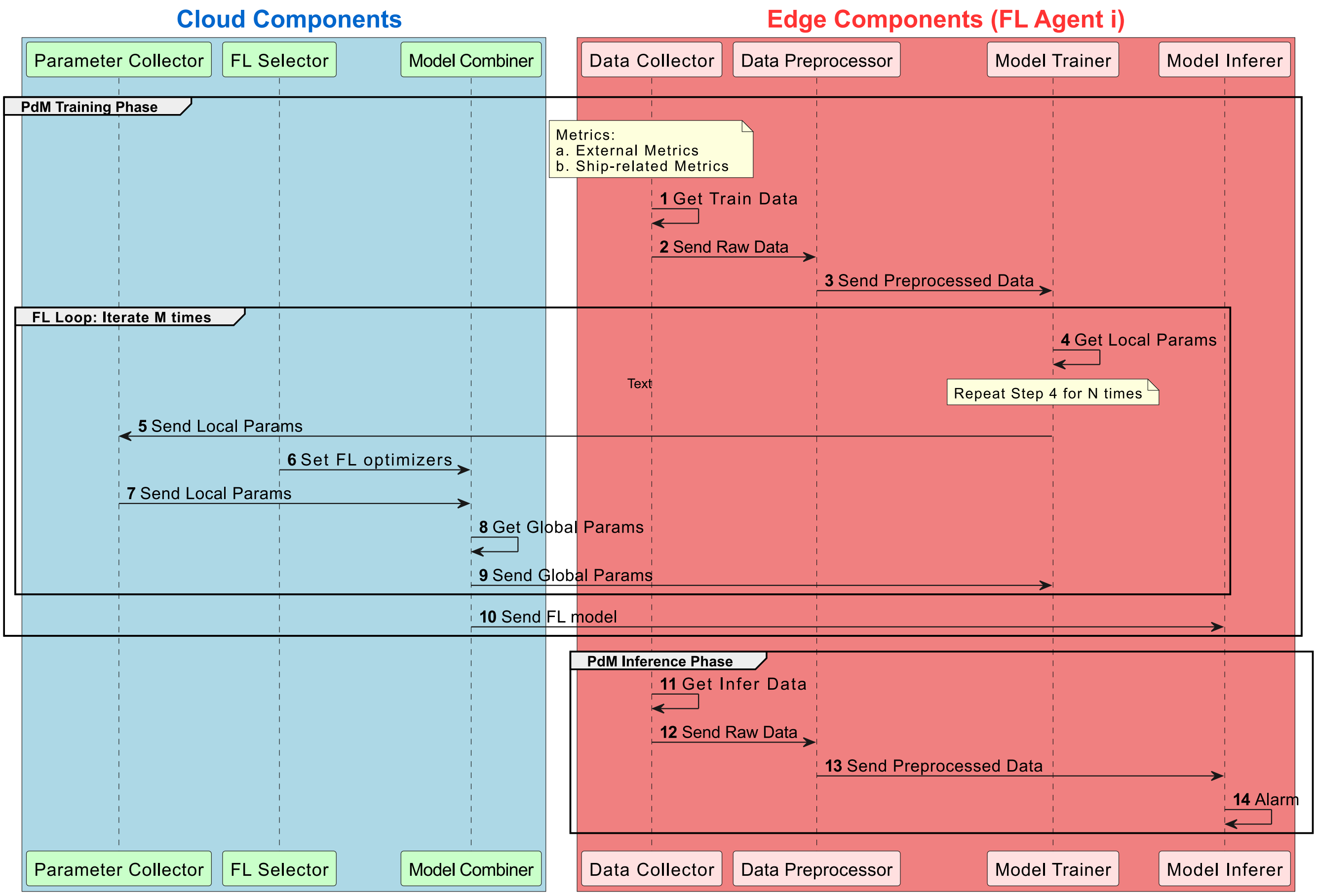


FIGURE 2. **Sequential diagram of the training and inference phases for enabling FL-aided PdM. 'Cloud' refers to the central entity, responsible for combining the local model updates, and 'Edge' refers to the local agents considered as FL participants.**

1) **Use Case 1 - Prediction of Naval Propulsion Gas Turbine Measures**: In this scenario, a multivariate regression ML model is constructed to enable Condition-based PdM of naval Vessels (Navy Frigate) equipped with Gas Turbines (GT). In greater detail, based on an open dataset [28], [29], an FL-based scheme is developed, targeting to collaboratively predict the GT performance degradation of frigates, as GT status has a serious impact on the vessel's propulsion system. The GT performance is represented by means of 2 GT parameters, namely the GT decay state coefficients of the compressor and the turbine, and is estimated using a 16-feature vector as input (or predictors).
2) **Use Case 2 - Prediction of Ship's Main Engine Condition**: This use case considers another important situation faced by the shipping crew and owners, concerning the continuous monitoring of the Primary Engine Condition (PEC). Given the strong influence of the PEC on the overall ship's fuel consumption and propulsion, here, we develop an FL-based multivariate classification model, capable of providing binary alarms ('Failure' or 'Normal') about the engine's status. This is achieved by exploiting 6 engine features as inputs and the categorical PEC parameter as desired output, according the open dataset in [30] (from IEEE DataPort).
3) **Use Case 3 - Prediction of Upcoming Main Engine Consumption**: The third use case concerns the time-series forecasting of the Primary Engine Propulsion Power (PEPP, usually expressed in kW) consumption, so as to allow, in cases of excessive fuel wastes, proactive actions that could be taken by either the engine's users or the ship captain. To that end, a dataset containing the temporal patterns related to both Engine Propulsion and Automatic Identification System (AIS) data of cargo ships was provided by a maritime enterprise [14]. The outcome of this use case is an FL-based multivariate Long-Short Term Memory Network (LSTM) that, given both ship- and weather-related information, is able to estimate the upcoming PEPP values, as a direct indication of the fuel waste.

For more detailed description of the datasets, see Section III-A.

To give a practical overview of the proposed workflow, Fig. 2 outlines the general-purpose sequential diagram required for supporting the employed use cases, involving

TABLE 2. Description of the Datasets used in the 3 use cases.

Dataset	Use case	Model[a]	Feature Metrics[b] (Acronym) [Unit]	Target(s) (Acronym) [Unit]
1	Multivariate regression for the prediction of the Gas Turbine status	FCNN	Lever position (lp); Ship speed (v); Gas Turbine (GT) shaft torque (GTT) [kN m]; GT rate of revolutions (GTn) [rpm]; Gas Generator rate of revolutions (GGn) [rpm]; Starboard Propeller Torque (Ts) [kN]; Port Propeller Torque (Tp) [kN]; Hight Pressure (HP) Turbine exit temperature (T48) [C]; GT Compressor inlet air temperature (T1) [C]; GT Compressor outlet air temperature (T2) [C]; HP Turbine exit pressure (P48) [bar]; GT Compressor inlet air pressure (P1) [bar]; GT Compressor outlet air pressure (P2) [bar]; GT exhaust gas pressure (Pexh) [bar]; Turbine Injection Control (TIC) [%]; Fuel flow (mf) [kg/s]	1. GT Compressor decay state coefficient (GTC coef) [0-1] 2. GT Turbine decay state coefficient (GTT coef) [0-1]
2	Multivariate classification for the prediction of the Engine Condition	FCNN	Engine rpm (rpm) [rpm]; Lub oil temperature (LOT) [C]; Coolant temperature (CT) [C]; Fuel pressure (FP) [bar]; Lub oil pressure (LOP) [bar], Coolant pressure (CP) [bar]	Primary Engine Condition (PEC) [binary]
3	Multivariate time-series forecasting for the prediction of the Propulsion Engine Power	LSTM	Speed over Ground (SOG) [kN]; Speed through Water (STW) [kN]; Heading (Head) [degrees]; Continuous Wind Speed (CWS) [m/s]; Discretized Wind Speed (DWS) [bft]; Wind Direction (WD) [degrees]; Draft Forward (DF) [m]; Draft Aft (DA) [m]; Trim (Trim) [m];	Primary Engine Propulsion Power (PEPP) [kW]

[a]FCNN: Fully-Connected Neural Network, LSTM: Long-Short Term Memory Network
[b]Here we present all the features contained in the datasets. Note that, some features have not been included in the final models, as they were rejected by running feature importance tests during preprocessing.

both Cloud and FL agents components. Components can be either physical or functional. Noteworthy, the Cloud representation could be replaced by any other central entity that is responsible for combining the local model updates, whereas the Edge components stand for any maritime node that is engaged in the FL process.

