applied sciences

MDPI

Article

Development of a Machine Learning-Based Framework for Predicting Vessel Size Based on Container Capacity

Indranath Chatterjee [1] and Gyusung Cho [2,*]

[1] Department of Computer Engineering, Tongmyong University, 428, Sinseon-ro, Nam-gu, Busan 48520, Korea
[2] Department of Port Logistics System, Tongmyong University, 428, Sinseon-ro, Nam-gu, Busan 48520, Korea
* Correspondence: gscho@tu.ac.kr

Abstract: Ports are important hubs in logistics and supply chain systems, where the majority of the available data is still not being fully exploited. Container throughput is the amount of work done by the TEU and the ability to handle containers at a minimal cost. This capacity of container throughput is the most important part of the scale of services, which is a crucial factor in selecting port terminals. At the port container terminal, it is necessary to allocate an appropriate number of available quay cranes to the berth before container ships arrive at the port container terminal. Predicting the size of a ship is especially important for calculating the number of quay cranes that should be allocated to ships that will eventually dock at the port terminal. Machine learning techniques are flexible tools for utilizing and unlocking the value of the data. In this paper, we used neighborhood component analysis as a tool for feature selection and state-of-the-art machine learning algorithms for multiclass classification. The paper proposes a novel two-stage approach for estimating and predicting vessel size based on container capacity. Our proposed approach revealed seven unique features of port data, which are the essential parameters for the identification of the vessel size. We obtained the highest average classification accuracy of 97.6% with the linear support vector machine classifier. This study paves a new direction for research in port logistics incorporating machine learning.

Keywords: port logistics; vessel size; Twenty-foot Equivalent Unit (TEU); machine learning; feature engineering; classification

check for updates

Citation: Chatterjee, I.; Cho, G. Development of a Machine Learning-Based Framework for Predicting Vessel Size Based on Container Capacity. *Appl. Sci.* **2022**, *12*, 9999. https://doi.org/10.3390/app12199999

Academic Editors: Teen-Hang Meen, Charles Tijus, Cheng-Chien Kuo, Kuei-Shu Hsu, Kuo-Kuang Fan and Jih-Fu Tu

Received: 31 August 2022
Accepted: 29 September 2022
Published: 5 October 2022

1. Introduction

A port is a place where ships are safely entered, anchored, and moored, natural or artificial, and various logistics activities are performed as a connection point between sea and land transportation. A container terminal is a place where cargo is loaded on a ship or cargo is unloaded from a ship, stored in a yard, and the loading and unloading of containers takes place. It is a connection point between land and sea transportation so that bulk cargo can be handled quickly and efficiently. It has a comprehensive system in which various systems, such as the export system, and information and management system, are organically operated.

According to the "ACT ON THE DEVELOPMENT, MANAGEMENT, ETC. OF MARINAS" of Korea, "a port is defined as a port equipped with facilities for entering and leaving ships, loading and unloading people, and loading and unloading cargo." In other words, a port is a starting point for maritime transportation and is defined as a connection point that connects the flow of cargo to each port, city, and factory using transportation means such as air, rail, and waterways [1].

Container throughput is the amount of work done by the TEU (Twenty-foot Equivalent Units) and the ability to handle containers at a minimal cost. This capacity of container throughput is the most important part of the scale of services provided by port terminals and is the most important factor in selecting port terminals for shipping companies. In addition, to improve container throughput, many container terminals are making efforts to provide various services through the introduction and operation of the latest equipment.

At the port container terminal, it is necessary to allocate an appropriate number of available quay cranes to the berth before container ships arrive at the port container terminal. The reason is that it is necessary to increase the operational efficiency of limited quay cranes and to ship on more ships. However, if container ships fail to arrive at the container terminal on time, complex problems arise that require the reallocation of existing planned quay cranes and the rearrangement of ships [2]. Therefore, in the future, it is necessary to allocate a quay crane in consideration of various future situations based on existing past data. Chatterjee and Cho show a way to solve various problems arising from port container terminals by analyzing real-time operation data generated by the terminals through a cloud system, as well as suggesting ways to streamline operations [3].

Predicting the size of a ship is very important for calculating the number of QCs that should be allocated to ships that will eventually dock at the port terminal. The reason is that, although the available QCs for each port terminal are limited, by optimally allocating the QC allocation according to the quantity of container loading and unloading, the goal is to increase the utilization of QC equipment and process containers within a set time for ships docked at the container terminal.

In many aspects, from berth scheduling to quay allocation, it is clear that artificial intelligence (AI) and machine learning (ML) are essential tools for port administration. As an alternative, ML is required for transportation systems to offer intelligent responses to a range of circumstances. Although optimization and simulation modeling has received a lot of attention in port studies, ML has contributed to the development of more complex prediction models for enhanced port operations. To evaluate the benefits of ML to port operations, we carried out this research. Steenken et al. (2004) defined and classified the main logistics activities and functions in container terminals and described a review of techniques for their optimization [4]. Bierwirth and Meisel (2011) reviewed the pertinent literature for identifying appropriate methods that help in modeling problems related to berth allocation and quay scheduling. New classification methods for issues with berth allocation and quay crane scheduling were devised. A special emphasis was placed on integrated solution methods, which are becoming more significant for terminal management [5]. Gharehgozli et al. (2015) reviewed the existing survey articles on container terminal operations. They mostly focused on two things: firstly, modern container terminal skills, and secondly, new operational research (OR) guidelines and models for present research fields [6].

In a study by Xie and Huynh (2010), two kernel-based ML methods were presented as Gaussian processes and ε-support vector machines. To assess their relative performance, they were contrasted with the multilayer feedforward neural network (MLFNN) model, which was applied in earlier investigations. Data from the Port of Houston's Bayport and Barbours Cut container terminals were used to build the model [7]. Gosasang et al. (2011) investigated the use of Linear Regression and Multilayer Perceptron to forecast future container throughput at Bangkok Port. The Bank of Thailand, the Office of the National Economic and Social Development Board, the World Bank, the Ministry of the Interior, and the Energy Policy and Planning Office were contacted to identify the factors impacting cargo throughput at the Bangkok Port. These variables were included in the forecasting MLP and linear regression models, which produced a projection of cargo throughput. The outcomes were then evaluated using mean absolute error and root mean squared error [8].

AI and ML are necessary for port management in many ways, from berth scheduling to quay allocation. According to the literature, there are few publications on the topic, and the most common application of machine learning techniques is to predict various port characteristics. Meanwhile, growing examples of prescriptive and autonomous machine learning approaches are seen in the literature. However, to the best of our knowledge, we find no study involving machine learning and feature engineering to overcome the challenge of predicting the vessel size based on container capacity.

In this paper, we used neighborhood component analysis (NCA) as a tool for feature selection and state-of-the-art machine learning algorithms for multiclass classification. The paper proposes a novel two-stage approach for estimating and predicting vessel size based on container capacity. To the best of our knowledge, for the first time, NCA is employed to select the best set of features in the area of port management. Alongside this, our proposed two-stage model is also novel in its kind.

2. Theoretical Background

When a container ship arrives at the quay, it carries out the loading and unloading of containers. The Figure 1 shows the unloading operation procedure.

