Water Research 196 (2021) 117001

Contents lists available at ScienceDirect

Water Research

journal homepage: www.elsevier.com/locate/watres

Prediction of antibiotic-resistance genes occurrence at a recreational beach with deep learning models

Jiyi Jang[a,1], Ather Abbas[a,1], Minjeong Kim[b], Jingyeong Shin[c], Young Mo Kim[c], Kyung Hwa Cho[a,*]

[a] Department of Urban and Environmental Engineering, Ulsan National Institute of Science and Technology (UNIST), 50, UNIST-gil, Eonyang-eup, Ulju-gun, Ulsan, 44919 South Korea

[b] Division of Radioactive Waste Disposal Research, Korea Atomic Energy Research Institute (KAERI), 989-111, Daedeok-daero, Yuseong-gu, Daejeon, 34057, South Korea

[c] Department of Civil and Environmental Engineering, Hanyang University, 222, Wangsimni-ro, Seongdong-gu, Seoul, 04763, South Korea

ARTICLE INFO

Article history:
Received 6 November 2020
Revised 27 February 2021
Accepted 1 March 2021
Available online 3 March 2021

Keywords:
Antibiotic-resistance genes (ARGs)
prediction model
deep neural network
long short-term memory (LSTM)
input attention
recreational beach

ABSTRACT

Antibiotic resistance genes (ARGs) have been reported to threaten the public health of beachgoers worldwide. Although ARG monitoring and beach guidelines are necessary, substantial efforts are required for ARG sampling and analysis. Accordingly, in this study, we predicted ARGs occurrence that are primarily found on the coast after rainfall using a conventional long short-term memory (LSTM), LSTM-convolutional neural network (CNN) hybrid model, and input attention (IA)-LSTM. To develop the models, 10 categories of environmental data collected at 30-min intervals and concentration data of 4 types of major ARGs (i.e., *aac(6′-Ib-cr)*, *bla*$_{\mathrm{TEM}}$, *sul*1, and *tet*X) obtained at the Gwangalli Beach in South Korea, between 2018 and 2019 were used. When individually predicting ARGs occurrence, the conventional LSTM and IA-LSTM exhibited poor R^2 values during training and testing. In contrast, the LSTM-CNN exhibited a 2–6-times improvement in accuracy over those of the conventional LSTM and IA-LSTM. However, when predicting all ARGs occurrence simultaneously, the IA-LSTM model exhibited a superior performance overall compared to that of LSTM-CNN. Additionally, the influence of environmental variables on prediction was investigated using the IA-LSTM model, and the ranges of input variables that affect each ARG were identified. Consequently, this study demonstrated the possibility of predicting the occurrence and distribution of major ARGs at the beach based on various environmental variables, and the results are expected to contribute to management of ARG occurrence at a recreational beach.

1. Introduction

The emergence of antibiotic resistance genes (ARGs) as aquatic environment contaminants (Pruden et al., 2006) has become a significant global threat to human public health. ARGs are released from landfills or sludge through runoff, and they can flow into recreational areas along the coast (Zhang et al., 2016b). Specifically, recreational beaches are susceptible to ARG contamination through various sources such as wastewater treatment plants (Proia et al., 2018), animal feed mills (Fang et al., 2018), and storm runoff (Joy et al., 2013). Hence, in a previous study, surfers were found to be 4.2 times more likely to be exposed to ARGs than non-surfers in the swimming areas in England (Leonard et al., 2018). The rainfall effect is known to naturally dilute ARGs; however, ARGs are not sufficiently managed in marine environments because of current global wastewater management practices (Bedri et al., 2015; Law and Tang, 2016).

