Maritime Federated Learning for Decentralized On-ship Intelligence

Anastasios Giannopoulos[1], Nikolaos Nomikos[1], Georgios Ntroulias[2], Theodoros Syriopoulos[1], and Panagiotis Trakadas[1]

[1] National and Kapodistrian University of Athens, Psachna, Evia, 34400, Greece
angianno@uoa.gr, nomikosn@pms.uoa.gr, tsiriop@pms.uoa.gr, ptrakadas@uoa.gr

[2] Hydrus Engineering S.A., Agia Paraskevi, Athens, 15343, Greece
g.ntroulias@hydrus-eng.com

Abstract. Maritime trade unavoidably influences the economy, transportation and market worldwide. Deploying efficient, privacy-preserving and environmental-friendly solutions in maritime environments requires a continuous global effort. There are many obstacles associated with the centralized optimization for on-sail ships, including data security violation, signaling overhead and high-latency response. Contradictorily, the emerging Federated Learning (FL) paradigm has been proposed as an efficient solution for promoting data privacy, low latency and high communication efficiency. In this context, this paper proposes a maritime-based FL scheme to ensure distributed, collaborative and secure optimization for enabling predictive on-ship intelligence. A scenario targeting the accurate prediction of the Primary Engine Power (PEP) of large cargo ships is considered, as the output PEP is directly proportional to fuel consumption. Based on real data, several Machine Learning (ML) models were tested and validated, in terms of their PEP prediction accuracy, with all models exploiting both weather- and ship-related data. Simulation results towards the selection of the PEP predictor, as well as the quantification of the FL scheme efficacy over traditional benchmarks were carried out, achieving a beneficial equilibrium between the prediction accuracy, data privacy and communication efficiency.

Keywords: Distributed Intelligence, Federated Learning, Fuel Consumption, Machine Learning, Maritime Communication Network.

1 Introduction

Maritime trade is responsible for carrying out the majority of global trade and a major driver of economic growth. Modern maritime systems necessitate significant improvements in energy efficiency and environmental-awareness with considerable reduction of their CO2 emissions. These requirements become more stringent due to the "Fit for 55" agreement, according to which, 173 countries declare to reduce emissions by 2050 produced by transport, buildings, agriculture and waste by about 55%, relative to the total emissions of 2008 [1]. Taking seriously this agreement and considering the maritime sector as one of the primary sectors being responsible for CO2 emissions, the maritime industry should be transformed in several cross-coupled domains, including the transportation [2], communication and intelligence layers [3].

A concerted global effort has been observed in developing novel solutions, being required in the domains of communication and intelligence. In this context, the introduction of advanced services and technologies, such as the Internet of Things (IoT), is expected to leverage optimum transportation solutions due to the huge amount of data that are collected, stored, and processed [3]. Hence, real-time monitoring and automated optimization procedures are expected to play a key role in future maritime transportation systems for ship path optimization or energy consumption-dependent and speed optimization purposes [4]. To this end, traditional optimization approaches might not be effective, since in general, they lead to non-convex problems, which are solved via iterative optimization approaches.

IoT solutions in the maritime sector include the transmission and processing of a large amount of heterogeneous data under harsh propagation conditions [5]. Therefore, reliable maritime communication and networking design becomes a critical and challenging task, due to the multiple constraints that need to be taken into consideration. The goal of this design should be the optimization of certain performance metrics (i.e., overall data throughput, network outages) with reduced transmission power and signaling burden [6]. In the vast majority of related works, the initial non-convex problem is decomposed into a discreet number of convex problems and solved via iterative optimization approaches. Still, performance loss is inherently introduced, while the frequent execution of iterative algorithms has a negative impact on the overall system response times [7]. Therefore, relying on conventional optimization to achieve ultra-reliable low latency communications (URLLC), being needed in maritime transportation applications is highly questionable.

