

Prediction of Vessel Arrival Time to Pilotage Area Using Multi-Data Fusion and Deep Learning

Xiaocai Zhang¹, Xiuju Fu¹, Zhe Xiao^{1*}, Haiyan Xu¹, Xiaoyang Wei¹, Jimmy Koh², Daichi Ogawa³ and Zheng Qin¹

Abstract—This paper investigates the prediction of vessels' arrival time to the pilotage area using multi-data fusion and deep learning approaches. Firstly, the vessel arrival contour is extracted based on Multivariate Kernel Density Estimation (MKDE) and clustering. Secondly, multiple data sources, including Automatic Identification System (AIS), pilotage booking information, and meteorological data, are fused before latent feature extraction. Thirdly, a Temporal Convolutional Network (TCN) framework that incorporates a residual mechanism is constructed to learn the hidden arrival patterns of the vessels. Extensive tests on two real-world data sets from Singapore have been conducted and the following promising results have been obtained: 1) fusion of pilotage booking information and meteorological data improves the prediction accuracy, with pilotage booking information having a more significant impact; 2) using discrete embedding for the meteorological data performs better than using continuous embedding; 3) the TCN outperforms the state-of-the-art baseline methods in regression tasks, exhibiting Mean Absolute Error (MAE) ranging from 4.58 min to 4.86 min; and 4) approximately 89.41% to 90.61% of the absolute prediction residuals fall within a time frame of 10 min.

I. INTRODUCTION

Pilotage is a mandatory requirement for many vessels entering and exiting ports worldwide and is an essential component of safe and efficient navigation [1]. Pilots are highly trained professionals aimed to guide vessel captains, ensuring vessels' safe passage in and out of a port. The pilotage area is a fixed water zone the pilot embarks on the vessel to commence pilotage service. The efficient operation of ports relies on a delicate balance of timing and coordination, and the Just-In-Time (JIT) arrival of vessels to pilot boarding areas is a critical component of this process. In order to facilitate JIT operations between pilotage service and arriving vessels, it is essential for the pilot operation center to accurately monitor vessel arrival times and adjust schedules accordingly. This not only has the potential to improve the overall efficiency of operations, but it also mitigates the risks associated with vessels waiting at pilot boarding areas

as a result of arriving too early. Given the diverse arrival behaviors that vessels exhibit [2], accurately predicting their arrival patterns becomes challenging due to the increased uncertainty. This uncertainty, such as movement behaviors and pilotage operation behaviors, makes it difficult to obtain robust predictions that can withstand potential disruptions or variations in the vessels' schedules.

In this study, we present an approach for predicting vessel arrival time to pilotage areas using multi-data fusion and deep learning. This method has the potential to enhance the efficiency of port operations by enabling Just-in-Time (JIT) operations at pilot boarding areas. The following are the key contributions to highlight:

- To the best of our knowledge, this is the first study to explore the forecasting of the arrival time of vessels to pilotage areas;
- We propose a hybrid approach based on statistical learning and machine learning to identify vessel arrival patterns by extracting vessel arrival contour using historical trajectory data;
- We first develop a Temporal Convolutional Network (TCN) framework that integrates residual blocks to model the problem of vessel arrival time prediction;
- We evaluate the performance of our proposed method on data sets obtained from Singapore and compare it with other advanced deep learning algorithms to validate its effectiveness and superiority.

The subsequent parts of this article are organized as follows. Section II presents a summary of previous studies on predicting ship arrival time. In Section III, we explain the proposed method in detail, including vessel arrival contour extraction and TCN-based model using multi-data fusion. Section IV details the results of the experiments and analyses. Finally, in Section V, we conclude this study.

II. RELATED WORK

The existing approaches for modelling vessel arrival time prediction can be categorized into two types: statistical models and machine learning models. The authors in [3] formulated a data-driven path-finding algorithm for vessel arrival time prediction. The algorithm involves a set of parameters conveniently optimized for the area under investigation. The authors in [4] introduced a path-finding algorithm to find possible vessel trajectories using Automatic Identification System (AIS) data. Then the Markov Chain property and Bayesian Sampling were introduced to estimate the Speed Over Ground (SOG) of a vessel. Finally, the Estimated Time

*Corresponding author: Zhe Xiao.

