IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 November 2023, accepted 15 November 2023, date of publication 21 November 2023, date of current version 29 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3335905

RESEARCH ARTICLE

A Multitask Learning Framework for Predicting Ship Fuel Oil Consumption

LOUKAS ILIAS, PANAGIOTIS KAPSALIS, SPIROS MOUZAKITIS, AND DIMITRIS ASKOUNIS
Decision Support Systems Laboratory, School of Electrical and Computer Engineering, National Technical University of Athens, 15780 Athens, Greece
Corresponding author: Loukas Ilias (lilias@epu.ntua.gr)

This work was supported by the European Union through the Horizon 2020 Research and Innovation Program, in the Context of the VesselAI Project under Agreement GA 957237.

ABSTRACT Predicting the ship fuel consumption constitutes a prerequisite for speed, trim, and voyage optimization. In spite of the rise of deep learning and transformers in many domains, research works train shallow machine learning (ML) algorithms for predicting ship fuel oil consumption (FOC). Although the auxiliary machinery is in support of the main propulsion engines and the emissions from ships' auxiliary engines contribute to the environmental pollution, most existing research initiatives train ML algorithms for predicting only the main engine FOC. Additionally, all the existing research initiatives use the mean squared error (MSE) as the loss function. However, recent studies have shown that neural network models tend to replicate the last observed value of the time series, thus limiting their applicability to real-world data. To address these limitations, this is the first study proposing transformer-based approaches and a multitask learning (MTL) framework. Firstly, the authors introduce Single-Task learning (STL) models consisting of BiLSTMs and MultiHead Self-Attention for predicting the main and auxiliary engine FOC. Secondly, the authors introduce the first MTL setting, which predicts the main and auxiliary engine FOC simultaneously allowing one task to inform the other. A loss function is introduced, which includes a regularization term for penalizing the replication of previously seen values. The authors evaluate the proposed approaches using data from three fishing ships and compare these approaches with traditional ML algorithms. Extensive experiments show that the introduced MTL models can improve the R^2 score, mean bias error, root mean squared error, and mean absolute error in comparison with shallow ML algorithms.

INDEX TERMS Maritime industry, main engine fuel oil consumption, auxiliary engine fuel oil consumption, deep learning, MultiHead self-attention, multi-task learning, loss function.

LIST OF ABBREVIATIONS AND ACRONYMS

In this section, the definitions of abbreviations and acronyms used in the paper are summarized in Table 1. Each abbreviation and acronym is ordered from A to Z.

I. INTRODUCTION

The maritime vessels emit around 940 million tonnes of CO_2 annually and are responsible for about 2.70% of global greenhouse gas (GHG) emissions [1]. Even worse, according to the 3rd International Maritime Organization (IMO) GHG study, shipping emissions are projected to increase between 50% and 250% by 2050, if business goes as usual, undermining the objectives of the Paris Agreement. The study of [2] also points out that ships consume large amounts of fuel oil and consequently GHGs are released causing serious damage to the environment, climate, and human health. Wave and wind conditions affect significantly the fuel consumption. Specifically, the study in [3] analyzed changes in wind and wave conditions over the last 27 years. After conducting numerical simulations for six voyages, the authors stated that the difference between fuel consumption, travel time, average ship speed is significant for to and fro direction. For instance, travelling from "Sydney to Valparaiso" requires more time and fuel than "Valparaiso to Sydney" in both summer and winter seasons due to the direction of wave

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

and wind. Slow steaming, i.e., sailing at reduced speed. constitutes one of the measures proposed throughout the years for mitigating CO_2 emissions [4], [5]. Slow steaming seems to be effective, since ship resistance is lower at the reduced speed. However, according to the study [4], one should analyze several locations along the sailing route and determine local sea states for quantifying the benefits of slow steaming more accurately. Additionally, sailing at reduced speed often leads to a deterioration of the main and auxiliary engines [6]. Apart from the careful examination of the operation of the main engine under lower loads, one should also carefully examine the operation of auxiliary engines, since the electrical consumption of auxiliary engines has a significant role in fuel consumption under slow steaming conditions [7]. Trim optimization can also lead to savings in fuel consumption. Grlj et al. [8] conducted a study for a containership and stated that trim has an effect on wind and air resistance. Specifically, findings suggested that trim by bow causes lower values of air resistance, while trim by stern leads to higher values in comparison to an even keel condition. To reduce CO_2 emissions in maritime transport, the IMO has introduced the Energy Efficiency Design Index (EEDI), which is used for new ships solely. Contrary to EEDI, the IMO has also introduced the Energy Efficiency Existing Ship Index (EEXI) [9]. Specifically,[1] from 1 January 2023 it is mandatory for all ships to calculate their attained EEXI to measure their energy efficiency.

Although deep learning approaches and transformer-based models are used in many tasks outperforming the traditional machine learning algorithms, existing research initiatives still train shallow machine learning regressors for predicting fuel oil consumption. In addition, only few works have proposed methods for estimating the auxiliary engines fuel oil consumption [10]. Although the auxiliary machinery is in support of the main propulsion engines [11] and the emissions of ships' auxiliary engines contribute to the environmental pollution, existing research initiatives train machine learning algorithms for predicting only the main engine fuel consumption. Additionally, existing research initiatives use limited sets of features for training machine learning algorithms and predicting fuel oil consumption [12], [13], [14]. In addition, all the existing research initiatives minimize the mean squared error (MSE) loss for predicting the fuel oil consumption. However, according to [15], one limitation of this approach is the fact that the model just replicates the last observed value of the time series. The authors in [15] define this problem as "mimicking" in time-series forecasting.

To tackle the aforementioned limitations, this is the first study employing BiLSTM and MultiHead Self-Attention layers in a multitask learning setting. Firstly, the authors introduce single-task learning (STL) models which predict the main and auxiliary engine fuel oil consumption separately.

[1] https://www.imo.org/en/MediaCentre/HotTopics/Pages/EEXI-CII-FAQ.aspx

TABLE 1. List of abbreviations and acronyms.

Abbreviation	Full Form
ANN	Artificial Neural Network
AIS	Automatic Identification System
BiLSTM	bidirectional long short-term memory
BPNN	Back-Propagation Neural Network
deadweight	DWT
ENN	elman neural network
ET	extremely randomized trees
ETR	Extra Trees Regressor
FFNN	feed-forward neural network
FOC	Fuel Oil Consumption
GHG	greenhouse gas
GMDH	group method of data handling
GP	Gaussian Process
GPR	Gaussian Process Regression
GRNN	general regression neural network
IMO	International Maritime Organization
kNN	k-Nearest Neighbours
LR	Linear Regression
LSTM	Long Short-Term Memory
MLR	Multiple Linear Regression
MRM	Multiple Regression Model
MSE	mean squared error
MTL	multi-task learning
nm	nautical mile
RBFNN	radial basis function neural network
RF	Random Forest
RFR	Random Forest Regressor
RNN	Recurrent Neural Network
RPM	revolution per minute
RVM	relevant vector machine
SOG	speed over ground
STL	single-task learning
SVM	Support Vector Machine
SVR	support vector regressor
XGB	XGBoost

These models comprise a BiLSTM layer and a MultiHead Self-Attention layer for allowing the model to jointly attend to information from different representation subspaces at different positions. Secondly, motivated by the fact that multitask learning (MTL) has been proved to be effective in many domains [16], [17], [18], including both related and unrelated tasks [19], the authors introduce the first MTL framework consisting of a primary task, i.e., prediction of main engine fuel oil consumption, and auxiliary task, i.e., prediction of auxiliary engine fuel oil consumption. MTL allows tasks to be learned jointly, thus sharing knowledge and features between the tasks. To address the phenomenon of "mimicking" in time series forecasting, the authors add a regularization term in the loss function. the authors exploit sensor data with a great number of features from three fishing ships for conducting the experiments. The authors train shallow machine learning algorithms, including BaggingRegressor, RandomForestRegressor, etc. and use them as baselines. Findings suggest that the introduced approaches offer valuable advantages over state-of-the-art ones.

The main contributions of this study can be summarized as follows:

- The authors utilize a multihead self-attention mechanism for allowing the model to jointly attend to information

from different representation subspaces at different positions.
- There is no prior work proposing a multitask learning framework for predicting main and auxiliary engine fuel oil consumption simultaneously.
- This is the first study addressing the problem of "mimicking" in maritime series data by adding a regularization term in the loss function.
- The authors use data from three vessels and exploit a large number of features in comparison with existing research initiatives.
- The authors compare their proposed approaches with shallow machine learning algorithms and show that the introduced architectures outperform the traditional ones.