Meanwhile, Machine learning (ML) comprises an ever-growing domain, mainly because it provides a powerful solution towards large-scale constrained optimization and predictive capabilities [8]. Most of the ML algorithms typically employed in the maritime sector require huge amounts of historically collected data. In order to ensure proper data processing and analytics, ML models are trained on a dataset to provide alarms, predictions or corrective actions, given a particular optimization task [9]. ML is classified into 3 main categories, namely (i) Supervised Learning (SL) in which labeled data are available as groundtruth to guide the model parameter adjustment during the training. SL algorithms are further divided, based on the type of the labels: classification refers to categorical labels, whereas regression refers to scalar-valued labels; (ii) Unsupervised Learning (UL) in which the goal is to identify hidden clusters with high feature similarity or to reduce the dimensions of the data through the usage of linear combinations of the features (i.e. components); (iii) Reinforcement Learning (RL) algorithms in which a cognitive agent finds an optimal decision-making policy through trial-and-error interactions with the maritime environment, aiming to maximize a reward (e.g., best possible behavior or path) in a specific situation [10].

However, there are several obstacles towards enabling ML functionalities in the maritime sector [3]. ML models require huge amounts of data in order to provide efficient solutions and predictions. From the maritime perspective, since the transmission and exchange of multi-source data in such an environment can be characterized by a high degree of heterogeneity, attributed to the distances that the components of a signal have to cover and the diverse transmission conditions, traditional centralized ML seems quite problematic. Moreover, centralized data collection raises privacy violation issues, considering that the training carries commercial sensitive information that requires protection against malicious attacks. Since the demanding transmissions and exchange of data, as well as privacy violation in such environments are mainly the case in centralized schemes, the solution of distributed cooperative ML is preferred.

Contrary to the aforementioned drawbacks, Federated Learning (FL) has been widely promoted, as an effective collaborative learning scheme [11]. FL training process can ensure fast ML execution, no need for data transmissions and privacy preservation of maritime sensitive data. In FL, maritime nodes exploit shared models trained from excessive amounts of data, without the need to centrally store it. In this context, instead of constantly updating a central node with data, the training of ML models is performed locally, and then, the global model is periodically updated. Therefore, this method can lead to four very important advantages: (i) reduction of the signaling load, (ii) reduction of the wireless transmission power, (iii) protection of security and privacy and (iv) enhanced model sharing abilities between maritime nodes, given the global observability of the FL-constructed global model that is finally deployed.

The present paper outlines the implementation and comparison of FL methodology against various ML benchmark schemes. Using a labeled commercial dataset from a maritime enterprise, a scenario focusing on the prediction of the Primary Engine Power (PEP) of large cargo ships is considered. As PEP is proportional to the fuel oil consumption, we target to demonstrate the potency of FL-assisted methods towards collaborative PEP prediction. Noteworthy, weather-related and on-sail ship-related features were used to accurately estimate the PEP required for the ship propulsion. To obtain optimal performance of FL, multiple SL ship-specific models were compared in the single-ship datasets, for the purpose of finding the best PEP regressor. The advantages and disadvantages of using FL against baseline methods are also identified. In summary, the contributions of this work can be identified as:

(i) The exploitation of historical knowledge of both weather- and ship-related features, extracted by real data from a maritime enterprise, to obtain accurate and ship-specific supervised learning models,

(ii) The proposition of a general-purpose FL-based architecture to enable privacy-preserving and low-latency predictive abilities for on-sail cargo ships.

(iii) The adoption of an end-to-end processing pipeline to implement FL in the maritime sector, including the training, testing and comparison of multiple regression models to properly find the best regressor for each ship.

(iv) The quantitative comparison of FL against centralized and decentralized team learning methods in maritime setups.

(v) The limitations of centralized ML solutions are tackled, facilitating the integration of ML in the maritime industry by addressing the privacy concerns of the involved stakeholders while avoiding excessive transmission overheads in the operation of the FL solution.

2 Architectural and Technical Methodology

2.1 Maritime Federated Learning

In this section, a three-tier Maritime Federated Learning (MFL) architecture is proposed (see Fig. 1). MFL is an adaptation of FL principles, properly embodied within a maritime setup, so as to provide an efficient handling of distributed ship-dependent datasets and to decouple model training from excessive transmissions to the central server. In the considered MFL scenario, local ship clients can be seen as common-goal agents, aiming to converge in an adequate solution of a given learning task. This process takes place with the supervision and collaboration of the central server, which is responsible for aggregating knowledge from all agents. Ship-specific data remain localized; hence, any data privacy issue can be met only in isolated ship sites, not affecting the federation.