¹Xiaocai Zhang, Xiuju Fu, Zhe Xiao, Haiyan Xu, Xiaoyang Wei and Zheng Qin are with the Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), 1 Fusionopolis Way, #16-16 Connexis, Singapore 138632, Republic of Singapore zhang_xiaocai@ihpc.a-star.edu.sg; xiaoz@ihpc.a-star.edu.sg

²Jimmy Koh is with the Pilotage and Digital Transformation Department, PSA Marine (Pte) Ltd, 70 West Coast Ferry Road, Singapore 126800, Republic of Singapore

³Daichi Ogawa is with the MTI Co., Ltd. Singapore Branch, 1 Harbourfront Place #14-01, Harbourfront Tower One, Singapore 098633, Republic of Singapore

of Arrival (ETA) was estimated from the derived SOG values for the predicted trajectory. Bodunov *et al.* [5] constructed a feed-forward neural network to forecast the arrival time of vessels. Ogura *et al.* [6] extracted the route and used Bayesian learning to calculate the speed along the voyage by considering the impact of weather. In the publication referenced as [7], the authors employed a Fuzzy Rule-Based Bayesian Network (FRBBN) to evaluate and forecast the on-time arrival of a liner container vessel at ports of call, even when faced with uncertain circumstances. In [8], the authors concluded that Gated Recurrent Unit (GRU) outperforms Gradient Boosting Decision Trees (GBDT) and Multi-Layer Perceptron (MLP) based on AIS data. Yu *et al.* [9] addressed the problem of ship arrival prediction (delay or advance arrival prediction) by using three methods, i.e., BPNN, Classification And Regression Tree (CART), and Random Forest (RF). The prediction results show that the RF model outperforms BP and CART models. In [10], the authors provided a supervised learning approach for ETA prediction for intermodal freight transport networks involving multiple modes of transportation. Mekkaoui *et al.* [11] showed a Long Short-Term Memory (LSTM)-based model for vessel arrival time prediction using different data sources, including vessel characteristics, AIS data, and weather data. Valero *et al.* [12] conducted a practical analysis of prior studies on predicting ETA for container vessels and determined that RF achieved the best performance. Meanwhile, Cammin *et al.* [13] utilized the name and shipping line type of vessels as features to forecast arrival delays using the Support Vector Regression.

III. METHOD

The vessel's arrival time is determined as the moment when the vessel reaches the designated pilot boarding areas of the port. Fig. 1 illustrates the framework for predicting the arrival time of vessels to the pilotage area. This framework consists of five modules: 1) AIS data, 2) arrival contour, 3) multi-data fusion and feature, 4) modelling, and 5) evaluation. The subsequent sections of this document delve into the specifics of each module. The first two modules, i.e., AIS data and arrival contour, are further explained in Section III-A, while the subsequent two modules, i.e., multi-data fusion and feature, and modelling, are elaborated in Section III-B.

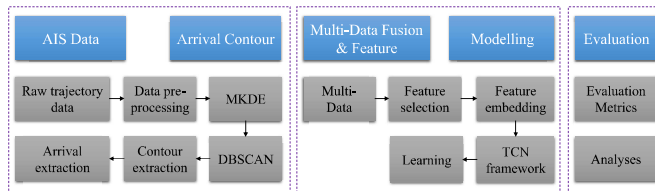


Fig. 1. The framework of prediction of the vessel arrival time to pilotage area.

A. Vessel Arrival Contour Extraction

Assume that there are a total of n historical trajectories that lead to a Pilot Boarding Ground (PBG), for each AIS

trajectory i , we can determine its nearest point to the PBG and denote it as X_i , as shown by

$$X = (X_1, X_2, \dots, X_n), \quad (1)$$

and

$$X_i = (x_{i1}, x_{i2})^T \in \mathbb{R}^2, \quad (2)$$

where x_{i1} and x_{i2} are the longitude and latitude, respectively. The following objective is to estimate the probability density function, represented by $f(x_1, x_2)$, of these points X . We propose to use the Multivariate Kernel Density Estimation (MKDE) method to derive an estimate of $f(x_1, x_2)$. The formulation of MKDE is expressed as

$$\hat{f}_h(x_1, \dots, x_d) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h_1 \dots h_d} K\left(\frac{x_1 - x_{i1}}{h_1}, \dots, \frac{x_d - x_{id}}{h_d}\right), \quad (3)$$

where d is the dimension of the variables, K is the kernel function, and h is the bandwidth vector. In this study, $d = 2$, and the Gaussian kernel is chosen as the kernel.