Monitoring of ARGs at recreational beaches is required for beach user safety. However, ARG monitoring has the following limitations. Conventional analysis methods are time-consuming; it takes 5.2 d on average to verify incubation results (McAdam et al., 2012). Current molecular biological techniques such as quantitative polymerase chain reaction (qPCR) have been used to identify and quantify certain ARGs (de Castro et al., 2014; Schmieder and Edwards, 2012). Although qPCR is simpler and faster compared to conventional techniques such as the culture method or traditional PCR (Kralik and Ricchi, 2017; Smith and Osborn, 2009), regular monitoring is restricted due to the high cost of qPCR analysis (Sakthivel et al., 2012). Although multiplex PCR has been developed to save time and effort by reacting multiple single PCRs si-

[*] Corresponding author.
E-mail address: khcho@unist.ac.kr (K.H. Cho).

[1] These authors contributed equally to this study.

https://doi.org/10.1016/j.watres.2021.117001
0043-1354/© 2021 Elsevier Ltd. All rights reserved.

multaneously, it is less accurate because it responds to nonspecific amplification products (Jansen et al., 2011; Sakthivel et al., 2012). Therefore, for preemptive responses within a limited timeframe for ARG occurrences at beaches, prediction through modeling can be more efficient than through monitoring.

Long short-term memory (LSTM), a type of recurrent neural network (RNN), has been widely used as an efficient tool to simulate and predict water quality due to an ability to extract features from time-series data (lin Hsu et al., 1997). For example, Barzegar et al. (2020) recently utilized LSTM and LSTM hybrid models to predict water quality variables in a lake. An advantage of LSTM is that it can use memory to learn features over time. Accordingly, it is considered a suitable neural network (NN) for predicting pollutant distributions and water quality over time (Wang et al., 2019; Wang et al., 2017). On the other hand, hydrological models suffer from higher uncertainties because of their inability to simulate complex mechanistic relationships among environmental variables (Abimbola et al., 2020). Although deep learning models are black box models, they can improve performance by training from observation data (Andrychowicz et al., 2016) and simulate nonlinear phenomena occurring in the environment. In particular, deep learning models have been widely used to enhance the prediction performance of hydrological models (Parmar et al., 2017; Sumi et al, 2012). Therefore, hypothetically, it is expected that the accuracy of deep learning will be higher than that of hydrological models to predict ARGs at a recreational beach of Korea affected by rain in a short period.

Based on the collected literature, however, the potential of LSTM has yet to be utilized to estimate ARGs released into the environment. We previously observed the occurrence of ARGs at a combined sewer overflow (CSO) site in Gwangalli Beach over time, which varied in relation to rainfall and tides (Jang et al., 2021). Recreational activities at the beach are concentrated in the summer and the beach is annually affected by monsoon weather. Therefore, ARG prediction is significant for preserving the health of beachgoers, and the application of LSTM would be promising in predicting the occurrence of ARGs over time. Therefore, in this study, we propose an approach based on NN techniques to predict ARGs occurrence quickly and accurately for managing and monitoring their occurrence in beach environments. This study compared conventional LSTM, LSTM-convolutional NN (CNN), and input attention (IA)-LSTM models (Fig. 1) with the following objectives: 1) to propose applicable models for predicting four major ARGs (i.e., *aac(6′-Ib-cr)*, *bla*$_{\mathrm{TEM}}$, *sul*1, and *tet*X) at a recreational beach, 2) to compare model accuracies when predicting single ARG individually and multiple ARGs simultaneously, and 3) to determine critical environmental features for predicting ARG occurrences.

2. Material and methods

2.1. Sampling location and period

Gwangalli Beach, a popular beach in South Korea, was selected as the study area. The eastern coast of Gwangalli Beach is adjacent to the Suyeong River estuary, which consists of urban areas and has a wastewater treatment plant and several sewer outlets along the river (Fig. 2). The total area of the beach is 82 000 m^2; the beach is 1.4 km in length and 25–110 m in width along the coastline (Choi et al., 2016). Seawater samplings were conducted at a CSO outfall on the right side of the beach (Fig. 2). Surface seawater samples (within 1 m in depth) were automatically collected using an ISCO 6712 portable water sampler (Teledyne ISCO Inc., USA) during the dry season and rainfall events from June 2018 to September 2019 (Table 1). In total, 218 samples were collected, and they were kept in the dark at 4°C in containers until they were pretreated for ARG analysis.