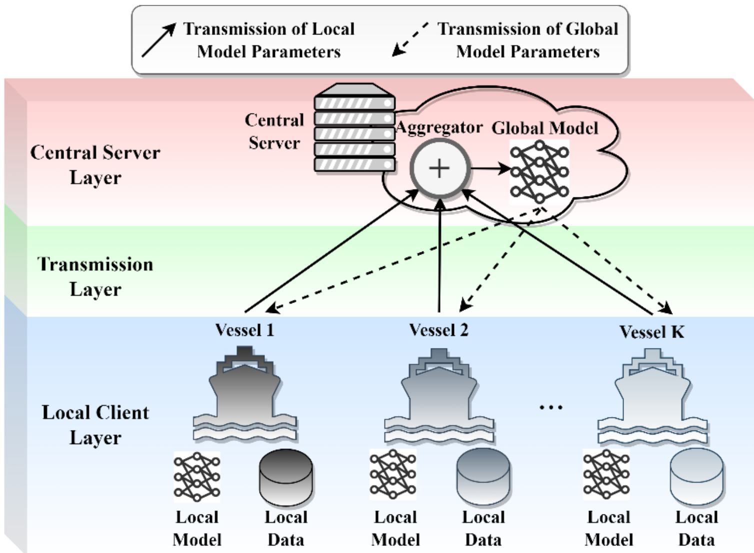

Fig. 1. Three-tier Federated Learning Architecture for on-sail maritime intelligence. Local Models are trained in a decentralized manned, based on local ship-specific data. Only model parameters are exchanged through the Transmission layer. The Central Server layer aggregates the local models before sending the averaged global model back to the local ship agents.

Driven by the aforementioned process, MFL can offer a unified distributed ML platform, allowing local models to exploit information observed and processed by other learners using the global model. Moreover, incoming new models can directly grasp the knowledge obtained by the global model, after long-term federated training among existing clients. MFL can also enable learning, based on non-independent and identical distributed (non-i.i.d) sets of data [11]. This attribute can be extremely beneficial in on-sail multi-ship collaborative learning, since different types of ships with diverse sizes of available datasets can also be harmonized according to MFL. Note that, although here, we illustrate only neural networks as local/global models, MFL models can be of any type, spanning from decision tree, logistic regression and ensemble models to recurrent and convolutional neural networks.

2.2 Cargo Ship Dataset

To train the ML schemes deployed in this study, a 1-year duration dataset of two twin cargo ships was available. For each cargo, ten time-varying variables are captured with a sampling frequency of 1 sample per minute. The temporal course of the variables refers to multiple voyages of the ships. The data were sampled from 01-07-2021 to 30-06-2022, resulting in 514,525 (for ship 1) and 525,005 (for ship 2) total instances per variable. Specifically, the dataset contained measures about both ship's on-sail features and wind characteristics which are detailed below:

i. *Speed over ground (SOG):* It quantifies the ship's speed in relation to the surface of the earth and is measured in knots.
ii. *Speed through water (STW):* Different from SOG, STW represents the ship's speed in relation to the water currents and is measured in knots.
iii. *Heading:* It denotes the actual on-sail direction in which the ship is pointing and it is expressed, as the angular distance relative to north (0°), clockwise through 359°.
iv. *Continuous wind speed (CWS):* It is the wind strength expressed in m/s.
v. *Discretized wind speed (DWS):* It is the discretized version of CWS, quantized according to the Beaufort scale (bft). DWS can be considered as a categorical variable ranging from 0 to 12 in steps of 1 and mapping with the 13 wind classes or weather flags (from "calm" to "hurricane-force").
vi. *Wind direction (WD):* It represents the true wind direction in relation to the ship's on-board heading. Thus, WD is expressed in degrees and varies between 0°-360° (wind on the bow at 0°, wind on the beam at 90°, wind on the stern at 180°).
vii. *Draft forward (DF):* The draft forward (bow) is the ship's depth in meters measured at the perpendicular of the bow based on predefined depth levels.
viii. *Draft aft (DA):* The draft aft (stern) is the ship's depth in meters, measured at the perpendicular of the stern, based on predefined depth levels.
ix. *Trim:* The trim is simply the difference between the DF and DA (depth in meters), relative to the designed waterline located at the middle of the ship. It determines the minimum depth of water a ship can safely navigate.
x. *Primary engine power (PEP):* It is the aggregated power supplied by the prime mover(s) which are responsible to provide useful propulsion to the ship and is expressed in kW. Obviously, PEP is positively correlated with the electricity requirements of the ship's diesel engine, therefore being directly proportional to the fuel oil consumption. Additionally, PEP has also bi-directional causal relationship with the ship's SOG and STW, as well as it is strongly affected by wind conditions, sea currents and ship's drafts.

