applied sciences

MDPI

Article

Time-Aware and Feature Similarity Self-Attention in Vessel Fuel Consumption Prediction

Hyun Joon Park [1], Min Seok Lee [1], Dong Il Park [2] and Sung Won Han [1,*]

[1] School of Industrial and Management Engineering, Korea University, 145, Anam-Ro, Seongbuk-Gu, Seoul 02841, Korea; winddori2002@korea.ac.kr (H.J.P.); karel@korea.ac.kr (M.S.L.)
[2] Seavantage, Nonhyeon-dong 201-6beon-ji, Seoul 06120, Korea; dipark@seavantage.com
* Correspondence: swhan@korea.ac.kr

Abstract: An accurate vessel fuel consumption prediction is essential for constructing a ship route network and vessel management, leading to efficient sailings. Besides, ship data from monitoring and sensing systems accelerate fuel consumption prediction research. However, the ship data consist of three properties: sequential, irregular time interval, and feature importance, making the predicting problem challenging. In this paper, we propose Time-aware Attention (TA) and Feature-similarity Attention (FA) applied to bi-directional Long Short-Term Memory (LSTM). TA acquires time importance by nonlinear function from irregular time intervals in each sequence and emphasizes data depending on the importance. FA emphasizes data based on similarities of features in the sequence by estimating feature importance with learnable parameters. Finally, we propose the ensemble model of TA and FA-based BiLSTM. The ensemble model, which consists of fully connected layers, is capable of simultaneously capturing different properties of ship data. The experimental results on ship data showed that the proposed model improves the performance in predicting fuel consumption. In addition to model performance, visualization results of attention maps and feature importance help to understand data properties and model characteristics.

Keywords: BiLSTM; deep learning; ensemble; feature similarity attention; self-attention; time-aware attention; vessel fuel consumption prediction

Citation: Park, H.J.; Lee, M.S.; Park, D.I.; Han, S.W. Time-Aware and Feature Similarity Self-Attention in Vessel Fuel Consumption Prediction. *Appl. Sci.* **2021**, *11*, 11514. https://doi.org/10.3390/app112311514

Academic Editors: Andrea Prati, Carlos A. Iglesias, Luis Javier García Villalba and Vincent A. Cicirello

Received: 9 November 2021
Accepted: 1 December 2021
Published: 4 December 2021

1. Introduction

Since the popularization of international trade, the shipping industry has made technical progress, making ships emerge as efficient transport [1]. Despite the efficiency of ships, various sailings consisting of approximately 90% of global trade result in environmental problems [2]. Due to environmental issues, the International Maritime Organization (IMO) announced a regulation, which mandates ships to use fuel containing less than 0.5% Sulfur. However, it is expensive to replace the vessel fuel, and this regulation increases the necessity of efficient sailings.

Including the efficiency of the ship, the technical progress of the shipping industry has made it feasible to get not only sufficient ship data from sensing and monitoring systems, but also weather data to consider external factors. These data have been used to develop methods for predicting vessel fuel consumption [3–5]. These studies have been essential for efficient sailings by constructing a ship route network for navigational strategies, route planning, and managing vessel operation. Several attempts have been made to predict fuel consumption through statistical and machine learning models. However, researchers have recently used deep learning models, and they have shown that it outperforms the previous methods [4,6].

For the past decades, many researchers in various domains have made significant progress and achieved remarkable performance in deep learning. In the vessel domain, some studies were conducted to predict fuel consumption using deep learning. The previous methods can be divided into two parts depending on the kinds of deep learning

models. First, the authors of [3–8] used *Multi-Layer Perceptron* (MLP). The authors of [3,5,8] showed that the deep learning models outperformed the machine learning and statistical models. They also discussed the usage of the proposed model for efficient sailings. The authors of [4,6] focused on increasing the model performance by handling important features and outliers. Although the authors of [7] adopted MLP, they addressed the importance of sequential property in ship data. On the other hand, the authors of [9–11] applied *Long Short-Term Memory* (LSTM) [12] to consider sequential ship data. Among them, the authors of [9,10] proved that LSTM outperformed MLP and other machine learning models which could not reflect sequential information. Although previous studies contributed to efficient sailing by predicting fuel consumption, they did not fully consider the ship data properties. Without understanding data, it is challenging to design a proper model for the data. Thus, we first explore ship data and describe their properties. Because ship data are acquired in time order, they have the characteristics of the temporal feature data. Based on the characteristics, we describe ship data properties in detail. The three main properties are given below:

1. Sequential data: Ship data acquisition from sensing or monitoring systems occurs consecutively during sailings.
2. Irregular time interval: Each data interval in the sequence (i.e., sailing) can vary from seconds to hours owing to the vessel type, sailing status or transport malfunction [7]. Besides, time irregularity can aggravate during processing of numerous noise and missing data. Figure 1 shows that the irregular time interval occurs also in the dataset in this paper.
3. Feature importance: The vessel fuel oil consumption is significantly affected by a few features, which have a high correlation with fuel oil consumption [7,10,13]. As shown in Figure 2, we deduce that fuel oil consumption has a high correlation with speed and draft compared to other features. As their correlation coefficients are positive, fuel oil consumption could increase as speed and draft become higher.

Although we describe the above three properties, there are some overlapped aspects with each property. For example, the irregular time interval is an implicit property of the sequential data. However, we distinguish the properties in detail to fully reflect individual properties on the models.

In this study, we propose each attention-based model and their ensemble model to accurately predict fuel oil consumption by representing the ship data properties. We use bi-directional LSTM [14] as the backbone model to consider sequential property. For irregular time interval property, we design **Time-aware Attention** (TA) based on self-attention. TA represents irregular time information and alleviates the problem of time irregularity in sequence models. To reflect the feature importance property, we propose **Feature-similarity Attention** (FA). FA, which estimates feature importance, emphasizes data based on feature similarity. Finally, we adopt the **ensemble** model of TA and FA-based BiLSTM, which makes it feasible to reflect two different properties simultaneously. Experimental results on the ship data show that the proposed models outperform the other sequence and attention-based models. Furthermore, attention maps of different scenarios and feature importance values illustrate the relationship between data properties and models, which increases interpretability.

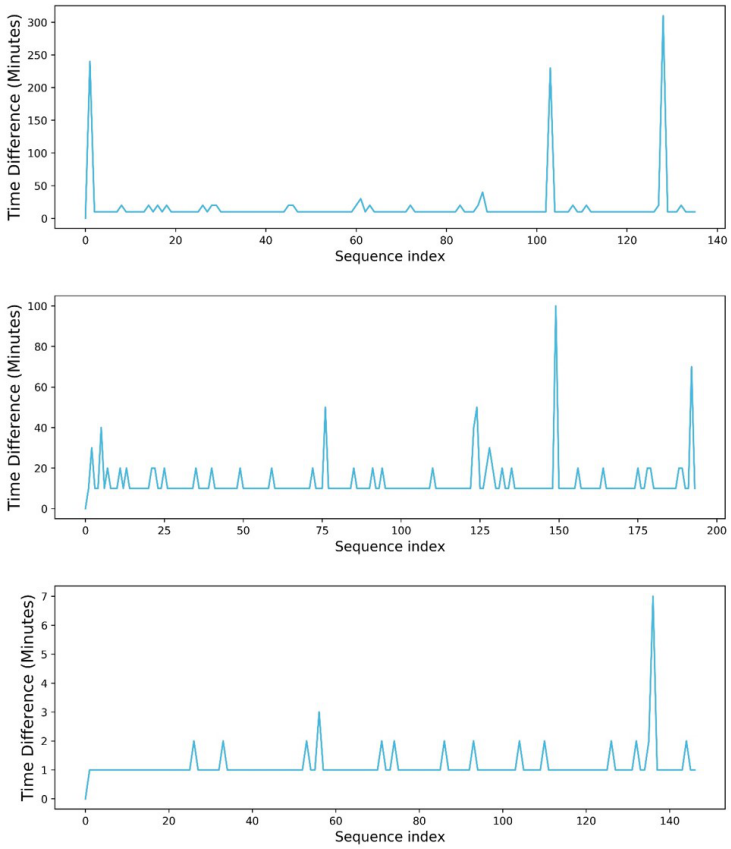

Figure 1. Irregular time interval plot examples. Each plot shows acquired sequential ship data during each voyage and their time difference. The time difference is calculated by $t_{(i+1)} - t_i$, where t_i is the time when the ith data was acquired during the voyage. The examples are sampled from the dataset used in this paper.

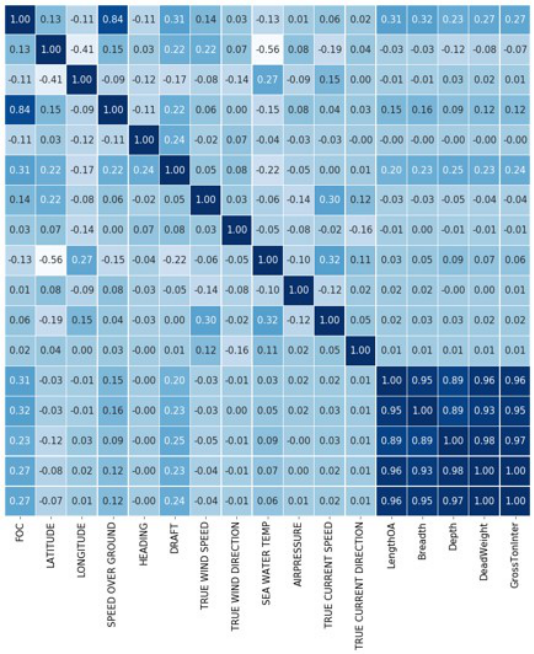

Figure 2. Correlations between features including a dependent variable, fuel oil consumption (FOC). In the correlation matrix, the correlation is higher as the color becomes darker. The correlation is calculated by the dataset used in this paper.

