sustainability

MDPI

Article

Machine Learning-Based Models for Accident Prediction at a Korean Container Port

Jae Hun Kim, Juyeon Kim, Gunwoo Lee * and Juneyoung Park

Department of Transportation & Logistics Engineering, Hanyang University, Ansan 15588, Korea; jhkim8182@hanyang.ac.kr (J.H.K.); lisa9780@hanyang.ac.kr (J.K.); juneyoung@hanyang.ac.kr (J.P.)
* Correspondence: gunwoo@hanyang.ac.kr

Abstract: The occurrence of accidents at container ports results in damages and economic losses in the terminal operation. Therefore, it is necessary to accurately predict accidents at container ports. Several machine learning models have been applied to predict accidents at a container port under various time intervals, and the optimal model was selected by comparing the results of different models in terms of their accuracy, precision, recall, and F1 score. The results show that a deep neural network model and gradient boosting model with an interval of 6 h exhibits the highest performance in terms of all the performance metrics. The applied methods can be used in the predicting of accidents at container ports in the future.

Keywords: container port; machine learning; accident prediction model; neural network; random forest; gradient boosting

Citation: Kim, J.H.; Kim, J.; Lee, G.; Park, J. Machine Learning-Based Models for Accident Prediction at a Korean Container Port. *Sustainability* **2021**, *13*, 9137. https://doi.org/10.3390/su13169137

Academic Editor: Sara Moridpour

Received: 20 July 2021
Accepted: 12 August 2021
Published: 16 August 2021

1. Introduction

Most of the existing studies on the risk assessment of maritime ports have focused on port security [1–3] and port safety [4–6]. Notably, most studies focusing on the security considered only unusual events such as hostile attacks [7] and the smuggling of weapons [8], and most studies pertaining to port safety focused on accidents that occurred during usual port activities such as loading, discharging, importing, and exporting. In this regard, research on port safety must be emphasized over that of port security.

In the maritime field, several researchers have examined safety by predicting vessel accidents on the waterway [9], forecasting coastal waves [10], and examining ship collisions [11,12], among other aspects. Moreover, although several researchers have implemented risk assessment methods to identify the risk factors associated with container ports [4–6], research regarding the prediction of container port accidents remains limited.

In a container port, several activities, including the loading, discharging, importing, and exporting of containers are performed by port workers using equipment such as yard tractors and container cranes. Owing to these extensive activities, container ports are prone to accidents, such as equipment–equipment collisions, equipment–container collisions, injuries, and container damages during discharging, loading, and moving. These accidents may result in damages to workers, equipment, and containers, as well as in economic loss. In particular, according to the statistics of occupational accidents in Korea [13], as shown in Table 1, the total number of accidents occurring at all container ports in Korea led to 4 fatalities and 91 injuries in 2015 and 3 fatalities and 96 injuries in 2019; the damages have gradually increased every year, except for in 2016. The estimated economic losses resulting from the accidents were approximately KRW 18.8 and 20.5 billion in 2015 and 2018, respectively. Moreover, the number of accidents involving minor injuries is considerably larger. Therefore, predicting accidents at a container port is essential for minimizing the economic loss in port operation and enhancing the port safety.

Table 1. Statistics of occupational accidents and economic loss associated with container ports in Korea (2015–2019).

		2015	2016	2017	2018	2019
Accidents	Fatality	4	1	1	5	3
	Injury	91	79	96	94	96
	Total	94	80	97	99	99
Economic loss (billion KRW)		18.8	17.2	20.8	20.5	NA

Source: Korea Port Logistics Association.

To predict uncommon events such as accidents, machine learning methods including neural network models, random forest, and gradient boosting have been used. In the transportation field, these methods have been widely applied to predict traffic accidents on roads. Specifically, neural networks have been used to predict vehicle crash accidents on roads [14] and the duration of traffic accidents [15]. Random forest models have been applied to detect traffic accidents [16] and identify taxi drivers with a high risk of accidents [17]. Gradient boosting models have been used to predict traffic accidents on roads [18,19] and railways [20]. Moreover, several comparative studies have been performed regarding machine learning methods, including the use of neural networks to predict the severity associated with traffic accidents [21,22].

In the field of maritime and port logistics, machine learning methods have been applied in certain studies to predict the risk associated with accidents. Several researchers have applied neural networks to predict vessel accidents on the waterway [9] and risk in maritime safety [23] and forecast coastal waves [10]. A neural network model was applied to develop a ship collision risk model [11]. Moreover, a random forest model was applied to predict the severity of ship collision accidents [12]. However, relatively few studies have applied machine learning methods to predict accidents in a container port.

In Korea, most studies associated with analyzing and predicting accidents have been conducted in road transportation fields, considering the characteristics of vehicles and pedestrians [24–27]. Most studies on container ports in Korea were focused on analyzing the safety factors related to loading and unloading activities [28], developing risk assessment methods based on the type of accidents occurring in the port [29,30], and examining the influence of education on the risk factors of accident occurrences in the ports [31]. Research on the prediction of accidents that occur in activities associated with the port is relatively limited.

Therefore, in this study, the accidents that can occur in a container port are predicted by applying machine learning methods, including a neural network model, random forest model, and gradient boosting model to a historical dataset of accidents that occurred at a container port in Busan, Korea.