The ML methodology presented below exploits the labelled dataset, considering the PEP as the target variable (model output) and the rest of the measures as potential predictors (or model inputs). This will allow the adoption of an accurate training and verification process of the developed ML model by measuring the error between the model-predicted outputs and the actual historical values of the target variable.

2.3 Machine Learning-based Pipeline for PEP Prediction

To ensure high signal-to-noise ratio and sufficient performance of the final-stage collaborative learning schemes, a gradual 3-phase pipeline was adopted, including the data preparation, the ship-specific local model selection and the comparison among collaborative methods for multi-ship PEP prediction (see Fig. 2).

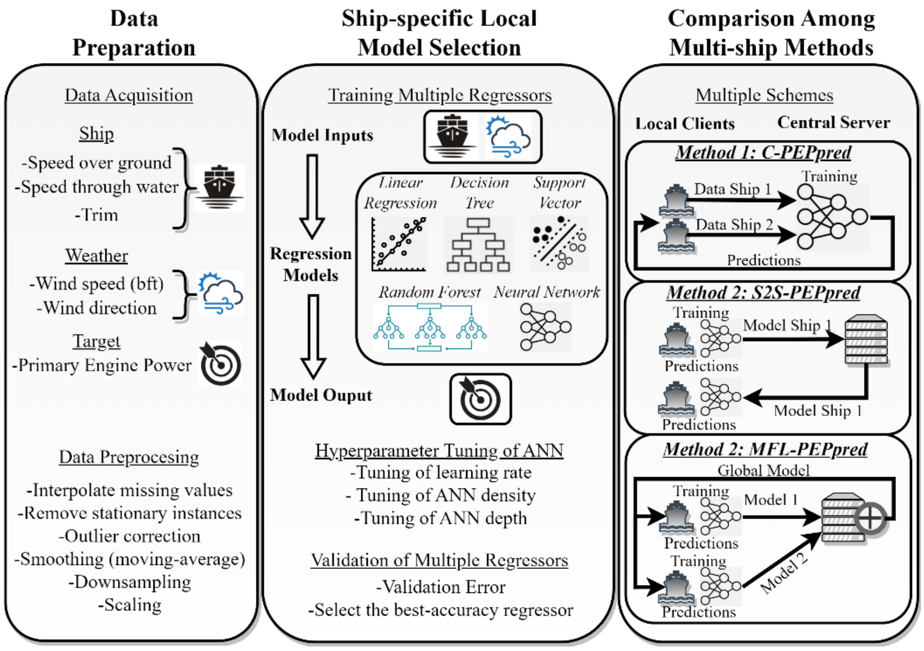

Fig. 2. Proposed pipeline for Collaborative Learning of ship's Primary Engine Power.