Following that, those points with high probability density values, e.g., greater than a threshold, are selected for the follow-up step to extract arrival contour and to define the arrival of vessels to the pilot boarding ground, as shown by

$$\hat{X} = \{X_i, \text{ if } \hat{f}_h(x_1, \dots, x_n) > k\text{th percentile}\}. \quad (4)$$

In this study, we established the threshold by considering the density values of all the points and selecting the value corresponding to the k th percentile.

Next, we utilize the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm, which is a density-based spatial clustering method, to detect the main cluster and eliminate outlier clusters. DBSCAN is able to find clusters with arbitrary shapes and clusters that contain noise, which is appropriate for processing the spatial data in this study. The bounding polygon of the main cluster is then extracted to form a contour. When a vessel enters this contour, it is recognized as arriving at the PBG.

Algorithm 1 Vessel arrival contour extraction

Input: vessels' historical trajectories to a PBG

Output: vessel arrival contour

- 1: Scan historical trajectories and get X based on Eq. (1);
 - 2: Apply MKDE to estimate the probability density function $f(x_1, x_2)$ with Eq. (3);
 - 3: Get points $\hat{X}$ with high density using Eq. (4);
 - 4: Apply DBSCAN to detect the main cluster and eliminate outlier clusters;
 - 5: Extract contour based on the main cluster.
-

B. TCN-Based Model Using Multi-Data Fusion

TCN is being used in this context due to its specific attributes. The notable components of TCN consist primarily of causal convolution, dilated convolution, and residual block [14]. Causal convolution is a specific type of convolution that is utilized in modelling temporal data to ensure that the temporal ordering is preserved. This means that when considering the output at time t , the convolution process can

only involve elements from time t or earlier in the previous layer. It cannot take into account any future time points beyond t . The purpose of using causal convolution in TCN is to prevent any leakage of information from the future to the past [15], as shown by Fig. 2.

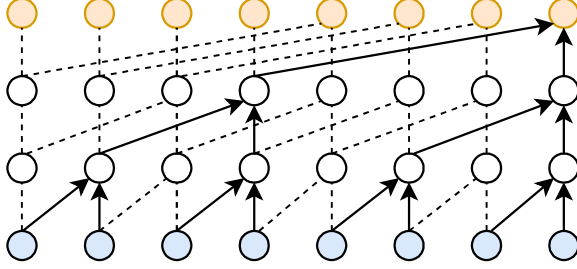


Fig. 2. Illustration of the dilated casual convolution.

Dilated convolution is a type of convolution where the kernel is expanded by creating holes between its elements. This expansion allows for a significant increase in the receptive field without sacrificing coverage [16]. The filters for dilated convolution can be represented by a function F with a size of f , while X represents the input time-series data. The convolved value at timestamp t is computed using the following equation:

$$\mathbf{v}(t) = (X *_d F)(t) = \sum_{i=0}^{f-1} F(i) \cdot X_{t-d \cdot i}, \quad (5)$$

where d is the dilation rate and f is the filter size. This equation essentially means that the value at time t is obtained by multiplying the filter elements with the input data at time points $t-d \cdot i$ for $i = 0, 1, \dots, f-1$ and summing the results.

The architecture of the residual block, which is a crucial component, is shown in Fig. 3. In a single residual block, the input is processed through two stacked dilated causal convolutional layers and two Rectified Linear Unit (ReLU) activation layers. Additionally, two layers of batch normalization are used to speed up the network convergence and prevent overfitting. The output of the combined transformations is denoted by $\tau(X, W)$, where X is the input and W represents the weights. If the dimensions of the input and output are the same, the final output (y) of the residual block is obtained by adding the output of the transformation $\tau(X, W)$ to the input X , as shown by

$$y = \tau(X, W) + X. \quad (6)$$

However, if the dimensions are different, a 1×1 convolution is first applied to the input X before the addition operation.