The rest of the paper is organized into various sections, beginning with a summary of related works. This is followed by the proposed methods which contain TA, FA, and its ensemble. Finally, we address the experimental results and analyze them using visualizations.

2. Related Work

In this section, we address three topics of related work (i.e., sequence model, attention mechanism, and feature importance). We also investigate the previous studies of applying deep learning models in the vessel fuel consumption domain.

2.1. Sequence Model

Recurrent Neural Network (RNN) [15,16] is one of the methods used to represent sequential information and handle a variable-length sequence. However, RNN has gradient vanishing problems in a long length sequence [17]. *Long Short-Term Memory* (LSTM) was proposed in [12], to capture short- and long-term memory. LSTM consists of three separated gates to update hidden and cell states. The process of updating states is given as follows:

$$\begin{aligned} i_t &= \sigma(W_i x_t + U_i h_t - 1 + b_i) \\ f_t &= \sigma(W_f x_t + U_f h_t - 1 + b_f) \\ o_t &= \sigma(W_o x_t + U_o h_t - 1 + b_o) \\ \tilde{c_t} &= tanh(W_c x_t + U_c h_{t-1}) \\ c_t &= f_t \odot c_t - 1 + i_t \odot \tilde{c_t} \\ h_t &= o_t \odot tanh(c_t) \end{aligned} \tag{1}$$

where x_t is the current input of time step t, and σ is sigmoid function. Each i_t, f_t, and o_t denotes the input, forget and output gates, and bias for b, respectively. These gates control the exposure of memory and assist in preserving short and long memory. After the success of LSTM, *Gated Recurrent Unit* (GRU) was proposed in [18] as another type of recurrent unit. RNNs achieved remarkable performance where sequential information was needed, such as *Natural Language Processing* (NLP) and time-series domain.

In the vessel fuel consumption prediction, sequence models are essential as data acquisition occurs consecutively during sailings. However, the irregularity of time interval made it challenging to apply sequence models [7]. This is because RNNs share parameters regardless of the irregular time interval. It means that RNNs equally update information in the sequence, even when the time interval is not equal. Other researchers applied LSTM in the vessel domain, but they did not consider irregularity of time [9,11]. This implies the necessity of a sequence model addressing irregular time intervals.

2.2. Attention Mechanism

The attention mechanism is developed, especially in the machine translation domain. It was introduced in [19] to preserve previous information and emphasize significant data in the sequence. This improved model performance and alleviated the gradient vanishing problem of RNNs since it used weighted previous information again. The achievement of attention accelerated the development of other attention. Ref. [20] compared various alignment functions and verified the global and local approach for attention. Ref. [21] used attention based *bi-directional* LSTM (BiLSTM) for sentiment classification.

Recently, Transformer [22] achieved state-of-the-art performance in machine translation. Transformer's self-attention consists of a scaled dot product between Q, K, and V, which refer to query, key, and value, respectively. Each Q, K, and V is projected by input X and its corresponding weights W_q, W_k, and W_v. The self-attention calculates the dot products between the query and key, divides them by $\sqrt{d_k}$ as a scaling factor, where d_k is the dimension of features. It additionally applies a softmax function to obtain the attention

weight, α, and is used to emphasize data on value. The process of self-attention is defined as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^\top}{\sqrt{d_K}})V \quad (2)$$

Using the outputs of the previous layer as the inputs of self-attention, it emphasizes significant information in a sequence. Due to good compatibility, other studies employed a part of Transformer's self-attention and applied it on RNNs [23,24]. Besides, the attention mechanism's effectiveness was not confined to machine translation, it was proved in diverse domains (e.g., NLP, recommendation, and time series) [25–27].

In the irregular time interval domain, several studies have been conducted to apply self-attention to address different time intervals in a sequence [28,29]. The authors of [29] used a Transformer encoder, a symmetric time interval matrix, and a position matrix as inputs to consider irregular time in the recommendation system. It also applied embedding to represent hidden information from the time interval and position information. The attention process is similar to the Transformer's self-attention. It calculates the scaled dot product between Q, K', and V', where $K' = K + E_K^R + E_K^P$ and $V' = V + E_V^P + E_V^P$. Each E^R and E^P is the result of embedding the time interval and position information. The attention process is as follows:

$$Attention(Q, K', V') = softmax(\frac{QK'^\top}{\sqrt{d_K}})V' \quad (3)$$

The embedding of the irregular time intervals was useful in obtaining the hidden information and time meaning of users when constructing the recommendation system. Inspired by the work in [29], the difference of the domain and time meaning motivated us to propose other approaches to represent the time interval. The irregular time interval occurs in the vessel domain mainly because of the problem of data acquisition and preprocessing for noise. The irregular time in the previous study reflects the behavior information of the specific user, whereas the irregular time in the vessel does not represent hidden information of the specific vessel. This indicates it is necessary to make representations considering discontinuous time steps above all things.

2.3. Feature Importance

Addressing significant features has been studied for a long time. One of the methods is feature selection based on criteria. The correlation coefficient and mutual information are used to measure the relationship between features [30]. The features are selected depending on the criteria values and their threshold [31].

The features can also be addressed during the model training phase. Various word embedding techniques [32] were adopted for the model to learn how to express the features (i.e., words). The authors of [33,34] focused on weighting features and forced to learn it. They regarded the frequencies or correlation with a target as the feature importance. As feature importance became a factor to weight features, the model could make better feature representations. Weighting features also made both the outstanding model performance and explanation of features feasible.

In the vessel fuel consumption prediction, there were efforts to handle important features [6,7]. They attempted to reflect the feature importance by selecting them based on correlation with a target. As the feature selection is just preprocessing to modify data, the model still regards the remaining features equally [33]. Besides, not only the feature selection requires the subjective deduction of the threshold, but also trials and errors to verify it. This indicates the necessity that the model itself regards features differently based on their importance. Researchers have recently shown that attention mechanisms can make the model itself handle important features [35–39]. However, there are difficulties in the application of the previous methods in the vessel domain. The authors of [37] only focused on replacing the feature selection method using attention modules. In addition, the

authors of [39] addressed the importance of feature interaction in the sparse feature space, and the authors of [35,38] used the estimated feature importance in the unstructured data. Although the authors of [36] proposed self-attention to estimate the feature importance in tabular data, the feature importance was acquired on the instance-level, which did not reflect the feature interaction in the sequence. Those backgrounds motivated us to develop attention modules to handle important features in ship data.

3. Methodology

In this section, we first address the base sequence model which represents sequential data before attention layers. To consider sequence enough, we use BiLSTM as the sequence model. Subsequently, we introduce Time-aware Attention (TA) which is capable of capturing importance based on the time interval. It consists of transformation to the time distance matrix, global time scaling, and function-based representation. After that, we propose Feature-similarity Attention (FA) which includes obtaining feature similarity based on feature importance. Finally, we propose the ensemble model of TA and FA, which can consider both irregular time interval and feature similarity. As shown in Figure 3, each two attention layer combines with self-attention and operates after the BiLSTM layer.

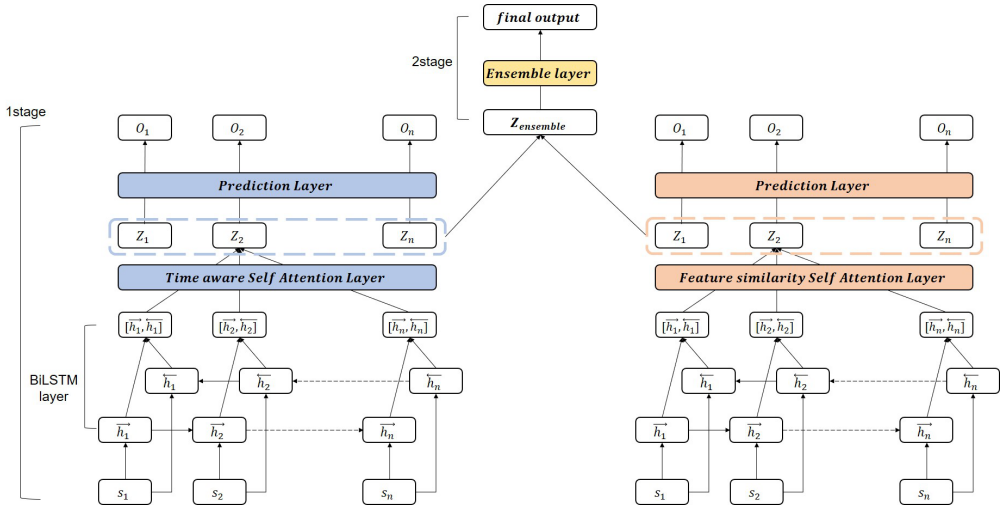

Figure 3. The structure of the ensemble model.

3.1. Sequence Model

3.1.1. Sequence Processing

Ship data have sequential property as the data acquisition occurs consecutively during the sailing. Before introducing the sequence model in this study, it is necessary to describe how we convert sailings into sequences, which are inputs of the model. Although we can roughly define sequences based on sailings, sequence-length has a large deviation depending on sailings. The deviation occurs owing to different sailing distances or preprocessing of noise and missing data. Thus, we use the truncated sequence per sailing, $S = \{s_1, s_2, \cdots, s_n\}$, where n indicates the fixed max-length to truncate the sequence. If the length of S does not satisfy fixed max-length, we pad them to zeros for guaranteeing the equal sequence length. Each s consists of the k number of features (e.g., speed, heading, and draft).