1. Data Preparation: In this phase, all the preliminary steps required to reduce the model dimensionality and mitigate the data size and noise were performed. Initially, a set of multi-collinearity criteria were investigated towards avoiding feature redundancy and mitigating the model input dimensions. As an independent factor of the ship's fuel consumption, Heading parameter was avoided from the regression inputs. In addition, both types of vessel drafts (DF and DA) were also ignored, given that retaining only their difference (Trim parameter) does not imply any information loss. Moreover, only one version of the wind speed was selected (i.e. DWS in bft) to reflect the wind strength. Noteworthy, since STW indirectly carries information about sea currents (whereas SOG is the ground-referenced speed), both types of speed (SOG and STW) were kept as model inputs. Then, the missing values were replaced, following an average-interpolation calculation, based on the previous and the next available samples. The final number of interpolated values was not more than 8% of the total samples per variable. Instances corresponding to near-zero vessel speed (SOG<0.2 knots) were also ditched. Afterwards, the moving-average filter was applied to slightly smooth the data and eliminate the aliasing-caused distortion that will follow upon downsampling. Given the slow variations observed in all time-series, a downsampling operation was applied to ensure a sampling frequency of 1 sample per 15 minutes. Finally, the values of each variable were scaled in the range [0,1] using a Min-Max scaling operation.

2. Ship-specific local model selection: In principle, the relationship between the ship's PEP and the model inputs (SOG, STW, DWS, WD and Trim) is unknown and it can be either linear or non-linear. ML algorithms can be employed to create a local regression model, using historical instance-specific features as inputs and the respective PEP values as labels. To reveal the best ship-specific regressor, several ML (linear and non-linear) regression algorithms were trained and tested, namely:

i. *Multiple Linear Regression (MLR)* assumes that a weighted linear combination of N independent features can predict the target. The resulting model is linear with coefficients w_i $(i = 1, 2, \dots N)$, graphically demonstrated by the best-fitting straight line, showing the minimum mean squared error between the real and MLR-predicted values (calculated over validation samples).

ii. *Decision Tree Regression (DTR)* is a non-parametric ML model which produces a rule-based tree by splitting the feature space properly, so as to guarantee low information entropy in each decision area. Then, using the separated decision areas as tree leafs, it constructs several rule paths, arriving at terminal leafs according to the prior feature space splitting rules. Each leaf predicts the average target value, computed across the target values of all samples, included in the respective decision area.

iii. *Support Vector Regression (SVR)* uses an ε-insensitive tube to allow for more flexibility in errors. Unlike *Linear Regression*, *SVR* penalizes only the data points that have at least ε distance from the fitting curve. The target of the SVR is to fit the best line within a threshold value (ε), reflecting the distance between the hyperplane and boundary line.

iv. *Random Forest Regression (RFR)* is an ensemble learning version of DTR. By splitting the dataset in N segments, multiple segment-specific DTRs are constructed. Given a new sample, the RFR predicts the average prediction, computed across all the individual trees.

v. *Artificial Neural Network (ANN)* is a supervised non-linear learning scheme of sequentially-stacked and fully-connected layers. Each layer consists of multiple units, each one calculating the weighted sum of all the previous-layer outputs and then applying an activation function (e.g. ReLu, SoftMax). ANN properly tunes the connections between units (i.e. weights), so as to minimize the prediction error (or loss function).

3. Collaborative methods for multi-ship PEP prediction: Three schemes were contrasted in terms of the provided PEP prediction accuracy preparation, including one centralized and two decentralized collaborative schemes, namely:

i. *Centralized PEP prediction (C-PEPpred)* is the traditional method to deploy ML models. Being sensitive to malicious attacks, C-PEPpred simply collects all the data in the central server, before building a powerful heavy model. Both ships can predict future PEP values upon request on the central model. Note that, this method can have near-optimal PEP estimation error, since the central model has been trained on the multi-ship dataset, thus obtaining global environment observability.

ii. *Ship-to-ship model transfer for PEP prediction (S2S-PEPpred)* is based on transferring a ship-specific already trained model to the rest of the ships, thus drastically reducing the training effort. Given adequate similarities across ships and their respective feature distributions, this method can have beneficial outcomes, leading both to extremely fast training and privacy-preservation, since the central server is used only for exchanging the models between local agents.

iii. *Maritime Federated Learning for PEP prediction (MFL-PEPpred)* is based on transmitting only the pretrained model parameters to the central server, where the conventional FedAvg [11] algorithm is running. Periodically, local model weights are replaced with a weighted average model derived by aggregation at the server site (usually located in ports). Apart from data privacy protection and low model dimensionality compared to C-PEPpred, MFL-based local models have finally a global knowledge and can be proactive in predicting previously-unseen PEP conditions, as they indirectly exchange information between each other.