II. RELATED WORK

A. FUEL OIL CONSUMPTION PREDICTION TASK

Reference [20] used two types of datasets, namely noon-reports and Automated Data Logging & Monitoring systems for predicting main engine fuel oil consumption (FOC). The authors exploited a limited set of feature set consisting of 12 features, including vessel speed, draft aft, engine speed, etc. Finally, the authors trained multiple regression algorithms, including Support Vector Machines (SVMs), Random Forest Regressors (RFRs), Extra Trees Regressors (ETRs), AdaBoost Regressors, Artificial Neural Networks (ANNs), Linear Regressors (LR), Ridge & Lasso Regressors, and k-Nearest Neighbours Regressors. The authors stated that the proposed models can accurately predict the FOC of vessels sailing under different conditions.

The study in [21] used noon-reports and exploited an ANN consisting of one hidden layer with 12 units for predicting fuel consumption. The authors used seven input variables, namely ship speed, revolutions per minute (RPM), mean draft, trim, cargo quantity on board, wind and sea effects.

The authors in [22] introduced a publicly available set of high-quality sensory data collected from a ferry over a period of two months. The authors introduced a non-linear ANN model to model ship fuel consumption efficiency.

A model for estimating the energy use and fuel consumption was proposed by [23]. The authors exploited Automatic Identification System (AIS) data and some technical information about cruise ships, including the service speed, total power, and number of engines. A multivariate regression model was trained.

Reference [12] trained a multiple regression model used for fuel consumption prediction. The authors used a limited feature set consisting of ship average speed, sailed distance, wind speed in knots, and displacement.

Similarly, the authors in [24] used a multiple linear regression model, where the amount of ship fuel consumed (in litres) was designated as the output dependent variable, while factors such as the travelled distance in Nautical Mile (nm), travelled hours (HRS), Ship speed (V), Deadweight in metric tonnes (DWT) and Wind Speed (W) in knots were designated as the input (independent) variables.

Reference [25] trained and compared multiple regression models for predicting CO_2 emissions. The authors used a set of 22 features, namely shaft generator power, speed over ground, arrival draught, departure/arrival trim, etc. Next, the authors exploited variable selection approaches (forward stepwise regression), penalized regression models, latent variable methods, and tree-based ensemble methods.

An artificial neural network was also proposed by [13]. After using data denoising methods, data clustering approaches, and data compression & expansion methods, the authors employed an ANN for predicting the ship's fuel consumption. The authors used seven features, including the average draft, trim, main engine power, shaft speed, speed through water, Speed over Ground (SOG), and relative wind speed in knots. Results showed that ANN is a more accurate and efficient model to predict the fuel consumption of the main engine than polynomial regression and support vector machine.

In [26], the authors used data from noon reports, engine logbook, and sensors to predict fuel oil consumption. They used the following features as inputs to machine learning algorithms: Bearing temperatures, Fuel mass flow, Air coolers cooling water temperature, Shaft power, and more. Finally, the authors trained Multiple Linear Regression, Kernel Ridge Regression, Ridge and Lasso Regression, Support Vector Regression (SVR), Tree-Based Regression as Random Forest Regression and Decision Tree Regression and Boosting Algorithms including AdaBoost Algorithm and Gradient Boost Algorithm.

In [27], the authors introduced a hybrid machine learning model consisting of a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) and an Elman Neural Network (ENN). The authors used as features the number of passengers, the average speed of the vessel, the wind force, etc. The authors compared the proposed architecture with: Radial Basis Function Neural Network (RBFNN), General Regression Neural Network (GRNN), Elman Neural Network (ENN), SVR, Group Method of Data Handling (GMDH)-based neural network, Relevant Vector Machine (RVM), Feed-Forward Neural Network (FFNN), and Multiple Regression Model (MRM).

The authors in [28] used two datasets for predicting the fuel consumption. With regards to the first dataset, the authors exploited the following set of features: ship shaft, speed, average draft, trims, current speed, current direction, wind speed knots, wind direction, wave height, and wave direction. In terms of the second dataset, the authors used the following feature set: ship shaft, speed, average draft, and trim. Finally, the authors used Back-Propagation Neural Network (BPNN) and Gaussian Process Regression (GPR) techniques.

Reference [29] used voyage, weather, and sea data for predicting fuel consumption. The authors trained a LASSO regression model and compared it with ANNs, SVR, and Gaussian Processes (GP). Findings showed that the proposed LASSO-based method outperforms other traditional methods.

A straightforward approach was also proposed by [30]. The authors trained traditional machine learning algorithms, including ANN, SVR, LASSO, and RF. They also used bayesian optimization for tuning the hyperparameters. The authors used as features the relative wind direction, main engine speed, fore and aft draft, and wind speed (knots).

In [31], the authors trained Gradient Boosting Regression (GBR), Random Forest (RF), BP Network (BP), LR, and K-Nearest Neighbor Regression (kNN) to estimate the energy consumption of ships in port. 15 features were exploited including inherent ship features and external port features. Findings showed that net tonnage, deadweight tonnage, actual weight, and efficiency of facilities are the top four features for predicting the energy consumption of ships.

A different approach was proposed by [32], where the authors exploited the engine temperature as feature for the estimation of fuel consumption for the first time and trained an LSTM neural network. They compared the LSTM with traditional machine learning algorithms, including linear regression, and ANNs and stated that the LSTM outperformed these approaches. Finally, the developed models have been implemented in optimisation of the engine speed to minimize the total fuel consumption and the total cost of the whole voyage. Specifically, the authors employed the Reduced Space Searching Algorithm for solving the optimization problems.

The study in [33] used in-service data collected from a 13,000 TEU class container ship to predict fuel consumption. The authors used 11 features, including Speed over Ground (SOG), speed through water, trim, displacement, etc. to train machine learning algorithms. They trained a Multiple Linear Regression model and an ANN and showed that ANNs yielded the best performance.

The authors in [34] used data of two real-world voyages from bulk carrier and introduced a hybrid model for predicting fuel consumption. Specifically, the introduced hybrid model is based on stacking theory consisting of two-level layers. The first layer is the base model, consisting of extremely randomized trees (ET), RF, and XGBoost (XGB). The second-level layer is a meta-model consisting of multiple linear regression (MLR). Finally, the authors presented a new method based on the developed hybrid model in combination with the enumeration method to optimize the fuel consumption from the perspective of trim adjustment.

A similar approach was proposed by [35], where the authors used sensor data collected from an ocean-going container ship. The first-level layer is composed of multiple base-models, namely ET, RF, and XGB, while the second-level layer is a meta model, i.e., multiple linear regression (MLR). The authors used nine features: GPS speed, mean draft, wave direction and height, wind direction, wind speed in knots, etc.

In [36], the authors introduced a two-stage fuel consumption prediction and reduction model. At the first stage, they trained a random forest regression model using 11 features, including weather conditions, sea conditions, wind force, wave height, and many more. Next, based on the random forest regression model, the authors introduced a speed optimization model between two ports.

The authors in [37] trained an ANN and polynomial regression models to predict ship's power and fuel consumption. The authors used data from two voyages and exploited a limited feature set, including the water depth, cargo conditions, and more.

Similarly, the authors in [38] trained an ANN consisting of one hidden layer with 10 units. The authors exploited voyage report data and used the following features as input to the deep neural network: sailing speed (knots), displacement (MT), trim (m), wave height (m), wave direction, wind force (Beaufort scale number), wind direction, and sea water temperature (°C). The output of the network was the fuel consumption rate.

In [39], the authors introduced methods for predicting energy efficiency and addressing the optimal energy efficiency route planning challenge. Specifically, for the energy efficiency prediction model, they utilized an ANN with a single hidden layer. For the optimal energy route planning issue, they presented an enhanced Ant Colony Algorithm.

Reference [14] introduced methods for estimating the required shaft power or main engine fuel consumption of a container ship sailing under varied conditions. The authors exploited data acquired from the operation of a container ship. This dataset consists of 14 features, including longitude, latitude, draft aft, speed over ground, speed through water, and many more. Next, the authors applied data preprocessing techniques, including time-series filtering, statistical outlier detection, smoothing, etc. Then, data quality control was performed. After that, the authors extracted a set of four features and applied feature selection approaches. Finally, the authors trained an ANN consisting of two hidden layers.