The PdM sequence unfolds as follows:

1) **Training Phase**: The use case sequence starts with the **Data Collector** gathering data for ship-related (measuring equipment in the ship) and ship-unrelated (sensors attached in the ship) parameters. The former refers to any measurement that is associated with the ship profile (position, speed, trim, heading, etc) and engine (rpm, torque, oil level, etc), whereas the latter concerns the external/environmental parameters affecting the ship voyage (wind direction, wind intensity, weather, etc). Upon the initiation of the FL process, raw data are sent to the **Data Preprocessor**, which includes the toolset to complete the preprocessing (cleaning, smoothing, data type transformation, dimensionality reduction, etc.). The **Model Trainer** should then initiate the hyperparameters of the local model and define the FL method that it will follow (FedProx, FedAvg, etc). Possible tuning iterations of the local model also take place in the Model Trainer. When the stabilization of the local model is achieved, the sequence enters the FL loop, where firstly, the Model Trainer executes N steps of the Stochastic Gradient Descent (SGD) algorithm to derive the model weights. Next, local parameters from all agents are collected by the **Parameter Collector**, which then forwards them to the **Model Combiner**. When Model Combiner receives the optimizer setup (FL method used for parameter combination, e.g. FedProx, FedAvg) from the **FL Selector**, it is able to produce the global model weights. Finally, the global model is returned back to the FL agents for continuing local training rounds and, when the aggregation round M is reached, the FL model is available for inference in the **Model Inferer**.
2) **Inference Phase**: To enable PdM with the deployed FL models, inference data should be first gathered and preprocessed (following the same steps as for the training data) and then, PdM predictions/alarms can be obtained by calling the deployed use case-specific model.

A. DATASET DESCRIPTIONS

This section tabulates the available data for implementing the considered use cases. Table 2 summarizes the information included in the datasets described in the second paragraph of Section III, containing the deployed local models, the features and the target variables per use case. Some of the metrics presented as features were finally excluded from the models input layer for dimensionality reduction purposes (features presenting multi-collinearity with other inputs, or independence with the outputs, were ditched as redundant).

B. FEDERATED LEARNING ALGORITHMS

There is a growing field deploying an FL algorithmic toolset, with each algorithm presenting advantages and disadvantages, depending on the optimization target and the dataset to which it is applied on. For example, intrinsic patterns of the data, data distributions and relationships, or a different degree of heterogeneity that exists between FL agents, can strongly affect the selection of the FL algorithm and the associated optimizers. We refer to optimizers, as the functions implementing the back-propagation algorithms (e.g. SGD, Adam). To this end, four different FL algorithms were used in this study, aiming to derive the optimal FL model per use case, namely:

1) **FedSGD** [31]: A classic approach for implementing the FL aggregation strategy, typically used as a baseline method in FL studies. According to this method, each local FL participant computes its gradient based on a set of training samples and transmits the computed gradient towards the cloud server. Once all gradients from all FL participants have been collected by the cloud server, they are averaged proportionally to the number of training samples on each participant and, then, sent back to all FL participants. Note that, for each batch of training data, FL participants should compute the gradient and, once the whole training dataset has been accessed, they forward the gradients towards the cloud. Formally, the procedure unfolds as follows: each FL participant $i = 1, 2, \ldots, N$ (where N is the total number of FL participants) selects a random data batch with size B of its local dataset and passes it to its local model. We let the local dataset of FL participant i having a size of D_i samples, whereas the total number of samples across all FL participants is D. The feed-forward passing steps result in an epoch loss function computation (i.e. the error between actual and predicted values of the batch) at FL participant i, which can be notated as $L_i(w_i)$, where w_i is the model weight matrix of FL participant i. Mathematically, the epoch loss function computed by FL participant i at time slot t can be given as:
$$L_i^t(w_i^t) = \frac{B}{D_i} \cdot \sum_{j}^{D_i/B} \sum_{k \in D_i} l_{j,k}^t(w_i^t), \tag{1}$$
where $l_{j,k}(w_i^t)$ denotes the loss function calculated for sample $k \in D_i$ belonging in the batch j of size B using the model weights w_i^t of FL participant i. Briefly, (1) computes the average loss amongst all batches that are selected in a single epoch (i.e. equal to the epoch loss of FL participant i). Then, the gradient of FL participant i at time slot t can be written as:
$$g_i^t = \nabla L_i^t(w_i^t), \tag{2}$$
where $\nabla(\cdot)$ stands for the gradient calculation. Finally, the weight matrix extracted by the cloud and received by all FL participants at the next time slot is the following:
$$w^{t+1} = w^t - \alpha \sum_{i=1}^{N} \frac{D_i}{D} g_i^t, \tag{3}$$
where w^t is the global model weights matrix at time slot t and α is a hyperparameter called learning rate.
2) **FedAvg** [13]: This aggregation approach is known as Federated Averaging, and is literally an immediate extension of the FedSGD to reduce to number of vertical communication rounds between the FL participants and the cloud. The idea behind FedAvg is to enforce each FL participant to locally compute the weights upon performing multiple local training rounds and, then, upload the resulting weights towards the cloud. Finally, the cloud computes a weighted mean across all local model weights and sends it back to all FL participants. Using identical notations as previously, we further define E as the number of the local training epochs performed by the FL participants before they send the local model updates. We also use the symbol t (or t') to denote the epoch ID counter (or the time slot for global aggregation). Thus, at every time slot t when an epoch ends, each FL participant i calculates its local weights as:
$$w_i^{t+1} = w_i^t - \alpha \cdot g_i^t, \tag{4}$$
where g_i^t is given by (2). Upon completion of E epochs for updating the local weights, assume at time slot t', FL participants transmit their matrices $w_i^{t'}$ to the cloud, with the latter returning back to all FL participants the new global model with the following weights:
$$w^{t'+1} = \sum_{i=1}^{N} \frac{D_i}{D} w_i^{t'}, \tag{5}$$
where $w^{t'}$ is the global model weights matrix at time slot t' and t' is the time slot of the aggregation step and is an integer multiplier of E (i.e. the local weights are uploaded for aggregation every E epochs).
3) **FedAvgM** [32]: This policy deals with non independent and identically distributed (non-IID) data across FL participants and is known as Federated Averaging with Server Momentum. FedAvgM is appropriate in cases where distributions and statistics differ significantly amongst local datasets, and provides convergence guarantees and mitigation of errors, introduced by the data heterogeneity via server momentum. The idea behind FedAvgM is that each FL participant i updates its local model update $w_i^{t'}$ at time slot t'. Note that local model updates follow the rule given in (4). Cloud server computes the new momentum according to the following formula:
$$m^{\text{new}} = \beta m^{\text{old}} + \sum_{i=1}^{N} \frac{D_i}{D} w_i^{t'}, \tag{6}$$
where m^{new} (or m^{old}) is the next (or previous) value of the momentum, and β is the momentum constant. Finally, the updated global model can be derived as:
$$w^{t'+1} = w^{t'} - m^{\text{new}}, \tag{7}$$
where $w^{t'}$ is the global model weights matrix at time slot t', which updated every E local training epochs as in typical FedAvg.
4) **FedProx** [33]: This policy, termed as FL with proximal term, is widely used for implementing FL under

statistical heterogeneity and non-IID data across the local FL participants. FedProx can be viewed as an empirical generalization of FedAvg, which makes some modifications to address the heterogeneity of data. Learning is again achieved in rounds, where at each round, the server randomly selects a subset of clients and sends them the current global model. At time slot t', the server sends to the selected FL participants the global model $w^{t'}$, which in turn perform E training epochs to minimize a regularized loss function, given by the formula below:

$$\min_{w_i^{t'}} \left\{ L_i^t(w_i^{t'}) - \frac{\mu}{2} ||w_i^{t'} - w^{t'}||^2 \right\}, \quad (8)$$

where $w_i^{t'}$ (or $w^{t'}$) are the local model parameters of FL participant i (or the global model parameters), and μ is the regularization constant. Noteworthy, FedProx is the same as FedAvg for $\mu = 0$. As implied by (8), FedProx properly adjusts the local weights of the selected agents, so as to ensure minimization of the regularized function, which is a function of the global weights. After performing local training rounds, the local weights are updated towards the cloud, where the global model is derived by applying (5), as in FedAvg.

All the aforementioned FL policies were implemented for comparison purposes.

Regarding the local models engaged in the FL loop, we used deep learning models in all use cases, as they have proven efficacy when the relationship between inputs/outputs is unknown and non-linear. Fully-Connected Neural Networks (FCNN) were used for use cases 1 and 2, and LSTM was considered in use case 3. Local training rounds targeted the minimization between the use case-specific loss function (error between the actual and predicted target values), as: (i) Loss of use case 1 was the Mean Squared Error (MSE) of the 2 GT measures, (ii) Loss of use case 2 was Binary Cross-Entropy (BCE) of the PEC and (iii) Loss of use case 3 was the MSE of the PEPP values. For the ease of exposure, Fig. 3 demonstrates the final structure of the ML models considered for the three use cases.