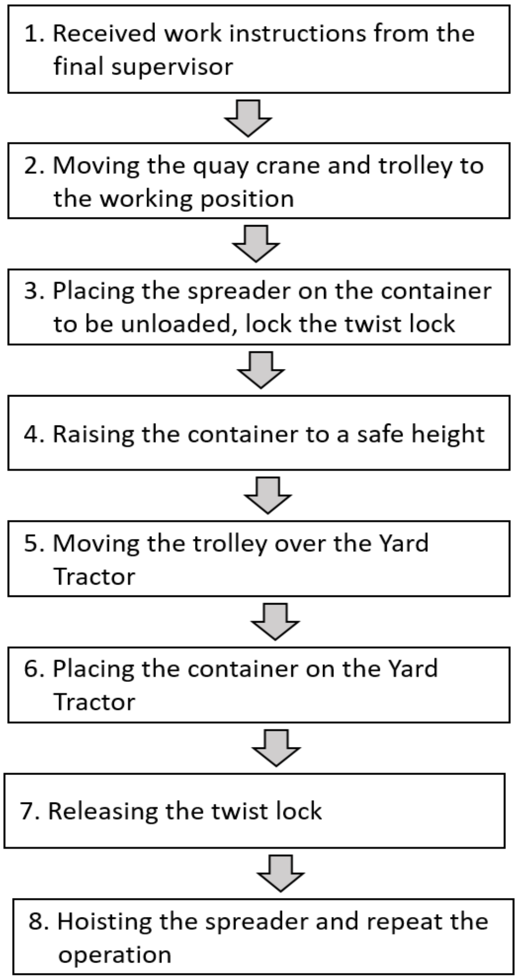
1. Received work instructions from the final supervisor
2. Moving the quay crane and trolley to the working position
3. Placing the spreader on the container to be unloaded, lock the twist lock
4. Raising the container to a safe height
5. Moving the trolley over the Yard Tractor
6. Placing the container on the Yard Tractor
7. Releasing the twist lock
8. Hoisting the spreader and repeat the operation

Figure 1. Workflow diagram for the port terminal.

In the unloading operation, the shipping supervisor is notified of the operation instructions, and after moving to the work location, the container is transported and moved, and loaded to the yard tractor. The unloading operation performs the reverse procedure of the unloading operation. In order to allocate QCs, both facility and operational aspects should be considered. In terms of facilities, quay walls, storage, gates, and loading and unloading equipment are included, and in terms of operation, the labor productivity of container port operators is included. To allocate a smooth QC in a container terminal, it is necessary to secure an appropriate size for carrying out the quantitative loading and to allocate the QC berth within the scope of not overloading the operation. After checking the availability of the berth of the arriving vessel, if all berths are occupied, it is placed on standby. Cho et al. examined how national research and development (R&D) in the domain of logistics has changed recently in the Republic of Korea [2]. Kim et al. examined waves, tide level and sea level fluctuations, design variable estimation, and morphological changes in several studies that applied ML in coastal engineering [9].

3. Materials and Methods

3.1. Dataset Details

The data considered in this study are based on one month's data carried out at the actual container terminal operating in the new port of Busan Port. In addition, the existing data considered a total of 718 operational data. A total of 39 parameters were defined in this study. Figure 2 shows the number of the vessels arrived at quay crane at different time. The following data show some of the operation data considered in this study.

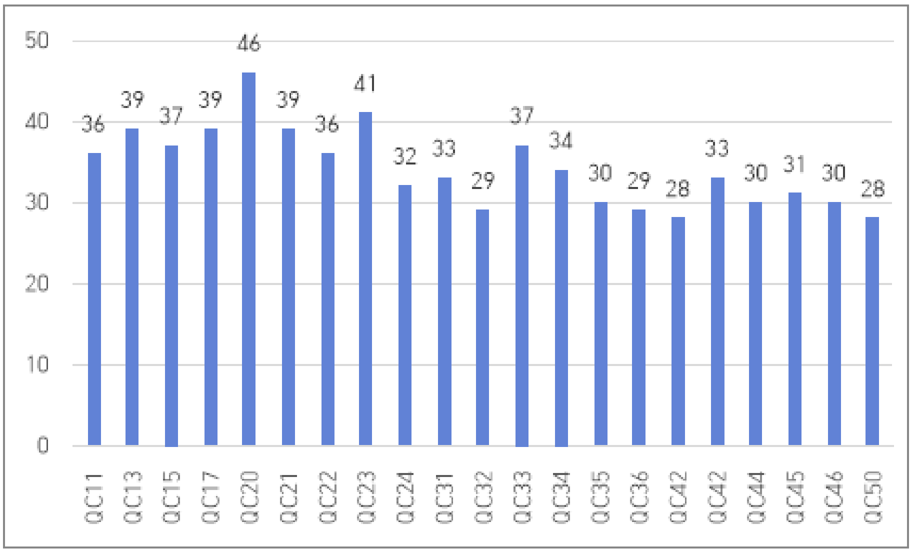

Figure 2. Count of vessels arrived at QC at various arrival times.

The results of the loading and unloading work in the port can be derived from a total of 39 parameters, as shown in Table 1, based on the number of quay cranes assigned to each ship and the processing results for each port equipment. Table 2 shows all the parameters considered for these study before applying.

Table 1. Sample of input data showing each vehicle's total stay time in the port.

ID	Arrival Time	Departure Time	Total Stay Time (s)
ANEA002/20XX	20XX-03-05 01:03:00	20XX-03-05 18:00:00	61,020
ANEA003/20XX	20XX-03-12 00:40:00	20XX-03-12 19:00:00	66,000
ANEA003/20XX	20XX-03-12 00:40:00	20XX-03-12 19:00:00	66,000
ANEA003/20XX	20XX-03-12 00:40:00	20XX-03-12 19:00:00	66,000
ANEA003/20XX	20XX-03-12 00:40:00	20XX-03-12 19:00:00	66,000
ANEA004/20XX	20XX-03-20 12:53:00	20XX-03-21 10:00:00	76,020
ANEA004/20XX	20XX-03-20 12:53:00	20XX-03-21 10:00:00	76,020
ANEA004/20XX	20XX-03-20 12:53:00	20XX-03-21 10:00:00	76,020
ANEA004/20XX	20XX-03-20 12:53:00	20XX-03-21 10:00:00	76,020
ANEA005/20XX	20XX-03-29 23:38:00	20XX-03-31 00:00:00	87,720
ANEA005/20XX	20XX-03-29 23:38:00	20XX-03-31 00:00:00	87,720
ANEA005/20XX	20XX-03-29 23:38:00	20XX-03-31 00:00:00	87,720

Table 1. *Cont.*

ID	Arrival Time	Departure Time	Total Stay Time (s)
ANEA005/20XX	20XX-03-29 23:38:00	20XX-03-31 00:00:00	87,720
APXX003/20XX	20XX-03-28 07:00:00	20XX-03-29 09:00:00	93,600
APXX003/20XX	20XX-03-28 07:00:00	20XX-03-29 09:00:00	93,600
APXX003/20XX	20XX-03-28 07:00:00	20XX-03-29 09:00:00	93,600
APXX003/20XX	20XX-03-28 07:00:00	20XX-03-29 09:00:00	93,600
APXX003/20XX	20XX-03-28 07:00:00	20XX-03-29 09:00:00	93,600
ATGI002/20XX	20XX-03-19 01:58:00	20XX-03-20 02:00:00	86,520
ATGI002/20XX	20XX-03-19 01:58:00	20XX-03-20 02:00:00	86,520
ATGI002/20XX	20XX-03-19 01:58:00	20XX-03-20 02:00:00	86,520
ATGI002/20XX	20XX-03-19 01:58:00	20XX-03-20 02:00:00	86,520
ATGI002/20XX	20XX-03-19 01:58:00	20XX-03-20 02:00:00	86,520
ATGI003/20XX	20XX-03-28 09:20:00	20XX-03-29 04:00:00	67,200
ATGI003/20XX	20XX-03-28 09:20:00	20XX-03-29 04:00:00	67,200
ATGI003/20XX	20XX-03-28 09:20:00	20XX-03-29 04:00:00	67,200
ATGI003/20XX	20XX-03-28 09:20:00	20XX-03-29 04:00:00	67,200
ATSO006/20XX	20XX-02-28 16:30:00	20XX-03-01 10:00:00	63,000
ATSO006/20XX	20XX-02-28 16:30:00	20XX-03-01 10:00:00	63,000
ATSO006/20XX	20XX-02-28 16:30:00	20XX-03-01 10:00:00	63,000
ATSO007/20XX	20XX-03-08 20:52:00	20XX-03-09 18:00:00	76,080
ATSO007/20XX	20XX-03-08 20:52:00	20XX-03-09 18:00:00	76,080

Table 2. Port data terminologies and their definitions.