Assuming that we have the historical features of a vessel at timestamps $(t+1, t+2, \dots, t+m)$, the objective is to predict when the vessel will arrive at the pilotage area. The input X of the model is represented by

$$\mathbf{x}_j = ((\varphi_{j1}, \lambda_{j1}, v_{j1}, c_{j1}, t_{j1}, r_{j1}, w_{j1}^s, w_{j1}^d), \dots, (\varphi_{jm}, \lambda_{jm}, v_{jm}, c_{jm}, t_{jm}, r_{jm}, w_{jm}^s, w_{jm}^d))^T, \quad (7)$$

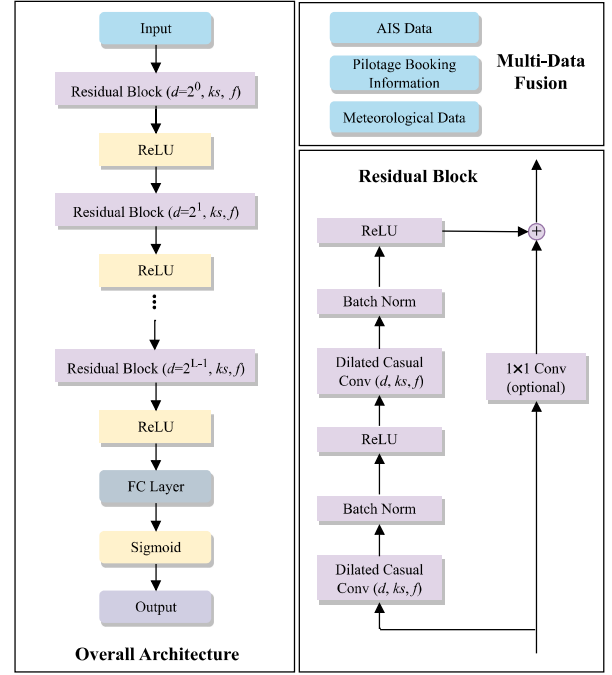


Fig. 3. The overall architecture of TCN for vessel arrival time prediction, where d represents the dilation rate, ks is the kernel size, f stands for the filter size, and L denotes the number of layers.

and

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N-1}, \mathbf{x}_N) \in \mathbb{R}^{N \times m \times 9}, \quad (8)$$

where $\varphi, \lambda, v, c, t, r, w^s$ and w^d stands for the latitude, longitude, speed, course, time left until the Confirmed Service Time (CST), rainfall, wind speed and wind direction. CST is defined as the confirmed service start time for the provision of pilotage service [17]. The sequence length is denoted by m , and the number of observations is denoted by N . The rainfall and wind speed features are transformed into discrete binary features using an embedding process, as demonstrated by

$$r_{ji} = \begin{cases} 1 & r_{ji} \geq \epsilon_r \\ 0 & r_{ji} < \epsilon_r, \end{cases} \quad (9)$$

and

$$w_{ji}^s = \begin{cases} 1 & w_{ji}^s \geq \epsilon_{w^s} \\ 0 & w_{ji}^s < \epsilon_{w^s}, \end{cases} \quad (10)$$

where ϵ_r and ϵ_{w^s} are the thresholds for rainfall and wind speed, respectively. Furthermore, the wind direction is embedded using the Canonical Encoding Rules, as formulated by

$$w_{ji}^d = \left(\frac{\sin(\theta_{ji}) + 1}{2}, \frac{\cos(\theta_{ji}) + 1}{2} \right). \quad (11)$$

The ultimate output of the entire network can be obtained using Eq. (6), and a Fully-Connected (FC) layer and a sigmoid activation layer are attached to generate the output within the range of $[0, 1]$, as indicated by

$$y = \text{FC}(y), \quad (12)$$

and

$$y = \text{sigmoid}(y). \quad (13)$$

The groundtruth arrival time is denoted by $\hat{y}_i$, and the output from TCN is denoted by y_i . Therefore, the loss can be computed as follows:

$$\mathcal{L}(W, b) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (14)$$

where N means the sample size, and W and b stand for the neural network's weights and biases, respectively.