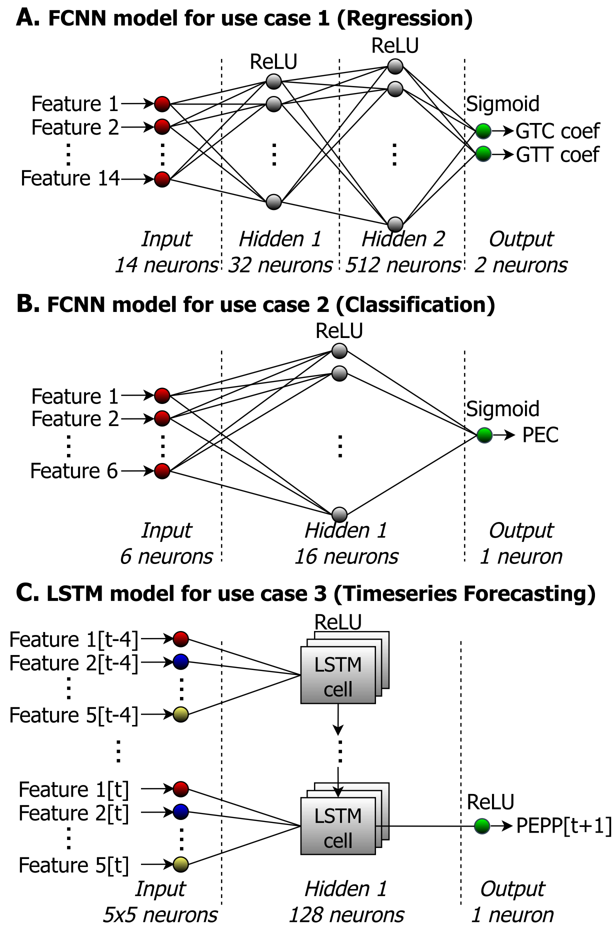


FIGURE 3. ML model structure and internal architecture for use case 1 (panel A), use case 2 (panel B) and use case 3 (panel C).

IV. RESULTS AND PERFORMANCE EVALUATION

The following subsections present the numerical results concerning the three use cases that were considered in this study. For each use case, there were six decentralized FL participants, together with their local datasets. The first five were considered as the FL local agents that are used to build the global FL model during the training phase, whereas the sixth was used for evaluating the FL model on unseen data. In this sense, the comparisons amongst the four FL policies (FedAvg, FedSGD, FedAvgM, FedProx) were conducted based on a dataset that was not encountered during the training for any of the FL schemes to ensure fairness and to assess the generalizability of the deployed schemes.

Simulations and algorithmic procedures ran on a personal PC with a processor Intel(R) Core(TM) i7-11800H, 2.30GHz, a 32 GB RAM, and a 64-bit operating system. The ML and FL programs were implemented using Python 3.9.12 and Tensorflow library, version 2.12.0, without GPU for model training acceleration.

A. USE CASE 1: PREDICTION OF NAVAL PROPULSION GAS TURBINE MEASURES

In the first use case, we use the Naval Vessel Condition dataset [29]. The dataset contains sixteen features that reveal the condition of a gas propulsion plant for Condition-based Maintenance. In the preprocessing stage, we neglected the features that showed non-significant Pearson's correlation coefficients ($-0.15 < r < 0.15$) with the target variables (GTC/GTT decay state coefficients). We then normalized all the dataset features in the range [0, 1] using the standard MinMax scaler. Afterwards, fourteen numerical features were used to predict the target variables, whereas the regression problem was approached with an FCNN model (per FL participant), since the relationship between the inputs and output was *a priori* unknown. The features that were excluded were the *GT Compressor inlet air temperature (T1)* and the *GT Compressor inlet air pressure (P1)*, as they have fixed

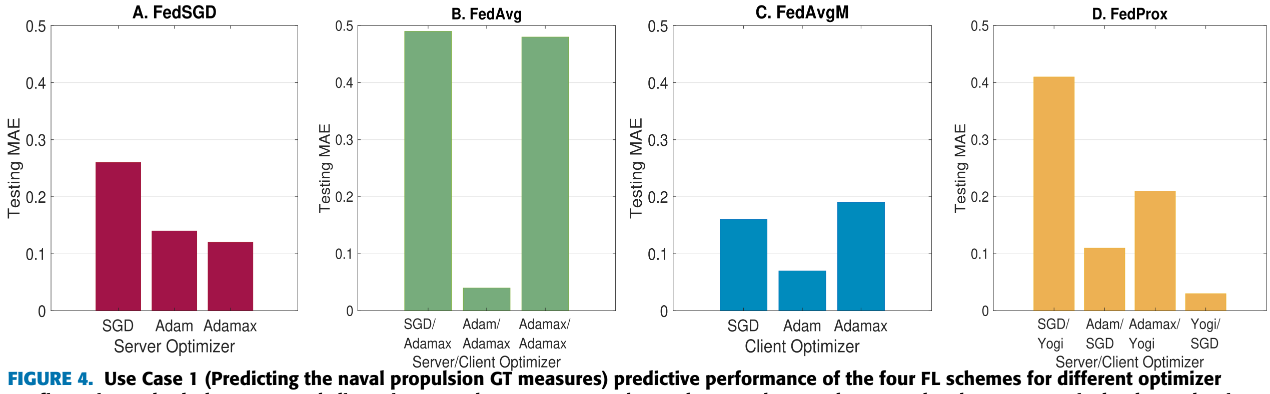


FIGURE 4. Use Case 1 (Predicting the naval propulsion GT measures) predictive performance of the four FL schemes for different optimizer configuration at both the server and client sites. Panels A-D correspond to FedSGD, FedAvg, FedAvgM and FedProx, respectively. The evaluation metric is the MAE loss between the actual and the predicted sample values drawn from the testing set.

values (took one unique value), thus not representing any feature variance.

1) TUNING OF LEARNING PARAMETERS

Initially, we conducted a series of simulation experiments for each FL agent to select the optimal configuration of the FL schemes in terms of the critical hyperparameters (learning rate, momentum, proximal strength, optimizers and number of local training epochs per aggregation round). First, the optimizers considered for implementing the backpropagation step of the FCNNs were the SGD, Adam, Adagrad, RMSprop, Adadelta, Adamax and Nadam (see Fig. 4). Note that, the backpropagation optimizer can be crucial for the final convergence of the model, since it influences the weights adjustment of the model, which in turn affect the produced model predictions. Notably, it is impossible to know in advance which backpropagation algorithm would be the best and, thus, multiple simulations are required.

Regarding the FL model deployment, we used the TensorFlow Federated (TFF) framework [34] for programming decentralized predictive maintenance algorithms in the shipping. We used four different FL policies (FedSGD, FedAvg, FedAvgM and FedProx) described in Section III-B for comparison purposes. In each FL algorithm, extensive simulations were carried out to fine-tune the optimizers both in the cloud server side and the FL participants' side. Different values of the learning rate α were also considered for the server and client optimizers, namely $\alpha = [0.001, 0.01, 0.1, 1]$. In the special cases of FedAvgM and FedProx policies, we experimented with different momentum values $\beta = [0.1, 0.3, 0.5, 0.7, 0.9, 0.96]$ and proximal strength $\mu = [0.1, 0.3, 0.5, 0.7, 0.9]$ values, respectively. The optimal values of the learning rate, momentum and proximal strength were set upon extensive simulations and are summarized in Table 3 for each scheme.

Considering the optimal values for both policies ($\beta = 0.96$ for FedAvgM, $\mu = 0.1$ for FedProx), Fig. 4 depicts, for each FL policy, the MAE loss, as a function of different optimizers. Note that, each ship-specific dataset was divided in training (80% of the whole dataset) and testing (20% of the whole dataset) sets. Thus, to compute the MAE loss, we used a combined testing set which was comprised of the individual testing sets of each ship-specific dataset. As such, the evaluation metric for all schemes was the MAE loss, calculated as the mean absolute difference between the actual and the predicted values of GTC/GTT decay state coefficients. The actual values were the ground truth target values, as they are included in the combined testing set, whereas the predicted values were the output target values, as they are derived by the global FL model, which was trained based on the training sets. As readily observed, the optimal optimizer for the FedSGD scheme was the Adamax which achieves a MAE of 0.12, whereas Adam was the best optimizer (MAE of 0.07) for FedAvgM. In addition, FedAvg showed the optimal accuracy (MAE of 0.03) when the cloud server uses the Adam and the FL participants use the Adamax optimizers, whereas FedProx exhibited the lowest prediction error (MAE of 0.02) for Yogi optimizer at the cloud server and SGD optimizers at the FL participants.