3 Simulation Results

For each of the following training setups, single-ship datasets were separated in training and testing sets after a 90/10 split ratio. All algorithmic implementations were conducted in Python 3.0, using the Tensorflow (v2.4) and Scikit-learn libraries [12].

3.1 Optimization of ML Model Hyperparameters

To ensure optimal configuration of the collaborative methods presented in subsection 3.3, a careful hyperparameter tuning is required for ANN and RFR. The former's convergence performance is considerably affected by the network depth and the learning rate parameters, whereas the latter is strongly influenced by the number of individual tree estimators that are deployed to comprise the forest. Extensive simulations with varying learning rates ($\alpha = 0.1, 0.01, 0.001, 0.0001, 0.00001$) and number of hidden layers ($H = 1, 2, 3, 4, 5$) were conducted to optimize the ANN performance. The ith hidden layer density was (6-i)×200 (where i=1, 2, …,5). For each (α, H) pair, the validation metric was the mean squared error (MSE) between the actual and predicted PEP, computed over the testing set. Fig. 3 shows the validation MSE for different values of

(α, H), as well as the loss curve for the optimal (α, H) configuration. The impact of the different number of estimators in the RFR is also depicted.

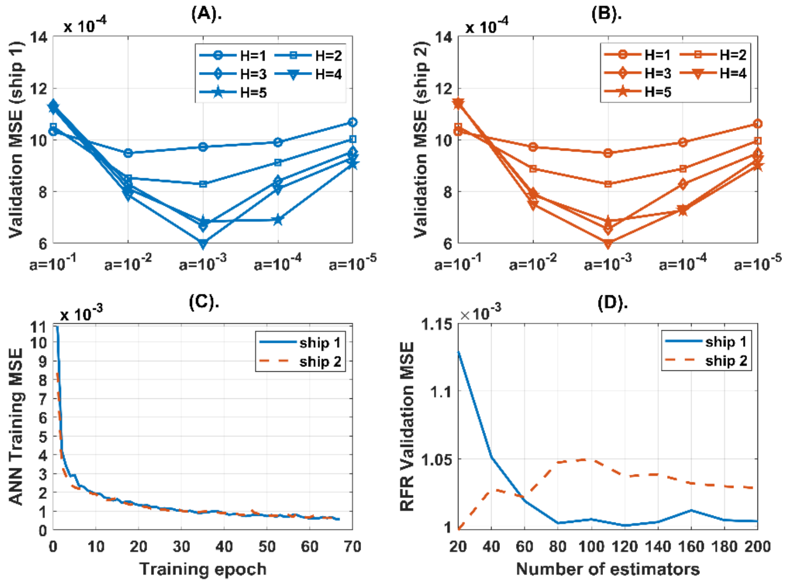

Fig. 3. Hyperparameter tuning of ANN and RFR models. **A – B.** Validation MSE for varying values of learning rate and number of hidden layers, for ship 1 and 2, separately. **C.** ANN training loss curves for ships 1 and 2 using the optimal hyperparameters. **D**. Validation MSE of RFR model, as a function of the number of estimators.

Evidently, both ship-specific ANN models are optimally trained for H=4 hidden layers and $a = 10^{-3}$ learning rate. In addition, the RFR algorithm showed the best outcomes for 120 (ship1) and 20 (ship 2) number of tree estimators, comprising the forest.

3.2 Ship-Specific Model Selection

Using the optimally configured RFR and ANN models for both ships, this section presents the performance comparisons between five ML model regressors, namely MLR, DTR, SVR, RFR and ANN. To decide which model is the best PEP predictor for representing the local models in an FL setup, all models were compared, in terms of their resulting validation error, calculated over the testing set (see Fig. 4).