3.1.2. BiLSTM Layer

We use BiLSTM as the base sequence model, which can fully consider both the forward and backward direction. BiLSTM is combined by each forward and backward directional LSTM. Each directional LSTM generates two hidden states, when input vector is the truncated sequence S. The concatenation of two hidden states is the result of BiLSTM layer, $h_i \in \mathbb{R}^{n \times d}$, where $i = \{1, 2, \cdots, n\}$, d is the hidden dimension size, and $h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}]$. h_i is

the sequential representation in the truncated sequence. As shown in Figure 3, the result of BiLSTM, h_i is used as the input of each attention layer.

3.2. Time-Aware Attention

As mentioned in Section 2.2, as self-attention does not represent irregular time in the sequence, we propose TA to represent irregular time property. In addition, instead of finding the hidden meaning of the time like in [29], TA uses time information to connect discontinuous time steps in sequence models and emphasize data depending on time importance. We assume that time importance would increase as the time interval decreases, and design TA to reflect the time importance. It is divided into two parts, function-based representation and combining with self-attention. First, we obtain time importance by function-based representation. It also consists of the process to get a time distance matrix and scale that matrix. After that, we get the time attention weight from the time importance by combining it with self-attention. We combine with self-attention to capture hidden information from data as TA is mainly designed to focus on time information.

3.2.1. Time Importance by Function-Based Representation

From each timestamp in the sequence S, we design a time distance matrix, $T \in \mathbb{R}^{n \times n}$, by calculating the time difference between each timestamp in the sequence. We can observe relative time distance in the sequence from t_{ij}, which indicates time difference between s_i and s_j.

$$T = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1n} \\ t_{21} & t_{22} & \cdots & t_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n1} & t_{n2} & \cdots & t_{nn} \end{pmatrix} \tag{4}$$

In the n by n symmetric matrix T, all of the diagonal elements are zero and others have as larger values as far from each element. We redefine the scaled T as $T' = \frac{1}{\log(e+T)}$ [40] to consider a global time relationship and represent time values into the range $[0, 1]$. Because of scaling, elements of T' are inversely represented by a time interval. We estimate the time importance, α_t, from T'. T' is transformed by the function representation and a softmax function. For the function representation, we use the sigmoid function, which transforms from scaled time interval to time importance. The sigmoid function consists of learnable parameters, gradient a and constant b.

$$f(x) = \frac{1}{1 + \exp^{-ax+b}}, \quad \text{s.t.} \quad a > 0 \tag{5}$$

In Equation (5), we force the range of the gradient a to be positive values. This is because we assume time importance would increase as the time interval decreases. Furthermore, the scaling process makes T have an inverse relationship between time interval and time importance. Thus, the assumption can be satisfied by forcing the range of a. Finally, we express the time importance with softmax, $\alpha_t = softmax(f(T'))$.

3.2.2. Time Attention Weight by Combining with Self-Attention

After getting the time importance α_t, we obtain time attention weight by combining with self-attention. Time attention weight is used to make time-aware representation. We adopt Transformer's self-attention to combine with our attention. $Q \in \mathbb{R}^{n \times d}$, $K \in \mathbb{R}^{n \times d}$, $V \in \mathbb{R}^{n \times d}$ of self-attention are represented by each weight, $W_{q,k,v} \in \mathbb{R}^{d \times d}$, and the hidden states of BiLSTM, $h \in \mathbb{R}^{n \times d}$. d is the hidden dimension size, and it is maintained during the attention process.

We transform Q, K into $Q_t = \alpha_t \cdot Q \in \mathbb{R}^{n \times d}, K_t = \alpha_t \cdot K \in \mathbb{R}^{n \times d}$. They are new representations considering time importance. As shown in Figure 4, each query–key pair, $Q_t - K_t$, $Q - K$, is used for scaled-dot product, where d_k is the scaling factor of dimension

size. The summation of each result is represented by hyperbolic tangent to get attention weight α_{TA}. We use hyperbolic tangent to expand the range of attention representation, considering the summation process.

$$\alpha_{TA} = tanh(\frac{QK^\top}{\sqrt{d_k}} + \frac{Q_t K_t^\top}{\sqrt{d_k}}) \tag{6}$$

$$Z_t = dropout(layernorm(FFN(V_t))) + h \tag{7}$$

The value of TA, $V_t \in \mathbb{R}^{n \times d}$, is produced by time attention weight, $V_t = \alpha_{TA} \cdot V$. After that, we get time-aware representation $Z_t \in \mathbb{R}^{n \times d}$ from Equation (7), which consists of layer normalization, feed-forward network, and dropout as the Transformer did [22]. The layer normalization is used to normalize the inputs and stabilize the learning process. As the hidden dimension size is maintained, we can apply the skip-connection and make the model to be optimized easily. Finally, Z_t gets through the prediction layer, which consists of three fully connected layers.

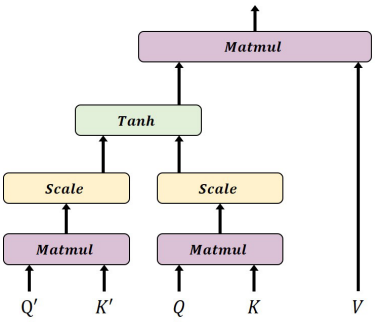

Figure 4. Attention process to combine with self-attention.

3.3. Feature-Similarity Attention

In this subsection, we introduce FA which emphasizes information based on feature similarity and importance. Unlike the previous attention module [38], FA estimates feature importance in sequential data and transforms it to make representations with the total feature similarity. In detail, we use learnable feature importance parameters to obtain the total similarity. This indicates that FA itself learns which feature is important, and reflects it through the total similarity. FA consists of two parts, total similarity and combining with self-attention. We first get a feature-wise distance which is the similarity between data. The total similarity is the weighted sum between feature-wise distance and feature importance. Through total similarity, we get feature attention weight and make feature similarity representation. As mentioned in Section 3.2, we combine with self-attention to obtain hidden information, which is different from FA.

3.3.1. Total Similarity Based on Feature Importance

To obtain the total similarity, we use feature-wise distance and the feature importance. The feature-wise distance is $L1$ distance between pairs of the features. We defined the truncated sequence, $S = \{s_1, s_2, \cdots, s_n\}$, which consists of nth sequential data. Each of s_i has the k number of features except timestamp. Among k, we only use the $k - p$ number of features to estimate feature similarity. k is the total number of vessel features, and p is the number of vessel formulation features. In truncated sequence S, vessel formulation features always have the same values since the types of vessels are the same. That is why we exclude p from total features. The feature-wise distance, $D \in \mathbb{R}^{n \times n \times |k-p|}$ is applied to each feature between data in the sequence S and the equation is as below:

$$D = \left| x_f^i - x_f^j \right| \tag{8}$$

where $i = \{1, 2, \cdots, n\}$, $j = \{1, 2, \cdots, n\}$, $f = \{1, 2, \cdots, k-p\}$, and x_f^i is fth feature of ith data in sequence S. From feature-wise distance D, we observe how each feature of data in sequence is different. However, it does not contain enough information since D is simply calculated by $L1$ distance. We estimate the total similarity $TS \in \mathbb{R}^{n \times n}$ to differently reflect feature-wise distance based on feature importance. TS can compare the overall distance between data. We apply a weighted sum on D to make the $k-p$ number of the features distance to total similarity, where the weight is the feature importance learnable parameters, $W = \{w_0, w_1, \cdots, w_{k-p}\}$.

$$TS = \sum_{f=1}^{k-p} \left| x_f^i - x_f^j \right| * w_f \tag{9}$$

From Equation (9), the total similarity between data is represented by the feature-wise distance and feature importance. Although each distance of the feature pair is equal, W parameters differently represent TS depending on the feature importance. Finally, TS is expressed as $\alpha_f = softmax(TS)$.

3.3.2. Feature Attention Weight by Combining with Self-Attention

As what we did in Time-aware Attention Section 3.2, we combine with self-attention to get attention weight and representation. $Q \in \mathbb{R}^{n \times d}, K \in \mathbb{R}^{n \times d}, V \in \mathbb{R}^{n \times d}$ are represented by each weight, $W_{q,k,v} \in \mathbb{R}^{d \times d}$, and the hidden states of BiLSTM, $h \in \mathbb{R}^{n \times d}$.

We make $Q_f = \alpha_f \cdot Q \in \mathbb{R}^{n \times d}, K_f = \alpha_f \cdot K \in \mathbb{R}^{n \times d}$, which are represented by the total similarity. The rest of the process is the same as TA, getting attention weight and making representation.

$$V_f = tanh(\frac{QK^\top}{\sqrt{d_k}} + \frac{Q_f K_f^\top}{\sqrt{d_k}})V \tag{10}$$

From Equation (10), we define the value $V_f \in \mathbb{R}^{n \times d}$. The feature similarity representation $Z_f \in \mathbb{R}^{n \times d}$ can be defined as $Z_f = dropout(layernorm(FFN(V_f))) + h$. Z_f is used as inputs of last prediction layer.

3.4. Ensemble

3.4.1. Necessity of Ensemble

In the previous sections, we proposed two different attentions: TA and FA. Each of them concentrates on the different properties of ship data. TA emphasizes data depending on time intervals, and it can be said to be time-dependent. On the other hand, the other features except for timestamp, which FA tries to focus on, tend to be time-independent. For instance, even though the time interval is large between data, their speed could be similar. This inconsistency can disturb each attention model from obtaining accurate information if we combine them at once. Thus, we adopt the ensemble model which combines different predictors and alleviates instability of the model prediction [41,42].