2. Methods and Dataset

2.1. Methods

2.1.1. Neural Network Model

A neural network is a machine learning model that has been widely applied in prediction applications. The basic element of a neural network model is the processing node [32]. In the model, each processing node performs two functions: (1) To sum the input values of the node; (2) To pass this information through an activation function to generate an output. All the processing nodes in a neural network are arranged in layers, and each layer is interconnected to the following layer. There is no interconnection among the nodes in the same layer. In general, a neural network model has an input layer that functions as a distribution structure for the data being used as the input, and this layer is not involved in any type of processing. The input layer is followed by the hidden layer, which consists of one or more processing layers. The final processing layer is the output layer.

The intersections between the nodes have a certain weight. When a value is passed through the input layer, the value is multiplied by the weight and summed to derive the total input n_j to the unit, as shown in Equation (1).

$$n_j = \sum_i w_{ji} o_i \quad (1)$$

where w_{ji} is the weight of the interconnection from the input unit i to another unit j, and o_i is the output of i. The total input calculated using Equation (1) is transformed by the activation function to produce an output o_j of j.

For a neural network, the parameters must be set by the users. These parameters include the number and type of hidden layers, number of nodes in each hidden layer, activation function for the output, weight initialization method, optimization algorithm, learning rate of the optimization algorithm, batch size (i.e., number of training samples used in one iteration), and number of epochs (one epoch is defined as the period in which an entire training dataset passes once through the neural network).

Neural networks are trained by searching for the optimal set of weights for the mapping function from the inputs to the outputs with the given dataset by initializing and updating the weights. In this study, a neural network with the adaptive moment estimation (Adam) optimizer is applied through the Keras Python package [33].

2.1.2. Random Forest Model

Random forests represent an ensemble machine learning technique. A random forest model employs an advanced decision tree analysis method to overcome overfitting issues, which is a drawback of decision tree analyses [34]. In the learning process, a random forest model generates classification trees by selecting subsets of the given dataset and randomly selecting subsets of variables for prediction. The number of trees is set in advance, and the average results for each tree are derived as the final outputs, based on the results generated in each tree. The learning process of random forests using bootstrap sampling consists of the following steps: (i) generate trees and datasets from the training dataset by sampling the bootstrap, (ii) train a basic sorter for the trees, (iii) combine the basic sorter (i.e., tree) into one sorter (i.e., random forest), and (iv) derive the final results of prediction by the majority voting rule. The observed values in the random forest that are not included in the learning process are considered out-of-bag (OOB) values, and they are used in the model validation. OOB values are used to estimate the predicted values and classify variables that cause anomalies. The number of times OOB values are selected in all trees varies for each tree, and the expected values are different for each tree. The probability of correctly predicting the OOB values for each observation in the original category, i.e., category k, can be calculated using Equation (2).

$$\mathrm{Prob}_k(x_i) = \frac{\sum j \in \overline{OOB_i} - I\left[y\left(x_i, T_j\right) = k\right]}{|OOB_i|}, \text{ for } k \quad (2)$$

where i is an indicator that is set as 1 and 0 when the value in the parenthesis is true and false, respectively. $y\left(x_i, T_j\right)$ is the predicted category, and T_j is the jth decision tree among the generated trees T, in the forest. OOB_i represents a group of decision trees that are not used in the learning process and are bagged as an observed variable. If a set of decision trees does not include x_i, the ratio of the number of decision trees predicting x_i to category k is $\mathrm{Prob}_k(x_i)$. For a random forest, the Gini importance is computed and used to indicate the importance of the independent variables. At each node τ within the binary trees t of the random forest, the optimal split is found using the Gini impurity $i(\tau)$, which indicates how well a potential split separates the samples of the two categories in a particular node.

Let $p_k = \frac{n_k}{n}$ represent the fraction of n_k samples from category k = {0, 1} among the total n samples at node τ. The Gini impurity $i(\tau)$ can be calculated using Equation (3).

$$i(\tau) = 1 - p_1^2 - p_0^2. \quad (3)$$

The change in $i(\tau)$, Δi, which can be attributed to the splitting and transmission of the samples to two subnodes τ_l and τ_r (with sample fractions $p_l = \frac{n_l}{n}$ and $p_r = \frac{n_r}{n}$, respectively) based on a threshold t_θ for variable θ, can be calculated using Equation (4).

$$\Delta i(\tau) = i(\tau) - p_l i(\tau_l) - p_r i(\tau_r) \quad (4)$$

In the search for all variables θ available at the node and all possible thresholds t_θ, the pair $\{\theta, t_\theta\}$ leading to the maximum Δi is determined. The change in the Gini impurity resulting from the optimal split, $\Delta i_\theta(\tau, T)$, is recorded and accumulated for all nodes τ in all trees T in the forest for all θ values, as shown in Equation (5).

$$I_G(\theta) = \sum_T \sum_\tau \Delta i_\theta(\tau, T) \quad (5)$$

The Gini importance I_G indicates how often a particular variable θ is selected for a split and the contribution of this value to the classification problem.

This study adopts the scikit-learn package in Python, an open-source programming language software that provides a user-customizable random forest model [35].