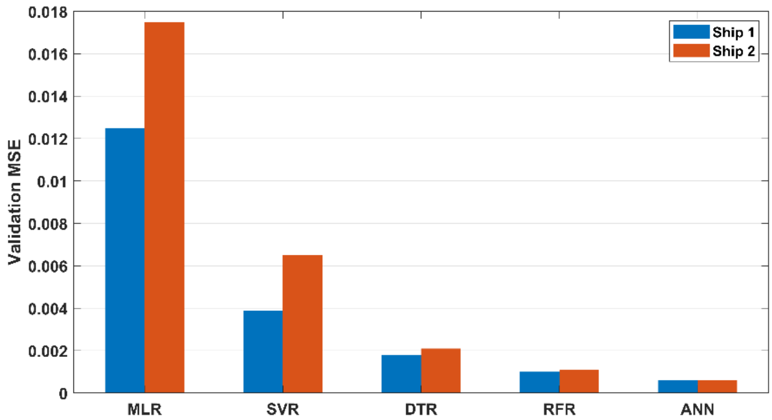

Fig. 4. Comparison among different local PEP prediction models for ship 1 and ship 2, in terms of the validation MSE.

As shown in Fig. 4, all models show a validation MSE constantly below 0.018. Given the presence of non-linearities in the PEP prediction, MLR showed the highest prediction error. Allowing non-linear mapping between the features and the target, SVR and DTR exhibited improved error performance (in the order of 10^{-3}). The ensemble version of DTR (i.e. RFR) improved further the MSE outcomes, being very close to the best regressor, which was the ANN models. Since ANN regressors outperform the rest of models, exhibiting an MSE of 0.0006 for both ships, they were finally selected to represent the local models of the subsequent analyses.

3.3 Multi-ship Collaborative ML Methods

In this section, a quantitative comparison is performed, amongst the considered collaborative between-ships learning schemes, namely the C-PEPpred, S2S-PEPpred and MFL-PEPpred (see Section 2.3). Based on an evaluation set of 100 data samples drawn from both ship 1 and ship 2 testing sets, the three collaborative methods were compared, in terms of their goodness of fit, based on normalized root MSE (NRMSE, degree of fitness between actual and predicted PEP values). To visually illustrate the fitting strength of all methods, Fig. 5 (panels A-C) depicts the actual PEP curve and the corresponding method-predicted series with all samples representing the evaluation set.

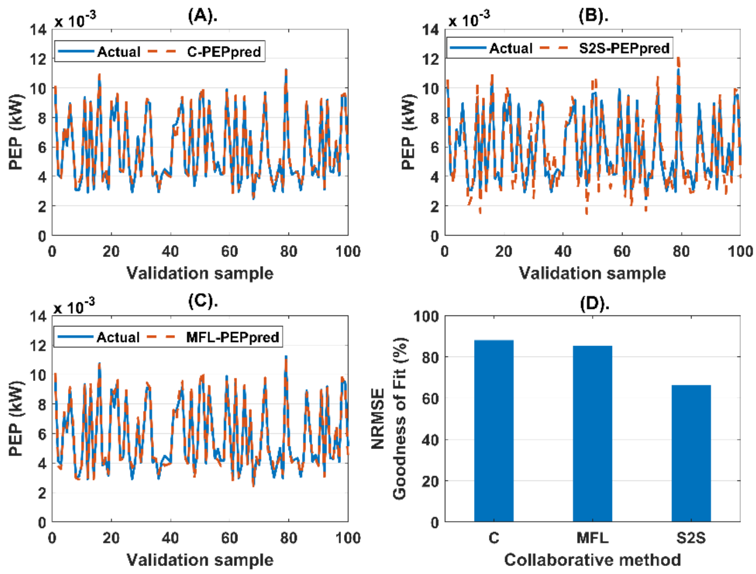

Fig. 5. Performance comparison between different collaborative PEP learning methods. **A – C.** Actual versus method-predicted PEP curves for C-, S2S- and MFL-PEPpred, respectively. **D**. Normalized Root MSE-base Goodness of Fit (%) for the three methods.