B. BACKGROUND

1) BIDIRECTIONAL LSTMS (BILSTMS)

Recurrent Neural Networks cannot capture the long-distance dependencies effectively. To address this issue, Long Short-Term Memory (LSTM) [40] neural models were proposed. An LSTM cell consists of three gates, namely the forget gate, the input gate, and the output gate. The structure of the LSTM unit is illustrated in Fig. 1 and is given by the equations below:

- The f_t is the forget gate:

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{1}$$

- The i_t is the input gate:

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \tag{2}$$

- The $\tilde{c_t}$ represents the candidate memory cell status at the current time-step.

$$\tilde{c_t} = \tanh\left(W_c x_t + U_i h_{t-1} + b_c\right) \tag{3}$$

- The c_t represents the state value of the current time-step in memory cell and is calculated as follows:

$$c_t = i_t \odot \tilde{c_t} + f_t \odot c_{t-1} \tag{4}$$

- The o_t is the output gate:

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \tag{5}$$

- h_t indicates the hidden layer state at time t:

$$h_t = o_t \odot \tanh\left(c_t\right) \tag{6}$$

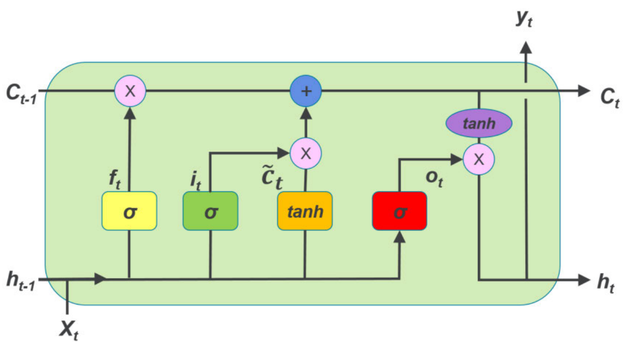

FIGURE 1. LSTM model.

However, the typical LSTM layer cannot capture effectively the contextual information. LSTM can only process sequence from one direction. For this reason, research works utilize the bidirectional LSTM consisting of a forward LSTM layer and a backward LSTM layer. A BiLSTM layer is illustrated in Fig. 2. The equations governing the internal mechanism of a BiLSTM layer are presented as follows:

$$\overrightarrow{h_t} = LSTM(x_t, \overrightarrow{h_{t-1}}) \tag{7}$$

$$\overleftarrow{h_t} = LSTM(x_t, \overleftarrow{h_{t+1}}) \tag{8}$$

$$H_t = \overrightarrow{h_t} \,||\, \overleftarrow{h_t}, \tag{9}$$

where x_t denotes the input of time t, $\overrightarrow{h_{t-1}}$ denotes the output of the forward hidden unit at time $t-1$, while $\overleftarrow{h_{t+1}}$ indicates the output of the backward hidden unit. || denotes the concatenation operation.

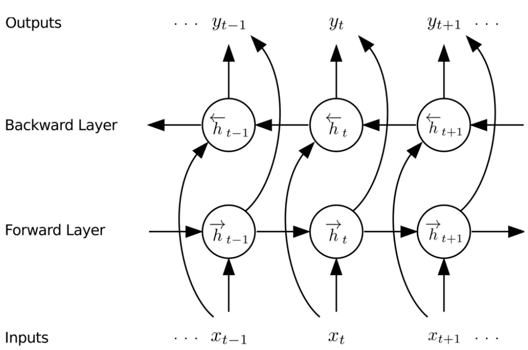

FIGURE 2. BiLSTM model.

2) MULTIHEAD ATTENTION MECHANISM

In this section, the multihead attention mechanism introduced in [41], is going to be described.

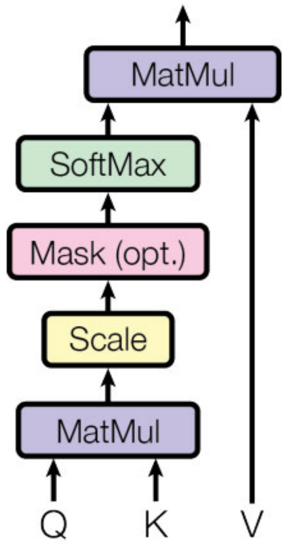

FIGURE 3. Scaled dot-product attention.

a: SCALED DOT-PRODUCT ATTENTION

A scaled dot-product attention can be described as mapping a query and a set of key-value pairs to an output. As illustrated in Fig. 3, the input consists of queries (Q), keys (K), and values (V). Specifically, queries and keys have a dimension of d_k, while values have a dimension of d_v. Firstly, the dot products of the query with all the keys are computed. Next, a scale operation is applied, where the result of the dot product is divided by $\sqrt{d_k}$. After that, a mask is applied optionally. Finally, a softmax function is applied for obtaining the weights on the values. It is worth noting that the scale operation is important, since the dot products grow large in magnitude, pushing the softmax function into regions where it has extremely small gradients.

The scaled dot-product attention can be calculated as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{D_k}}\right)V \tag{10}$$

3) MULTIHEAD ATTENTION

To address the issue of performing one single attention, the work in [41] introduces the multi-head attention mechanism, which is illustrated in Fig. 4. Specifically, the multi-head attention mechanism captures information from different subspaces and strengthens the feature discrimination by converting the original query matrix Q, key matrix K, and value matrix V into H submatrices of the same size as described in the equations presented below:

$$\begin{cases} Q^i = QW_i^Q, \\ \vdots \\ K^i = KW_i^K, \quad , \\ \vdots \\ V^i = VW_i^V, \end{cases} \tag{11}$$

where $Q^i \in R^{n\times d_q}$, $K^i \in R^{n\times d_k}$, and $V^i \in R^{n\times d_v}$ represent the i^{th} subspaces of Q, K, and V respectively. Usually, $d_q = d_k = \frac{d_{model}}{H}$ and $d_v = \frac{d_{model}}{H}$.

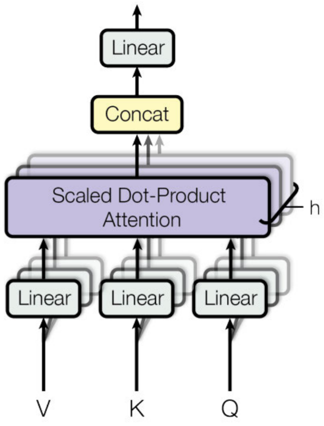

FIGURE 4. Multi-head attention.

Next, scaled dot-product attention operations are performed on H subspaces in parallel as follows:

$$\begin{cases} head_1 = Attention(Q^1, K^1, V^1), \\ \vdots \\ head_i = Attention(Q^i, K^i, V^i), \\ \vdots \\ head_H = Attention(Q^H, K^H, V^H), \end{cases} \tag{12}$$

Finally, the results learned by the multi-head attention are concatenated as output and projected to dimensionality d_{model}. Formally:

$$p = MHA(Q, K, V) = Concat(head_1, head_2, \cdots, head_H)W^0, \tag{13}$$

where $W^0 \in R^{Hd_v\times d_{model}}$.

In case $Q = K = V$, then this attention can be called as multi-head self-attention.

C. RELATED WORK REVIEW FINDINGS

From the research works mentioned above, it is evident that existing research initiatives rely on the feature extraction and train of shallow machine learning algorithms. Only few research works have experimented with deep learning approaches. However, these approaches introduce simple architectures. For instance, the authors of some research works have employed ANNs and BiLSTMs instead of transformers, which have achieved state-of-the-art results in many domains. Additionally, all the research works minimize the mean square error loss function without taking into consideration the effect of ''mimicking''. Finally, the existing research initiatives mainly predict the main engine fuel oil consumption neglecting the auxiliary engine. However, the auxiliary machinery operates as a support of the main propulsion engines, while at the same time the auxiliary engines contribute to the air pollution. Therefore, the prediction of both the main and auxiliary engine fuel consumption is crucial.

Therefore, the present study is significantly different from the state-of-the-art approaches, since the authors *(a)* introduce approaches for predicting both the main and auxiliary engine fuel oil consumption, *(b)* exploit for the first time the multihead self-attention, *(c)* introduce a multitask learning framework which jointly learns to predict the main and auxiliary engine fuel oil consumption, *(d)* introduce a new loss function in the task of main and auxiliary engine fuel oil consumption which mitigates the effect of ''mimicking'', and *(e)* evaluate the approaches on three publicly available datasets with a great number of features.

III. DATASET & TASK

A publicly available dataset [42] is used, which includes sensor data from three different fishing ships for a period of one month. This data has been used in Work Package (WP) 3 of DataBio project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732064. Also, this data is used in the context of the VesselAI project. The authors resample the dataset into 5 minute bins and sum the values of the timestamps falling into a bin. A lag variable accounting for 10 is used.

Given a feature set, the tasks are to predict: *(i)* main engine fuel oil consumption and *(ii)* auxiliary engine fuel oil consumption. Table 2 describes the features used for the task of main engine fuel oil consumption, while Table 3 reports the features used in terms of the task of the auxiliary engine fuel oil consumption. As a preprocessing step, constant columns are removed. Therefore, in these Tables, the features used per dataset are mentioned.

IV. METHODOLOGY

In this section, the proposed models for predicting the main engine and auxiliary engine fuel oil consumption are described. Specifically, Section IV-A describes the Single-Task Learning (STL) models that utilize either the main engine Fuel Oil Consumption task or the auxiliary engine Fuel Oil Consumption task as the sole optimization objective. Section IV-B describes the proposed architectures for jointly learning to predict main and auxiliary engine fuel oil consumption. Finally, in IV-C the authors describe the proposed loss function used in the experiments.