2.1.3. Gradient Boosting Decision trees

Gradient boosting decision trees are decision tree models that can prevent overfitting and demonstrate an enhanced prediction accuracy [36]. In gradient boosting decision trees, F(x) is assumed to be an approximation function of the output y based on a set of input variables x. The squared error function is applied as the loss function *L* to estimate the approximation function, as indicated in Equation (6).

$$L(y, F(x)) = [y - F(x)]^2 \quad (6)$$

Assuming that the number of splits is *J* for each regression tree, each tree partitions the input space into *J* disjoint regions $R_{1m}, \ldots, R_{jm}$ and predicts a constant value b_{jm} for region R_{jm}. In this case, each decision tree exhibits the additive form, as indicated in Equation (7).

$$h_m(x) = \sum_{j=1}^{J} b_{jm} I(x \in R_{jm}) \quad (7)$$

$$I = \begin{cases} 1 & \text{if } x \in R_{jm} \\ 0 & \text{otherwise} \end{cases}$$

Using the training data, the gradient boosting model iteratively constructs *M* decision trees $h_1(x), \ldots, h_M(x)$. The updating approximation function $F_m(x)$ and gradient descent step size ρ_m can be defined using Equation (8) and (9).

$$F_m(x) = F_{m-1}(x) + \rho_m \sum_{j=1}^{J} b_{jm} I(x \in R_{jm}) \quad (8)$$

$$\rho_m = \underset{\rho}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \rho \sum_{j=1}^{J} b_{jm} I(x \in R_{jm})) \quad (9)$$

With a separate optimal γ_{jm} for each region R_{jm}, b_{jm} can be discarded. Equation (8) can be expressed as Equation (10):

$$F_m(x) = F_{m-1}(x) + \sum_{j=1}^{J} \gamma_{jm} I(x \in R_{jm}) \quad (10)$$

and the optimal γ_{jm} can be calculated using Equation (11).

$$\gamma_{jm} = \underset{\gamma}{\operatorname{argmin}} \sum_{x \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma) \tag{11}$$

$$= \underset{\gamma}{\operatorname{argmin}} \sum_{x \in R_{jm}} \left(\tilde{y}_i - \gamma\right)^2$$

$$\text{where } \tilde{y}_i = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F_m(x) = F_{m-1}(x)}$$

Gradient boosting decision trees build the model sequentially and update it by minimizing the expected value of the loss function. To avoid overfitting and increase the prediction accuracy, a learning rate strategy is applied. The learning rate is used to scale the contribution of each tree model by introducing a factor ξ, $(0 < \xi \leq 1)$, as indicated in Equation (12).

$$F_m(x) = F_{m-1}(x) + \xi \cdot \sum_{j=1}^{J} \gamma_{jm} I(x \in R_{jm}),\ 0 < \xi \leq 1 \tag{12}$$

In Equation (12), a smaller ξ corresponds to a higher learning rate. Through the learning rate strategy, the overfitting issue can be avoided by reducing the impact of an additional tree. A smaller learning rate leads to a higher reduction in the loss function value. However, a larger number of trees may be added to the model. In this case, another parameter C, which refers to the number of splits, can be used for fitting each decision tree. This parameter represents the depth of variable interaction in a tree. Increasing C can help capture more complex interactions among variables and exploit the strength of gradient boosting decision trees. Depending on the value of the learning rate and C, the optimal number of trees can be identified by examining how well the model fits the test dataset. The performance of gradient boosting decision trees depends on the combination of the learning rate and tree complexity. In this study, the gradient boosting model is applied through the scikit-learn package in Python, which provides a user-customizable model [35].

2.1.4. Synthetic Minority Oversampling Technique (SMOTE)

Despite their importance in analyzing the risk to safety, accident data are usually a minority class owing to their relative unavailability. Therefore, if oversampling to the minority class is not performed, the results from machine learning models may be skewed to the majority class (i.e., non-accident data), leading to the inferior performance of the models. SMOTE is an oversampling method that can be used to overcome this imbalanced data issue [37]. In this method, a minority class is oversampled by creating synthetic samples. SMOTE generates synthetic samples in the following order: 1) Consider the difference between the feature vector (sample) and its nearest neighbor; 2) Multiply this difference by a random number between 0 and 1; 3) Add this value to the feature vector under consideration. Through this process, SMOTE effectively enables the enhanced generalization of the minority class. Because SMOTE can address the imbalanced data issue, several studies on the predicting of uncommon events such as accidents have applied this method to enhance the model performance [38–40]. This study uses time-series datasets, in which the amount of historical data of accidents is small. Consequently, the datasets contain accident data as a minority class, which leads to an imbalanced data issue. To address this problem, SMOTE is applied to the dataset in the model training phase.

2.2. Dataset

Three datasets were aggregated and used in the analysis: (1) An operation dataset for 2017, 2018, and 2020, formulated at container port A in Busan, Korea; (2) A historical

dataset of the accidents that occurred at container port A in 2017, 2018, and 2020, formulated at container port A; (3) Weather observation dataset for Busan for 2017, 2018, and 2020, collected by the Korea Meteorological Administration [41]. Notably, because the data for 2019 are missing in the second dataset, the analysis was performed using the data for 2017, 2018, and 2020.

The first dataset contained information regarding the movements of containers (i.e., loading, discharging, importing, and exporting) and equipment (e.g., yard trucks and container cranes). The second dataset contained information of the accidents, including the time at which an accident occurred and type of accident (i.e., injury, collision, etc.). The weather dataset included data of the temperature, humidity, wind speed, and precipitation.

Tables 2, A1 and A2 present the results of the basic statistical analysis of the datasets. As shown in Table 2, at container port A, 26, 39, and 78 accidents occurred in 2017, 2018, and 2020, respectively. Therefore, the number of accidents has increased from 2017 to 2020. Tables A1 and A2 present the results for the third and first datasets, respectively. These datasets are integrated into a single time-series dataset. Five datasets are generated according to the intervals of 1 h, 3 h, 6 h, 12 h, and 24 h.

Table 2. Basic statistical analysis of the accidents that occurred in container port A.