During network training, the objective is to minimize the overall loss function by adjusting the weights and biases using the training data. This allows the network to learn and improve its predictions over time, as shown by

$$(W^*, b^*) = \arg \min_{W, b} \mathcal{L}(W, b), \quad (15)$$

where $\mathcal{L}(W, b)$ is the overall loss function of Eq. (14). To find the optimal parameters (W^*, b^*) , the widely-used back-propagation algorithm is employed in a similar way to train a basic deep neural network. An Adam optimizer is used during the back-propagation process.

IV. EXPERIMENTS AND EVALUATION

A. Data Sets

TABLE I
DATASET DESCRIPTION

PBG	Trajectory	Training Sample	Test Sample	Pred. Start Time (h)
PEBGA	3,658	171,798	42,950	1.14
PEBGC	8,901	334,716	83,680	1.30

Notes: "Pred. Start Time" refers to the average time it takes for vessels to travel from the area where the predictions begin to the PBG.

Two real-world data sets, including AIS, pilotage booking data, and local meteorological data, from the Singapore Strait are used as experimental data in this study. The data spans 1 year, from January 1, 2018 to December 31, 2018. These data sets collect the trajectories of vessels heading to two PBGs in the Port of Singapore: the Eastern Boarding Ground "A" (PEBGA) and the Eastern Boarding Ground "C" (PEBGC). More details about the PBGs can be found at the PSA Marine's website¹. The trajectories of vessels heading to these PBGs are extracted to construct the training and test data, with a preprocessing interpolation time interval of 1 minute. The earlier 80% data are used for model training, and the later 20% are used for tests. This division is made because the data follows a time-series pattern, and it is necessary to use only the preceding data for training [18]. The predictions are consistently generated at intervals of 1 minute. Consequently, a single trajectory can produce multiple samples for both training and testing purposes. It takes approximately 1.14-1.30 hours of travel time to reach the PBG from the prediction starting point. More information about the data sets can be referred to Table I.

¹<https://www.psamarine.com/wp-content/uploads/2017/08/Information-for-Merchant-Ships.pdf>

B. Experimental Settings

Table II shows the specific values used for each parameter. For the extraction of the arrival contour, the value for k (Eq. (4)) was set to 75. In the case of the TCN framework, we set the values of m (Eq. (7)), ks (Fig. 3), f (Eq. (5)) and L (Fig. 3) to be 10, 15, 5, and 6 respectively. For feature embedding, we used values of 7.6 mm and 22 knots for ϵ_r (Eq. (9)) and ϵ_{ws} (Eq. (10)) respectively, which were scaled to represent heavy rain and strong breeze. The performance of the models was assessed using the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Coefficient of Determination (R^2). Python and TensorFlow were used to develop the models in this study.

TABLE II
PARAMETER SETTING

Parameter	k	h	m	ϵ_r (mm)
Value	75	0.001	10	7.6
Parameter	ϵ_{ws} (knot)	ks	f	L
Value	22	15	5	6

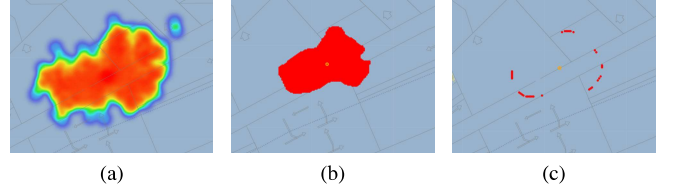


Fig. 4. Vessel arrival contour extraction for PEBGA. (a) heatmap of the closest points to PEBGA. (b) refined area after clustering and removing outlier clusters. (c) extracted contour (bounding polygon).

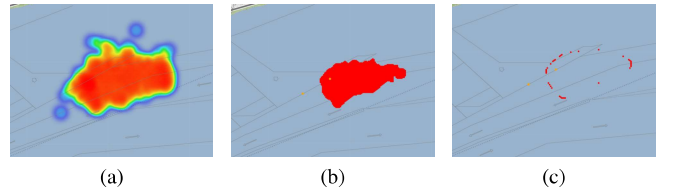


Fig. 5. Vessel arrival contour extraction for PEBGC. (a) heatmap of the closest points to PEBGC. (b) refined area after clustering and removing outlier clusters. (c) extracted contour (bounding polygon).