A. SINGLE-TASK LEARNING

The proposed STL architecture is illustrated in Fig. 5. Below, the layers of the proposed architecture are mentioned.

1) INPUT LAYER

Let $x \in \mathbb{R}^{n\times T}$ be the input representation, where n denotes the lag variable and is equal to 10, while T denotes the number of features. The dimension corresponding to the batch size is not mentioned for the sake of simplicity.

TABLE 2. Description of features - main engine.

Feature	Description	Ship_1	Ship_2	Ship_3
cyl_chargeair_press		✓	✓	✓
draught_aft_side	Depth of ship aftward side from reference condition.from zero in ship	✓	✓	
draught_fore_side	Depth of ship foreward side from reference condition.from zero in ship	✓	✓	
engine_speed	Engine speed	✓	✓	✓
CAC_CW_HT_pressure		✓	✓	✓
CAC_in_Low_Temperature_CW_temp	If temperature is high, then probably cooling water cooler is dirty or flow is not enough dut to faulty pump.	✓	✓	✓
propeller_shaft_output	Product of Torque and RPM in shaft.	✓	✓	✓
propeller_shaft_rpm	Turning speed of propeller shaft. Constant ratio with engine rpm.	✓	✓	✓
propeller_shaft_thrust	Shaft compression, thrust in shaft.	✓	✓	✓
cyl_chargeair_temp	A high value means that air cooler is not working properly because it is dirty or water flow is not correct. If temperature is low could be cause by low Charge Air Pressure.	✓	✓	✓
ship_speed_actual	Speed Over Ground from ships GPS.	✓		
Ship_SpeedLOG	Speed over water, not over ground. This means that includes currents.	✓		✓
cyl_exh_gas_temp_mean	The engine has 9 cylinders. Each cylinder has exhaust gas temperature measurement in the outlet of the engine. This mean value.	✓	✓	✓
torque	Torque in propeller shaft	✓	✓	✓
Eng_in_HTCW_press	Water flow to engine pressure.	✓	✓	✓
Eng_in_Jacket_HTCW_temp	Water flow to engine temperature.	✓	✓	✓
Eng_out_Jacket_HTCW_temp	Water flow from engine temperature.	✓	✓	✓
Eng_Relative_load	Value for monitoring engine load in each working condition. Could be used as reference value for the rest of parameters. I.e., with a certain value of relative load rest of values should be in a certain range.	✓	✓	✓
FO_Rack_position	Indicated amount of fuel injected to each cylinder.	✓	✓	✓
FO_inlet_press	Fuel oil inlet pressure	✓	✓	✓
fueloil_inlet_temperature	Fuel oil inlet temperature	✓	✓	✓
ME_FO_inlet_Temp	Measurement of inlet temperature to engine	✓	✓	✓
ME_FO_inlet_flow	Measurement of inlet flow to engine coming from the fuel daily tank in the tank temperature.	✓	✓	✓
ME_FO_outlet_Temp	Measurement of outlet temperature from engine	✓	✓	✓
ME_FO_outlet_flow	Measurement of outlet flow from engine	✓	✓	✓
LO_Filter_P	Pressure drop in filter	✓	✓	✓
LO_filter_in_press	Pressure is measured before filter to evaluate lubricating oil condition.	✓	✓	✓
LO_in_temp	Lubricating oil inlet temperature	✓	✓	✓
LO_in_press	Lubricating oil inlet pressure	✓	✓	✓
LO_out_temp_TC	Lubricating oil outlet temperature	✓	✓	✓
LO_cooler_CW_out_temp	Water cooler outlet temperature	✓	✓	✓
ship_inclination	Inclination of ship taking as reference ship metacentre (approximate).	✓	✓	✓

In terms of the prediction of the main engine fuel oil consumption, the authors use the feature set described in Table 2.

Regarding the prediction of the auxiliary engine fuel oil consumption, the authors exploit the feature set reported in Table 3.

2) BILSTM LAYER

x is passed through a BiLSTM layer. A BiLSTM consists of two LSTMs, a forward LSTM $\overrightarrow{f}$ which processes the input sequence from left to right, and a backward LSTM $\overleftarrow{f}$, which processes the input sequence from right to left. Formally,

$$h_i = [\overrightarrow{h_i}\,;\,\overleftarrow{h_i}], \tag{14}$$

where $\overrightarrow{h_i}, \overleftarrow{h_i} \in \mathbb{R}^l$ and $h_i \in \mathbb{R}^{n\times 2l}$, where l denotes the hidden dimensionality of the BiLSTM.

3) MULTIHEAD SELF-ATTENTION LAYER

The authors exploit a MultiHead Self-Attention layer introduced by [41]. Specifically, h_i is transformed into a Query $Q \in \mathcal{R}^{n\times D_q}$, Key $K \in \mathcal{R}^{n\times D_k}$, and Value $V \in \mathcal{R}^{n\times D_v}$ matrix as described via the equations below:

$$Q = QW^Q, K = KW^K, V = VW^V, \tag{15}$$

where $W^Q \in \mathcal{R}^{2l\times D_q}$, $W^K \in \mathcal{R}^{2l\times D_k}$, and $W^V \in \mathcal{R}^{2l\times D_V}$ are learnable parameters. As mentioned in [43], the authors set $D_q = D_k = D_v = 2l$. The self-attention mechanism can be calculated as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{D_k}}\right)V \tag{16}$$

As an improved self-attention mechanism, MultiHead self-attention divides self-attention into h heads to learn the different levels of long-term information in the input sequence.

The equation for calculating the attention of the i-th head is described below:

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V), \tag{17}$$

where $W_i^Q \in \mathbb{R}^{2l\times d_q}$, $W_i^K \in \mathbb{R}^{2l\times d_k}$, $W_i^V \in \mathbb{R}^{2l\times d_v}$. $d_q = d_k = d_v = \frac{2l}{h}$

Finally, the results learned by the multi-head attention are concatenated as output and projected to dimensionality d_0. Formally:

$$p = MHA(Q, K, V) = Concat(head_1, head_2, \cdots, head_h)W^0, \tag{18}$$

where $W^0 \in \mathbb{R}^{2l\times d_0}$. $d_0 = 2l$.

TABLE 3. **Description of features - auxiliary engine.**

Feature	Description	Ship_1	Ship_2	Ship_3
AE_FO_inlet_flow	Measurement of inlet flow to auxiliary engines coming from the fuel daily tank in the tank temperature.	✓	✓	✓
AE_FO_inlet_Temp	Temperature of inlet flow to auxiliary engines coming from the fuel daily tank in the tank temperature.	✓	✓	✓
DG_1_power	electric kilowatt in generator	✓	✓	✓
DG_2_power	electric kilowatt in generator	✓	✓	✓
DG_3_power	electric kilowatt in generator		✓	✓
DG_4_power	electric kilowatt in generator	✓	✓	✓
DG_5_power	electric kilowatt in generator	✓	✓	✓
AE_FO_outlet_flow	Measurement of outlet flow from auxiliary engine going to the fuel daily tank in the outlet temperature. The fuel in the engine warms up, this means that density is reduced and same mass occupies more volume.	✓	✓	✓
AE_FO_outlet_Temp	Temperature of outlet flow from auxiliary engine going to the fuel-	✓	✓	✓
DG_1_condition	Aux engine generator on or off	✓	✓	✓
DG_2_condition	Aux engine generator on or off	✓	✓	✓
DG_3_condition	Aux engine generator on or off		✓	✓
DG_4_condition	Aux engine generator on or off	✓	✓	✓
DG_5_condition	Aux engine generator on or off	✓	✓	✓

Next, p is passed through a Global Average Pooling layer. Let the output of the Global Average Pooling layer be $z \in \mathbb{R}$.

4) OUTPUT LAYER

Finally, a dense layer consisting of 1 unit is used, in order to get the final prediction.

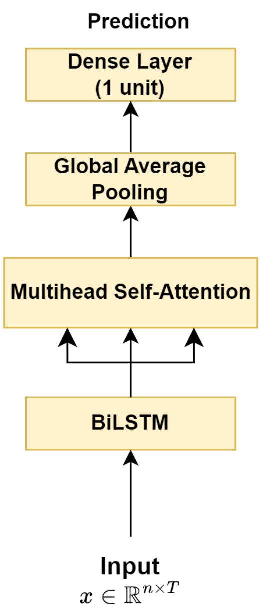

FIGURE 5. **The proposed STL model.**

B. MULTI-TASK LEARNING

According to [11], the auxiliary machinery operates as a support of the main propulsion engines. Auxiliary engine is correlated with main engine as it can also assist the main propulsion engines by incorporating heat exchangers and compressed air, aid in ship and cargo handling through propellers, shafting, steering gear, and deck cranes, or support ship services like ballast water arrangements and sewage systems. Additionally, multitask learning has been proved to be effective in many domains [16], [17], [18], including both related and unrelated tasks [19].