Year	Month	Number of Accidents	Year	Month	Number of Accidents	Year	Month	Number of Accidents
2017	1	1	2018	1	4	2020	1	7
	2	1		2	1		2	2
	3	1		3	4		3	1
	4	2		4	2		4	7
	5	3		5	4		5	4
	6	2		6	1		6	4
	7	2		7	4		7	7
	8	3		8	4		8	8
	9	2		9	3		9	10
	10	3		10	2		10	7
	11	3		11	5		11	9
	12	3		12	5		12	12
	Total	**26**		**Total**	**39**		**Total**	**78**

Source: Busan Container Port A, Republic of Korea.

3. Results

The data of 2017 and 2018 were used for the model training, and the data of 2020 were used for testing the models. As described in Section 2.1.4, SMOTE was applied to the dataset for training to overcome an imbalanced data issue and enhance the model performances. From the training and testing datasets, hourly weather data (i.e., temperature, precipitation, wind speed, and humidity) from the third dataset and hourly operation data for terminal A (i.e., number of ships in berth, number of containers loaded/unloaded from the ships, number of containers imported/exported from the port, number of trucks entering/exiting the port, and number of container cranes/yard equipment/yard trucks in operation) were used as the input variables. The occurrence of accidents (or lack thereof) in the time intervals was used as the output variable. Moreover, as described in Section 2, the model hyperparameters were the learning rate, max depth, max features, min samples leaf, min samples split, n-estimator, and subsample. Tables 3–5 list the values of hyperparameters for each model that can enhance the model performance.

Table 3. Values of hyperparameters for the deep neural network model.

Hyperparameter	Value or Method
Optimizer	Adam
Loss	Binary cross-entropy
Callbacks	Early stopping
Batch size	128
Epochs	100
Monitor	Val loss

Table 4. Values of hyperparameters for the random forest model.

Hyperparameter	Definition	Values
Max depth	Maximum depth of decision trees	5
Min samples leaf	Minimum samples in a leaf node	18
Min samples split	Minimum samples in a node to be considered for splitting	12
N-estimator	Number of decision trees in the random forest	100

Table 5. Values of hyperparameters for the gradient boosting model.

Hyperparameter	Definition	Values
Learning rate	Impact of each tree on the final outcome	1.0
Max depth	Maximum depth of decision trees	5
Max features	Number of features to consider while searching for the best split	0.25
Min samples leaf	Minimum samples required in a leaf node	18
Min samples split	Minimum samples to be considered for splitting	12
N-estimator	Number of sequential trees to be modeled	100
Subsample	Fraction of observations to be selected for each tree	0.90

To determine the model accuracy in the training process, a 10-fold cross-validation method was applied to the training dataset. Specifically, every model in this study was trained 10 times with 10 datasets for each time interval. The results of the cross validations for the models are presented in Tables 6–8.

As shown in Tables 6–8, the average accuracies of the models in the training phase were higher than 90%, except for the accuracy of the deep neural network with a 24 h interval. This finding shows that the model performance is acceptable in terms of its accuracy. In the testing phase, the model performance was evaluated using the test data. The model performance indicators, specifically, the accuracy, precision, recall, and F1 score, were calculated using Equations (13)–(16), respectively.

Table 6. Results of cross-validation for deep neural network model (accuracy).

	1 h	3 h	6 h	12 h	24 h
1	0.98	0.94	0.94	0.93	0.84
2	0.99	0.94	0.93	0.91	0.72
3	0.98	0.92	0.92	0.86	0.90
4	0.97	0.96	0.91	0.86	0.69
5	0.98	0.94	0.88	0.83	0.69
6	1.00	1.00	0.99	0.97	0.91
7	1.00	1.00	0.94	0.96	0.95
8	1.00	0.99	0.99	0.99	0.99
9	1.00	1.00	0.99	0.98	0.94
10	1.00	1.00	0.98	0.98	0.97
Average	0.99	0.97	0.95	0.93	0.86

Table 7. Results of cross-validation for the random forest model (accuracy).

	1 h	3 h	6 h	12 h	24 h
1	0.96	0.97	0.94	0.90	0.90
2	0.92	0.93	0.93	0.93	0.93
3	0.93	0.95	0.92	0.92	0.93
4	0.92	0.94	0.94	0.98	0.96
5	0.93	0.93	0.95	0.94	0.96
6	0.97	0.96	0.96	0.94	0.93
7	0.91	0.89	0.91	0.92	0.92
8	0.93	0.91	0.93	0.94	0.95
9	0.93	0.94	0.92	0.93	0.95
10	0.93	0.96	0.95	0.94	0.93
Average	0.93	0.94	0.94	0.93	0.93

Table 8. Results of cross-validation for the gradient boosting model (accuracy).