Table 1
Sampling intervals and amount and maximum intensity of rainfall during sampling periods (mean ± standard deviation).

Event	Date & Time	ARGs Sampling interval (No. of ARG data)	Rainfall (No. of rainfall data)	Maximum rainfall intensity
(a) Dry season sampling				
Dry event I	June 19, 2018 (12:00) – June 22, 2018 (00:00)	3 hrs (5)	0 mm (145)	-
Dry event II	Aug 20, 2019 (00:00) – Aug 21, 2019 (19:00)	1~3 hrs (22)	0 mm (87)	-
Dry event III	Sep 4, 2019 (15:30) – Sep 8, 2019 (00:00)	1~15 hrs (33)	0 mm (186)	-
(b) Wet season sampling				
Rainfall event I	June 26, 2018 (00:00) – July 1, 2018 (00:00)	3~5 hrs (14)	138.5 mm (241)	32.5 mm/h
Rainfall event II	May 17, 2019 (00:00) – May 21, 2019 (00:00)	1~31 hrs (42)	46.6 mm (241)	5.3 mm/h
Rainfall event III	Aug 21, 2019 (19:30) – Aug 24, 2019 (00:00)	1 hr (15)	4.0 mm (106)	2.5 mm/h
Rainfall event IV	Aug 25, 2019 (00:00) –Aug 31, 2019 (00:00)	1~22 hrs (40)	5.0 mm (289)	1.0 mm/h
Rainfall event V	Sep 1, 2019 (00:00) – Sep 4, 2019 (15:00)	1~12 hrs (47)	68.0 mm (151)	17.0 mm/h
Total number of data		218	1446	-