Evidently, C-PEPpred shows the best accuracy, as the large ANN used (5 hidden layers) has been trained on the datasets of both ships (global observability). Contrary to the high demands of C-PEPpred to acquire the training data (massive data transmission to/from central server, privacy violation), S2S-PEPpred method shows the most relaxed requirement of having finally a model at each ship site, however providing multiple selfish models. This is attributed to the fact that the transferred model has been only trained on the source model data (low global observability). Towards achieving a reasonable trade-off between the accuracy of C-PEPpred and the flexibility of S2S-PEPpred, the MFL-PEPpred model showed a goodness of fit very close to that of C-PEPpred, offering also low-dimensionality and privacy-preserving benefits. The adequate performance of MFL-PEPpred relies on the ability to obtain general observability for all the ship agents, given the aggregation operation at the central server, across all individual FL contributors.

4 Conclusions

In this paper, a Maritime Federated Learning scheme for decentralized on-ship learning is proposed, targeting to overcome the limits of the current centralized approaches. Considering a fuel consumption prediction use case, several models are compared, as local ship learners, with ANNs outperforming the benchmark ML algorithms. Using a real dataset that includes both on-sail ship features and wind information, multiple simulations were conducted for the optimal configuration of the ML models. The Federated

Learning method was compared against a centralized and a decentralized transfer learning-based method, in terms of the collaborative performance of multi-ship prediction. Results showed that Federated Learning comprises a promising solution for enabling accurate, collaborative and team learning abilities in maritime setups, overcoming the obstacles faced by centralized schemes.

Acknowledgment

This work was partially supported by the TARDIS Project, funded by EU HORIZON EUROPE program, under grant agreement No 101093006.

References

1. Bagoulla, C., & Guillotreau, P.: Maritime transport in the French economy and its impact on air pollution: An input-output analysis. *Marine Policy, 116*, 103818 (2020).
2. Lister, J., Poulsen, R. T., & Ponte, S.: Orchestrating transnational environmental governance in maritime shipping. *Global Environmental Change, 34,* 185-195 (2015).
3. Nomikos, N., Gkonis, P. K., Bithas, P. S., & Trakadas, P.: A survey on UAV-aided maritime communications: Deployment considerations, applications, and future challenges. *IEEE Open Journal of the Communications Society* (2022).
4. Jimenez, V. J., Kim, H., & Munim, Z. H.: A review of ship energy efficiency research and directions towards emission reduction in the maritime industry. *Journal of Cleaner Production*, 132888 (2022).
5. Zhang, J., Wang, M. M., Xia, T., & Wang, L.: Maritime IoT: An architectural and radio spectrum perspective. *IEEE Access, 8*, 93109-93122 (2020).
6. Giannopoulos, A., Spantideas, S., Tsinos, C., & Trakadas, P.: Power control in 5G heterogeneous cells considering user demands using deep reinforcement learning. In *Artificial Intelligence Applications and Innovations*. *AIAI 2021 IFIP WG 12.5* (pp. 95-105). Springer (2021, June).
7. Spantideas, S. T., Giannopoulos, A. E., Kapsalis, N. C., Kalafatelis, A., Capsalis, C. N., & Trakadas, P.: Joint energy-efficient and throughput-sufficient transmissions in 5G cells with deep Q-learning. In *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*. IEEE, pp. 265-250 (2021, September).
8. Kaloxylos, A., Gavras, A., Camps Mur, D., Ghoraishi, M., & Hrasnica, H. (2020).: AI and ML—Enablers for beyond 5G networks. *Zenodo, Honolulu*, *HI, USA, Tech. Rep.*
9. Angelopoulos, A., Giannopoulos, A., Spantideas, S., Kapsalis, N., Trochoutsos, C., Voliotis, S., & Trakadas, P.: Allocating orders to printing machines for defect minimization: A comparative machine learning approach. In *Artificial Intelligence Applications and Innovations: 18th IFIP WG 12.5 International Conference, AIAI 2022* (pp. 79-88). Springer (2022, June).
10. Giannopoulos, A., Spantideas, S., Kapsalis, N., Karkazis, P., & Trakadas, P.: Deep reinforcement learning for energy-efficient multi-channel transmissions in 5G cognitive hetnets: Centralized, decentralized and transfer learning based solutions. *IEEE Access, 9*, 129358-129374 (2021).
11. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A.: Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273-1282). PMLR (2017, April).