	1 h	3 h	6 h	12 h	24 h
1	0.99	0.97	0.96	0.94	0.86
2	0.91	1.00	1.00	0.98	0.98
3	1.00	1.00	0.99	0.97	0.97
4	0.89	1.00	1.00	0.99	0.96
5	1.00	1.00	0.99	0.97	0.96
6	0.96	1.00	0.99	0.98	0.95
7	1.00	1.00	0.99	0.97	0.89
8	0.96	0.99	0.99	0.95	0.92
9	0.84	1.00	0.98	0.98	0.92
10	1.00	0.99	0.98	0.94	0.83
Average	0.96	0.99	0.99	0.97	0.92

$$\text{Accuracy} = \frac{\text{True Accidents} + \text{True NoAccidents}}{\text{True Accidents} + \text{True NoAccidents} + \text{False Accidents} + \text{False NoAccidents}} \quad (13)$$

$$\text{Precision} = \frac{\text{True Accidents}}{\text{True Accidents} + \text{False Accidents}} \quad (14)$$

$$\text{Recall} = \frac{\text{True Accidents}}{\text{True Accidents} + \text{False NoAccidents}} \quad (15)$$

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (16)$$

Table 9 summarizes the results of implementing the models over the testing dataset. All the models exhibit the highest accuracy for the 1 h interval. As the time intervals increase, the accuracies decrease, although the precision, recall, and F1 score increase. For the deep neural network model, random forest model, and gradient boosting model, the accuracy for the 1 h interval is approximately 98.2%, 90.4%, and 98.3%, respectively, which decreases to approximately 76.5%, 76.8%, and 62.3% for the 24 h interval, respectively. The gradient boosting model exhibits the highest increase in the precision, recall, and F1 score as the time interval increases (i.e., from 1.4%, 1.3%, and 1.4% for the 1 h interval to 20.2%, 39.1%, and 26.6% for the 24 h interval, respectively). The second-largest increase pertains to the deep neural network model (i.e., from 2.3%, 2.6%, and 2.4% for the 1 h interval to 17.7%, 21.9%, and 19.6% for the 24 h interval, respectively), and the lowest increase pertains to the random forest model (i.e., from 1.1%, 11.5%, and 2.1% for the 1 h interval to 18.2%, 9.4%, and 12.4% for the 24 h interval, respectively).

Table 9. Testing results for all models.

Model	Time Intervals	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Deep neural network	1 h	98.2	2.3	2.6	2.4
	3 h	93.6	4.2	6.5	5.1
	6 h	90.9	7.4	6.7	7.0
	12 h	81.4	10.0	11.1	10.5
	24 h	76.5	17.7	21.9	19.6
Random forest	1 h	90.4	1.1	11.5	2.1
	3 h	91.3	2.7	6.5	3.8
	6 h	86.9	4.7	8.0	5.9
	12 h	83.6	11.3	9.7	10.4
	24 h	76.8	18.2	9.4	12.4
Gradient boosting	1 h	98.3	1.4	1.3	1.4
	3 h	90.0	3.9	11.7	5.8
	6 h	85.1	8.7	20.0	12.1
	12 h	76.6	13.3	25.0	17.4
	24 h	62.3	20.2	39.1	26.6

4. Discussion

The results presented in Section 3 show that all the considered models exhibit different performances in predicting accidents in terms of their accuracy, precision, recall, and F1 score under various time intervals. As shown in Table 9, the precision, recall, and F1 score of the models increase as the time intervals increase, whereas the accuracy decreases. In addition to the accuracy, the precision, recall, and F1 score are important measures of the model performance. Higher precision, recall, and F1 score values correspond to a higher model performance. In comparison, a model with an accuracy of at least 85% is considered to have a high performance. Several studies that have applied machine learning methods to predict the accidents indicated that the highest accuracy of the existing models in predicting accidents was 85% [42–44]. Therefore, in terms of the precision, recall, and F1 score, as well as the accuracy, the models using the input data with a time interval of 6 h exhibit a reasonable performance. For the 6 h interval, the deep neural network exhibits an accuracy, precision, recall, and F1 score of 90.9%, 7.4%, 6.7%, and 7.0%, respectively. The corresponding values for the random forest model are 86.9%, 4.7%, 8.0%, and 5.9%. The corresponding values for the gradient boosting model are 85.1%, 8.7%, 20.0%, and 12.1%. The gradient boosting model exhibits the best F1 score of 12.1%, followed by the deep neural network (11.2%). Therefore, the gradient boosting model and deep neural network model are preferable for predicting the accident occurrence at a container port.

In this study, accident data for container port A for three years was used in the analysis. In Korea, the accident data for each container port is collected and managed by the container terminal operator of that port, and there is no legal organization that manages the overall accident data for all container ports nationwide. The accident data for each container port is classified as confidential, and so it is not fully available to the public, except for occupational accidents that are compensated by insurance, whereas the terminal operating data for the port is partially available. Moreover, it is difficult to obtain data because accidents are considered to be a sensitive issue in port operations, and so the terminal operator rarely provides it. This makes it difficult to collect a sufficient amount of accident data at container ports nationwide. As a result, this study used an accident dataset of limited size that was available for analyzing whether accidents occurred or not, rather than focusing on the types of accident that occurred.

5. Conclusions

This study adopts machine learning methods to predict the accidents that can occur in a container port. Time-series datasets with various time intervals are applied, and the model

performance is evaluated based on these intervals. According to the results, as the time interval increases, the accuracy in predicting accidents decreases and the precision, recall, and F1 score increases. In terms of all the indicators, the models using the dataset with a 6 h interval exhibit the highest performance. Under the same time interval, the gradient boosting model and deep neural network model are the best in predicting accidents at the container port. These results demonstrate that machine learning methods can be applied to predict accidents at container ports.