In this section, two deep learning architectures based on multi-task learning are introduced [44]. Each architecture consists of two tasks, namely the main task and the auxiliary one. The main objective is to explore whether the auxiliary task helps the main task in increasing its performance. The main task constitutes the task of the main engine fuel oil consumption, while the auxiliary task constitutes the task of the auxiliary engine fuel oil consumption. The introduced architectures are trained on the two tasks and updated at the same time with a joint loss:

$$L = (1 - \alpha)\, L_{MEFOC} + \alpha L_{AEFOC}, \tag{19}$$

where L_{MEFOC} denotes the loss of the main engine fuel oil consumption, L_{AEFOC} indicates the loss of the auxiliary engine fuel oil consumption, and α is a parameter denoting the importance the authors place to each task.

1) MTL-SIMPLE

In this architecture, the authors merge the features reported in Tables 2 and 3 into one feature vector. The merging is based on the correlation between the main engine and auxiliary engines features, as the auxiliary engines, among others, are in support of the main engine [11]. A lag variable of 10 is used. Let $x \in \mathbb{R}^{n \times T}$ be the input representation. n denotes the lag variable, while T denotes the number of features. As illustrated in Fig. 6, the input representation is passed through a shared BiLSTM layer as described in Eq. 14. Let $z \in \mathbb{R}^{n \times 2h}$ be the output of the BiLSTM layer, where h denotes the hidden dimensionality of the BiLSTM. Next, z is passed through a MultiHead Self-Attention layer as described via the Equations.

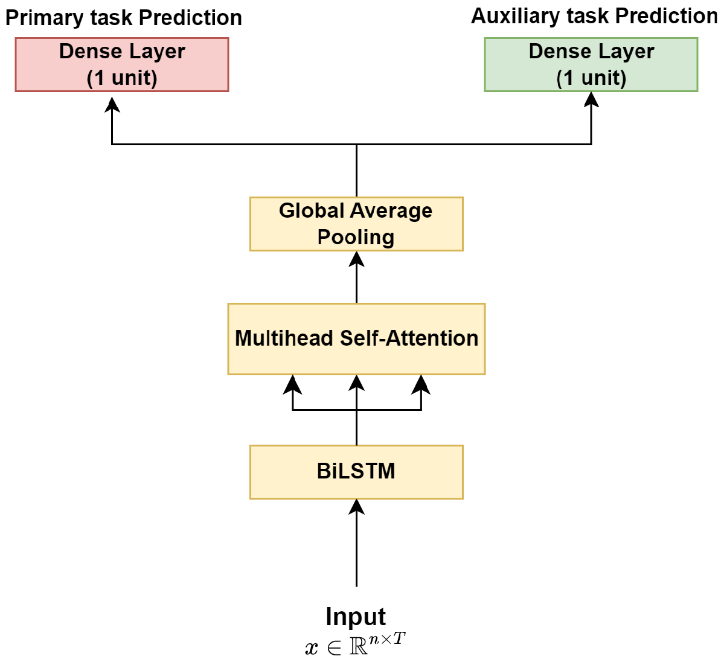

FIGURE 6. The proposed MTL-simple model.

Let $p \in \mathbb{R}^{n \times 2h}$ be the output the MultiHead Self-Attention layer. Next, the authors pass p through a global average pooling layer and obtain $s \in \mathbb{R}^{2h}$. Finally, s is passed through two dense layers, where each dense layer consists of one unit, which give the final prediction per task.

2) MTL-DOUBLE ENCODERS

The introduced architecture is illustrated in Fig. 7. Similar to the *MTL-Simple* architecture, the authors merge the features reported in Tables 2 and 3 into one feature vector. A lag variable of 10 is used. Let $x \in \mathbb{R}^{n \times T}$ be the input representation. First, the authors pass x through a shared BiLSTM layer, which is updated by both tasks during training. Let $z \in \mathbb{R}^{n \times 2h}$ be the output of the BiLSTM layer, where h denotes the hidden dimensionality of the BiLSTM.

- **Primary Task Prediction:** This is a task-specific branch pertinent to the primary task. Specifically, z is passed through a task-specific BiLSTM, a MultiHead Self-Attention layer, a Global Average Pooling layer, and a Dense layer consisting of one unit which gives the final output.
- **Auxiliary Task Prediction:** Here, the authors describe the task-specific branch related to the prediction of the auxiliary task. Specifically, z is passed through a MultiHead Self-Attention layer followed by a global average pooling layer. Finally, a dense layer consisting of one unit is used, which gives the final output, i.e., auxiliary engine fuel oil consumption.

C. LOSS FUNCTION

In [15], the authors define the problem of "mimicking" in time series forecasting. Specifically, neural networks minimizing the MSE loss, become often sensitive to noise. This might result into the problem of predicting previously seen values (usually the last seen observation in the time series), rather than making predictions based on long-term extracted patterns. To tackle this limitation, the authors in [15] introduce a regularization term for mitigating to some degree the effect of "mimicking". For this reason, the following loss function is proposed for a sequence of n time-steps.

$$\mathcal{L} = \sum_{i=1}^{n}(z_i - \hat{z}_i)^2 + \lambda \sum_{i=1}^{n}[(z_i - z_{i-1})(z_i - \hat{z}_i)]^2, \quad (20)$$

where λ is a parameter used for controlling the importance of the regularization term, i.e., how much penalty needs to be imposed to alleviate "mimicking".

The loss function described in Eq. 20 is adopted in STL and MTL frameworks. To be more precise, in terms of the MTL setting, this specific loss function is exploited regarding both L_{MEFOC} and L_{AEFOC} (see Eq. 19).

V. EXPERIMENTS

A. BASELINES

The introduced approaches are compared with shallow machine learning algorithms, namely AdaBoostRegressor, BaggingRegressor, ExtraTreesRegressor, GradientBoostingRegressor, and RandomForestRegressor.

B. EXPERIMENTAL SETUP

The authors scale the features and the output variable to a [0,1] scale during training. After training, a simple post-processing step is applied where the predicted values are scaled to the

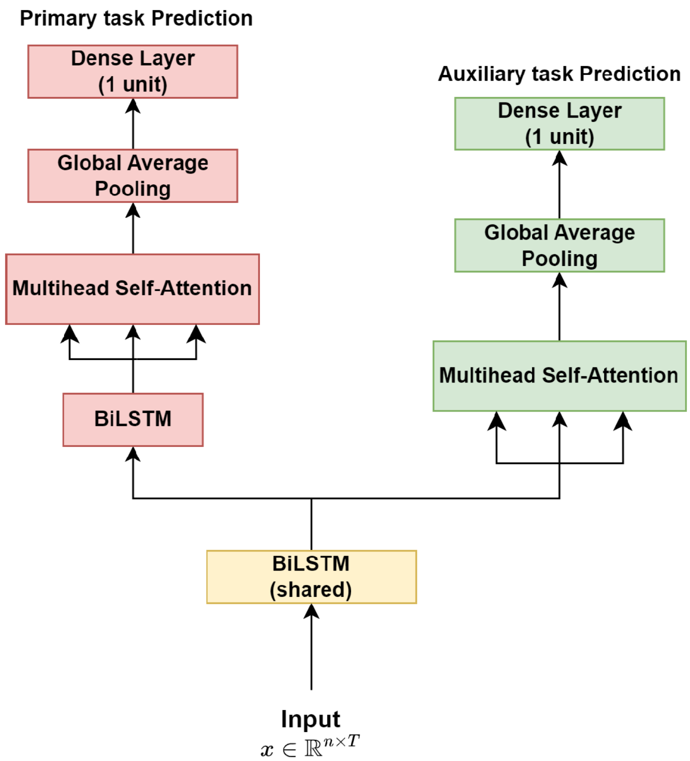

FIGURE 7. The proposed MTL-double encoders model. This model consists of a shared BiLSTM layer and two task-specific branches. The shared BiLSTM layer is updated by both tasks, while each task-specific branch is updated by the corresponding task.

actual range of values. α of Eq. 19 is set equal to 0.1. λ of Eq. 20 is set equal to 1. For STL and MTL-Simple models, the hidden size of the BiLSTM layer is $h = 150$. For the MTL-Double Encoders model, the hidden size of the shared BiLSTM layer is $h = 150$, while the hidden size of the BiLSTM corresponding to the task-specific branch of the primary task is $h = 200$. The authors split the dataset into a train and test set. Additionally, the authors divide the train set into a train and validation set. A batch size of 64 is used. The authors use *EarlyStopping*, where training is stopped if the validation loss has stopped decreasing for 10 consecutive epochs. Adam optimizer [45] is used for all the experiments. The models are trained for a maximum of 200 epochs. The authors repeat the experiments 20 times and report the mean results. The authors use the t-test for significance testing. The authors use PyTorch [46] for performing the experiments. All experiments are trained on a single Tesla P100-PCIE-16GB GPU.