Number	Parameters	Definition
1	IMPORT_BOXES	Number of Import Boxes
2	EXPORT_BOXES	Number of Export Boxes
3	TEU	Number of TEU
4	Bay	Number of Bays in Yard
5	TIERS	Number of Tiers in Yard
6	Discharging (%)	The ratio of Discharging Containers
7	4000% Discharging	Number of Discharging Containers
8	MTY Discharging	Number of Empty Discharging Containers
9	Reefer Discharging	Number of Reefer Discharging Containers
10	Dangerous Discharging	Number of Dangerous Discharging Containers
11	Over Discharging	Number of Over Discharging Containers
12	% Loading	The ratio of Loading Containers
13	4000% Loading	Number of Loading Containers
14	MTY Loading	Number of Empty Loading Containers
15	Reefer Loading	Number of Reefer Loading Containers
16	Dangerous Loading	Number of Dangerous Loading Containers
17	Over Loading	Number of Over Loading Containers

Table 2. *Cont.*

Number	Parameters	Definition
18	2000% Discharging single lift	Number of Discharging Single Containers
19	2000% Discharging twin	Number of Discharging Two Containers
20	4000% Discharging single	Number of Discharging Single Containers
21	2000% Loading Single lift	Number of Loading Single Containers
22	4000% Loading twin	Number of Loading Two Containers
23	Loading single	Number of Loading Single Containers
24	DS	Number of Discharging Containers
25	20BOX	Number of 20ft Containers
26	40BOX	Number of 40ft Containers
27	MTY	Number of Empty Containers
28	IMDG	Number of International Maritime Dangerous Goods Containers
29	REEFER	Number of Reefer Containers
30	OOG	Number of Discharging Gages
31	TW	Number of Twin Containers (Two 20ft Containers)
32	LD	Number of Loading Containers
33	L_20BOX	Number of Loading 20boxes
34	L_40BOX	Number of Loading 40boxes
35	L_MTY	Number of Loading MTY Containers
36	L_IMDG	Number of Loading IMDG Containers
37	L_REEFER	Number of Loading Reefer Containers
38	L_OOG	Number of Loading OOG Containers
39	L_TW	Number of Loading TW (Twin Containers)

3.2. Methodology

In this paper, we propose a two-stage model consisting of a feature selection method in the first stage, followed by the application of state-of-the-art machine learning algorithms in the second stage (as shown in Figure 3). The first stage employs neighborhood component analysis as a tool for feature selection. As mentioned, the dataset contains 39 parameters, describing the three different kinds of vessels based on their size. The proposed two-stage feature-selection-cum-classification approach is meant to identify the most relevant parameters to estimate the vessel size, instead of using all the unnecessary parameters. In the second stage, we applied advanced machine learning algorithms as a tool for classification. The classification process estimates the efficacy of the identified features (or parameters) selected at stage I. In stage II of the proposed framework, we used 11 classifiers, chosen as per the data compatibility.

3.2.1. Stage I: Feature Selection

In stage I of our approach, we employed NCA for the identification of the relevant parameters useful for proper classification of the vessel size based on container capacity. In this study, the number of quay crane berths allocated using machine learning is calculated based on the arrival time data of container ships operated in existing container terminals, and quay crane berth allocation is performed in consideration of various characteristics of container ship arrival times. A study was conducted to calculate future data based on existing data through machine learning, and a total of 11 machine learning techniques were applied to find the efficacy of the identified features.

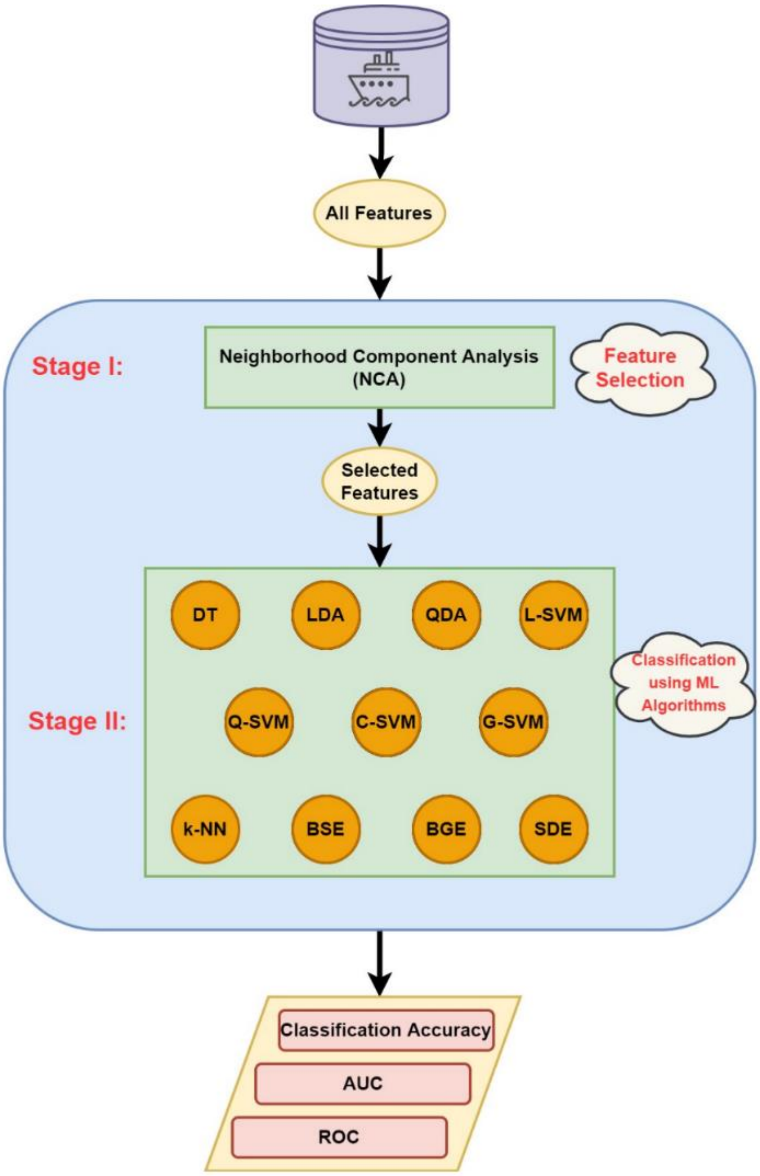

Figure 3. Workflow diagram for our proposed two-stage ML approach.

A supervised learning technique called neighborhood components analysis divides multivariate data into discrete groups based on a predetermined distance measure. It performs the same functions as the K-nearest neighbors method and directly applies the related idea of stochastic nearest neighbors. It is a non-parametric strategy for feature selection that aims to increase the predictive power of regression and classification algorithms [10,11].

By identifying a linear transformation of the input data such that the average leave-one-out (LOO) classification performance is maximized in the converted space, neighborhood components analysis seeks to "learn" a distance measure. The main idea behind the approach is that, by creating a differentiable objective function for matrix A and using an iterative solution such as conjugate gradient descent, one may find a matrix A that corresponds to the transformation. The ability to calculate the number of k classes as a function of A up to a scalar constant is one advantage of this technique. Thus, the problem of model selection is addressed by this use of this method.

Consider predicting a single data point's class label based on the agreement of its k-nearest neighbors using a certain distance measure. This method of categorization is called leave-one-out cross-validation. Assume a problem of multiclass classification with n observations in the training set, as shown in Equation (1):

$$S = \{(x_i,\ y_i),\ i = 1, 2, \ldots, n\} \tag{1}$$

where $xi \in \mathbb{R}p$ stands for feature vectors, $yi \in \{1, 2, \ldots, c\}$ stands for class labels, and c denotes the classes.

The goal is to develop a classifier $f : \mathbb{R}p \to \{1, 2, \ldots, c\}$ that takes a feature vector as input and predicts the true label of x using the formula $f(x)$.

Suppose a randomized classifier that:

- As the *reference point* for x, $Ref(x)$ is chosen at random from S.
- Label x using the reference point's label $Ref(x)$.

This method is comparable to a $1 - NN$ classifier in which the reference point is selected as the new point's close neighbor. Every point in S has a chance of being picked as the reference point, since in NCA, the reference point is chosen at random [11].