Nevertheless, this study involves certain limitations that must be addressed in future work. The operation dataset of the container port and weather dataset are considered as independent variables affecting the occurrence of accidents. However, other variables, such as the accident type, cause, and time of incidence, can also directly affect the occurrence of accidents. Accident data for container ports, especially in Korea, are of significance, and can affect the port operation. However, although the annually aggregated statistics of accidents are available [13], it is difficult to collect an adequately large dataset including the raw data from the port. The accident dataset from container port A contains accident types, including vehicle collision, container damage, injury, and death. However, considering the total number of accidents (i.e., 26 accidents in 2017, 39 accidents in 2018, and 78 accidents in 2020), it is difficult to specify the accidents by type. Therefore, in this study, the number of accidents is considered as the output variable. In future work, by categorizing adequate accident data according to the accident type (i.e., injury, collision, and container damage, among other events), it may be possible to predict the accident type, analyze the factors affecting the accidents, and assess the risks in a container port. Moreover, accidents at a container port happen during work in any environment. Operational data for a container terminal is stored as an hourly dataset, and shows the changes in working status at the port well. However, these data and accident data together cannot provide enough information about the situation in which an accident occurred, because the accident data contains reasons for the occurrence (i.e., carelessness during working, rough driving, and so on) and the hourly aggregated operations dataset cannot describe the dynamic situation in which the accident happened. Therefore, to analyze accidents with their causes, the dynamic operating situation at the container port when the accident happened should be considered, rather than the hourly aggregated data. In a future study, with disaggregated operational data and accident data for a container port, it may be possible to not only predict accident occurrence but also to analyze accident risk.

Author Contributions: Conceptualization, J.H.K. and G.L.; methodology, J.H.K. and G.L.; software, J.K.; validation, J.H.K. and J.K.; formal analysis, J.H.K.; writing—original draft preparation, J.H.K., J.K., G.L. and J.P.; writing—review and editing, J.H.K., G.L. and J.P.; supervision, G.L.; project administration, J.P.; funding acquisition, G.L. and J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was a part of the project titled 'Smart Port IoT convergence and operation technology development [20190399]', funded by the Ministry of Oceans and Fisheries, Korea.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Restrictions apply to the availability of these data. Data was obtained from container port A in Korea and are available with the permission of container port A.

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

Appendix A. Basic Statistical Analysis of the Dataset Associated with Container Port A in Korea

Table A1. Basic statistical analysis of the weather dataset pertaining to Busan, Korea.

Year	Month	Average Temperature (°C)	Average Precipitation (mm)	Average Wind Speed (m/s)	Average Humidity (%)
2017	1	4.1	0.0	3.3	46.1
	2	5.5	0.0	3.5	44.7
	3	8.9	0.1	3.1	52.8
	4	14.9	0.1	3.8	63.4
	5	19.0	0.1	3.3	63.1
	6	21.2	0.1	2.9	70.3
	7	26.0	0.2	3.5	82.2
	8	27.0	0.1	2.9	73.6
	9	22.6	0.5	2.8	68.9
	10	18.1	0.2	2.9	67.6
	11	11.3	0.0	2.9	49.4
	12	3.5	0.0	3.0	40.0
2018	1	1.9	0.1	3.1	44.7
	2	3.8	0.1	3.4	41.8
	3	9.9	0.3	3.9	63.1
	4	14.5	0.2	4.1	62.4
	5	17.9	0.2	3.9	72.5
	6	21.5	0.4	3.1	75.2
	7	26.7	0.2	3.0	77.8
	8	27.9	0.2	3.6	74.8
	9	22.0	0.4	3.0	76.2
	10	16.5	0.2	2.0	61.8
	11	12.4	0.1	2.3	57.1
	12	5.7	0.1	2.9	48.5
2020	1	6.4	0.2	3.0	56.3
	2	7.1	0.1	3.0	55.2
	3	10.4	0.1	3.5	57.0
	4	12.6	0.1	3.7	53.0
	5	17.9	0.1	3.1	72.3
	6	22.4	0.4	3.0	73.5
	7	22.1	1.1	2.7	87.2
	8	27.0	0.5	4.0	82.4
	9	22.0	0.3	3.6	73.9
	10	17.3	0.0	2.7	58.0
	11	12.4	0.1	2.8	52.0
	12	4.4	0.0	2.9	42.7

Source: Average values are calculated based on data collected from the Korea Meteorological Administration Weather Data Service (data.kma.go.kr).

Table A2. Basic statistical analysis of the operation dataset pertaining to container port A.

Year	Month	Average Number of Ships in Berth	Average Number of Containers Unloaded	Average Number of Containers Loaded	Average Number of Containers Imported	Average Number of Containers Exported	Average Number of Trucks Entering	Average Number of Trucks Exiting	Average Number of Operation Container Cranes	Average Number of Operating Yard Equipment	Average Number of Operating Yard Trucks
2017	1	5	83	80	76	79	66	68	7	32	39
	2	5	84	82	81	87	72	76	7	34	38
	3	5	91	83	87	92	76	80	8	34	40
	4	5	94	89	88	91	76	78	8	34	42
	5	5	91	88	85	88	74	76	8	34	41
	6	5	94	89	86	92	75	79	8	34	42
	7	5	92	90	83	87	72	75	8	34	43
	8	5	88	84	83	85	73	74	8	33	40
	9	5	91	90	89	90	79	79	8	33	42
	10	5	87	81	76	81	66	70	8	33	41
	11	5	98	100	96	96	84	84	10	35	50
	12	5	102	93	90	91	78	80	9	36	51
2018	1	5	90	94	86	91	76	79	9	35	48
	2	5	94	90	86	87	75	76	9	35	48
	3	6	92	91	88	89	78	79	9	36	48
	4	5	102	100	90	91	79	81	10	36	50
	5	5	100	96	89	93	78	81	10	37	50
	6	6	103	100	92	95	81	83	11	37	53
	7	6	100	95	93	97	83	86	10	36	52
	8	6	93	89	86	92	77	82	10	36	47
	9	6	95	94	86	89	76	78	10	36	50
	10	6	99	94	94	100	83	88	10	37	50
	11	5	102	96	96	101	85	88	10	39	51
	12	5	99	89	89	94	78	83	9	37	49
2020	1	5	80	82	83	88	72	75	8	36	46
	2	5	85	85	80	83	70	72	8	37	48
	3	5	92	89	84	90	74	79	9	36	51
	4	5	92	85	84	87	74	76	9	38	48
	5	5	87	83	81	84	71	73	8	37	46
	6	5	91	82	81	90	72	78	8	38	46
	7	5	89	84	78	88	69	77	9	38	47
	8	5	88	85	74	81	66	71	9	36	48
	9	6	89	86	90	88	79	78	10	36	51
	10	5	94	87	85	93	76	82	10	38	51
	11	6	100	94	89	94	79	83	10	39	54
	12	5	97	88	93	95	82	84	10	39	53