C. EVALUATION METRICS

The authors utilize the metrics mentioned below for evaluating the results of the introduced approaches. Specifically, y_t refers to the real value of the fuel oil consumption, while $\hat{y}_t$ refers to the predicted value of the fuel oil consumption. $\bar{y}$ denotes the mean of the real values of fuel oil consumption.

- **Coefficient of determination:**

$$R^2 = 1 - \frac{\sum_{t=1}^{n} \left(y_t - \hat{y}_t\right)^2}{\sum_{t=1}^{n} \left(y_t - \bar{y}\right)^2} \tag{21}$$

- **Mean Bias Error:**

$$MBE = \frac{1}{n} \sum_{t=1}^{n} (y_t - \hat{y}_t) \tag{22}$$

- **Root Mean Squared Error:**

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_t - \hat{y}_t)^2} \tag{23}$$

- **Mean Absolute Error:**

$$MAE = \frac{\sum_{i=1}^{n} |y_t - \hat{y}_t|}{n} \tag{24}$$

VI. RESULTS

The results of the proposed approaches mentioned in Section IV are reported in Tables 4 and 5.

Regarding the Dataset 1, the authors observe in Table 4 that MTL-Double Encoders constitutes the best performing model achieving the best results across all metrics for predicting the main engine fuel oil consumption. Specifically, the best performing model improves R^2 score by 0.22-17.06%, RMSE presents an improvement of 0.05-2.37, while MAE is also

TABLE 4. Performance comparison among traditional ML algorithms and proposed STL and MTL models on primary prediction task (prediction of main engine fuel oil consumption). Results for STL and MTL approaches are averaged across 20 runs. Best results per evaluation metric are underlined. † indicates significant improvement over STL and MTL-Simple (t-test, p-value<0.05).

	Dataset 1				Dataset 2				Dataset 3			
Architecture	**R^2(%)**	**MBE**	**RMSE**	**MAE**	**R^2(%)**	**MBE**	**RMSE**	**MAE**	**R^2(%)**	**MBE**	**RMSE**	**MAE**
AdaBoostRegressor	77.91	2.19	4.53	4.14	89.19	-0.23	3.49	3.07	90.08	-0.17	4.17	3.85
BaggingRegressor	93.02	0.64	2.55	1.60	95.24	0.19	2.31	1.10	98.88	0.32	1.40	0.57
ExtraTreesRegressor	94.34	0.44	2.29	1.35	95.20	-0.16	2.32	1.02	99.13	0.41	1.23	0.58
GradientBoostingRegressor	94.75	-0.21	2.21	1.19	95.33	0.07	2.29	1.09	95.66	0.36	2.76	1.71
RandomForestRegressor	93.23	0.65	2.51	1.56	95.15	-0.22	2.34	1.08	99.04	0.37	1.30	0.62
STL	94.65	-0.07	2.23	0.99	94.82	-0.26	2.41	1.10	99.21	0.44	1.18	0.62
MTL-Simple	94.74	-0.14	2.21	1.01	94.97	-0.24	2.38	1.12	99.28	0.39	1.13	0.58
MTL-Double Encoders	94.97†	-0.02†	2.16†	0.99	95.39†	-0.09†	2.28†	1.02†	99.45†	0.16†	0.99†	0.36†

TABLE 5. Performance comparison among traditional ML algorithms and proposed STL and MTL models on auxiliary prediction task (prediction of auxiliary engine fuel oil consumption). Results for STL and MTL approaches are averaged across 20 runs. Best results per evaluation metric are underlined.

	Dataset 1				Dataset 2				Dataset 3			
Architecture	**R^2(%)**	**MBE**	**RMSE**	**MAE**	**R^2(%)**	**MBE**	**RMSE**	**MAE**	**R^2(%)**	**MBE**	**RMSE**	**MAE**
AdaBoostRegressor	88.88	-0.12	0.44	0.29	89.92	-0.21	0.53	0.38	81.59	-0.21	0.40	0.32
BaggingRegressor	92.14	0.06	0.37	0.21	92.73	-0.04	0.45	0.24	70.11	-0.29	0.51	0.37
ExtraTreesRegressor	93.00	0.01	0.35	0.19	93.80	-0.01	0.42	0.22	88.57	-0.09	0.32	0.23
GradientBoostingRegressor	92.85	0.003	0.35	0.19	93.60	0.01	0.43	0.21	87.85	-0.15	0.33	0.25
RandomForestRegressor	92.37	0.0004	0.36	0.19	93.40	-0.02	0.43	0.22	74.80	-0.24	0.47	0.33
STL	93.15	0.02	0.34	0.19	94.37	-0.01	0.39	0.19	90.79	-0.12	0.28	0.21
MTL-Simple	92.07	0.02	0.37	0.21	92.52	-0.07	0.46	0.29	83.55	-0.16	0.37	0.28
MTL-Double Encoders	88.57	0.003	0.43	0.26	91.62	-0.07	0.48	0.30	85.70	-0.13	0.35	0.26

improved by 0.02-3.15. The authors observe that the AdaBoost Regressor obtains the worst evaluation results yielding an R^2 score of 77.91%. MTL-Double Encoders outperforms significantly both MTL-Simple and STL in terms of R^2 score by 0.23% (p-value=9.26e-14) and 0.32% (p-value=1.50e-07) respectively. The best performing model obtains better RMSE than MTL simple by 0.05 (p-value=3.50e-05) and STL by 0.07 (p-value=1.50e-07). In terms of MBE, the authors observe that the best performing model achieves the lowest MBE accounting for -0.02 with an improvement over STL and MTL-Simple of 0.05 (p-value=1.10e-05) and 0.12 (p-value=0.00278) respectively. Observing Table 5, the authors observe that the proposed STL model outperforms the introduced MTL approaches. This can be justified by the fact that the authors have set α of Eq. 19 equal to 0.1, placing in this way importance to the task of the prediction of main engine fuel oil consumption. As one can observe in Table 5, the proposed STL model surpasses the performance of traditional ML algorithms in R^2 score by 0.15-4.27% and in RMSE by 0.01-0.10. Although the differences in performance are limited, the authors believe that even a small improvement can make the difference in this field. In terms of the MTL approaches, although the authors do not place enough importance in this task, the authors observe that their approaches yield competitive results with traditional ML algorithms. Specifically, the authors observe that MTL-Simple outperforms AdaBoostRegressor in terms of R^2, MBE, RMSE, and MAE. For instance, MTL-Simple outperforms AdaBoostRegressor in R^2 score by 3.19%, in MBE by 0.10, in RMSE by 0.07, and in MAE by 0.08.

In terms of the Dataset 2, the authors observe in Table 4 that MTL-Double Encoders constitutes the best performing model surpassing the rest of the approaches, i.e., traditional ML algorithms and introduced approaches, in R^2 score by 0.06-6.20% and in RMSE by 0.01-1.21. It achieves a MAE of 1.02 which is equal with the one obtained by ExtraTreesRegressor, while it outperforms the rest approaches by 0.06-2.05. Compared with the introduced approaches, MTL-Double Encoders outperforms STL and MTL-Simple in R^2 score by 0.57% (p-value=1.99e-12) and 0.42% (p-value=6.51e-09) respectively, in MBE by 0.17 (p-value=7.71e-12) and 0.15 (p-value=3.97e-11) respectively, in RMSE by 0.13 (p-value=2.81e-12) and 0.10 (p-value=7.57e-09) respectively, and in MAE by 0.08 (p-value=1.89e-08) and 0.10 (p-value=1.14e-09) respectively. As one can observe in Table 5, the proposed STL model attains an R^2 score of 94.37% outperforming the traditional ML algorithms by 0.57-4.45%. Additionally, STL improves RMSE over traditional ML algorithms by 0.04-0.14 and MAE by 0.02-0.19.

With regards to the Dataset 3, the authors observe in Table 4 that MTL-Double Encoders constitutes the best performing model obtaining an R^2 score of 99.45%, a MBE of 0.16, an RMSE of 0.99, and a MAE of 0.36. Specifically, MTL-Double Encoders outperforms MTL-Simple in R^2 score by 0.17% (p-value=7.89e-09), in MBE by 0.23 (p-value=7.03e-15), in RMSE by 0.14 (p-value=3.11e-09), and in MAE by 0.22 (p-value=7.96e-16). Similarly, MTL-Double Encoders outperforms STL in R^2 score by 0.24% (p-value=4.89e-08), in MBE by 0.28 (p-value=2.09e-13), in RMSE by 0.19 (p-value=4.33e-09), and in MAE by 0.26 (p-value=8.27e-16). In comparison with the traditional ML regressors, MTL-Double Encoders yields a better R^2 score by 0.32-9.37%, improves RMSE by 0.24-3.18, and MAE by 0.21-3.49. As one can easily observe in Table 5, the proposed STL model can predict the auxiliary engine fuel oil consumption attaining an R^2 score of 90.79% which is better than 2.22-20.68% in comparison with the traditional ML algorithms. Regarding the proposed approaches in the MTL framework, one can observe that they achieve competitive results. Specifically, both MTL approaches outperform three ML algorithms in R^2 score.