The closer x_j is to x, as determined by the distance function d_w, the higher the chance $P\big(Ref(x) = x_j \mid \text{S}\big)$ that point x_j is chosen from S as the reference point for x. Here, the distance function d_w is shown in Equation (2):

$$d_w(x_i, x_j) = \sum_{r=1}^{p} w_r^2 |x_{ir} - x_{jr}| \tag{2}$$

where w_r signifies the feature weights. Let us say in Equation (3),

$$P\big(Ref(x) = x_j | s\big)\ \alpha\ k\big(d_w(x, x_j)\big) \tag{3}$$

where k signifies a random kernel that assumes big values when $dw(x, xj)$ is irrelevant (Equation (4)). Presume it is

$$k(z) = \exp\left(-\frac{z}{\sigma}\right) \tag{4}$$

The goal of neighborhood component analysis is to maximize $F(w)$ concerning w, shown in Equation (5).

$$F(w) = \frac{1}{n} \textstyle\sum_{i=1}^{n} F_i(w) \tag{5}$$

The collection of closest neighbors Ci, however, might alter significantly after all the points have been subjected to a linear transformation. Particularly, any objective function $f(*)$ based on the neighbors of a point is piecewise-constant and, hence, not differentiable, since the set of neighbors for a point might experience discrete changes in response to smooth changes in the components of A [11].

3.2.2. Stage II: Classification

Stage II of the proposed approach deals with the application of advanced ML algorithms as a tool for classification. The features identified after the application of NCA are fed into each of the classifiers to measure the efficacy of the selected features. Here, we used 11 state-of-the-art machine learning classifiers [12,13], namely, Decision Tree (DT), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Linear Support Vector Machine (L-SVM), Quadratic Support Vector Machine (Q-SVM), Cubic Support Vector Machine (C-SVM), Gaussian Support Vector Machine (G-SVM), K-Nearest Neighbours (k-NN), Boosted Tree Ensemble (BSE), Bagged Tree Ensemble (BGE), and Subspace Discriminant Ensemble (SDE). Here, instead of using the generic algorithms, we tuned the hyperparameters of all the algorithms to provide optimal results.

A decision tree is a tree structure that constructs classification or regression models. It progressively divides a dataset into smaller and smaller sections, while also developing an associated decision tree. The result is a tree containing leaf nodes and decision nodes [14].

In this study, we used a tuned decision tree, where we took the Gini Diversity Index as a split criterion, with a maximum number of 100 splits and fine tree type.

Linear Discriminant Analysis is a linear classification model. Data from a D dimensional feature space are projected using LDA into a D′ (D > D′) dimensional space to increase variability across classes, while minimizing variability within classes. LDA works well while dealing with multiclass classification [15].

Here, in the case of linear discriminant analysis, we used the “full” covariance structure.

Quadratic Discriminant Analysis is a generative model. According to QDA, each class is thought to have a Gaussian distribution. The fraction of data points that belong to the class is the class-specific prior. The average of the input variables that are part of the class makes up the mean vector particular to that class. Simply put, the covariance of the class-specific vectors makes up the class-specific covariance matrix [16].

In this work, we have used a “full” covariance structure in the case of quadratic discriminant analysis.

Linear Support Vector Machine is a linear model that may be used to solve classification and regression issues. It can handle linear and non-linear problems and is useful for a wide range of practical applications. The concept of SVM is straightforward: The method draws a line or a hyperplane to divide the data into classes [17].

In this experiment, we tuned the hyperparameters of the linear SVM for the betterment of the classification results. To obtain the optimal results, we considered the “Linear Kernel” with an “automatic” kernel scaling, keeping the box constraint level as 1, and considering the multiclass method as one vs one.

Quadratic kernel-free non-linear support vector machine (Q-SVM) is a quadratic decision function capable of separating data non-linearly. The geometrical margin is shown to be equal to the inverse of the gradient of the decision function’s norm. The equation of the quadratic function is the functional margin. It is demonstrated that Q-SVM may be used in a quadratic optimization context [17].

Here, in this study, we tuned the hyperparameters of the SVM for the betterment of the classification results. To obtain the optimal results, we considered the “Quadratic Kernel” with an “automatic” kernel scaling, keeping the box constraint level as 1, and considering the multiclass method as one vs one.

Cubic support vector machine (C-SVM): when dealing with a memory space constraint, C-SVM is an effective SVM approach because it locates a hyperplane in a multidimensional space that best separates the classes, whereas Q-SVM has low memory utilization for binary classification and high memory utilization for multiclass classification during its training phase. Prediction speed is also fast for binary classification but sluggish for multiclass classification [17].

In this experiment, we tuned the hyperparameters of the SVM for the betterment of the classification results. To obtain the optimal results, we have considered the “cubic Kernel” with an “automatic” kernel scaling, keeping the box constraint level as 1, and considering the multiclass method as one vs one.

Gaussian support vector machine (G-SVM) is another prominent Kernel approach used in Support Vector Machine models, using the Gaussian RBF (Radial Basis Function). The RBF kernel is a function whose value is proportional to the distance between the origin and some point [17].

We tuned the hyperparameters of the SVM for the betterment of the classification results. To obtain the optimal results, we have considered the “Gaussian Kernel” with an “automatic” kernel scaling, keeping the box constraint level as 1, and considering the multiclass method as one vs one.

K-nearest neighbor (k-NN) is a variant of k-nearest neighbors. The selection of the hyperparameter k is one of several factors that influence the performance of the KNN algorithm. If k is too little, the algorithm becomes more susceptible to outliers [18].

In this experiment, to optimize the algorithm, we tuned the k-NN model with a hyperparameter tuning, empirically. In this study, we used the Cosine variant of the k-NN classifier, with the number of neighbors as 10, considering the distance weight as "equal" and the distance metric as "cosine".

Boosted trees ensemble learner (BSE) is used to reduce training time errors. Boosting is an ensemble learning strategy that combines a group of weak learners into strong learners. A random sample of data is chosen, fitted with a model, and then trained sequentially—that is, each model attempts to compensate for the shortcomings of its predecessor [19,20].

This study attempts to employ the ensemble learner to check the variability of the results for the selected set of features. We also tuned the hyperparameters of the BSE learner during the training phase of the model. Here, we considered the ensemble method as "AdaBoost", learner type as "decision tree", the maximum number of allowed splits as 20, the number of learners as 30, and the rate of learning as 0.1.

Bagged trees ensemble learner (BSE) is an acronym for Bootstrap Aggregation; it is an ensemble method, which is essentially a mechanism for overlaying diverse models, data, and methods. [20].

Here, in this study, for the hyperparameter tuning of the BSE, we set the following hyperparameter for the optimal result condition. Here, we used the ensemble type as "Bag", learner type as "Decision tree", and the number of learners as 30.

Subspace Discriminant Ensemble learner (SDE): The majority voting rule was utilized to create the subspace discriminant ensemble, which employed the random subspace ensemble approach with 30 linear discriminant learners and two subspace dimensions [21].

Cross-Validation Method

A statistical technique called cross-validation is used to evaluate the competence of machine learning models. Since it is simple to comprehend, simple to implement, and produces ability estimates that often have a smaller bias than other approaches, it is frequently used in applied machine learning to compare and select a model for a specific predictive modeling issue. The process contains a single parameter, k, that designates how many groups should be created from a given data sample. As a result, the process is frequently referred to as k-fold cross-validation. If k is decided to be a certain value, k in reference to the model may be replaced by that value.

This study incorporates the k-fold cross-validation method for training and validating the trained model. Here, we used a 10-fold cross-validation method, where nine folds were taken for training and one fold for testing. In this study, for every iteration, around 10% of the total samples were hidden from all the training data, which were considered unseen testing samples, and others were used for training. This process was repeated 10 times to avoid any possibilities of bias in the study.

Evaluation Metrics

In this study, we used three metrics for the purpose of evaluating the efficacy of our proposed approach. The three evaluation metrics used are accuracy, receiver operating characteristics (ROC) curve, and area under curve (AUC).