Source: Busan Container Port A, Republic of Korea.

References

1. Abdelfattah, M.; Elsayeh, M.; Abdelkader, S. A proposed port security risk assessment approach, with application to a hypothetical port. *Aust. J. Marit. Ocean. Aff.* **2021**, 1–18. [CrossRef]
2. Andritsos, F.; Mosconi, M. Port Security in EU: A systemic approach. In Proceedings of the 2010 International WaterSide Security Conference, Carrara, Italy, 3–5 November 2010; p. 11863527. [CrossRef]
3. Orosz, M.D.; Southwell, C.; Barrett, A.; Chen, J.; Ioannou, P.; Abadi, A.; Maya, I. PortSec: A port security risk analysis and resource allocation system. In Proceedings of the 2010 IEEE International Conference on Technologies for Homeland Security, Boston, MA, USA, 8–10 November 2010; p. 11679258. [CrossRef]
4. Budiyanto, M.A.; Fernanda, H. Risk Assessment of Work Accident in Container Terminals Using the Fault Tree Analysis Method. *J. Mar. Sci. Eng.* **2020**, *8*, 466. [CrossRef]

5. Chlomoudis, C.I.; Pallis, P.L.; Tzannatos, E.S. A Risk Assessment Methodology in Container Terminals: The Case Study of the Port Container Terminal of Thessalonica, Greece. *J. Traffic Transp. Eng.* **2016**, *4*, 251–258.
6. Hamka, M.A. Safety Risks Assessment on Container Terminal Using Hazard Identification and Risk Assessment and Fault Tree Analysis Methods. *Procedia Eng.* **2017**, *194*, 307–314. [CrossRef]
7. Rosoff, H.; von Winterfeldt, D. A risk and economic analysis of dirty bomb attacks on the ports of Los Angeles and Long Beach. *Risk Anal.* **2007**, *27*, 533–546. [CrossRef]
8. Concho, A.L.; Ramirez-Marquez, J.E. An evolutionary algorithm for port-of-entry security optimization considering sensor thresholds. *Reliab. Eng. Syst. Saf.* **2010**, *95*, 255–266. [CrossRef]
9. Hashemi, R.R.; Le Blanc, L.A.; Rucks, C.T.; Shearry, A. A Neural Network for Transportation Safety Modeling. *Expert Syst. Appl.* **1995**, *9*, 247–256. [CrossRef]
10. Doong, D.; Chen, S.; Chen, Y.; Tsai, C. Operational Probabilistic Forecasting of Coastal Freak Waves by Using an Artificial Neural Network. *J. Mar. Sci. Eng.* **2020**, *8*, 165. [CrossRef]
11. Wang, Y.; Wang, L.; Jiang, J.; Wang, J.; Yang, Z. Modelling ship collision risk based on the statistical analysis of historical data: A case study in Hong Kong waters. *Ocean Eng.* **2020**, *197*, 106869. [CrossRef]
12. Tang, L.; Tang, Y.; Zhang, K. Prediction of Grades of Ship Collision Accidents Based on Random Forests and Bayesian Networks. In Proceedings of the the 5th International Conference on Transportation Information and Safety, Liverpool, UK, 14–17 July 2019; pp. 1377–1381.
13. Korea Port Logistics Association. *Statistics and Cases of Cargo Handling and Accident Situation*; Korea Port Logistics Association: Seoul, Korea, 2018.
14. Dong, C.; Shao, C.; Li, J.; Xiong, Z. An Improved Deep Learning Model for Traffic Crash Prediction. *J. Adv. Transp.* **2018**, *2018*, 1–13. [CrossRef]
15. Yu, B.; Wang, Y.T.; Yao, J.B.; Wang, J.Y. A comparison of the performance of ANN and SVM for the prediction of traffic accident duration. *Neural Netw. World* **2016**, *26*, 271–287. [CrossRef]
16. Dogru, N.; Subasi, A. Traffic Accident Detection Using Random Forest Classifier. In Proceedings of the 2018 15th Learning and Technology Conference, Jeddah, Saudi Arabia, 25–26 February 2018; pp. 40–45.
17. Lee, S.; Kim, J.H.; Park, J.; Oh, C.; Lee, G. Deep-Learning-Based Prediction of High-Risk Taxi Drivers Using Wellness Data. *Int. J. Environ. Res. Public Health.* **2020**, *17*, 9505. [CrossRef] [PubMed]
18. Parsa, A.B.; Movahedi, A.; Taghipour, H.; Derrible, S.; Mohammadian, A. Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis. *Accid. Anal. Prev.* **2020**, *136*, 105405. [CrossRef]
19. Zhang, Z.; Yang, W.; Wushour, S. Traffic Accident Prediction Based on LSTM-GBRT Model. *J. Contr. Sci. Eng.* **2020**, *2020*, 1–10. [CrossRef]
20. Lu, P.; Zheng, Z.; Ren, Y.; Zhou, X.; Keramati, A.; Tolliver, D.; Huang, Y. A Gradient Boosting Crash Prediction Approach for Highway-Rail Grade Crossing Crash Analysis. *J. Adv. Transp.* **2020**, *2020*, 1–10. [CrossRef]
21. Cuenca, L.G.; Puertas, E.; Aliane, N.; Andres, J.F. Traffic Accidents Classification and Injury Severity Prediction. In Proceedings of the 2018 3rd International Conference on Intelligent Transportation Engineering, Singapore, 3–5 September 2018; pp. 52–57.
22. AlMamlook, R.E.; Kwayu, K.M.; Alkasisbeh, M.R.; Frefer, A.A. Comparison of Machine Learning Algorithms for Predicting Traffic Accident Severity. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; pp. 272–276.
23. Ung, S.T.; Williams, V.; Bonsall, S.; Wang, J. Test case based risk predictions using artificial neural network. *J. Safety Res.* **2006**, *37*, 245–260. [CrossRef] [PubMed]
24. Park, Y. An analysis of Traffic Accidents in Terms of Human Factors: The Case of Bus-Drivers. *Korean J. Ind. Organ. Psychol.* **2000**, *13*, 75–90.
25. Choi, J.; Kim, S.; Hwang, K.; Baik, S. Severity Analysis of the Pedestrian Crash Patterns Based on the Ordered Logit Model. *Int. J. Highw. Eng.* **2009**, *11*, 153–164.
26. Choi, J.; Kim, S.; Kim, S.; Yeon, J.; Kim, C. A Study on Pedestrian Crashes Contributing Factors During Jaywalking—Focused on the case of Seoul. *J. Korea Inst. Intell. Transp. Syst.* **2015**, *14*, 38–49.
27. Lee, H.; Kum, K.; Son, S. A Study on the factor analysis by grade for highway traffic accident. *Int. J. Highw. Eng.* **2011**, *13*, 157–165. [CrossRef]
28. Kim, B.; Park, S.; Gong, J.; Yeo, G. A Study on the Safety Factor Analysis of Bulk Cargo Handling Using Fuzzy-AHP: Focused on steel cargo. *J. Digit. Converg.* **2018**, *16*, 179–188.
29. Kim, D. A risk analysis of accidents for improving port logistics productivity—A case study of a container operator of a port. *Product. Rev.* **2016**, *30*, 53–79.
30. Yeun, D.; Choi, Y.; Kim, S. An Assessment & Analysis of Risk Based on Accident Category for Container Terminals. *J. Shipp. Logist.* **2014**, *30*, 843–858.
31. Cha, S.; Noh, C. The Accidents Analysis for Safety Training in the Container Terminal. *J. Korean Navig. Port Res.* **2016**, *40*, 197–205. [CrossRef]
32. Singh, G.; Pal, M.; Yadav, Y.; Singla, T. Deep neural network-based predictive modeling of road accidents. *Neural Comput. Appl.* **2020**, *32*, 12417–12426. [CrossRef]
33. Keras: The Python Deep Learning API. Available online: http://keras.io (accessed on 3 March 2021).