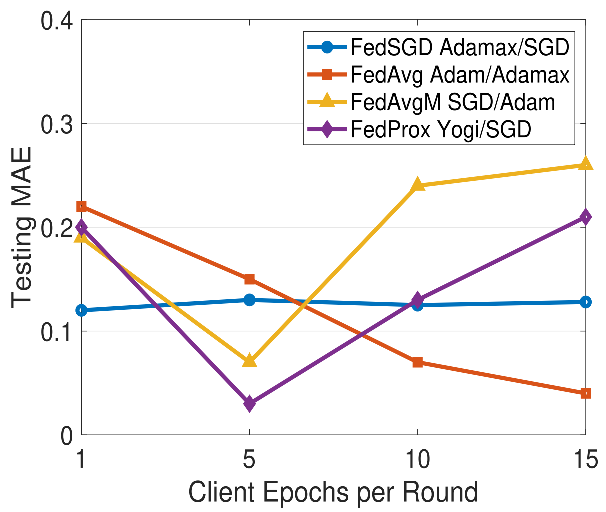


FIGURE 5. The impact of the number of Epochs per Round on the FL performance in Use Case 1. The MAE of FedSGD, FedAvgM, FedAvg and FedProx schemes are depicted, respectively. All schemes are configured with their optimal optimizer, whereas MAE metrics are calculated on the testing samples.

TABLE 3. Hyperparameter configuration for the four FL schemes considered in the comparisons of use case 1.

Parameter	**FedSGD**	**FedAvg**	**FedAvgM**	**FedProx**
Cloud/Client optimizer	Adamax/ SGD	Adam/ Adamax	SGD/ Adam	Yogi/ SGD
Cloud/Client learning rate α	0.01/ 0.001	0.1/ 0.01	1/ 0.001	0.1/ 0.001
Cloud/Client momentum β	1/ 1	1/ 1	0.96/ 1	1/ 0.96
Proximal strength μ	0	0	0	0.1
Local epochs per client	15	15	5	5

Using the optimal configuration of optimizers for each FL scheme, Fig. 5 shows the MAE loss (calculated as described above) as a function of the number of client epochs per round. In specific, the number of client epochs per round defines how many local training epochs are performed at each FL participant (to update the local model weights) between any two successive aggregation steps (at the cloud server). Noteworthy, this analysis allowed us to show whether an FL scheme requires frequent or rare communication rounds between the FL participants and the cloud, since, if an FL scheme shows the lowest error for low (or high) number of client epochs per round, this means that the aggregation steps should be performed frequently (or rarely) in time. Evidently from Fig. 5, FedAvg requires 15 local training epochs per aggregation round to achieve the highest accuracy, whereas FedAvgM and FedProx exhibit the best prediction accuracy for 5 local training epochs per aggregation round. FedSGD policy was independent of the number of local epochs per round (i.e. quasi-constant MAE loss) and, hence, we selected the highest number (15) of local training epochs per aggregation round to decrease the communication overhead between the FL clients and the cloud.

For the rest, each FL policy is configured with the optimal learning hyperparameters, as they were derived by the analyses of this subsection.

2) PERFORMANCE COMPARISON

This subsection presents the performance comparison amongst the considered FL policies, in terms of the achieved prediction accuracy of the GTC/GTT decay state coefficients. The MAE loss considered for the validation comparisons was calculated, as the mean absolute difference between the actual (i.e. the GTC/GTT decay state coefficients of the sixth ship dataset) and the predicted values (i.e. GTC/GTT decay state coefficients derived by inferring the global FL model that was constructed by the datasets of the first five ships). Note that, the sixth ship dataset was kept outside of the FL training phase, so as to be agnostic to each of the compared schemes of this subsection. For ease of exposure, Table 3 tabulates the final hyperparameter configuration of each FL policy for conducting the comparisons.

In Fig. 6, we demonstrate the MAE (computed over the samples of the sixth ship), as a function of the training aggregation rounds. Specifically, each point of the curves included in Fig. 6 has been derived as follows: for a given round, the FL model is inferred with the samples of the sixth ship, and the MAE of this round is equal

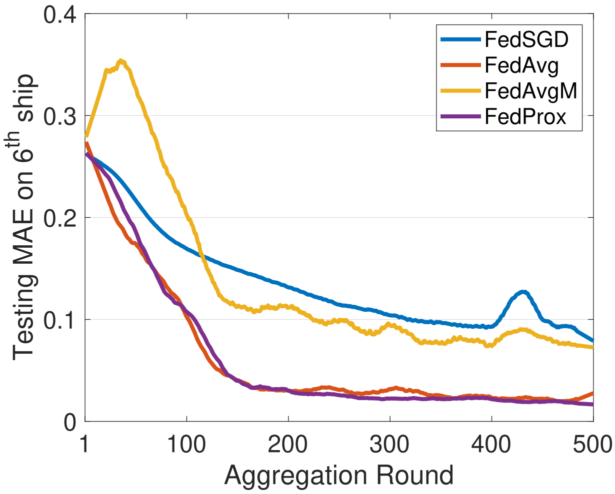


FIGURE 6. Learning convergence of the four FL schemes, as a function of the training rounds. In each marker of the curves, the evaluation metric is the MAE computed over the testing set.

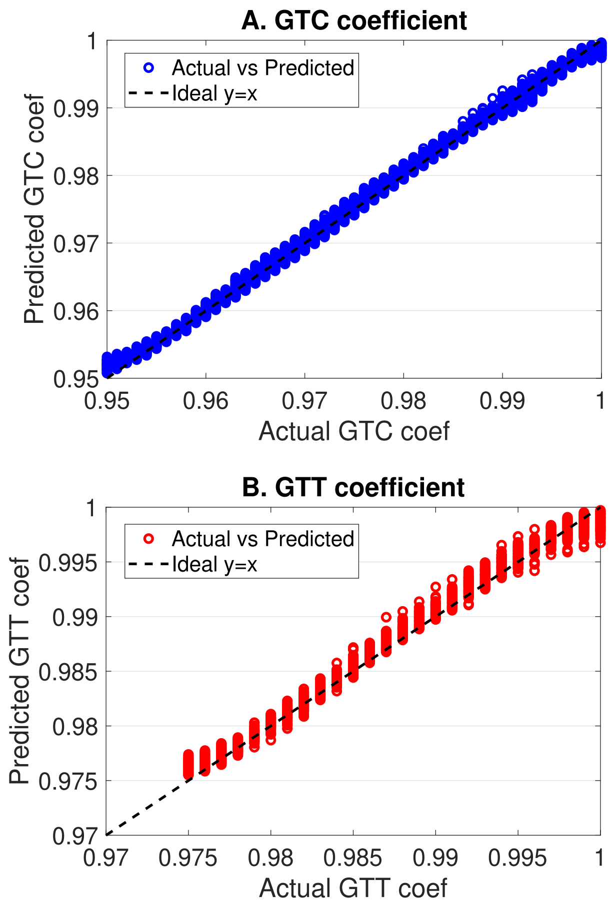


FIGURE 7. Actual and predicted values of the testing samples of the GT Compressor decay state coefficient (panel A) and GT Turbine decay state coefficient (panel B) as scatter plots. Both schemes refer to the FedAvg algorithm. The dashed line represents the ideal 'actual vs predicted' matching (y = x).

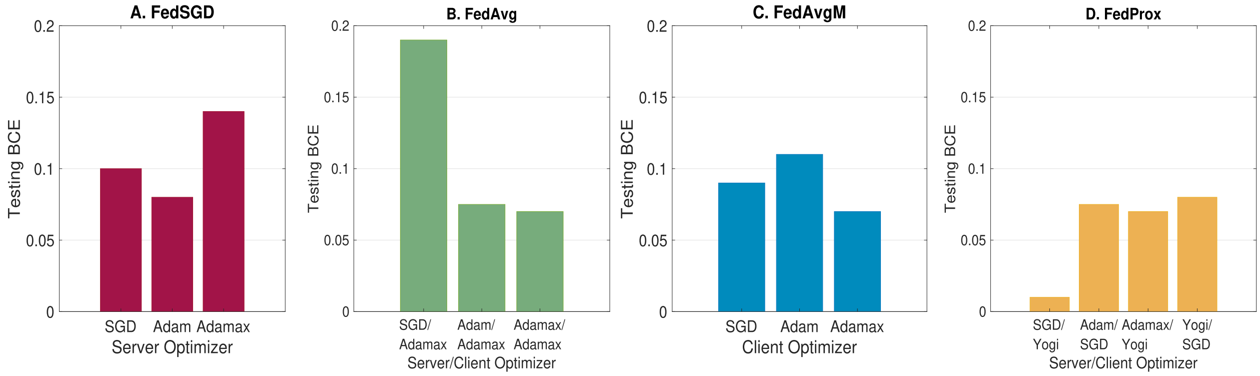


FIGURE 8. Use Case 2 (Predicting the engine condition) predictive performance of the four FL schemes for different optimizer configuration at both server and client sites. Panels A-D correspond to FedSGD, FedAvg, FedAvgM and FedProx, respectively. The evaluation metric is the Binary Cross-Entropy (BCE) loss between the actual and the predicted sample classes drawn from the testing set.