1. Accuracy: One parameter for assessing classification models is accuracy. The percentage of predictions that our model correctly predicted is known as accuracy. The following is the actual definition of accuracy (as shown in Equation (6)):

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{6}$$

where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

2. Receiver Operating Characteristics (ROC): The receiver operating characteristic curve is a chart that displays how well a classification model performs across all categorization levels. Two parameters are shown on this curve:

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN} \tag{7}$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN} \tag{8}$$

   TPR (Equation (7)) vs. FPR (Equation (8)) are shown on a ROC curve at various classification levels. More items are classified as positive when the classification threshold is lowered, which raises the number of both False Positives and True Positives.

3. Area Under Curve (AUC): Integral calculus is used to calculate the AUC; it measures the full two-dimensional region beneath the entire ROC curve from (0,0) to (1,1). An overall assessment of performance across all potential classification criteria is provided by the AUC. The AUC may be seen as the likelihood that the model values a randomly chosen positive example higher than a randomly chosen negative example.

4. Results

With the help of the proposed two-staged approach, we obtained the set of most relevant parameters or features out of 39 inputted features. These seven features are capable of distinguishing the type of vessel in terms of size while considering the container capacity. Our proposed feature selection method revealed the set of seven unique features, which are as follows:

- Import box
- 4000% Loading
- Loading single
- 20_Box
- 40_Box
- TW
- LD

The qualitative details of these seven identified features are mentioned above in the Dataset Details section. In addition, we obtained impressive results in terms of classification accuracy while performing the multiclass classifiers, as mentioned above. We obtained the highest average classification accuracy of 97.6% with the linear support vector machine classifier. While applying the decision tree classifier, we obtained an accuracy of 87.5%, whereas, while applying the ensemble tree with the boosted and bagged methods, we obtained a high accuracy of 90.4% and 92.6%, respectively. Other than the linear kernel, all kernels of the support vector machine showed promising results. We obtained 96.8%, 95.5%, and 93.9% accuracy with quadratic, cubic, and Gaussian kernels of the support vector machine, respectively. The complete classification results are shown in Table 3, where accuracy and the AUC for each algorithm are given.

Alongside this, here we show the ROC curve and the confusion matrix for each of the classifiers. Figure 4 depicts the confusion matrix for each of the algorithms used. Here, the order of appearance of sub-figures (from left to right) is a = Decision Tree (Model 6.1), b = Linear Discriminant Analysis (Model 6.4), c = Quadratic Discriminant Analysis (Model 6.5), d = Linear Support Vector Machine (Model 6.6), e = Quadratic Support Vector Machine (Model 6.7), f = Cubic Support Vector Machine (Model 6.8), g = Gaussian Support Vector Machine (Model 6.10), h = K-Nearest Neighbor (Model 6.15), i = Boosted Tree Ensemble (Model 6.18), j = Bagged Tree Ensemble (Model 6.19), and k = Subspace Discriminant Ensemble Learner (Model 6.20). From the confusion matrix, we can note down the cases of true classes and predicted classes; thereby, we are able to trace the true positive rate

(TPR) and false positive rate (FPR). Our results showpromising results in terms of correct prediction of vessel size. Figure 5 depicts the ROC curves for each of the algorithms used. The order of the appearance of each subfigure is the same as in Figure 4.

Table 3. Results show the classification accuracy and area under curve (AUC) for each of the classifiers. DT = Decision Tree, LDA = Linear Discriminant Analysis, QDA = Quadratic Discriminant Analysis, L-SVM = Linear Support Vector Machine, Q-SVM = Quadratic Support Vector Machine, C-SVM = Cubic Support Vector Machine, G-SVM = Gaussian Support Vector Machine, k-NN = K-Nearest Neighbor, BSE = Boosted Tree Ensemble, BGE = Bagged Tree Ensemble, and SDE = Subspace Discriminant Ensemble Learner.

	DT	LDA	QDA	L-SVM	Q-SVM	C-SVM	G-SVM	k-NN	BSE	BGE	SDE
Accuracy	87.50%	90.90%	92.60%	97.60%	96.80%	95.50%	93.90%	91.40%	90.40%	92.60%	91.80%
AUC	0.95	1	0.99	1	1	1	1	0.98	0.99	0.99	1