Overall, one can observe that MTL-Double Encoders constitutes the best performing model across all datasets. We speculate that this is attributable to the fact that this architecture consists of both shared and task-specific branches.

VII. CONCLUSION & FUTURE WORK

In this paper, the authors present the first study exploiting transformer-based approaches for predicting both the main and auxiliary engine fuel oil consumption. Specifically, the authors introduce both single-task and multi-task learning models. In terms of the single-task learning setting, the proposed models consist of BiLSTMs and MultiHead Self-Attention layers. The authors further use a multi-task learning framework to jointly model the main engine fuel oil consumption and auxiliary engine fuel oil consumption as an auxiliary task. In order to address the phenomenon of "mimicking" in time-series forecasting which is a consequence of minimizing the MSE loss, the authors add a regularization term in the loss function. The authors evaluate the proposed approaches in three publicly available datasets, which include sensor data from fishing vessels. Findings show that the proposed MTL approaches outperform significantly the traditional ones.

The significantly increased performance and accuracy of the introduced MTL approach can result to a number of measures and technologies that can contribute to the overall fuel oil consumption and thus the operating costs of shipping companies. By predicting fuel oil consumption through the feature sets of the main and auxiliary engine, more insights can be provided for the ship performance with data based on real operating conditions that can further impact different sectors and activities of the maritime industry. For instance, a better fuel consumption prediction model can further improve the accuracy and performance fuel prediction services of commercial fleet monitoring and stability management software that has been tailored to specific clients/ships (main and auxiliary engines features) and thus enables them for better weather routing optimization planning. Moreover, they can directly contribute to supporting the decision-making process for ship energy design systems for feeding ship energy systems simulations and optimization. Furthermore, it can provide guidance and insights for adjusting a vessel's trim, ballast, and cargo distribution in order to improve its hydrodynamics and reduce fuel consumption usage and optimal trim. For example, an accurate fuel consumption prediction can inform operators about the most fuel-efficient trim for the vessel under different conditions. Moreover, by correlating accurate fuel consumption predictions with varying ballast levels, operators can identify the optimal amount and distribution of ballast. Additionally uneven or suboptimal cargo distribution can result in increased resistance and, consequently, higher fuel consumption. Accurate predictions of fuel consumption, based on different cargo distributions, can guide operators in distributing cargo more efficiently, optimizing the underwater part of the ship and its hydrodynamics. Lastly, it can contribute to proactive maintenance and performance monitoring of the vessel's engines as well as their optimal tuning.

However, this study comes with some limitations. Specifically, the authors did not apply hyperparameter tuning, which often leads to a performance improvement. In addition, the authors did not apply explainability techniques for rendering the proposed approaches explainable. Finally, the authors experimented only with fishing ships and did not test other ship types.

In the future, the authors aim to propose deep learning approaches for the task of route optimization. Also, the prediction of CO_2 emissions is one of their future plans. Additionally, the authors aim to apply the proposed models to other ship types. Finally, the authors aim to contribute to this field by proposing explainable deep neural networks.

REFERENCES

[1] H. N. Psaraftis and C. A. Kontovas, "Speed models for energy-efficient maritime transportation: A taxonomy and survey," *Transp. Res. C, Emerg. Technol.*, vol. 26, pp. 331–351, Jan. 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X12001246

[2] C. Bagoulla and P. Guillotreau, "Maritime transport in the French economy and its impact on air pollution: An input–output analysis," *Mar. Policy*, vol. 116, Jun. 2020, Art. no. 103818. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0308597X19305408

[3] P. P. Vinayak, C. S. K. Prabu, N. Vishwanath, and S. O. Prakash, "Numerical simulation of ship navigation in rough seas based on ECMWF data," *Brodogradnja*, vol. 72, no. 1, pp. 19–58, Mar. 2021.

[4] A. Farkas, N. Degiuli, I. Martić, and A. Mikulić, "Benefits of slow steaming in realistic sailing conditions along different sailing routes," *Ocean Eng.*, vol. 275, May 2023, Art. no. 114143. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801823005279

[5] N. Degiuli, I. Martić, A. Farkas, and I. Gospić, "The impact of slow steaming on reducing CO_2 emissions in the Mediterranean Sea," *Energy Rep.*, vol. 7, pp. 8131–8141, Nov. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S235248472100144X

[6] A. Farkas, N. Degiuli, I. Martić, and C. G. Grlj, "Is slow steaming a viable option to meet the novel energy efficiency requirements for containerships?" *J. Cleaner Prod.*, vol. 374, Nov. 2022, Art. no. 133915. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959652622034874

[7] C. Dere and C. Deniz, "Load optimization of central cooling system pumps of a container ship for the slow steaming conditions to enhance the energy efficiency," *J. Cleaner Prod.*, vol. 222, pp. 206–217, Jun. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959652619307139

[8] C. G. Grlj, N. Degiuli, Ž. Tuković, A. Farkas, and I. Martić, "The effect of loading conditions and ship speed on the wind and air resistance of a containership," *Ocean Eng.*, vol. 273, Apr. 2023, Art. no. 113991. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S002980182300375X

[9] M. Kalajdžić, M. Vasilev, and N. Momčilović, "Power reduction considerations for bulk carriers with respect to novel energy efficiency regulations," *Brodogradnja: Teorija i praksa brodogradnje i pomorske tehnike*, vol. 73, no. 2, pp. 79–92, 2022.

[10] C. Sui, P. de Vos, D. Stapersma, K. Visser, and Y. Ding, "Fuel consumption and emissions of ocean-going cargo ship with hybrid propulsion and different fuels over voyage," *J. Mar. Sci. Eng.*, vol. 8, no. 8, p. 588, Aug. 2020. [Online]. Available: https://www.mdpi.com/2077-1312/8/8/588

[11] A. F. Molland, "Chapter 6—Marine engines and auxiliary machinery," in *The Maritime Engineering Reference Book*, Oxford: Butterworth-Heinemann, 2008, pp. 344–482. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780750689878000068

[12] D. Bocchetti, A. Lepore, B. Palumbo, and L. Vitiello, "A statistical control of the ship fuel consumption," in *Proc. Int. Conf. Design, Construct. Operation Passenger Ships, Roy. Inst. Nav. Architects*, Nov. 2013, pp. 20–21.

[13] M. Jeon, Y. Noh, Y. Shin, O.-K. Lim, I. Lee, and D. Cho, "Prediction of ship fuel consumption by using an artificial neural network," *J. Mech. Sci. Technol.*, vol. 32, no. 12, pp. 5785–5796, Dec. 2018.

[14] P. Karagiannidis and N. Themelis, "Data-driven modelling of ship propulsion and the effect of data pre-processing on the prediction of ship fuel consumption and speed loss," *Ocean Eng.*, vol. 222, Feb. 2021, Art. no. 108616. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801821000512

[15] C. Kosma, G. Nikolentzos, N. Xu, and M. Vazirgiannis, "Time series forecasting models copy the past: How to mitigate," in *Proc. Artif. Neural Netw. Mach. Learn. (ICANN)*, E. Pimenidis, P. Angelov, C. Jayne, A. Papaleonidas, and M. Aydin, Eds. Cham, Switzerland: Springer, 2022, pp. 366–378.

[16] L. Ilias and D. Askounis, "Explainable identification of dementia from transcripts using transformer networks," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 8, pp. 4153–4164, Aug. 2022.

[17] S. Rajamanickam, P. Mishra, H. Yannakoudakis, and E. Shutova, "Joint modelling of emotion and abusive language detection," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4270–4279. [Online]. Available: https://aclanthology.org/2020.acl-main.394

[18] M. Jin and N. Aletras, "Modeling the severity of complaints in social media," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, Jul. 2021, pp. 2264–2274. [Online]. Available: https://aclanthology.org/2021.naacl-main.180

[19] B. R. Paredes, A. Argyriou, N. Berthouze, and M. Pontil, "Exploiting unrelated tasks in multi-task learning," in *Proc. 15th Int. Conf. Artif. Intell. Statist.*, vol. 22, N. D. Lawrence and M. Girolami, Eds. La Palma, Canary Islands: PMLR, Apr. 2012, pp. 951–959. [Online]. Available: https://proceedings.mlr.press/v22/romera12.html

[20] C. Gkerekos, I. Lazakis, and G. Theotokatos, "Machine learning models for predicting ship main engine fuel oil consumption: A comparative study," *Ocean Eng.*, vol. 188, Sep. 2019, Art. no. 106282. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801819304561

[21] E. B. Beşikçi, O. Arslan, O. Turan, and A. I. Ölçer, "An artificial neural network based decision support system for energy efficient ship operations," *Comput. Oper. Res.*, vol. 66, pp. 393–401, Feb. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054815000842

[22] J. P. Petersen, O. Winther, and D. J. Jacobsen, "A machine-learning approach to predict main energy consumption under realistic operational conditions," *Ship Technol. Res.*, vol. 59, no. 1, pp. 64–72, Jan. 2012, doi: 10.1179/str.2012.59.1.007.