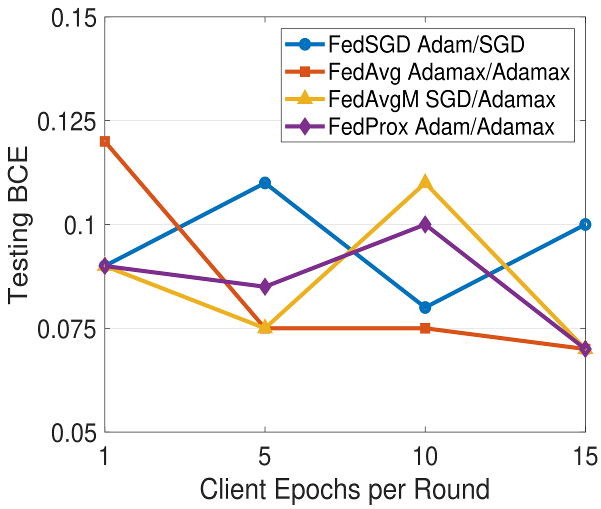


FIGURE 9. The impact of the number of epochs per round on the FL performance in Use Case 2. The binary cross-entropy (BCE) loss of FedSGD, FedAvgM, FedAvg and FedProx schemes are depicted, respectively. All schemes are configured with their optimal optimizer, whereas BCE metrics are calculated on the testing samples.

to the mean absolute difference between the ground truth and the FL model-predicted values of the GTC/GTT decay state coefficients. Note that, local training epochs take place within two successive rounds.

Evidently, all FL schemes show decreased prediction errors, as the number of aggregation rounds increases. FedAvg and FedProx outperform the rest of the schemes, achieving a MAE of about 0.02. Noteworthy, both methods also exhibit faster convergence than FedSGD and FedAvgM. In this use case, the similarity of FedAvg and FedProx, in terms of their prediction performance had been implied by the proximal strength $\mu = 0.1$ that was found to be the optimal for FedProx. Note that, FedProx is the general case of FedAvg, with FedProx being the same as FedAvg for $\mu = 0$ (see also (8)). We conclude that, the FedAvg is the optimal FL scheme for predicting the GTC/GTT decay state coefficients both accurately and efficiently, as it requires less local training epochs per round than FedProx, thus offering optimal performance with rare vertical communication between clients and cloud.

To concretely quantify the performance of the best FL scheme (FedAvg), Fig. 7 depicts the actual (x-axis) and predicted (y-axis) values of the GT Compressor decay state coefficient and GT Turbine decay state coefficient as scatter plots. Both target variables were accurately predicted with a prediction error less than 0.004.

B. USE CASE 2: PREDICTION OF SHIP'S MAIN ENGINE CONDITION

In order to classify the condition of the ship's main engine, we used a data-set from [30] which contains 6 numerical engine-related features and the binary PEC variable (target).

1) TUNING OF LEARNING PARAMETERS

As previously, we firstly fine-tuned the four FL schemes in terms of the crucial learning hyperparameters. The tuning procedures were identical to those presented for use case 1. The goal was to obtain optimally-configured FL schemes (FedSGD, FedAvg, FedAvgM and FedProx) prior the comparisons by properly regulating the hyperparameters towards accurate PEC prediction.

For each FL policy, we conducted extensive simulations with different server/client optimizers, different learning rates $\alpha = [0.001, 0.01, 0.1, 1]$. We also tested the convergence performance of FedAvgM and FedProx for different momentum constants ($\beta = 0.1, 0.3, 0.5, 0.7, 0.9, 0.96$) and proximal strength ($\mu = 0.1, 0.3, 0.5, 0.7, 0.9$). Table 4 lists the optimal values of the learning hyperparameters.

The evaluation metric used for assessing the classification error was the BCE, which quantifies the ability of the models to correctly classify the samples. As depicted in Fig. 8, there were different optimizer selections that correspond to low prediction error, depending on the FL scheme. To effectively predict the PEC testing values, FedSGD and FedAvgM require the Adam (at the cloud server side) and Adamax (at the clients side), respectively. The optimal selections

TABLE 4. Hyperparameter configuration for the four FL schemes considered in the comparisons of use case 2.

Parameter	**FedSGD**	**FedAvg**	**FedAvgM**	**FedProx**
Cloud/Client optimizer	Adam/ SGD	Adamax/ Adamax	SGD/ Adamax	Adam/ Adamax
Cloud/Client learning rate α	0.1/ 0.001	1/ 0.01	0.1/ 0.001	0.1/ 0.001
Cloud/Client momentum β	1/ 1	1/ 1	0.96/ 1	1/ 1
Proximal strength μ	0	0	0	0.1
Local epochs per client	10	15	15	15

for server/clients optimizers were Adamax/Adamax (for FedAvg) and Adam/Adamax (for FedProx). Note that the BCE in Fig. 8 has been computed over the combined testing set (i.e. the concatenated testing sets of the five training ships), given that each ship-specific dataset had been separated into 80%/20% training/testing subsets.

To investigate the impact of the number of local epochs per round, Fig. 9 shows the BCE for varying values of the local training epochs. All schemes presented optimal classification accuracy for 15 local training epochs per aggregation round, except FedSGD which showed the best performance for 10 epochs per round.

For the rest of the comparisons concerning use case 2, all FL schemes have been set to their optimal parameters according to Table 4 to ensure fair comparative analysis.

2) PERFORMANCE COMPARISON

This subsection presents the comparison amongst the FL policies in classifying the engine condition as normal or abnormal. For the comparisons, using the parameters listed in Table 4, we measured the classification of the FL schemes, as BCE computed over the data samples of the sixth ship. This means that, each FL scheme, which was trained according the federation across the five ship-specific datasets, was inferred by the samples (the 6 input features) of the sixth ship, whereas the BCE was computed based on the actual (as included in the dataset of the sixth ship) and FL model-predicted PEC values.

As shown in Fig. 10, the learning curve of the FedAvg converges to the lowest BCE scores relative to the rest of the FL schemes. This implies that, when considering PEC prediction PdM problems in a federated or collaborative learning scenario, FedAvg would be the optimal scheme, noticing a classification BCE of 0.075 on unseen data. Noteworthy, in this use case, all methods presented also have low communication-wise footprint, since the local weights are uploaded to the server quite rarely (every 15 local epochs).

To further quantify the performance of the FedAvg in this use case, Fig. 11 depicts the confusion matrix, which was obtained by inferring the FedAvg model with the sixth ship's input data. It is evident that 34.04% of the samples corresponding to the 'normal' PEC class were correctly classified, and 62.91% of the 'abnormal' samples were also correctly classified. This outcome results into an overall classification accuracy 96.95% for FedAvg model or, in other words, we conclude that FedAvg exhibits a false positive/negative ratio of 3.05% in the PEC prediction.

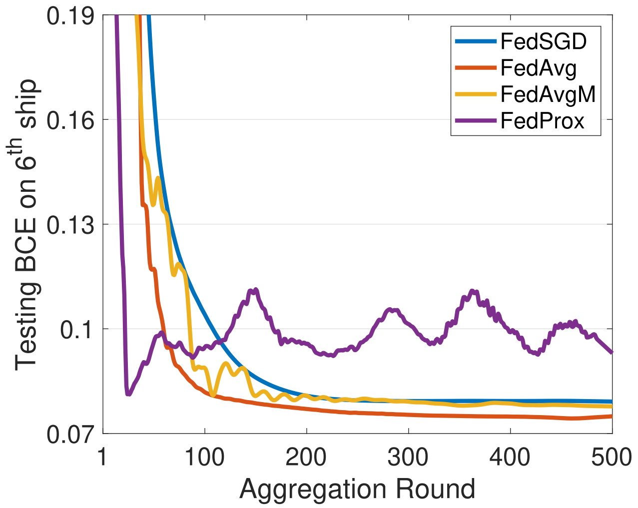


FIGURE 10. Learning convergence of the four FL schemes as a function of the training rounds. In each marker of the curves, the evaluation metric is the Binary Cross-Entropy (BCE) computed over the testing set.