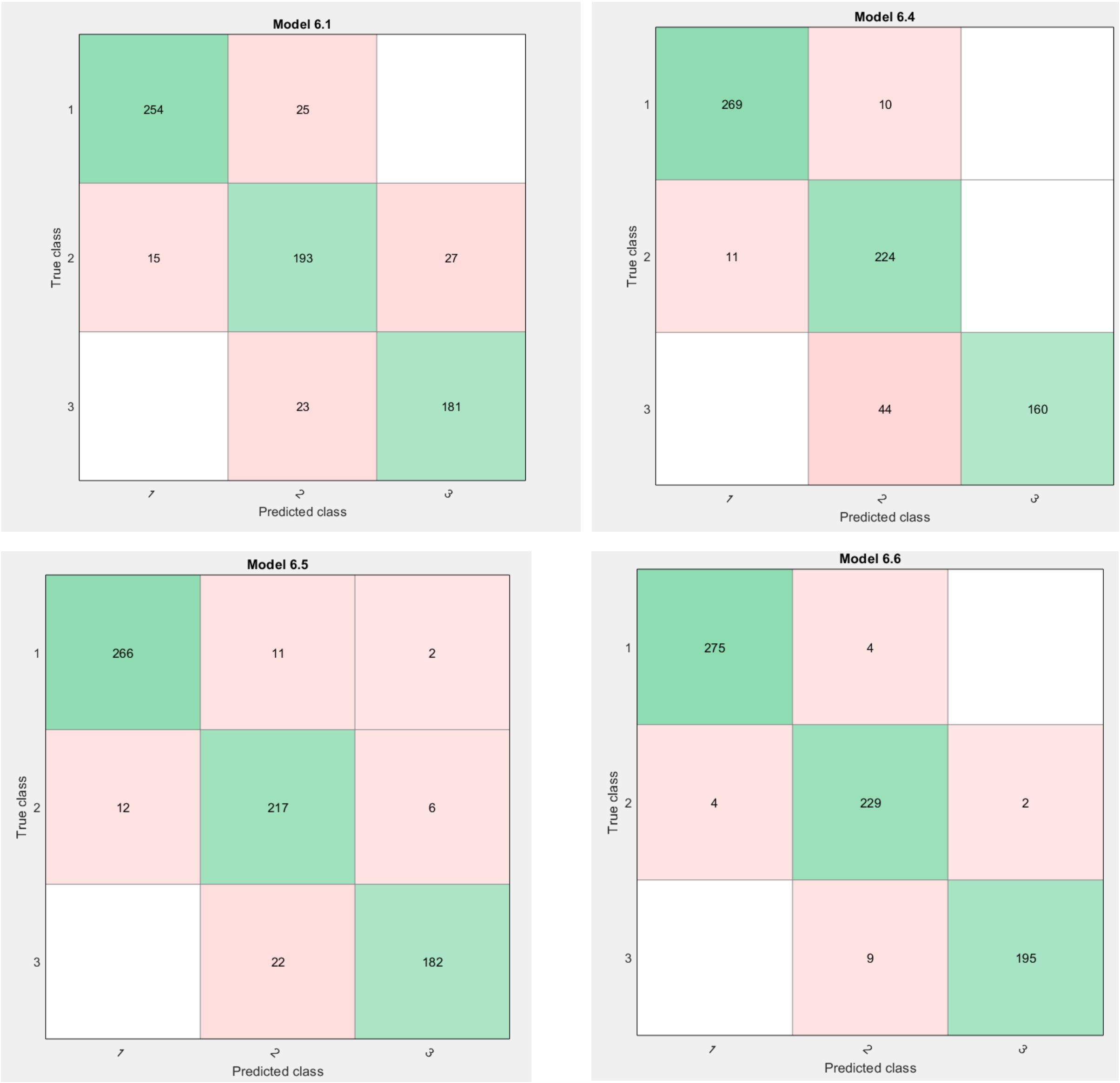

Figure 4. *Cont.*

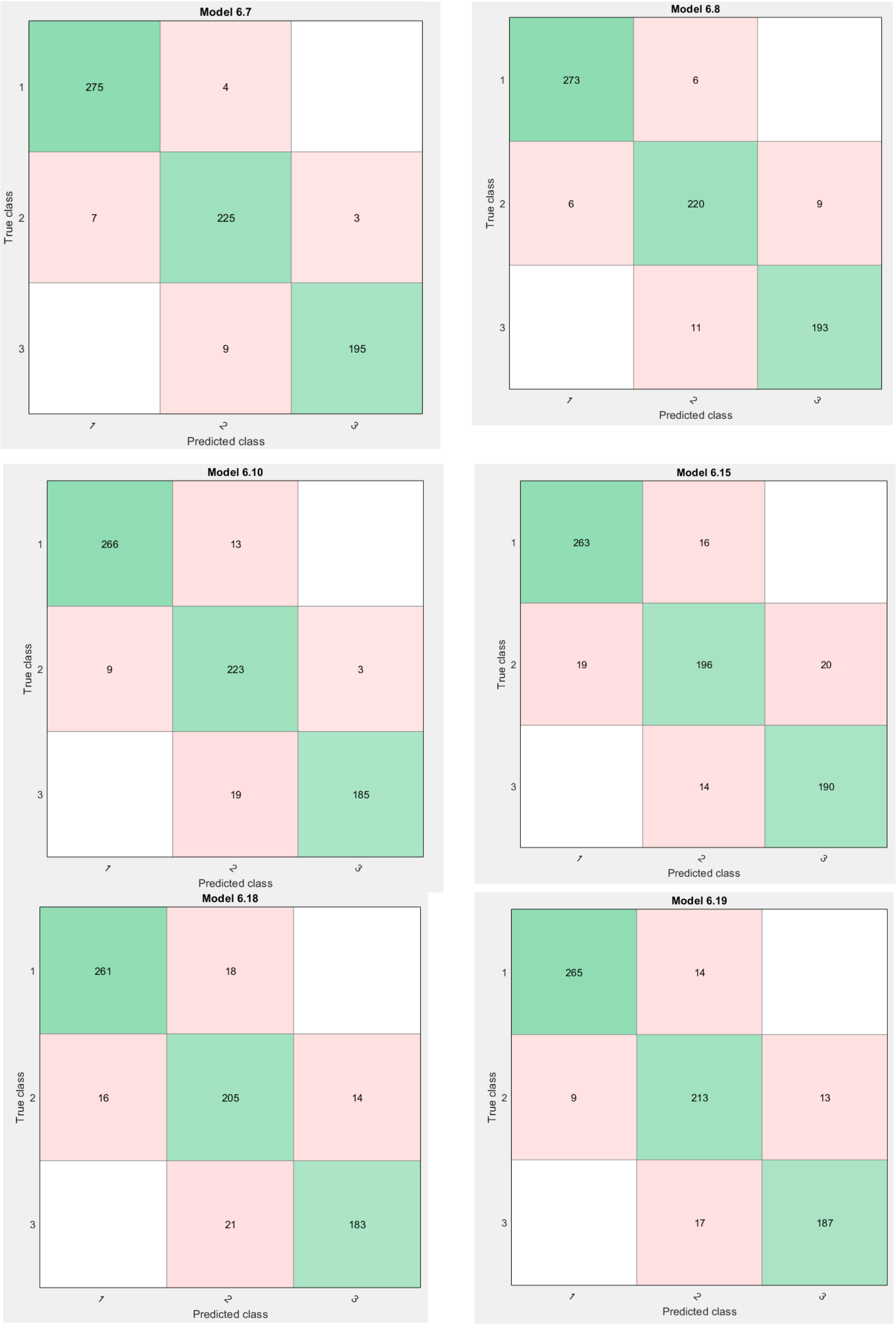

Figure 4. *Cont.*

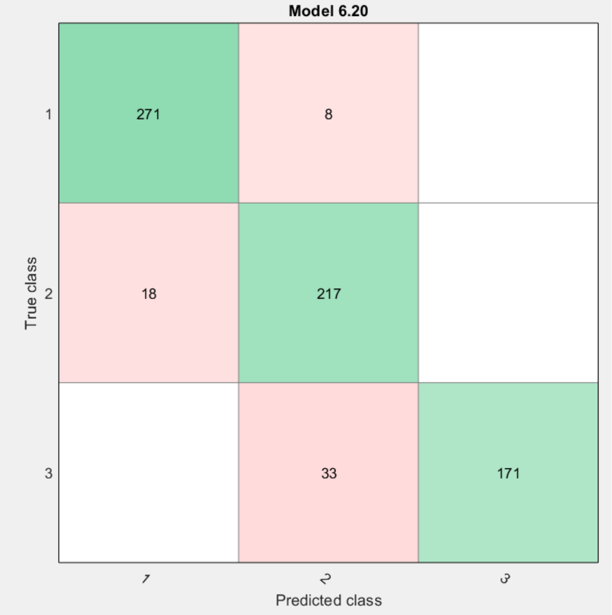

Figure 4. Confusion matrix for each classifier. Order of appearance of sub-figures (from left to right): a = Decision Tree (Model 6.1), b = Linear Discriminant Analysis (Model 6.4), c = Quadratic Discriminant Analysis (Model 6.5), d = Linear Support Vector Machine (Model 6.6), e = Quadratic Support Vector Machine (Model 6.7), f = Cubic Support Vector Machine (Model 6.8), g = Gaussian Support Vector Machine (Model 6.10), h = K-Nearest Neighbor (Model 6.15), i = Boosted Tree Ensemble (Model 6.18), j = Bagged Tree Ensemble (Model 6.19), and k = Subspace Discriminant Ensemble Learner (Model 6.20).