34. Breiman, L. Random Forests. *Mach Learn.* **2001**, *45*, 5–32. [CrossRef]
35. Scikit-learn: Machine Learning in Python. Available online: http://scikit-learn.org (accessed on 3 March 2021).
36. Ma, X.; Ding, C.; Luan, S.; Wang, Y.; Wang, Y. Prioritizing Influential Factors for Freeway Incident Clearance Time Prediction Using the Gradient Boosting Decision Trees Method. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2303–2310. [CrossRef]
37. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell.* **2002**, *16*, 321–357. [CrossRef]
38. Parsa, A.B.; Taghipour, H.; Derrible, S.; Mohammadian, A. Real-time accident detection: Coping with imbalanced data. *Accid. Anal. Prev.* **2019**, *129*, 202–210. [CrossRef]
39. Li, P.; Abdel-Aty, M.; Yuan, J. Real-time crash risk prediction on arterials based on LSTM-CNN. *Accid. Anal. Prev.* **2020**, *135*. [CrossRef] [PubMed]
40. Bobbili, N.P.; Cretu, A. Adaptive Weighting with SMOTE for Learning from Imbalanced Datasets: A Case Study for Traffic Offense Prediction. In Proceedings of the 2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), Ottawa, ON, Canada, 12–13 June 2018. [CrossRef]
41. Open MET Data Portal. Korea Meteorological Administration (KMA) Weather Data Service. Available online: data.kma.go.kr (accessed on 15 January 2021).
42. Reichenbach, A.; Navaro-Barrientos, J.E. A model for traffic incident prediction using emergency braking data. *arXiv* **2021**, arXiv:2102.06674.
43. Varghese, V.; Chikaraishi, M.; Urata, J. Deep Learning in Transport Studies: A Meta-analysis on the Prediction Accuracy. *J. Big Data Anal. Transp.* **2020**, *2*, 199–220. [CrossRef]
44. Zhang, Z.; He, Q.; Gao, J.; Ni, M. A deep learning approach for detecting traffic accidents from social media data. *Transp. Res. Part C Emerg. Technol.* **2018**, *86*, 580–596. [CrossRef]