3.4.2. Ensemble Layer

We use the ensemble model after learning each attention model, not the end-to-end model, to avoid the inconsistency of the two attention models. In addition, we expect the improvement of the model performance by designing the ensemble model of attention-based models [43,44]. For the ensemble input, we concatenate the representations of each attention model since there are only two models to ensemble.

$$Z_{ensemble} = concat(Z_t, Z_f) \tag{11}$$

The concatenated representation, $Z_{ensemble} \in \mathbb{R}^{n \times 2d}$, gets through the ensemble model, which consists of three fully connected layers. As the ensemble model uses the representations from TA and FA, it does not require an additional process. In addition, TA and FA

handle time information and important features during the learning process without the preprocess, the overall pipeline can be simply developed.

Despite discussing the inconsistency of the two attention models in this subsection, we need to verify it. In Section 4, we address the inconsistent relationship between each model. Besides, we demonstrate the drawbacks of the end-to-end model by comparing two attention results of ensemble and end-to-end models.

4. Experiments

In this section, we introduce the details of our dataset and preprocessing. It is followed by experimental settings and results. After comparing and verifying the proposed model performance, we analyze the results with the visualization of attention maps and feature importance.

4.1. Dataset

We use 2.5 million units of ship and weather data from 19 types of containers. The data are collected by the container sensors from 2016 to 2019. We use ship spec data from Lloyd List Intelligence. The nominal *twenty-foot equivalent unit* TEU of containers is from 4000 to 13,000. A dependent variable, *fuel oil consumption* (FOC), is acquired from the main diesel engine of the containers. Before cleansing the data, we select the features that we can use. However, more than half of the features consist of many missing values. This disturbs the model learning and it is difficult to replace the missing values. To avoid these problems, we exclude unusable features. The feature draft is divided into four parts (i.e., forward, starboard, port, and aft) depending on the position. We integrate four draft features into the draft as their values are slightly different. Through the exclusion and integration feature process, 19 features were left for use. In reference to the work in [9], we also divide the features into several categories (i.e., navigational status, formulation, performance, and weather data). Table 1 presents the list of features. In the learning process, we remove the vessel code feature to generalize the model regardless of the vessel types.

Table 1. List of features selected after exclusion and integration process.

Category	Features	Unit
Performance	Fuel oil consumption	tons/hour
	Speed over ground	knot/hour
	Heading	degree
Navigational	Vessel code	string
	Timestamp	UTC
	Latitude	degree
	Longitude	degree
	Draft	meter
Formulation	Length overall	meter
	Breadth	meter
	Depth	meter
	Dead weight	tons
	Gross ton inter	tons
Weather	Wind speed	m/s
	Wind direction	degree
	Sea water temperature	Celsius degree
	Airpressure	hPA
	Current speed	m/s
	Current direction	degree

We set two criteria for removing abnormal data and imputation of missing data since there are still missing and abnormal data. First, we detect abnormal data from the criteria.

The detected data is defined as removal candidates. If they satisfy the imputation criteria, we replace them with other values. The criteria for removal of abnormal data is as follows:

(1) Illogical values. Some values that are not possible to exist are defined as removal candidates.
   (a) Out of range values in latitude, longitude, and direction features.
   (b) Negative values in FOC, *speed over groud* (SOG), and draft features.
   (c) Meaningless symbols in features.
(2) Abnormal values. Some values are possible to exist, but strongly out of the distribution. We define them as removal candidates.
   (a) Zero values in FOC, SOG, and draft.
   (b) Strongly out of the distribution in SOG and draft features.
   (c) Sequential abnormal data through moving average.

By applying the removal criteria, we can define abnormal data as removal candidates. We subsequently check the possibility of imputation on the removal candidates and missing data based on the imputation criteria. The imputation criteria are as below:

1. Sequential values. Ship data are sequential during sailings. In the sequence, if abnormal or missing data are between other normal data, we replace the data with the moving average value.
2. Static values. If the data are static like the water temperature feature, we use the hourly or daily average value for imputation.

We replace the removal candidates and missing data based on the criteria. However, if the data are dynamic like the direction feature or there is no normal data nearby in a sequence, we remove them. With the removal and imputation processes, the datasets are approximately 2 million. We could only apply imputation in some cases since the malfunction of the sensing systems tends to last for a certain period of time. We use the processed data in the experiments.

4.2. Benchmark Models

In this subsection, we introduce base models and attention modules to compare the performance with the proposed models. We selected them from the previous studies and modified the details to fit our task. First, to verify the importance of the sequential property, we compare between a simple MLP like [7] and sequence models. The previous studies in vessel domain used LSTM [12], and we add GRU [18]. We also consider a bi-directional version for sequence models. We use three layers for MLP and two layers for sequence models. Each sequence model has a prediction layer, which consists of three fully connected layers. Regarding sequence models, we apply other attention modules to compare with the proposed attention module. Each attention module is located before the prediction layer. The authors of [21] applied attention to BiLSTM, and used a dot-product between hidden states and random initial vectors. We regard this attention as *Attention-based BiLSTM* (AB) in experiments. Additionally, we adopt Transformer's self-attention in sequence models as in [23,24]. We call it *Self-Attention* (SA) in the experiments.

4.3. Implementation Details

We add our attention types based on the benchmark models. We regard our attention types as TA, FA, and ensemble (ENS). Each TA and FA is applied to the sequence models and is located before the prediction layer. For the ENS layer, we use three fully connected layers. The models are backpropagated by a loss function, *mean squared error* (MSE). We first conduct base experiments without an attention layer. We set the learning rate, batch size, and sequence length of 0.001, 200, and 50, respectively. In addition, we compare the models on the different hyperparameter combinations. Based on the results without the attention layer, we determine the hidden size and dropout for the attention experiments.

4.4. Evaluation Metrics

As our data are time-series data, we split the data in time order to avoid data leakage. The portions of each train, validation, and test set are 70%, 15%, and 15%, respectively. The validation and test sets consist of ship data in 2019 as the total portion of data is the most in that year. We use *mean average error* (MAE) and *root mean squared error* (RMSE) to evaluate the model performance on the test set.

4.5. Experimental Results

We conduct two experiments in this subsection. First, we determine the hyperparameters in the base models. Besides, we verify the importance of the sequential property by comparing MLP and sequence models. As attention modules need the backbone model, we use the sequence model with the best settings of the previous sequence models. We add attention modules on the sequence models and compare the performance based on attention types. Simultaneously, we verify the performance of attention modules by comparing them with sequence models without attention modules. In the experiments, MAE and RMSE show the error between the prediction values and real FOC.

Base Experiment. Table 2 shows the prediction performance of the base models, MLP, LSTM, GRU, and their bi-directional models. As mentioned for the sequential property of ship data, sequence models achieve better performance than MLP. Besides, BiGRU shows the best performance among the other base sequence models. Based on the performance of the models with the different hyperparameters, we select the hyperparameters to use for further experiments.

Table 2. Comparison between base models.

Model	Hidden Size	Dropout = 0.1		Dropout = 0.3	
		MAE	RMSE	MAE	RMSE
MLP	32	0.375	0.518	0.457	0.598
LSTM	4	0.350	0.466	0.417	0.528
	8	0.348	0.475	0.358	0.489
BiLSTM	4	0.348	0.467	0.366	0.470
	8	0.353	0.475	0.362	0.476
GRU	4	0.344	0.462	0.378	0.488
	8	0.333	0.447	0.376	0.488
BiGRU	4	0.325	0.434	0.393	0.498
	8	0.334	0.447	0.428	0.524

Attention Experiment. After the experiments of base models, we compared each attention type on sequence models. Our attentions are TA, FA, ENS, and others are AB and SA. Table 3 presents the prediction performance of each attention on the sequence models. Generally, it records better performance besides AB when we apply attention. For SA, we notice the improvement of performance except BiGRU. For our models, at least one of TA and FA outperforms SA and sequence models without attention modules. Besides, ENS achieves the best performance. The MAE of ENS with BiLSTM is 0.3, which indicates that the prediction error at a point in time is 0.3 tons/hour. Considering the range of the FOC is from 0 to 10, the prediction is approximate to the real FOC.

Table 3. Comparison between different attention types.

Model	Metric	Attention Type					
		-	AB	SA	TA	FA	ENS
LSTM	MAE	0.348	0.360	0.332	0.329	0.323	0.315
	RMSE	0.475	0.484	0.454	0.448	0.441	0.436
BiLSTM	MAE	0.348	0.356	0.321	0.309	0.309	0.301
	RMSE	0.467	0.468	0.435	0.426	0.425	0.417
GRU	MAE	0.326	0.333	0.319	0.319	0.326	0.315
	RMSE	0.447	0.447	0.433	0.431	0.441	0.429
BiGRU	MAE	0.325	0.337	0.323	0.336	0.320	0.314
	RMSE	0.434	0.445	0.446	0.448	0.429	0.424

As there is a significant improvement from TA, FA, and ENS in BiLSTM relative to the other methods, we select BiLSTM as the backbone model. From the above experiments, we observe the improvement of the performance by capturing the ship data properties. It is more useful to consider the sequential property than applying MLP. TA, FA, and ENS designed to capture data properties are effective to predict fuel consumption accurately.

4.6. Further Experiments

We verified the performance of the proposed methods from the previous experiments. However, ENS performance is dependent on TA and FA performance. In this subsection, we evaluate and verify the methods to improve TA and FA performance. We expect it to increase the performance of ENS. Last, we apply our attention modules on the recent backbone model, Transformer [22]. By adopting the other backbone model, we also verify the compatibility of the proposed attention modules. For further experiments, we keep the settings that we found the most proper from the previous experiments.

Sequence Length Experiment. For the previous experiments, we fixed the sequence length $n = 50$ as a default value. However, it is necessary to verify the results depending on different sequence lengths, since the information for sequence models depends on the sequence length. We set $n = \{25, 50, 75, 100\}$ and compared the performance of each attention model. As shown in Table 4, the model performance decreases as the sequence length increases. In contrast, the attention type ENS achieves the best performance regardless of sequence lengths. For our dataset and models, $n = 50$ is suitable and we keep the setting for the other experiments.