C. Results and Analyses

Fig. 4 and Fig. 5 depict the processes involved in obtaining the arrival contours for PEBGA and PEBGC. Fig. 4(a) and Fig. 5(a) show a heatmap that displays the closest points to the pilot boarding ground. Fig. 4(c) and Fig. 5(c) depict the refined area after applying MKDE and DBSCAN clustering. The final arrival contours for PEBGA and PEBGC are shown in Fig. 4(b) and Fig. 5(b), respectively. These contours are utilized to define the arrival of the vessel and to determine the groundtruth arrival time for each trajectory.

TABLE III
PREDICTION PERFORMANCE FOR PEBGA

Model	MAE (min)	RMSE (min)	R ²
w/o rainfall & wind	4.66	7.99	0.8790
w/o CST	6.37	9.87	0.8154
wth rainfall & wind (continuous)	4.75	7.41	0.8960
wth rainfall & wind (discrete)	4.58	6.82	0.9117

Notes: "w/o" and "wth" are short for "without" and "with", respectively.

TABLE IV
PREDICTION PERFORMANCE FOR PEBGC

Model	MAE (min)	RMSE (min)	R ²
w/o rainfall & wind	4.97	6.83	0.8813
w/o CST	6.08	8.96	0.7981
wth rainfall & wind (continuous)	5.03	6.79	0.8828
wth rainfall & wind (discrete)	4.86	6.61	0.8890

Notes: "w/o" and "wth" are short for "without" and "with", respectively.

The prediction results of our method for PEBGA and PEBGC are presented in Table III and Table IV, respectively. Incorporating meteorological data improves the R² from 0.8790 to 0.9117 for PEBGA and from 0.8813 to 0.8890 for PEBGC. However, pilotage information (i.e., CST) contributes more to the improvement of R², enhancing it by 9.63% (0.9117 vs. 0.8154) for PEBGA and by 9.09% (0.8890 vs. 0.7981) for PEBGC. Additionally, using discrete vectors to embed the wind and rainfall features improves the prediction performance for both data sets. Specifically, compared to using continuous features, it reduces the MAE from 4.75 min to 4.58 min for PEBGA and from 5.03 min to 4.86 min for PEBGC.

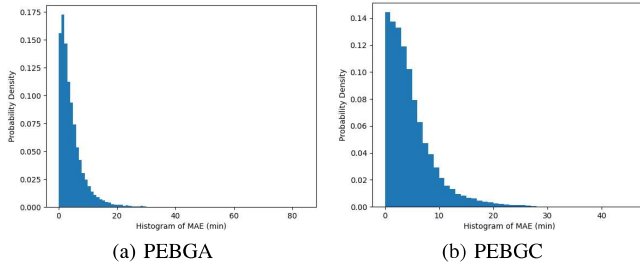


Fig. 6. Histogram of MAE on the test data.

The proposed TCN framework is benchmarked with eight commonly used baseline models for time-series regression, including LSTM [11], GRU, Bidirectional LSTM (Bi-LSTM), Bidirectional GRU (Bi-GRU), Convolutional Neural Network (CNN) [19], LSTM_BiLSTM [20], LSTM Seq2Seq [21], and Stacked Bi-GRU (SBI-GRU) [22]. The prediction results of TCN as well as the baseline models for PEBGA and PEBGC have been presented in Tables V and VI, respectively. Our TCN model shows superior accuracy, with an MAE of only 4.58 min for PEBGA and 4.86 min for PEBGC, outperforming the best baseline model by at least 0.96%

(0.9117 vs. 0.9021 for PEBGA) in terms of R². LSTM, GRU, Bi-LSTM, and Bi-GRU are popular RNN-based frameworks for sequential data modelling because of their strong capability of learning long-term interdependencies [23]. Bi-LSTM and Bi-GRU can learn the sequential input data in both forward and backward directions, enabling a higher learning capability on sequential data. Fig. 6(a) and Fig. 6(b) show the histograms of MAE for PEBGA and PEBGC, respectively. For PEBGA, 90.61% of the predictions' MAEs are within [0, 10 min], and 68.09% are within [0, 5 min]. For PEBGC, 89.41% of the predictions' MAEs are within [0, 10 min], and 63.65% are within [0, 5 min].