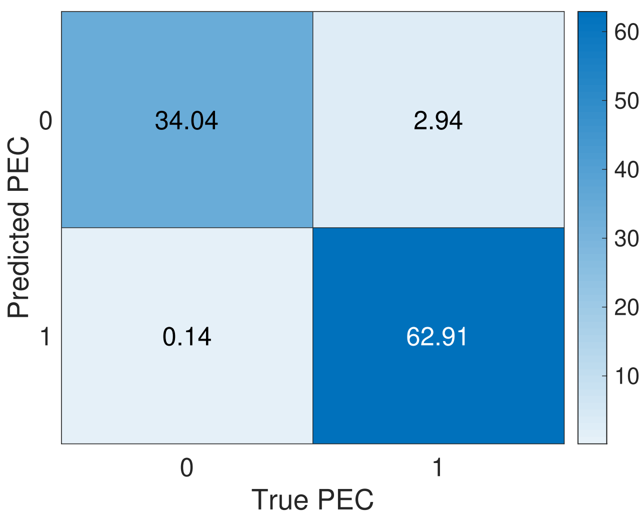


FIGURE 11. Confusion matrix illustrating the classification performance of the FedAvg scheme (Server/Client optimizers are Adamax/Adamax) in the prediction of the engine condition.

C. USE CASE 3: PREDICTION OF UPCOMING MAIN ENGINE CONSUMPTION

In this use case, we used a real dataset with AIS data to define and solve a 9-variable regression problem, whose solution provides forecasts of the upcoming PEPP over time, under the principles of FL (see Table 2 for the feature list). The dataset contained AIS data for 6 twin cargo ships, including multiple trajectories of the ships. Firstly, we reduced the number of the input features to six, based on objective criteria as: (i) heading variable was excluded since it is independent on the PEPP, (ii) DA and DF were also neglected because they are

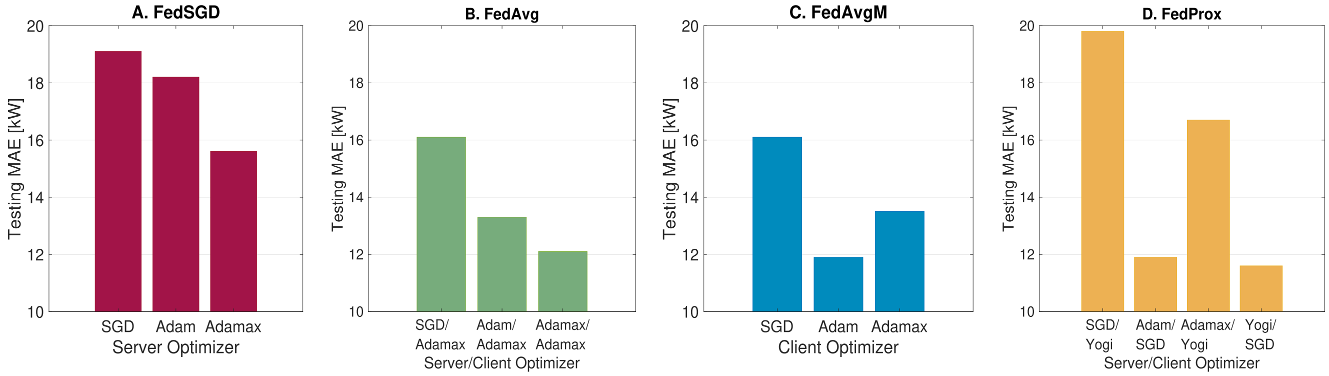


FIGURE 12. Use Case 3 (Prediction of upcoming main engine consumption) predictive performance of the four FL schemes for different optimizer configuration at both server and client sites. Panels A-D correspond to FedSGD, FedAvg, FedAvgM and FedProx, respectively. The evaluation metric is the MAE loss between the actual and the predicted sample values drawn from the testing set.

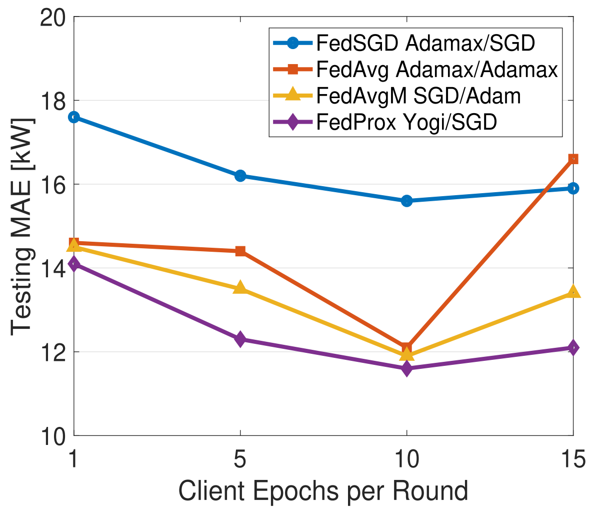


FIGURE 13. The impact of the number of Epochs per Round on the FL performance in use Case 3. The Mean Absolute Error (MAE) loss of FedSGD, FedAvgM, FedAvg and FedProx schemes are depicted, respectively. All schemes are configured with their optimal optimizer, whereas MAE metrics are calculated on the testing samples.

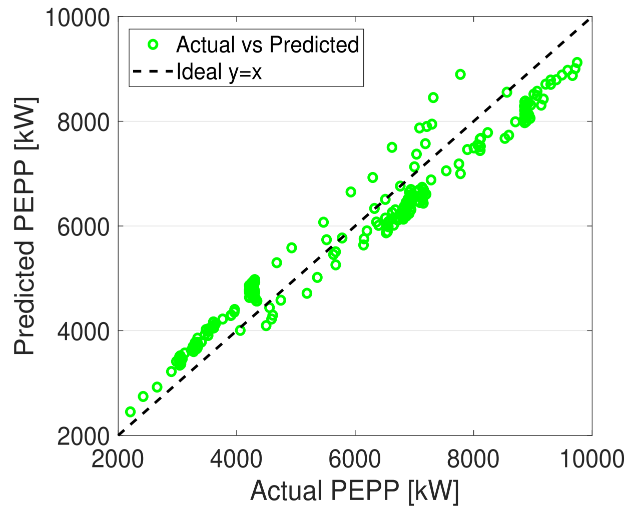


FIGURE 14. Actual and predicted values [in kW] of the testing samples of the Primary Engine's Propulsion Power (PEPP) as scatter plots. Predictions refer to the FedProx algorithm. The dashed line represents the ideal 'actual vs predicted' matching ($y = x$).

correlated with the Trim by definition (Trim $=$ DF $-$ DA), and (iii) from the two versions of the wind speed (DWS and CWS), we kept only DWS. As such, we considered only six features (SOG, STW, DWS, WD, Trim), as inputs and PEPP, as the target output to solve the timeseries forecasting problem. Specifically, for any given time instance t, the goal of each local model was to predict the upcoming PEPP value (at $t+1$), based on the 5 previous values of each feature (i.e. $\mathrm{SOG}_{t-4,t-3,\ldots,t}$, $\mathrm{STW}_{t-4,t-3,\ldots,t}$, $\mathrm{DWS}_{t-4,t-3,\ldots,t}$, $\mathrm{WD}_{t-4,t-3,\ldots,t}$, $\mathrm{Trim}_{t-4,t-3,\ldots,t}$) (see Fig. 3). Notably, PEPP values have been collected every 15 minutes, thus PEPP_{t+1} refers to the upcoming engine's propulsion power in the next 15 minutes (after the current time instance t). Each FL participant trained an LSTM network, which was chosen because in general, it can model linear and non-linear complex functions between inputs and expected outputs, and it can deal with timeseries with temporal dependencies. A 80%/20% splitting into training/testing sets was adopted for each local dataset.

1) TUNING OF LEARNING PARAMETERS

Before comparing the four FL schemes considered in this study, a learning parameter tuning procedure was followed, as in the previous subsections. By training each of the FL schemes with varying values of the learning rate $\alpha = [0.0001, 0.001, 0.01, 0.1]$ and number of hidden layers $N_{layers} = [1, 2, 3, 4]$, we obtained an optimal MAE of the predictions for $\alpha = 0.001$ and $N_{layers} = 1$ (128 neurons), as shown in Fig. 3. Note that, the simulations regarding the learning rate and hidden layers tuning are not depicted for purposes of saving figure space. Also, for this subsection, the MAE refers to the mean of squared differences between the predicted and the actual PEPP values, with the latter referring to the combined testing set of all training ships (concatenated testing sets of the training ship datasets). Specially for

FedAvgM and FedProx, we ran similar simulation for tuning the momentum $\beta = [0.1, 0.3, 0.5, 0.7, 0.9, 0.96]$ and the proximal strength $\mu = [0.1, 0.3, 0.5, 0.7, 0.9]$, respectively. FedAvgM presented the lowest MAE for $\beta = 0.9$, whereas FedProx exhibited the lowest MAE for $\mu = 0.7$, implying that there was data heterogeneity amongst the local datasets (non-IID distributions of the features across local FL participants).