Table 4. Experiment results of different sequence lengths.

Model	Attention Type	*n*	MAE	RMSE
BiLSTM	TA	25	0.316	0.435
	FA	25	0.311	0.427
	ENS	25	0.311	0.425
BiLSTM	TA	50	0.309	0.426
	FA	50	0.309	0.425
	ENS	50	0.301	0.417
BiLSTM	TA	75	0.323	0.438
	FA	75	0.333	0.449
	ENS	75	0.312	0.427
BiLSTM	TA	100	0.336	0.454
	FA	100	0.343	0.461
	ENS	100	0.326	0.439

Time Masking Experiment. In attention modules, we can also adjust the amount of information by the masking technique. We considered the time-based masking technique [27,29] for TA. This masking technique forces attention to consider only information satisfying the masking standard. For our data, the range of time interval accumulation is from 0 to 1500. The time interval accumulation refers to the accumulating time difference between right before data in the sequence. Based on the range, we define a hyperparameter masking time, m_t. TA uses only the data for which the time interval accumulation is under m_t. m_t = None indicates that we did not adopt the masking technique and it is equal to $m_t = 1500$. As shown in the time masking of Figure 5, if $m_t = 800$, TA ignores the data over m_t and uses only the data under m_t to apply the attention. Table 5 shows a slight improvement compared to TA without m_t, when m_t is 800. TA achieves better performance when considering the sequence data within a certain period of time. In addition, there is a tendency to decrease the model performance when m_t decreases. This is because TA uses only a small portion of data in sequence as the m_t reduces.

Table 5. Experiment results of TA using time masking parameter.

Model	Metric	m_t						
		200	**400**	**600**	**800**	**1000**	**1200**	**None**
BiLSTM-TA	MAE	0.31	0.314	0.316	0.304	0.311	0.315	0.309
	RMSE	0.427	0.431	0.431	0.422	0.428	0.431	0.426

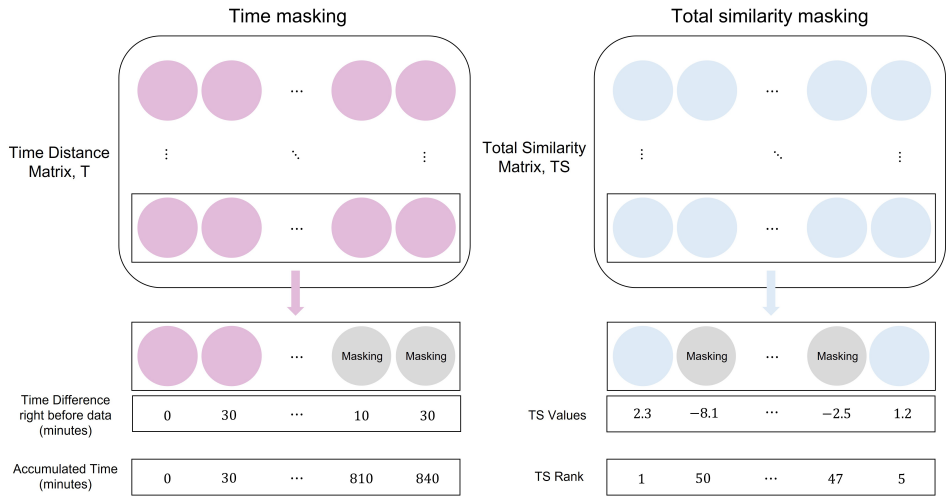

Figure 5. Masking techniques for time (**left**) and total similarity (**right**).

Total Similarity Masking Experiment. FA captures important data in sequence through the total feature similarity TS. Unlike time, TS has no specific range. Based on the idea of local attention [20], we used bottom N% masking, on the hyperparameter m_f. It means that we ignored the bottom N% data based on the TS values in the sequence when applying the attention. The TS masking of Figure 5 ignores the five numbers of data with the lowest TS values when $m_f = 10\%$. Table 6 shows that there is slight improvement when we ignore only 0% to 20% data. Similar to the results of the masking TA experiment, the performance of FA decreases as the portion of ignoring data increases.

Time Function Experiment. In Section 3.2, we handled function representation $f(x)$ used for making time importance. We used the sigmoid function with the learnable gradient and constant. For this experiment, we verified the performance of TA based on other functions, such as linear, quadratic, cubic, and exponential functions. Table 7 presents the summary of the results. Nonlinear functions are more suitable for estimating time importance.

Table 6. Experiment results of FA using N% parameter

Model	Metric	m_f (%)						
		30	**25**	**20**	**15**	**10**	**5**	**None**
BiLSTM-FA	MAE	0.319	0.318	0.309	0.311	0.308	0.309	0.309
	RMSE	0.437	0.435	0.422	0.424	0.421	0.423	0.425

Table 7. Experiment results of TA depending on functions.

Model	Metric	Function				
		Linear	**Quadratic**	**Cubic**	**Exponential**	**Sigmoid**
BiLSTM-TA	MAE	0.314	0.316	0.312	0.307	0.309
	RMSE	0.431	0.432	0.428	0.424	0.426

Ensemble Combination Experiment. From the previous result Tables 5–7, we observed the performance depending on the masking and function types. We designed combinations of TA and FA to determine the optimal ENS in the searching space. We selected the top of the two models from each previous result. We also included naive TA and FA, which used the sigmoid function for TA and did not apply any masking. As shown in Figure 6, we found the effect of ENS depending on different functions and masking. In $f(x) = sigmoid$ or $m_t = 800$, it generally proves better performance than others. On the other hand, the performance decreases as m_f increases. Finally, it achieves the best performance when we combined TA with $m_t = 800$, $f(x) = sigmoid$, and FA with $m_f = 10\%$. It is also the same compared to all other experimental results.

Module Compatibility Experiment. The previous experiments show the results when the backbones are RNNs. In this experiment, we use Transformer [22] as the backbone model to verify the compatibility of the proposed attention modules and the effect of the backbone. As the number of features is limited, we adopt the 2 and 4 layers of Transformer with small hidden sizes. In addition, we apply the proposed attention modules instead of the Transformer's attention module.

Table 8 shows the results of Transformer depending on different hyperparameters. The single Transformer with four layers shows the best performance when the hidden size is 32. When we compare the results with the previous experiment, Table 3, Transformer outperforms some of RNNs. However, the single Transformer does not get to the performance of the RNNs with the proposed attention modules. As the data are numeric, the embedding for representation in Transformer is not effective. In addition, the limited number of features is the other reason for the low performance of Transformer. Even in this situation, we notice the improvement of the performance, when we adopt the proposed attention modules, especially for the ENS. When we apply ENS, Transformer-4 with 64 hidden sizes shows the best performance including the results of the previous experiments. From this experiment, we verified the compatibility of the proposed attention modules. In the vessel domain, TA, FA, and ENS can replace the previous attention modules and exhibit better performance by considering ship data properties.

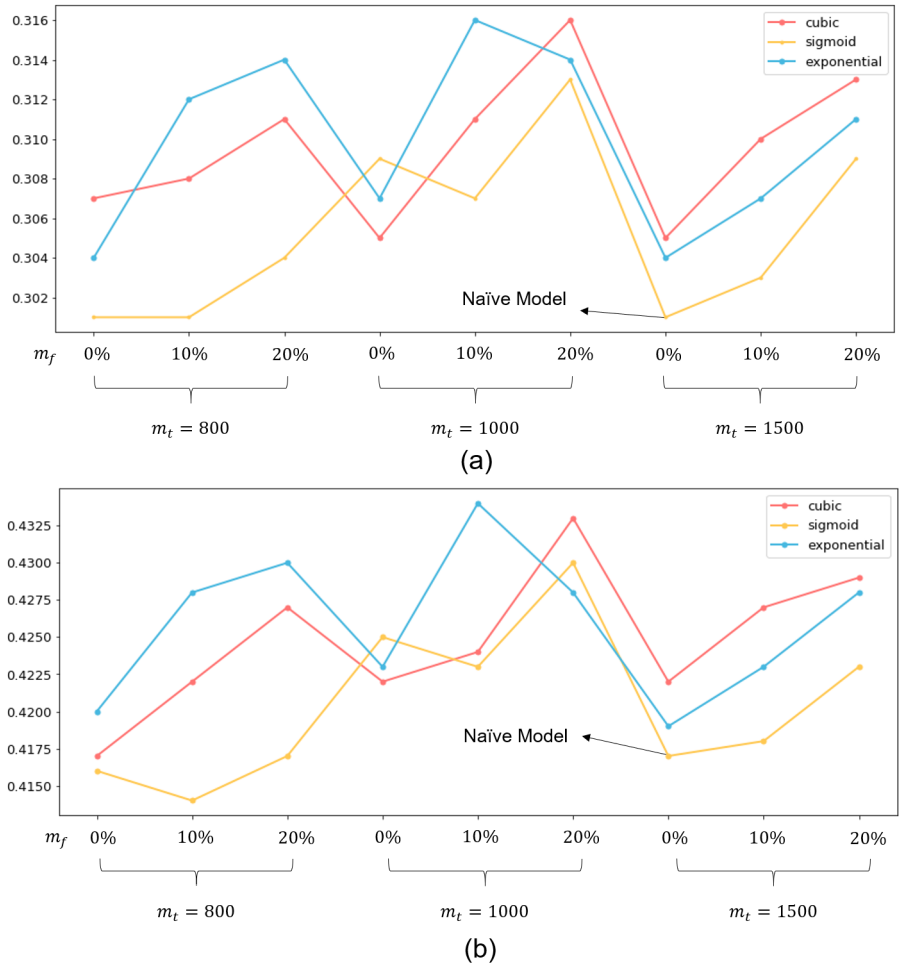

Figure 6. Experiment results of ensemble combinations. (**a**) MAE and (**b**) RMSE.