TABLE V
BENCHMARKING OF PREDICTION PERFORMANCE FOR PEBGA

Model	MAE (min)	RMSE (min)	R ²
LSTM	4.74	7.34	0.8978
GRU	4.76	7.37	0.8971
Bi-LSTM	4.79	7.35	0.8976
Bi-GRU	4.67	7.18	0.9021
CNN	5.16	7.70	0.8876
LSTM_BiLSTM	4.92	7.50	0.8934
LSTM seq2seq	5.60	8.21	0.8723
SBI-GRU	5.05	7.62	0.8898
TCN	4.58	6.82	0.9117

TABLE VI
BENCHMARKING OF PREDICTION PERFORMANCE FOR PEBGC

Model	MAE (min)	RMSE (min)	R ²
LSTM	5.07	7.04	0.8739
GRU	5.13	7.16	0.8698
Bi-LSTM	5.08	7.08	0.8724
Bi-GRU	5.08	6.98	0.8760
CNN	5.64	7.98	0.8383
LSTM_BiLSTM	5.09	7.11	0.8714
LSTM seq2seq	5.55	7.74	0.8478
SBI-GRU	5.13	7.14	0.8703
TCN	4.86	6.61	0.8890

V. CONCLUSIONS

This paper proposes a method to predict vessel arrival time to pilot boarding grounds based on multi-data fusion and deep learning. Initially, vessel arrival is defined using MKDE and DBSCAN clustering, which helps in extracting the vessel arrival contour. Subsequently, deep learning is applied using multi-data fusion, which incorporates AIS data, pilotage booking information, and meteorological data as inputs. A multi-layer TCN framework is utilized to recognize the vessel arrival pattern and predict the arrival time. The experiments carried out in Singapore, using two real-world data sets, demonstrate that the TCN framework for arrival time prediction is highly effective, achieving a high R² value of 91.17% and 88.90%, with about 89.41%-90.61% of the absolute prediction residuals located within 10 minutes. Additionally, the proposed method outperforms other models such as LSTM, GRU, Bi-LSTM, Bi-GRU, CNN, LSTM_BiLSTM, LSTM seq2seq, and SBI-GRU. The inclusion of pilotage booking information and meteorological data

has been found to enhance prediction accuracy, with pilotage booking information being the more significant contributor. The use of discrete embedding for meteorological data is more effective than using continuous embedding.

VI. ACKNOWLEDGEMENT

This study is partially supported by the Institute of High Performance Computing (IHPC)'s research project titled "Digital Intelligence Research Based on Big Data for Improving Efficient Navigation of Vessels to Singapore Port Waters" (Grant number SMI-2021-MTP-01 funded by Singapore Maritime Institute) and is also partially supported under Maritime AI Research Programme (Grant number SMI-2022-MTP-06 funded by Singapore Maritime Institute).