Fig. 12 shows the performance of each FL scheme, in terms of MAE for different server/client optimizers. The MAE was computed, as the mean absolute difference between the predicted and actual PEPP values, which were taken by a combined testing set. This combined testing set was comprised of all the testing sets of the ships (each ship-specific dataset had been divided into 80%/20% training/testing sets). It is evident that FedSGD performed optimally for Adamax at the server side, FedAvg for Adamax/Adamax at the server/clients sides, FedAvgM for Adam at the client side and FedProx for Yogi/SGD at the sever/clients sides.

Considering the optimal optimizer per FL scheme, we also assessed the impact of the number of local epochs per round, as depicted in Fig. 13. It is shown that, in this use case, the optimal local epochs per aggregation round was 10 for all schemes, with the FedProx outperforming the other methods, noticing a MAE of 11.6 kW. Note that, the presented MAE (11.6 kW) was considerably low with respect to the overall variance of the testing PEPP values ranging from 2000 to 13000 kW. In this sense, the MAE metrics are orders of magnitude lower than the total variance of the PEPP data, implying that the PEPP predictions are low-error. Notably, FedProx constantly outperformed the FedAvg scheme, which implicitly means that the local data distribution of the features presented some degree of heterogeneity across FL participants. Moreover, FedAvgM exhibited more accurate predictions than FedAvg regardless of the local epochs per round, but failed to exceed the FedProx, in terms of the PEPP prediction performance.

2) PERFORMANCE COMPARISON

To quantitatively evaluate the FedProx performance in forecasting the PEPP values under unknown conditions, we used the data of the sixth dataset belonging to the sixth ship. This dataset had not been encountered in the FedProx model training phase, thus being able to index how well the model generalize its predictions and whether it can be reused by new ships (transfer learning) for purposes of predicting their upcoming PEPP without retraining. Fig. 14 shows the relationship between the actual and the FedProx model-predicted PEPP values of the sixth ship's dataset. Note that, the perfect relation would be the line $y = x$, which would mean that the model produces the perfectly correct predictions. Evidently, we observed that FedProx was able to accurately forecast the PEPP values of the unknown dataset, showing semi-linear relationship between predicted and ground truth PEPP values, even considering totally unknown ship dataset. Note that, if the FedProx is softly retrained using a few training samples of the sixth ship's dataset, the accuracy would be higher relative to the worst-case results that we considered in Fig. 14 where no training with the sixth ship's data has been considered.

Overall, the results confirm that FedProx may be the most suitable FL scheme in cases that local data distributions present heterogeneity across FL agents, while also ensuring data privacy preservation (only model parameters are communicated), low communication footprint (since it requires quite rare aggregation rounds) between the cloud and the local ship clients and reusability of the model by new ships without hard retraining. The latter is in principle crucial in maritime applications when environmental footprint mitigation comes into play.

V. CONCLUSION AND FUTURE DIRECTIONS

Shipping 4.0 will revolutionize maritime operations by integrating advanced automation and digitization technologies. A major enabling technology in this field is ML-aided predictive maintenance, addressing inefficiencies, costs, and unexpected failures of conventional maintenance practices. This study focused on the use of FL in predictive maintenance use cases in the context of Shipping 4.0, relying on real datasets from the maritime industry. We evaluated and compared various FL algorithms across three maritime use cases were investigated, including regression for predicting naval propulsion gas turbine, classification for predicting ship engine condition, and time-series regression for predicting ship fuel consumption. The efficiency of the proposed federated learning-based predictive maintenance demonstrated its potential to enhance maintenance decision-making, reduce shipping industry downtime, and improve operational efficiency for smart ships. It is expected that our findings will contribute to the advancement of predictive maintenance methodologies in Shipping 4.0, offering valuable insights for maritime stakeholders to embrace federated learning as a viable and privacy-preserving solution. Moreover, it will facilitate model sharing within the shipping industry and encourage collaboration among maritime stakeholders.

This work can be extended in various directions, including:

1) Clustering of the federated learning agents before building the FL global model. This means that, an agent grouping (as a preliminary step) based on the features of the local agent datasets may improve the final FL-per-group performance, since agents with quasi-common data distributions and high similarity can converge to a homogeneous high-accuracy model.
2) Evaluation of additional use cases that are envisioned in Shipping 4.0 era, exploiting real datasets concerning dense maritime areas and diverse types of maritime nodes (UAVs, buoys, underwater vehicles,

port facilities) towards coordinating with the foreseen 6G communications.

3) Deployment of communication-efficient methods [35] with the aim of lowering the overhead required for model exchange rounds and reducing the energy consumption footprint of the FL schemes in the maritime sector (e.g. selection of nodes participating in the FL training, energy-efficient and bandwidth-aware methods for controlling the transmitting power of the maritime entities [36], [37]).

REFERENCES

[1] T. Xia, M. M. Wang, J. Zhang, and L. Wang, "Maritime Internet of Things: Challenges and solutions," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 188–196, Apr. 2020.

[2] W. Tao et al., "Coordination and optimization control framework for vessels platooning in inland waterborne transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 15667–15686, Dec. 2023.

[3] H. Yu, A. T. Murray, Z. Fang, J. Liu, G. Peng, M. Solgi, and W. Zhang, "Ship path optimization that accounts for geographical traffic characteristics to increase maritime port safety," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5765–5776, Jun. 2022.

[4] N. Nomikos, P. K. Gkonis, P. S. Bithas, and P. Trakadas, "A survey on UAV-aided maritime communications: Deployment considerations, applications, and future challenges," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 56–78, 2023.

[5] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140699–140725, 2020.

[6] O. A. Wahab, A. Mourad, H. Otrok, and T. Taleb, "Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1342–1397, 2nd Quart., 2021.

[7] A. Giannopoulos, P. Gkonis, P. Bithas, N. Nomikos, G. Ntroulias, and P. Trakadas, "Federated learning for maritime environments: Use cases, experimental results, and open issues," *TechRxiv*, Feb. 2023.

[8] P. Trakadas, N. Nomikos, E. T. Michailidis, T. Zahariadis, F. M. Facca, D. Breitgand, S. Rizou, X. Masip, and P. Gkonis, "Hybrid clouds for data-intensive, 5G-enabled IoT applications: An overview, key issues and relevant architecture," *Sensors*, vol. 19, no. 16, p. 3591, Aug. 2019.

[9] A. Angelopoulos, E. T. Michailidis, N. Nomikos, P. Trakadas, A. Hatziefremidis, S. Voliotis, and T. Zahariadis, "Tackling faults in the Industry 4.0 era—A survey of machine-learning solutions and key aspects," *Sensors*, vol. 20, no. 1, p. 109, Dec. 2019.

[10] G. Aiello, A. Giallanza, and G. Mascarella, "Towards shipping 4.0. A preliminary gap analysis," *Proc. Manuf.*, vol. 42, pp. 24–29, Jan. 2020.

[11] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM J. Res. Develop.*, vol. 44, nos. 1–2, pp. 206–226, Jan. 2000.

[12] S. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, pp. 1345–1359, Nov. 2010.

[13] H. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, Apr. 2017, pp. 1–10.

[14] A. Giannopoulos, N. Nomikos, G. Ntroulias, T. Syriopoulos, and P. Trakadas, "Maritime federated learning for decentralized on-ship intelligence," in *Artificial Intelligence Applications and Innovations*, I. Maglogiannis, L. Iliadis, J. MacIntyre, and M. Dominguez, Eds. Cham, Switzerland: Springer, Jun. 2023, pp. 195–206.

[15] K. Skianis, A. Giannopoulos, P. Gkonis, and P. Trakadas, "Data aging matters: Federated learning-based consumption prediction in smart homes via age-based model weighting," *Electronics*, vol. 12, no. 14, p. 3054, Jul. 2023.

[16] P. Boobalan, S. P. Ramu, Q.-V. Pham, K. Dev, S. Pandya, P. K. R. Maddikunta, T. R. Gadekallu, and T. Huynh-The, "Fusion of federated learning and industrial Internet of Things: A survey," *Comput. Netw.*, vol. 212, Jul. 2022, Art. no. 109048.