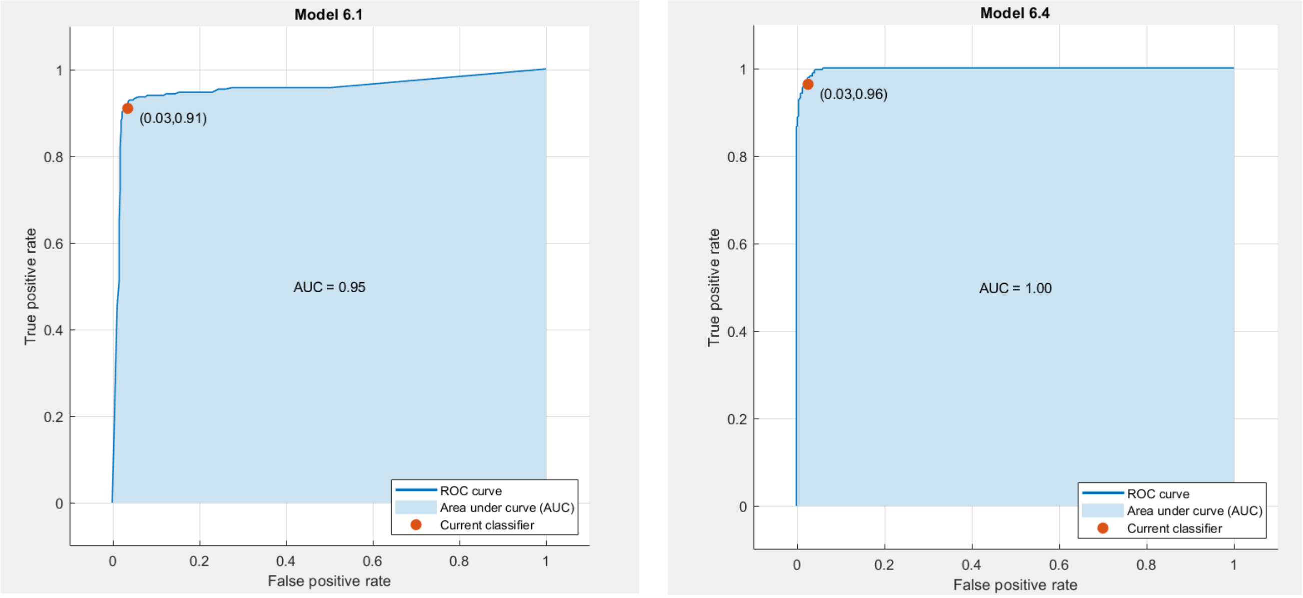

Figure 5. *Cont.*

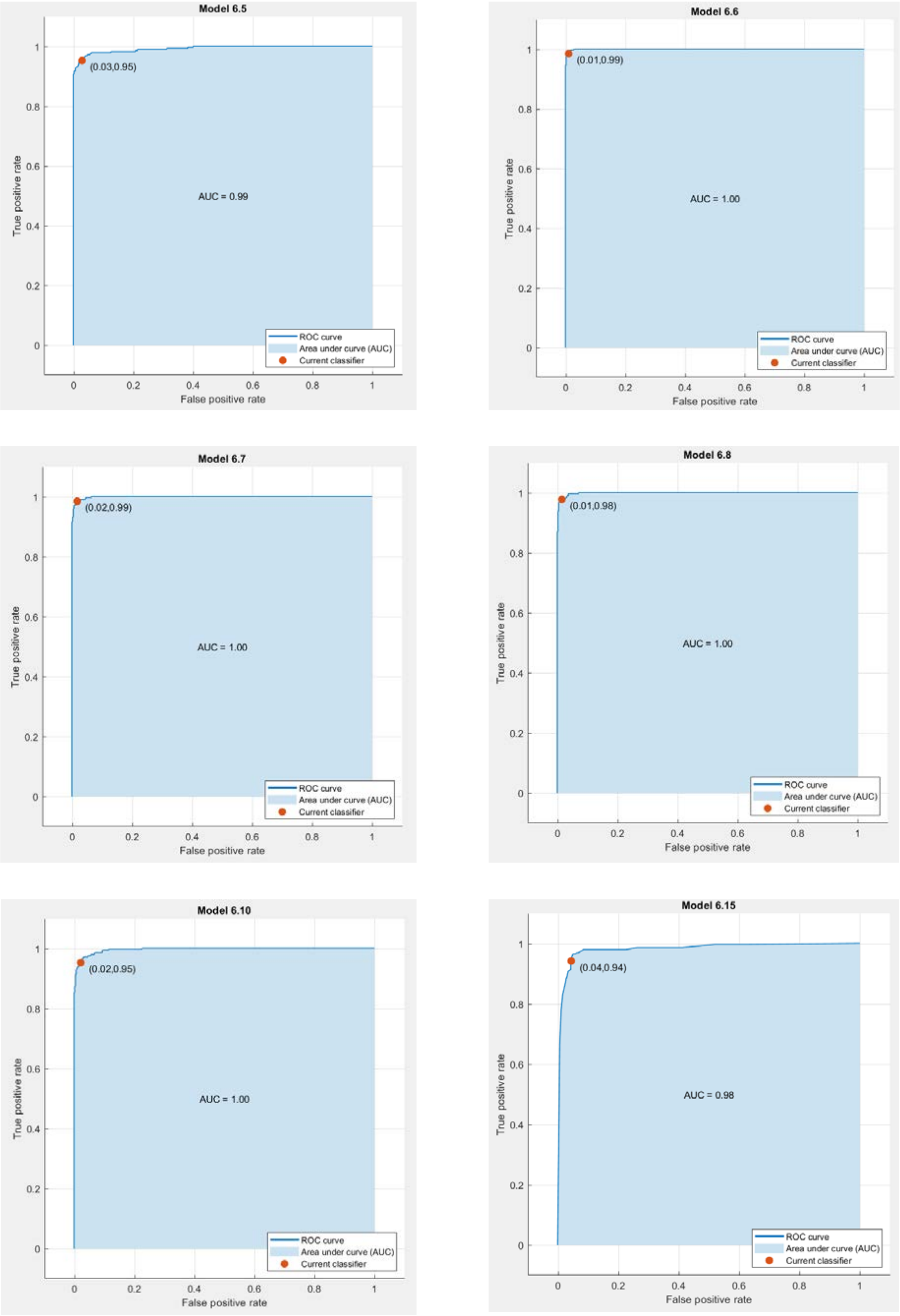

Figure 5. *Cont.*

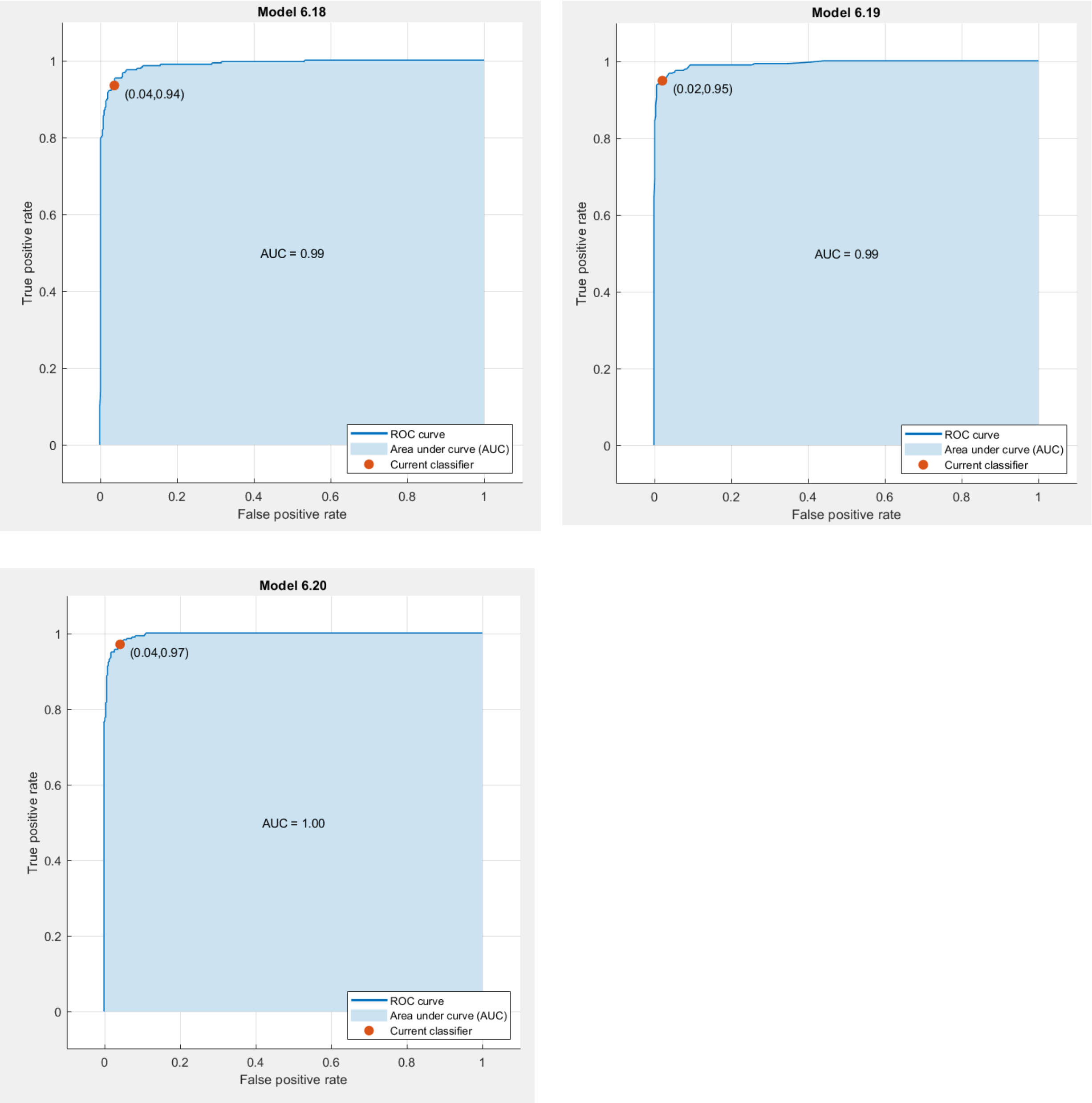

Figure 5. The ROC Curves show AUC values. Order of appearance of sub-figures (from left to right): a = Decision Tree (Model 6.1), b = Linear Discriminant Analysis (Model 6.4), c = Quadratic Discriminant Analysis (Model 6.5), d = Linear Support Vector Machine (Model 6.6), e = Quadratic Support Vector Machine (Model 6.7), f = Cubic Support Vector Machine (Model 6.8), g = Gaussian Support Vector Machine (Model 6.10), h = K-Nearest Neighbor (Model 6.15), i = Boosted Tree Ensemble (Model 6.18), j = Bagged Tree Ensemble (Model 6.19), and k = Subspace Discriminant Ensemble Learner (Model 6.20).