REFERENCES

- [1] T. Hutabarat, S. Ma'arif, N. Zulfainarni, and P. Yusgiantoro, "The use of integrated piloting strategy to achieve a balance of economic efficiency and security effectiveness in the strait of malacca," *International Journal of Management*, vol. 11, no. 9, 2020.
- [2] X. Zhang, X. Fu, Z. Xiao, H. Xu, and Z. Qin, "Vessel trajectory prediction in maritime transportation: current approaches and beyond," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 19980–19998, 2023.
- [3] A. Alessandrini, F. Mazzarella, and M. Vespe, "Estimated time of arrival using historical vessel tracking data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 7–15, 2018.
- [4] K. Park, S. Sim, and H. Bae, "Vessel estimated time of arrival prediction system based on a path-finding algorithm," *Maritime Transport Research*, vol. 2, p. 100012, 2021.
- [5] O. Bodunov, F. Schmidt, A. Martin, A. Brito, and C. Fetzer, "Real-time destination and eta prediction for maritime traffic," in *Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems*, 2018, pp. 198–201.
- [6] T. Ogura, T. Inoue, and N. Uchihira, "Prediction of arrival time of vessels considering future weather conditions," *Applied Sciences*, vol. 11, no. 10, p. 4410, 2021.
- [7] N. H. M. Salleh, R. Riahi, Z. Yang, and J. Wang, "Predicting a container ship's arrival punctuality in liner operations by using a fuzzy rule-based bayesian network (frbnn)," *The Asian Journal of Shipping and Logistics*, vol. 33, no. 2, pp. 95–104, 2017.
- [8] A. A. Noman, A. Heuermann, S. A. Wiesner, and K.-D. Thoben, "Towards data-driven gru based eta prediction approach for vessels on both inland natural and artificial waterways," in *2021 IEEE International Intelligent Transportation Systems Conference*. IEEE, 2021, pp. 2286–2291.
- [9] J. Yu, G. Tang, X. Song, X. Yu, Y. Qi, D. Li, and Y. Zhang, "Ship arrival prediction and its value on daily container terminal operation," *Ocean Engineering*, vol. 157, pp. 73–86, 2018.
- [10] A. Balster, O. Hansen, H. Friedrich, and A. Ludwig, "An eta prediction model for intermodal transport networks based on machine learning," *Business & Information Systems Engineering*, vol. 62, pp. 403–416, 2020.
- [11] S. El Mekkaoui, L. Benabbou, and A. Berrado, "Deep learning models for vessel's eta prediction: bulk ports perspective," *Flexible Services and Manufacturing Journal*, vol. 35, no. 1, pp. 5–28, 2023.
- [12] C. I. Valero, Á. Martínez, R. Oltra-Badenes, H. Gil, F. Boronat, and C. E. Palau, "Prediction of the estimated time of arrival of container ships on short-sea shipping: A pragmatcal analysis," *IEEE Latin America Transactions*, vol. 20, no. 11, pp. 2354–2362, 2022.
- [13] P. Cammin, M. Sarhani, L. Heilig, and S. Voß, "Applications of real-time data to reduce air emissions in maritime ports," in *Lecture Notes in Computer Science*, vol. 12202. Springer, 2020, pp. 31–48.
- [14] X. Zhang, H. Peng, J. Zhang, and Y. Wang, "A cost-sensitive attention temporal convolutional network based on adaptive top-k differential evolution for imbalanced time-series classification," *Expert Systems With Applications*, vol. 213, p. 119073, 2023.
- [15] J. Deng, X. Chen, R. Jiang, X. Song, and I. W. Tsang, "A multi-view multi-task learning framework for multi-variate time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–16, 2022.
- [16] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proceedings of 2016 International Conference on Learning Representations (ICLR)*, 2016.
- [17] "Psam general operating conditions," <https://www.psamarine.com/wp-content/uploads/2022/11/PSAM-General-Operating-Conditions-Updated-1-Nov-2022.pdf>, accessed: 2023-05-17.
- [18] X. Zhang, Z. Zhao, Y. Zheng, and J. Li, "Prediction of taxi destinations using a novel data embedding method and ensemble learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 68–78, 2020.
- [19] Y. Geng and X. Luo, "Cost-sensitive convolutional neural networks for imbalanced time series classification," *Intelligent Data Analysis*, vol. 23, no. 2, pp. 357–370, 2019.
- [20] C. Ma, G. Dai, and J. Zhou, "Short-term traffic flow prediction for urban road sections based on time series analysis and lstm_bilstm method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5615–5624, 2022.
- [21] N. Forti, L. M. Millefiori, P. Braca, and P. Willett, "Prediction of vessel trajectories from ais data via sequence-to-sequence recurrent neural networks," in *Proceedings of 2020 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 8936–8940.
- [22] Y. Xu, J. Zhang, Y. Ren, Y. Zeng, J. Yuan, Z. Liu, L. Wang, and D. Ou, "Improved vessel trajectory prediction model based on stacked-bigrus," *Security and Communication Networks*, vol. 2022, 2022.
- [23] X. Zhang, Z. Zhao, and J. Li, "Arde-n-beats: An evolutionary deep learning framework for urban traffic flow prediction," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2391–2403, 2023.