[17] H. T. Truong, B. P. Ta, Q. A. Le, D. M. Nguyen, C. T. Le, H. X. Nguyen, H. T. Do, H. T. Nguyen, and K. P. Tran, "Light-weight federated learning-based anomaly detection for time-series data in industrial control systems," *Comput. Ind.*, vol. 140, Sep. 2022, Art. no. 103692.

[18] S. Becker, K. Styp-Rekowski, O. V. L. Stoll, and O. Kao, "Federated learning for autoencoder-based condition monitoring in the industrial Internet of Things," in *Proc. IEEE Int. Conf. Big Data*. Los Alamitos, CA, USA: IEEE Computer Society, Dec. 2022, pp. 5424–5433.

[19] S. Bharti and A. Mcgibney, "Privacy-aware resource sharing in cross-device federated model training for collaborative predictive maintenance," *IEEE Access*, vol. 9, pp. 120367–120379, 2021.

[20] T. Ranathunga, A. McGibney, S. Rea, and S. Bharti, "Blockchain-based decentralized model aggregation for cross-silo federated learning in Industry 4.0," *IEEE Internet Things J.*, vol. 10, no. 5, pp. 4449–4461, Mar. 2023.

[21] Z. Zhang, C. Guan, H. Chen, X. Yang, W. Gong, and A. Yang, "Adaptive privacy-preserving federated learning for fault diagnosis in Internet of Ships," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6844–6854, May 2022.

[22] T. Tinga, W. Tiddens, F. Amoiralis, and M. Politis, "Predictive maintenance of maritime systems: Models and challenges," in *Proc. Eur. Saf. Rel. Conf. (ESREL)*, Jun. 2017, pp. 421–429.

[23] G. Makridis, D. Kyriazis, and S. Plitsos, "Predictive maintenance leveraging machine learning for time-series forecasting in the maritime industry," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–8.

[24] M. Cheliotis, I. Lazakis, and G. Theotokatos, "Machine learning and data-driven fault detection for ship systems operations," *Ocean Eng.*, vol. 216, Nov. 2020, Art. no. 107968.

[25] A. Bakdi, N. B. Kristensen, and M. Stakkeland, "Multiple instance learning with random forest for event logs analysis and predictive maintenance in ship electric propulsion system," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 7718–7728, Nov. 2022.

[26] V. J. Jimenez, N. Bouhmala, and A. H. Gausdal, "Developing a predictive maintenance model for vessel machinery," *J. Ocean Eng. Sci.*, vol. 5, no. 4, pp. 358–386, Dec. 2020.

[27] S. Liu, H. Chen, B. Shang, and A. Papanikolaou, "Supporting predictive maintenance of a ship by analysis of onboard measurements," *J. Mar. Sci. Eng.*, vol. 10, no. 2, p. 215, Feb. 2022.

[28] F. Cipollini, L. Oneto, A. Coraddu, A. J. Murphy, and D. Anguita, "Condition-based maintenance of naval propulsion systems with supervised data analysis," *Ocean Eng.*, vol. 149, pp. 268–278, Feb. 2018.

[29] A. Coraddu, L. Oneto, A. Ghio, S. Savio, D. Anguita, and M. Figari, "Condition based maintenance of naval propulsion plants," *UCI Mach. Learn. Repository*, Oct. 2014, doi: 10.24432/C5K31K.

[30] D. Mohakul, "Predictive maintenance on ship's main engine using AI," *IEEE Dataport*, Nov. 2022.

[31] M. Zinkevich, M. Weimer, L. Li, and A. Smola, "Parallelized stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, Jan. 2010, pp. 1–9.

[32] T.-M. Harry Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," 2019, *arXiv:1909.06335*.

[33] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proc. Mach. Learn. Syst.*, vol. 2, pp. 429–450, Mar. 2020.

[34] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.

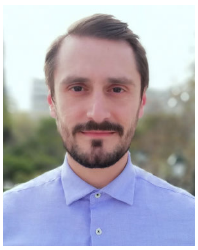
[35] N. Nomikos, A. Giannopoulos, P. Trakadas, and G. K. Karagiannidis, "Uplink NOMA for UAV-aided maritime Internet-of-Things," in *Proc. 19th Int. Conf. Design Reliable Commun. Netw. (DRCN)*, Apr. 2023, pp. 1–6.

[36] C. Tsinos, S. Spantideas, A. Giannopoulos, and P. Trakadas, "Over-the-air computation with quantized CSI and discrete power control levels," *Wireless Commun. Mobile Comput.*, vol. 2023, pp. 1–8, Nov. 2023.

[37] S. T. Spantideas, A. E. Giannopoulos, N. C. Kapsalis, A. Kalafatelis, C. N. Capsalis, and P. Trakadas, "Joint energy-efficient and throughput-sufficient transmissions in 5G cells with deep Q-learning," in *Proc. IEEE Int. Medit. Conf. Commun. Netw. (MeditCom)*, Sep. 2021, pp. 265–270.

ANGELOS ANGELOPOULOS received the bachelor's degree in electrical engineering from the Technological Educational Institute of Chalkida, in 2012, and the M.Sc. degree in the intelligent management of renewable energy systems from the Technological Educational Institute of Central Greece, in 2017. He is currently pursuing the Ph.D. degree with the General Department, National and Kapodistrian University of Athens, with a focus on artificial intelligence and more particularly on employing zero defects in industrial manufacturing by applying artificial intelligence methods. He is currently working on machine learning and deep learning algorithms, with a focus on federated learning. His research interests include concept drift and assisted learning.

ANASTASIOS GIANNOPOULOS (Member, IEEE) received the Dipl.-Ing. degree in electrical and computer engineering and the Ph.D. degree from the National Technical University of Athens (NTUA). Currently, he is an Electrical and Computer Engineering Researcher with the Department of Ports Management and Shipping, National and Kapodistrian University of Athens. He has more than 35 scientific publications. His research interests include wireless network optimization, machine learning, maritime communications, and multidimensional analysis.



NIKOLAOS NOMIKOS (Senior Member, IEEE) received the Diploma degree in electrical engineering and computer technology from the University of Patras, Greece, in 2009, and the M.Sc. and Ph.D. degrees from the Information and Communication Systems Engineering Department, University of the Aegean, Samos, Greece, in 2011 and 2014, respectively. He is currently a Senior Researcher with the Department of Ports Management and Shipping, National and Kapodistrian University of Athens. His research interests include 6G communications, NOMA, and machine learning-aided wireless networks. He is a member of the IEEE Communications Society and the Technical Chamber of Greece. He is an Associate Editor of *Frontiers in Communications and Networks*.

ALEXANDROS KALAFATELIS (Graduate Student Member, IEEE) received the B.Sc. (Hons.) degree in biomedical sciences from the University of East London and the B.Eng. degree in electrical engineering and the M.Sc. degree in intelligent management of renewable energy systems from the National and Kapodistrian University of Athens (NKUA). He is currently pursuing the Ph.D. degree with the Department of Ports Management and Shipping, NKUA. Since 2021, he has been a Research Scientist and an Engineer with FOUR DOT INFINITY. Part of his research has been conducted in the framework of several European-funded research and development projects. His current research interests include predictive maintenance applications utilizing federated learning, focusing on the development of secure aggregations tailored for the maritime industry.

ANTONIOS HATZIEFREMIDIS received the degree in electrical and computer engineering and the Ph.D. degree from the National Technical University of Athens. He is currently a Professor with the Department of Aerospace Science and Technology. His research interests include the design and development of optical logic devices, photonic signal processing subsystems for optical networks, OTDM/DWDM high-speed optical sources, wireless optical communications, satellite applications, telecommunications, and unmanned aerial vehicles.

PANAGIOTIS TRAKADAS received the Dipl.-Ing. degree in electrical and computer engineering and the Ph.D. degree from the National Technical University of Athens (NTUA). In the past, he was worked at Hellenic Aerospace Industry (HAI), as a Senior Engineer, on the design of military wireless telecommunications systems, and the Hellenic Authority for Communications Security and Privacy, where he held the position of the Director of the Division for the Assurance of Infrastructures and Telecommunications Services Privacy. He is currently an Associate Professor with the National and Kapodistrian University of Athens. He has been actively involved in many EU FP7 and H2020 research projects. He has published more than 130 papers in magazines, journals, and conference proceedings. His research interests include the fields of wireless and mobile communications, wireless sensor networking, network function virtualization, and cloud computing. He is a Reviewer of several journals, including IEEE Transactions on Communications and IEEE Transactions on Electromagnetic Compatibility journals.