[23] M. Simonsen, H. Walnum, and S. Gössling, "Model for estimation of fuel consumption of cruise ships," *Energies*, vol. 11, no. 5, p. 1059, Apr. 2018. [Online]. Available: https://www.mdpi.com/1996-1073/11/5/1059

[24] K.-K. Kee, B.-Y. L. Simon, and K.-H. Y. Renco, "Prediction of ship fuel consumption and speed curve by using statistical method," *J. Comput. Sci. Comput. Math.*, vol. 8, no. 2, pp. 19–24, Jun. 2018.

[25] A. Lepore, M. S. dos Reis, B. Palumbo, R. Rendall, and C. Capezza, "A comparison of advanced regression techniques for predicting ship CO_2 emissions," *Qual. Rel. Eng. Int.*, vol. 33, no. 6, pp. 1281–1292, Oct. 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/qre.2171

[26] T. Uyanık, Ç. Karatuğ, and Y. Arslanoğlu, "Machine learning approach to ship fuel consumption: A case of container vessel," *Transp. Res. D, Transp. Environ.*, vol. 84, Jul. 2020, Art. no. 102389. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361920920305769

[27] I. Panapakidis, V.-M. Sourtzi, and A. Dagoumas, "Forecasting the fuel consumption of passenger ships with a combination of shallow and deep learning," *Electronics*, vol. 9, no. 5, p. 776, May 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/5/776

[28] Z. Hu, Y. Jin, Q. Hu, S. Sen, T. Zhou, and M. T. Osman, "Prediction of fuel consumption for enroute ship based on machine learning," *IEEE Access*, vol. 7, pp. 119497–119505, 2019.

[29] S. Wang, B. Ji, J. Zhao, W. Liu, and T. Xu, "Predicting ship fuel consumption based on LASSO regression," *Transp. Res. D, Transp. Environ.*, vol. 65, pp. 817–824, Dec. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361920917302109

[30] T. Zhou, Q. Hu, Z. Hu, and R. Zhen, "An adaptive hyper parameter tuning model for ship fuel consumption prediction under complex maritime environments," *J. Ocean Eng. Sci.*, vol. 7, no. 3, pp. 255–263, Jun. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2468013321000759

[31] Y. Peng, H. Liu, X. Li, J. Huang, and W. Wang, "Machine learning method for energy consumption prediction of ships in port considering green ports," *J. Cleaner Prod.*, vol. 264, Aug. 2020, Art. no. 121564. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959652620316115

[32] Z. Yuan, J. Liu, Q. Zhang, Y. Liu, Y. Yuan, and Z. Li, "Prediction and optimisation of fuel consumption for inland ships considering real-time status and environmental factors," *Ocean Eng.*, vol. 221, Feb. 2021, Art. no. 108530. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801820314372

[33] Y.-R. Kim, M. Jung, and J.-B. Park, "Development of a fuel consumption prediction model based on machine learning using ship in-service data," *J. Mar. Sci. Eng.*, vol. 9, no. 2, p. 137, Jan. 2021. [Online]. Available: https://www.mdpi.com/2077-1312/9/2/137

[34] Z. Hu, T. Zhou, R. Zhen, Y. Jin, X. Li, and M. T. Osman, "A two-step strategy for fuel consumption prediction and optimization of ocean-going ships," *Ocean Eng.*, vol. 249, Apr. 2022, Art. no. 110904. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801822003407

[35] Z. Hu, T. Zhou, M. T. Osman, X. Li, Y. Jin, and R. Zhen, "A novel hybrid fuel consumption prediction model for ocean-going container ships based on sensor data," *J. Mar. Sci. Eng.*, vol. 9, no. 4, p. 449, Apr. 2021. [Online]. Available: https://www.mdpi.com/2077-1312/9/4/449

[36] R. Yan, S. Wang, and Y. Du, "Development of a two-stage ship fuel consumption prediction and reduction model for a dry bulk ship," *Transp. Res. E, Logistics Transp. Rev.*, vol. 138, Jun. 2020, Art. no. 101930. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1366554519308555

[37] Y. B. A. Farag and A. I. Ölçer, "The development of a ship performance model in varying operating conditions based on ANN and regression techniques," *Ocean Eng.*, vol. 198, Feb. 2020, Art. no. 106972. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801820300536

[38] Y. Du, Q. Meng, S. Wang, and H. Kuang, "Two-phase optimal solutions for ship speed and trim optimization over a voyage using voyage report data," *Transp. Res. B, Methodol.*, vol. 122, pp. 88–114, Apr. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0191261517305738

[39] C. Zhang, D. Zhang, M. Zhang, and W. Mao, "Data-driven ship energy efficiency analysis and optimization model for route planning in ice-covered Arctic waters," *Ocean Eng.*, vol. 186, Aug. 2019, Art. no. 106071. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801819302744

[40] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[42] P. Siltanen and Z. U. Arrue, "Sensor data from three different fishing ships for a period of one month," Zenodo, Version 1.0.0, Dec. 2019. [Online]. Available: https://zenodo.org/records/3563390, doi: 10.5281/zenodo.3563390.

[43] C. Chen, D. Han, and C.-C. Chang, "CAAN: Context-aware attention network for visual question answering," *Pattern Recognit.*, vol. 132, Dec. 2022, Art. no. 108980. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320322004605

[44] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.
[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[46] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

LOUKAS ILIAS received the integrated master's degree from the School of Electrical and Computer Engineering (SECE), National Technical University of Athens (NTUA), Athens, Greece, in June 2020, where he is currently pursuing the Ph.D. degree with the Decision Support Systems (DSS) Laboratory, SECE. He has completed a Research Internship with University College London (UCL), London, U.K. He is a Researcher with the DSS Laboratory, NTUA, where he is involved in EU-funded research projects. He has published in numerous journals, including IEEE Journal of Biomedical and Health Informatics, IEEE Transactions on Computational Social Systems, *Knowledge-Based Systems* (Elsevier), *Expert Systems With Applications* (Elsevier), *Applied Soft Computing* (Elsevier), *Computer Speech and Language* (Elsevier), *Online Social Networks and Media* (Elsevier), IEEE Access, and *Frontiers in Aging Neuroscience*. His research has also been accepted for presentation at international conferences, including the IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI'22) and the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2023. His research interests include speech processing, natural language processing, social media analysis, and the detection of complex brain disorders.

PANAGIOTIS KAPSALIS received the Dipl.-Ing. degree in computer and informatics engineering from the University of Patras, in 2016, and the M.Sc. degree in applied data science from the Athens University of Economics and Business, in 2017. He is currently pursuing the Ph.D. degree with the National Technical University of Athens. Since 2017, he has been a Research Software Engineer in EU funded projects. He has presented his research at international conferences. His research interests include information systems, big data systems and data management, and software engineering.

SPIROS MOUZAKITIS received the Ph.D. degree from the School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece, in 2009. He is currently a Senior Research Analyst with the Decision Support Systems Laboratory, National Technical University of Athens. He has more than ten years of industry experience in software engineering for banking, e-commerce systems, and energy suppliers. He was a Project Manager for 16 years in more than 20 projects (H2020, FP7, FP6, and FP5 research projects) related to big data, open data, market/impact analysis in the context of ICT technologies, web development, and enterprise interoperability. He has published in numerous journals and presented his research at international conferences. His current research interests include decision analysis in the field of decision support systems based on machine learning/deep learning, big and linked data analytics, and optimization systems and algorithms.

DIMITRIS ASKOUNIS for a number of years, was an Advisor with the Minister of Justice and the Special Secretary of the Digital Convergence for the introduction of information and communication technologies in public administration. Since June 2019, he has been the President of the Information Society SA, Kallithea, Greece. He is currently a Professor with the School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), Athens, Greece, and the Deputy Director of the Decision Support Systems Laboratory. He has over 25 years of experience in decision support systems, intelligent information systems and manufacturing, e-business, e-government, open and linked data, big data analytics, artificial intelligence (AI) algorithms, and the application of modern information technology (IT) techniques in the management of companies and organizations. He was the Scientific Director of over 50 European research projects in the above areas (FP7 and Horizon2020).