Table 8. Attention experiment on the different backbone model. The two backbones of the table refer to Transformer models with two and four layers.

Backbone	Hidden Size	Attention Type	MAE	RMSE
Transformer-2	32	-	0.357	0.467
		TA	0.364	0.461
		FA	0.352	0.463
		ENS	0.333	0.452
	64	-	0.355	0.473
		TA	0.364	0.480
		FA	0.363	0.480
		ENS	0.325	0.447
Transformer-4	32	-	0.338	0.451
		TA	0.345	0.452
		FA	0.348	0.456
		ENS	0.326	0.437
	64	-	0.361	0.471
		TA	0.357	0.474
		FA	0.349	0.467
		ENS	0.303	0.415

4.7. Visualization and Results Analysis

In this subsection, we interpret the attention models and data properties through the visualization of feature importance and attention maps. By visualizing the attention maps, we observe the characteristics of TA, FA, and how it is different from SA. In addition, we verify the necessity of ENS by showing the drawbacks of the end-to-end model.

4.7.1. Feature Importance Analysis

We addressed the feature importance parameter W in Section 3.3. W adjusted the total similarity between the data by multiplication with the feature-wise distance. Figure 7 shows the estimated feature importance from FA. We observed that especially draft and SOG have the largest absolute values. This indicates that draft and SOG considerably affect total similarity compared to other features. As their values are negative, the total similarity would decrease as the difference of draft and SOG in the sequence increases. This means that, if the draft and SOG are different from other data, that data would be less considered for FA. As we find the importance of draft and SOG from FA, the importance can also be interpreted generally. SOG is significant as the speed directly affects FOC. We can interpret the importance of draft into two parts. First, draft shows the relative weight of cargo, which affects FOC. In addition, sailing stability or hidden weather information, such as wind wave, can be demonstrated through draft [45]. Draft is more useful because our dataset does not include the wind wave feature. This analysis shows that FA is effective not only for prediction but also for determining important features.

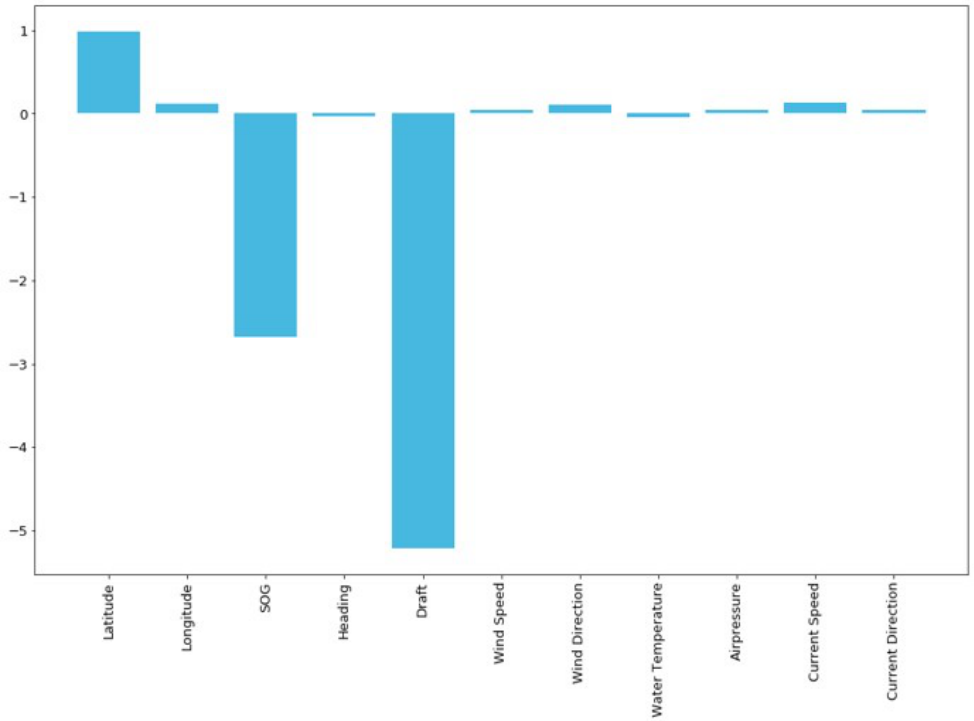

Figure 7. Estimated feature importance from Feature-similarity Attention.

4.7.2. Time-Aware Attention Map Analysis

TA Scenario 1. The first case is when the time interval increases rapidly. As shown in Figure 8, they are attention maps of each SA and TA, when sequence length is 50. Time plots show the time difference right before the data and accumulated time in the sequence. In this case, the time interval increases above 300 min compared to right before, at the 8th data of the sequence. TA specifically ignores these data. However, SA emphasizes the around area together, which means that SA cannot represent the exact time information. In addition, we can interpret the TA attention map in two parts, (i.e., row-wise and column-wise). Row-wise indicates the influence of data on other data when estimating attention

weight. In this case, 8th and 9th rarely affect the other data. Column-wise is the influence of other data on the data. We observe that both 8th and 9th are affected by themselves than others.

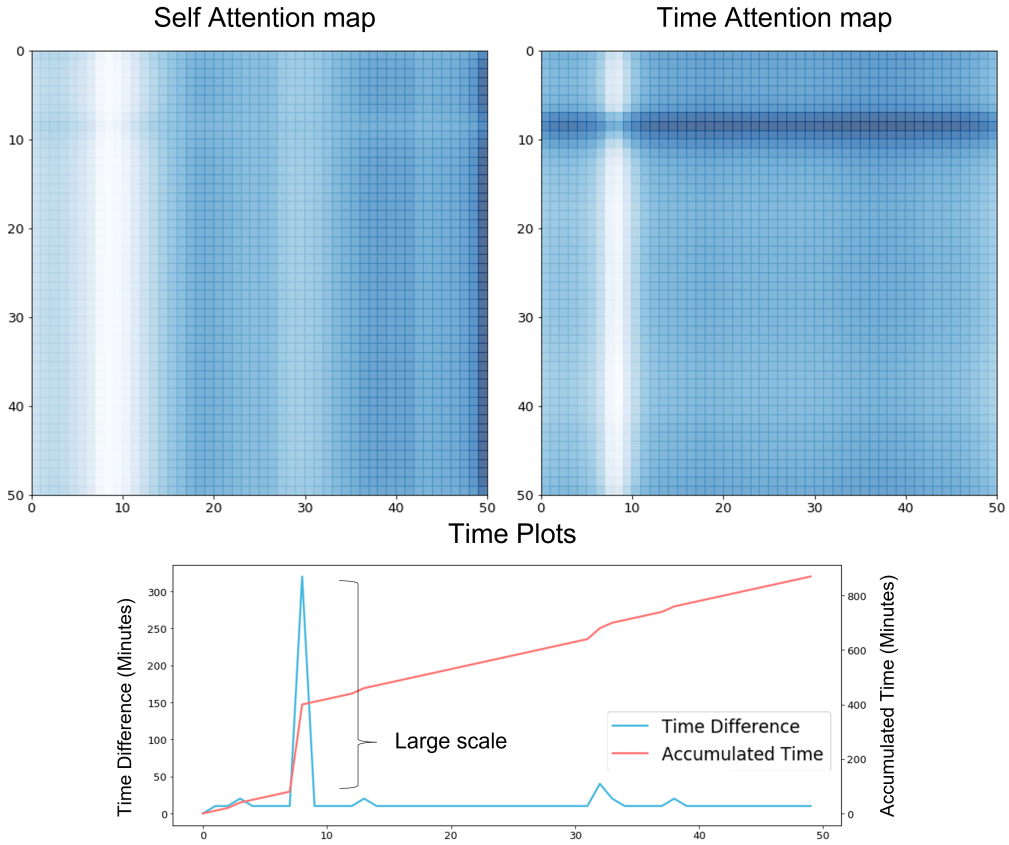

Figure 8. TA Scenario 1. The case when the time difference increases rapidly.

TA Scenario 2. The other case of TA is when the time difference is approximately 10 min and steady. As shown in time plots of Figure 9 (TA scenario 2.1.), the accumulation of time interval increases gradually. Compared to SA, TA equally emphasizes the area where the time difference is steady. For example, TA emphasizes data similarly in a boxed area. In Figure 9 (TA scenario 2.2.), although the time difference is regular, its scale is small and approximately 1 min. In this case, TA emphasizes the overall data similarly. TA also divides the areas where the time difference is steady. However, the areas are not distinct compared to TA scenario 2.1. as the scale of the time difference is small.

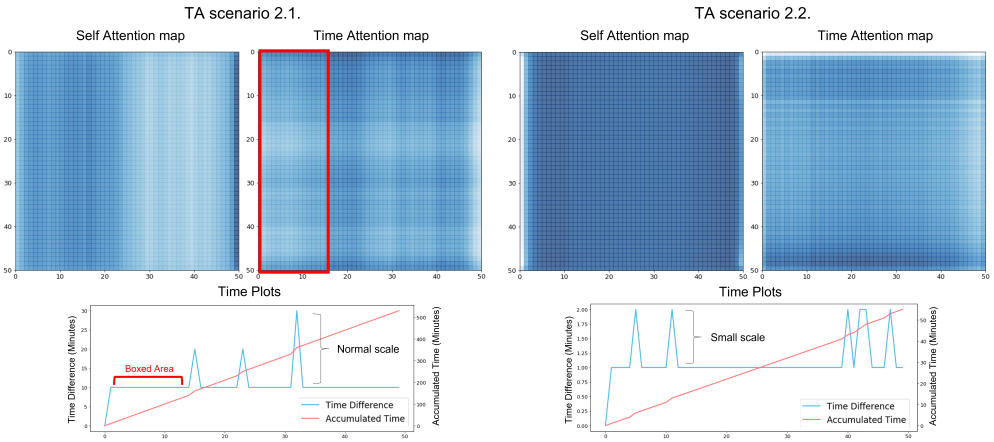

Figure 9. TA Scenario 2. The case when the time difference is steady (TA scenario 2.1.). The case when the time difference is small and steady (TA scenario 2.2.).