5. Discussion

Ports are important hubs in logistics and supply chain systems, where the majority of the available data is still not being fully exploited. Machine learning techniques are flexible tools for utilizing and unlocking the value of the data. The port sector is significantly behind other forms of transportation in this change, which is surprising given the rapidly expanding usage of machine learning as a tool for data-driven prediction.

In this paper, we proposed a two-stage model consisting of a feature selection method involving neighborhood component analysis, followed by a classification task using 11 state-of-the-art machine learning algorithms. In step I of the proposed approach, we obtained seven unique features, which are the essential parameters for the identification of the vessel size. The selected features are Import box, 4000% Loading, Loading single, 20_Box, 40_Box, TW, and LD. Here, the container capacity plays a vital role in distinguishing the size of the vessel. Out of these seven features, we also made an effort to tune the algorithm with different permutations. However, we found that the set of these seven parameters can achieve high classification accuracy. It is worth mentioning that we achieved the highest classification accuracy of 97.6% and AUC = 1 with the support vector machine classifier with a linear kernel. Alongside this, we also achieved promising classification accuracies with other algorithms (as shown in Table 3).

Figures 3 and 4 show the confusion matrixes and the ROC curves for all the state-of-the-art machine learning algorithms used in this study. The promising results, as shown in the figures, prove the efficacy of our proposed method. Most of the ML classifiers showed high classification accuracy for these seven distinct parameters. The AUC values of 1 or close to one signify the classifiers are well-trained and well-chosen for this dataset.

The finally selected 7 out of 39 parameters are essential data derived through the quantitative work of berthed ships, and these data are the basic items for calculating the operational performance and throughput of the port terminal.

The seven parameters selected through this study are ultimately used as the basic data for the calculation of throughput performed at the container terminal. QC allocation prediction for each ship was performed through the seven parameters derived in this study. In the future, container terminals are changing into unmanned and automated ports. Therefore, it will be possible to use basic data to calculate the container throughput in the unmanned automated port and the number of ships that can be handled in the unmanned automated port through the seven parameters presented in this study. This study, which incorporates a machine learning-based study, paves the way for future research in port logistics.

6. Conclusions

Whenever container ships arrive at the port container terminal, it is important to assign an adequate number of available quay cranes to the berth. To allocate quay cranes to ships that will eventually dock at the port terminal, it is particularly crucial to predict the size of a ship. In this study, we chose features using neighborhood component analysis and classified classes using cutting-edge machine learning techniques. To estimate and anticipate the vessel size based on container capacity, the research suggests an innovative two-stage technique. Our suggested method identified seven distinctive characteristics of port data that are crucial indicators of vessel size.

Author Contributions: Conceptualization, I.C. and G.C.; methodology, I.C. and G.C.; software, I.C. and G.C.; validation, I.C. and G.C.; formal analysis, I.C. and G.C.; investigation, I.C. and G.C.; resources, I.C. and G.C.; data curation, I.C. and G.C.; writing—original draft preparation, I.C. and G.C.; writing—review and editing, I.C. and G.C.; visualization, I.C. and G.C.; supervision, I.C. and G.C.; administration, G.C.; funding acquisition, G.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research work and the APC for the publication were funded by the National Research Foundation of Korea (NRF) grant funded by the Government of Korea (MSIP) with grant number NRF-2021R1I1A3046794.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: I.C. and G.C. are thankful to Tongmyong University, Busan, Korea for providing the platform to perform this research work. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Government of Korea (MSIP).

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

References

1. Republic of Korea. Act on the Development, Management, etc. of Marinas. Available online: https://www.law.go.kr (accessed on 1 April 2021).
2. Jeong, J.Y.; Cho, G.; Yoon, J. Trend analysis on Korea's National R&D in logistics. *J. Ocean. Eng. Technol.* **2020**, *34*, 461–468.
3. Chatterjee, I.; Cho, G. Port Container Terminal Quay Crane Allocation Based on Simulation and Machine Learning Method. *Sens. Mater.* **2022**, *34*, 843–853. [CrossRef]
4. Steenken, D.; Voß, S.; Stahlbock, R. Container terminal operation and operations research-a classification and literature review. *OR Spectr.* **2004**, *26*, 3–49.
5. Bierwirth, C.; Meisel, F. A survey of berth allocation and quay crane scheduling problems in container terminals. *European J. Oper. Res.* **2010**, *202*, 615–627. [CrossRef]
6. Gharehgozli, A.H.; Roy, D.; De Koster, R. Sea container terminals: New technologies and OR models. *Marit. Econ. Logist.* **2016**, *18*, 103–140. [CrossRef]
7. Xie, Y.; Huynh, N. Kernel-based machine learning models for predicting daily truck volume at seaport terminals. *J. Transp. Eng.* **2010**, *136*, 1145–1152. [CrossRef]
8. Gosasang, V.; Chandraprakaikul, W.; Kiattisin, S. A comparison of traditional and neural networks forecasting techniques for container throughput at Bangkok port. *Asian J. Shipp. Logist.* **2011**, *27*, 463–482. [CrossRef]
9. Kim, T.; Lee, W.D. Review on Applications of Machine Learning in Coastal and Ocean Engineering. *J. Ocean. Eng. Technol.* **2022**, *36*, 194–210. [CrossRef]
10. Yang, W.; Wang, K.; Zuo, W. Fast neighborhood component analysis. *Neurocomputing* **2012**, *83*, 31–37. [CrossRef]
11. Yang, W.; Wang, K.; Zuo, W. Neighborhood component feature selection for high-dimensional data. *J. Comput.* **2012**, *7*, 161–168. [CrossRef]
12. Bonaccorso, G. *Machine Learning Algorithms*; Packt Publishing Ltd.: Birmingham, UK, 2017.
13. Kumar, A.; Chatterjee, I. Data Mining: An experimental approach with WEKA on UCI Dataset. *Int. J. Comput. Appl.* **2016**, *138*, 13. [CrossRef]
14. Quinlan, J.R. Learning decision tree classifiers. *ACM Comput. Surv.* **1996**, *28*, 71–72. [CrossRef]
15. Izenman, A.J. Linear discriminant analysis. In *Modern Multivariate Statistical Techniques*; Springer: New York, NY, USA, 2013; pp. 237–280.
16. Srivastava, S.; Gupta, M.R.; Frigyik, B.A. Bayesian quadratic discriminant analysis. *J. Mach. Learn. Res.* **2007**, *8*, 1277–1305.
17. Suthaharan, S. Support vector machine. In *Machine Learning Models and Algorithms for Big Data Classification*; Springer: Boston, MA, USA, 2016; pp. 207–235.
18. Hu, Q.; Yu, D.; Xie, Z. Neighborhood classifiers. *Expert Syst. Appl.* **2008**, *34*, 866–876. [CrossRef]
19. Zięba, M.; Tomczak, S.K.; Tomczak, J.M. Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert Syst. Appl.* **2016**, *58*, 93–101. [CrossRef]
20. Dietterich, T.G. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 1–15.
21. Shin, J. Random subspace ensemble learning for functional near-infrared spectroscopy brain-computer interfaces. *Front. Hum. Neurosci.* **2020**, *14*, 236. [CrossRef] [PubMed]