4.7.3. Feature-Similarity Attention Map Analysis

FA Scenario 1. Now, we address the attention map of FA in Figure 10 (FA scenario 1.). As FA does not reflect the time, it emphasizes data regardless of time. By comparing attention maps and SOG, we observe that both SA and FA divide areas based on SOG. In other words, both models equally emphasize data in which SOG is similar. In this case, the main difference between SA and FA is that FA divides more detailed symmetric areas. FA keeps an equal relationship between affecting (row-wise) and being affected (column-wise). However, SA is only concerned with affecting (row-wise). It means that the data differently affect other data based on SOG, but the data are just affected by others equally every time. Unlike FA, this unequal relation results in the limited representation of SA.

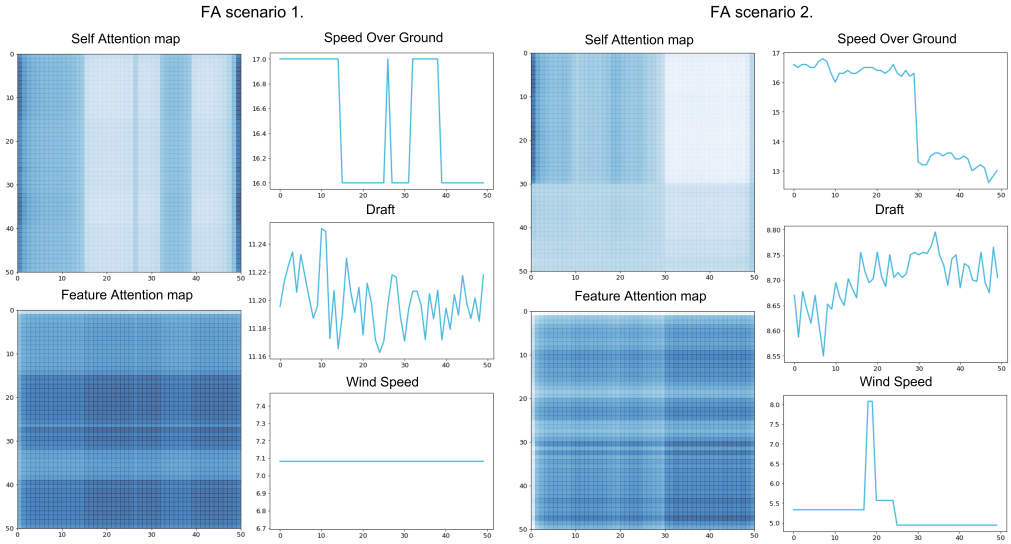

Figure 10. FA Scenario. The case when the features are stable (FA scenario 1.). The case when the features are unstable (FA scenario 2.).

FA Scenario 2. Figure 10 (FA scenario 2.) shows the other case of FA. Unlike FA scenario 1., features are unstable in the sequence. In addition, SOG suddenly changes at the index 30 in the SOG plot. Although the features are changing, SA splits the areas discretely based on only SOG and equally emphasizes data in the same areas. However, FA divides specific areas and emphasizes them differently depending on the feature values, even in the same areas. Thus, FA considers the change of other feature values, whereas SA cannot reflect the change of SOG or other features.

From the above comparison between the proposed and the previous attention modules, we observe that TA and FA reflect ship data properties. However, in the perspective of time and feature, SA shows a limited representation. The above attention maps and the lower prediction performance reveal this limited representation. In addition, compared to TA and FA, SA does not give an accurate interpretation perspective.

4.7.4. Comparison between Ensemble and End-to-End Model

Last, we compare ENS and *End-to-End* (E2E) results and verify the necessity of ENS. Figure 11 shows each attention map of TA and FA from ENS and E2E. Here, there is an inconsistency between time intervals and features. In the time plot, from the 30 to 35 indices, the time interval increases. However, SOG keeps similar values at the point where the time interval increases rapidly. In this inconsistency, TA and FA focus on their properties when we use ENS. TA ignores the point where the time interval increases, whereas FA emphasizes the same point because the feature values are similar. However, TA and FA of E2E do not fully consider the ship data properties. As shown in the TA attention map of E2E, TA of E2E does not ignore the data where the time interval increases. This indicates that when using E2E, TA and FA affect each other during the learning process. This interaction

causes TA and FA not to capture the different properties. It also results in lower prediction performance of E2E. The RMSE of ENS is 0.417, whereas that of E2E is 0.445. The lower performance also occurs when we replace the backbone model as Transformer. Thus, ENS performs better than E2E in our case since each TA and FA can focus on capturing different properties without interactions.

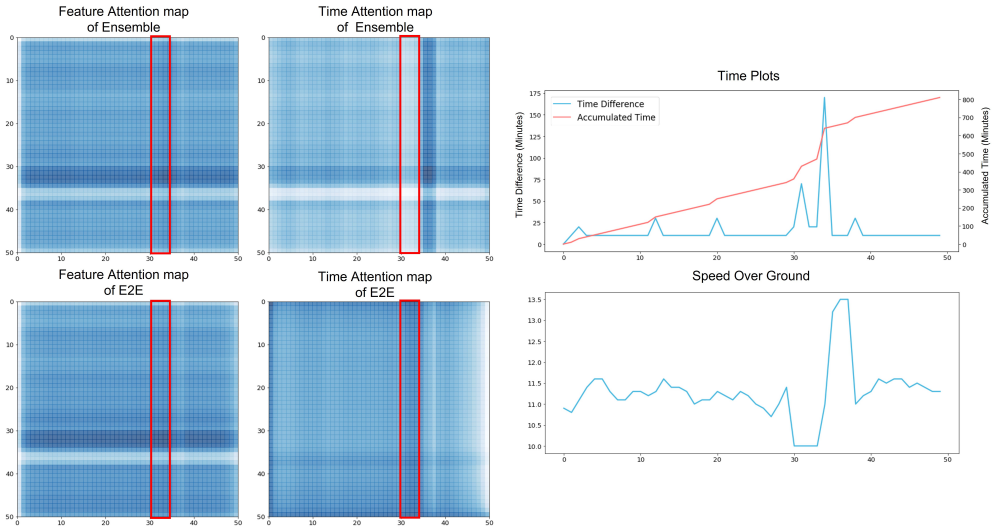

Figure 11. Comparison between Ensemble and E2E.

5. Discussion

In this section, we discuss how the proposed models can contribute to efficient sailings in the shipping industry. We suggest that the proposed models can support sea route planning by predicting the FOC of the voyage. Among the data collected by the container sensors, we bring three voyage data, which head to the same destination from the same departure but take different sea routes. Then, we compare the predicted FOC (tons) of each route and verify the model would decide a more efficient route.

As shown in Figure 12, each ship heads to Brisbane from Singapore. The FOC comparison result of Figure 12 shows the real FOC and the predicted FOC. Based on the model prediction result, we can infer that routes A and B are more efficient. Even considering the prediction error, the model suggested the efficient route as the real FOC of route A and B were lower than route C. It indicates that the proposed models can support deciding the efficient routes. Including the decision-making for the efficient route, the proposed models can contribute to constructing a sea route network of even unseen ships. For example, if there are new ship data with other specs, it is challenging to estimate the new ship's FOC of the specific route. In this case, the proposed models can predict FOC by substituting the new ship's spec into the existing route. Those applications can be more useful with Automatic identification system (AIS) data which is easy to access. AIS data have the same features as ship data but do not contain FOC. Thus, with AIS data, the proposed models can construct a sea route network for deciding an efficient route and even covering unseen ships.

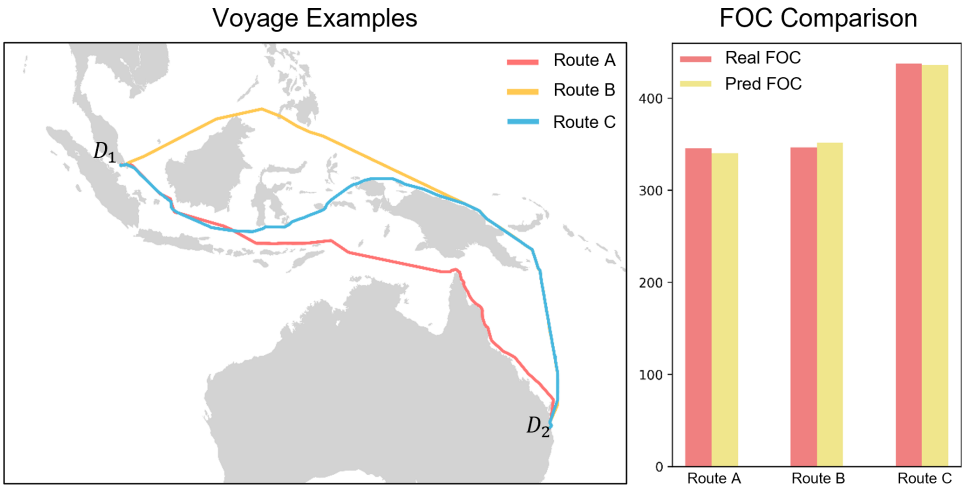

Figure 12. Voyage examples and FOC comparison of the voyages. In voyage examples, D_1 and D_2 indicate departure and destination, respectively. The predicted FOC of each route is estimated by the ensemble model of TA and FA-based BiLSTM.

6. Conclusions

In this paper, we described three main properties of ship data (i.e., sequential, irregular time interval, and feature importance) for prediction. We used BiLSTM as a backbone model to capture sequential property. For the irregular time interval and feature importance property, we proposed TA, FA, and their ensemble model. The experimental results showed that the model performance is improved by considering the ship data properties. Each attention also provided information to interpret the model and data. TA was useful for understanding the time relationship in the sequence. It captured the point where the time interval increases rapidly. FA emphasized data in the sequence by considering the feature-based similarities. It provided the estimated feature importance that can be useful to understand data.

As we designed the model to reflect the ship properties, it can be more appropriate when the ship data properties are apparent. For example, preprocessing ship data can result in more aggravated time irregularity in the sequence. Even in this situation, the proposed model can operate well by considering the time information. Besides, by estimating the feature importance, the model can determine and reflect important features without a feature selection process. These advantages of the model lead to predicting fuel consumption accurately and flexibly in different data situations. Finally, we expect that an accurate prediction of the model can improve the efficiency of sailing using the methods, such as constructing the sea route network. In addition, the model can provide attention maps and feature importance information. Attention maps identify the effect of features on the FOC at the specific route during sailing. This analysis can be done in detail by considering the important features from the estimated feature importance. The sea route network constructed by the model contains not only the accurate FOC but also information to understand the route situation. Thus, by providing information to understand the networks in diverse, the sea route network can support decision-making for navigational strategies and routing planning.

Author Contributions: Conceptualization, H.J.P.; methodology, H.J.P.; software, H.J.P.; formal analysis, H.J.P.; writing—original draft preparation, H.J.P.; visualization, H.J.P.; validation, M.S.L.; data curation, D.I.P.; writing—review and editing, M.S.L. and S.W.H.; supervision, M.S.L. and S.W.H.; project administration, S.W.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Brain Korea 21 FOUR. This research was also supported by Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government(MOTIE) (P0008691, The Competency Development Program for Industry Specialist).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Trancossi, M. What price of speed? A critical revision through constructal optimization of transport modes. *Int. J. Energy Environ. Eng.* **2016**, *7*, 425–448. [CrossRef]
2. Pallotta, G.; Vespe, M.; Bryan, K. Vessel pattern knowledge discovery from AIS data: A framework for anomaly detection and route prediction. *Entropy* **2013**, *15*, 2218–2245. [CrossRef]
3. Beşikçi, E.B.; Arslan, O.; Turan, O.; Ölçer, A.I. An artificial neural network based decision support system for energy efficient ship operations. *Comput. Oper. Res.* **2016**, *66*, 393–401. [CrossRef]
4. Jeon, M.; Noh, Y.; Shin, Y.; Lim, O.K.; Lee, I.; Cho, D. Prediction of ship fuel consumption by using an artificial neural network. *J. Mech. Sci. Technol.* **2018**, *32*, 5785–5796. [CrossRef]
5. Hu, Z.; Jin, Y.; Hu, Q.; Sen, S.; Zhou, T.; Osman, M.T. Prediction of fuel consumption for enroute ship based on machine learning. *IEEE Access* **2019**, *7*, 119497–119505. [CrossRef]
6. Le, L.T.; Lee, G.; Park, K.S.; Kim, H. Neural network-based fuel consumption estimation for container ships in Korea. *Marit. Policy Manag.* **2020**, *47*, 615–632. [CrossRef]
7. Liang, Q.; Tvete, H.A.; Brinks, H.W. Prediction of vessel propulsion power using machine learning on AIS data, ship performance measurements and weather data. *J. Phys.* **2019**, *1357*, 012038. [CrossRef]
8. Uyanik, T.; Arslanoglu, Y.; Kalenderli, O. Ship Fuel Consumption Prediction with Machine Learning. In Proceedings of the 4th International Mediterranean Science and Engineering Congress, Antalya, Turkey, 25–27 April 2019.
9. Yuan, Z.; Liu, J.; Liu, Y.; Zhang, Q.; Liu, R.W. A multi-task analysis and modelling paradigm using LSTM for multi-source monitoring data of inland vessels. *Ocean Eng.* **2020**, *213*, 107604. [CrossRef]
10. Panapakidis, I.; Sourtzi, V.M.; Dagoumas, A. Forecasting the Fuel Consumption of Passenger Ships with a Combination of Shallow and Deep Learning. *Electronics* **2020**, *9*, 776. [CrossRef]
11. Liu, Y.; Duan, W.; Huang, L.; Duan, S.; Ma, X. The input vector space optimization for LSTM deep learning model in real-time prediction of ship motions. *Ocean Eng.* **2020**, *213*, 107681. [CrossRef]
12. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
13. Borkowski, T.; Kasyk, L.; Kowalak, P. Assessment of ship's engine effective power, fuel consumption and emission using the vessel speed. *J. KONES* **2011**, *18*, 31–39.
14. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
15. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
16. Werbos, P.J. Backpropagation through time: What it does and how to do it. *Proc. IEEE* **1990**, *78*, 1550–1560. [CrossRef]
17. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks* **1994**, *5*, 157–166. [CrossRef]
18. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
19. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
20. Luong, M.T.; Pham, H.; Manning, C.D. Effective approaches to attention-based neural machine translation. *arXiv* **2015**, arXiv:1508.04025.
21. Yao, X. Attention-based BiLSTM neural networks for sentiment classification of short texts. *Proc. Int. Conf. Inf. Sci. Cloud Comput.* **2017**, 110–117.
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
23. Merity, S. Single headed attention rnn: Stop thinking with your head. *arXiv* **2019**, arXiv:1911.11423.
24. Gao, Y.; Fang, C.; Ruan, Y. A novel model for the prediction of long-term building energy demand: LSTM with Attention layer. In Proceedings of the IOP Conference Series: Earth and Environmental Science, Tokyo, Japan, 6–7 August 2019; Volume 294, p. 012033.
25. Song, H.; Rajan, D.; Thiagarajan, J.J.; Spanias, A. Attend and diagnose: Clinical time series analysis using attention models. *arXiv* **2017**, arXiv:1711.03905.
26. Chaudhari, S.; Polatkan, G.; Ramanath, R.; Mithal, V. An attentive survey of attention models. *Acm Trans. Intell. Syst. Technol.* **2019**, *12*, 1–32. [CrossRef]
27. Wu, N.; Green, B.; Ben, X.; O'Banion, S. Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case. *arXiv* **2020**, arXiv:2001.08317.

28. Darabi, S.; Kachuee, M.; Fazeli, S.; Sarrafzadeh, M. TAPER: Time-Aware Patient EHR Representation. *IEEE J. Biomed. Health Informatics* **2020**, *24*, 3268–3275. [CrossRef] [PubMed]
29. Li, J.; Wang, Y.; McAuley, J. Time Interval Aware Self-Attention for Sequential Recommendation. In Proceedings of the 13th International Conference on Web Search and Data Mining, Houston, TX, USA, 3–7 February 2020; pp. 322–330.
30. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
31. Chandrashekar, G.; Sahin, F. A survey on feature selection methods. *Comput. Electr. Eng.* **2014**, *40*, 16–28. [CrossRef]
32. Lai, S.; Liu, K.; He, S.; Zhao, J. How to generate a good word embedding. *IEEE Intell. Syst.* **2016**, *31*, 5–14. [CrossRef]
33. Iqbal, R.A. Using feature weights to improve performance of neural networks. *arXiv* **2011**, arXiv:1101.4918.
34. Jiang, L.; Li, C.; Wang, S.; Zhang, L. Deep feature weighting for naive Bayes and its application to text classification. *Eng. Appl. Artif. Intell.* **2016**, *52*, 26–39. [CrossRef]
35. Zheng, J.; Wang, Y.; Xu, W.; Gan, Z.; Li, P.; Lv, J. GSSA: Pay attention to graph feature importance for GCN via statistical self-attention. *Neurocomputing* **2020**, *417*, 458–470. [CrossRef]
36. Škrlj, B.; Džeroski, S.; Lavrač, N.; Petkovič, M. Feature importance estimation with self-attention networks. *arXiv* **2020**, arXiv:2002.04464.
37. Gui, N.; Ge, D.; Hu, Z. AFS: An attention-based mechanism for supervised feature selection. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 3705–3713.
38. Lee, K.H.; Park, C.; Oh, J.; Kwak, N. LFI-CAM: Learning Feature Importance for Better Visual Explanation. *arXiv* **2021**, arXiv:2105.00937.
39. Xiao, J.; Ye, H.; He, X.; Zhang, H.; Wu, F.; Chua, T.S. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv* **2017**, arXiv:1708.04617.
40. Baytas, I.M.; Xiao, C.; Zhang, X.; Wang, F.; Jain, A.K.; Zhou, J. Patient subtyping via time-aware lstm networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 65–74.
41. Dietterich, T.G. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 1–15.
42. Qiu, X.; Zhang, L.; Ren, Y.; Suganthan, P.N.; Amaratunga, G. Ensemble deep learning for regression and time series forecasting. In Proceedings of the 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL), Orlando, FL, USA, 9–12 December 2014; pp. 1–6.
43. Tan, Z.; Wang, M.; Xie, J.; Chen, Y.; Shi, X. Deep semantic role labeling with self-attention. *arXiv* **2017**, arXiv:1712.01586.
44. Kim, W.; Goyal, B.; Chawla, K.; Lee, J.; Kwon, K. Attention-based ensemble for deep metric learning. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 736–751.
45. Wang, W.; Wu, T.; Zhao, D.; Guo, C.; Luo, W.; Pang, Y. Experimental–numerical analysis of added resistance to container ships under presence of wind–wave loads. *PLoS ONE* **2019**, *14*, e0221453. [CrossRef]