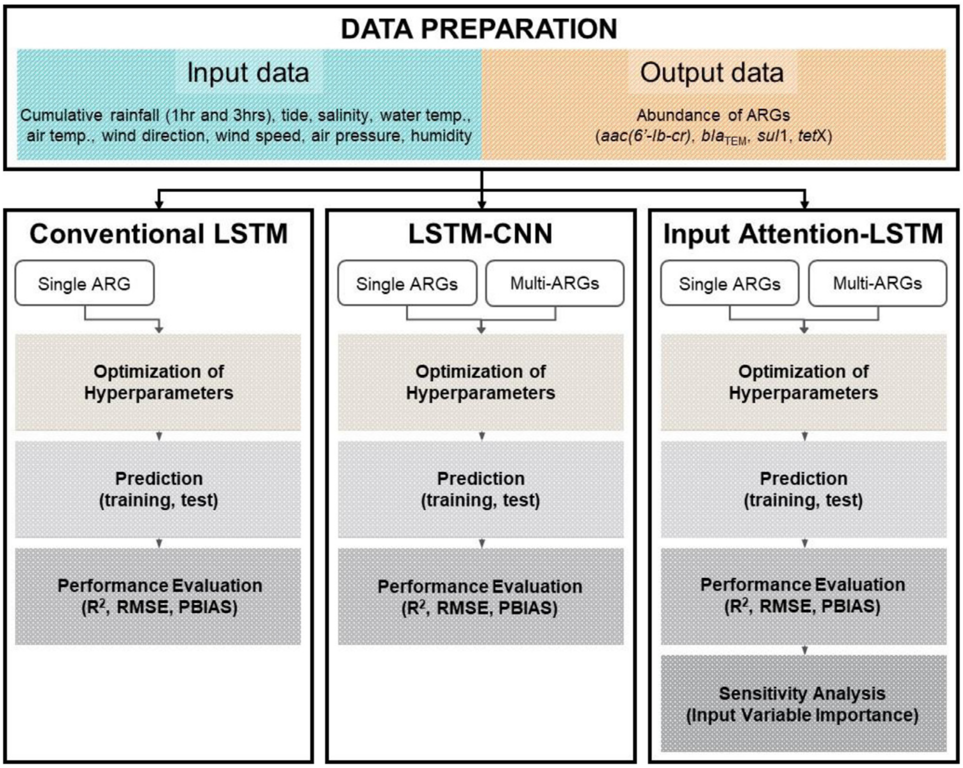

DATA PREPARATION
Input data
Cumulative rainfall (1hr and 3hrs), tide, salinity, water temp., air temp., wind direction, wind speed, air pressure, humidity
Output data
Abundance of ARGs
(aac(6'-Ib-cr), bla_{TEM}, sul1, tetX)
Conventional LSTM
Single ARG
LSTM-CNN
Single ARGs
Multi-ARGs
Input Attention-LSTM
Single ARGs
Multi-ARGs
Optimization of Hyperparameters
Prediction (training, test)
Performance Evaluation (R^2, RMSE, PBIAS)
Sensitivity Analysis (Input Variable Importance)

Fig. 1. Summary diagram for predicting ARGs at a recreational beach.

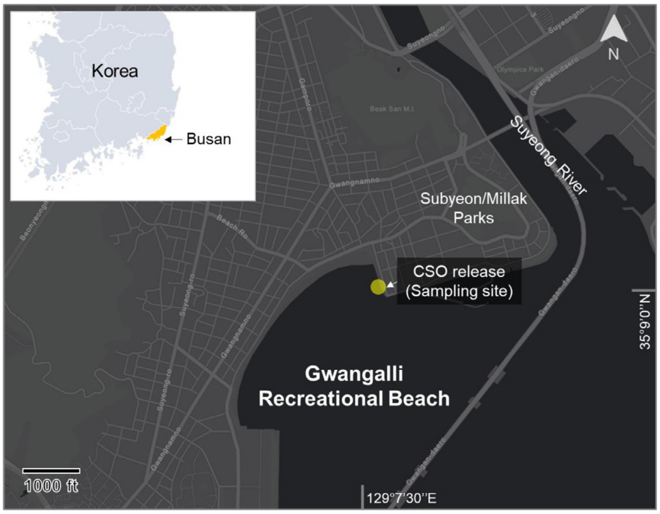


Fig. 2. Map of study area and sampling site in Gwangalli Beach in Busan, South Korea. CSO, combined sewer overflows.

2.2. Data acquisition

2.2.1. Environmental variables

A total of ten categories of environmental data including seven categories of meteorological variables (i.e., cumulative rainfall for 1 h and 3 h, wind direction, wind speed, air pressure, air temperature, and relative humidity) and three categories of aquatic variables (i.e., water temperature, tides, and salinity) were used to develop the models. A total of 1,446 data points were collected at 30-min intervals for model training, and the number of data points for each event and their trends are summarized in Table 1 and Fig. S1, respectively. Rainfall data for all sampling events were obtained from the Korea Meteorological Administration (KMA) (KMA, 2015). Because antecedent rainfall has previously been found to be an important environmental factor for bacteria occurrence in Gwangalli Beach (Park et al., 2018), cumulative rainfall data for 1 hour and 3 hours were used as input variables for the model development. The total rainfall for each wet season and maximum rainfall intensity in an hour are presented (Table 1). Other meteorological data were also collected from the KMA while aquatic environmental data were obtained from the South Korea Hydrographic and Oceanographic Agency (KHOA, 2020).

2.2.2. Enumeration of ARGs

Seawater samples (50–500 mL) were filtered through 0.45-μm membrane filters (Nylon, Whatman) in a vacuum to determine relative abundance of ARG. The membranes were stored in 2-mL tubes, and the tubes were kept at −20°C until analysis. Sample deoxyribonucleic acid (DNA) was extracted from the membrane filters using an Exgene^{TM} soil kit (GeneAll Biotechnology, South Korea) according to manufacturer instructions. The ARGs were quantified using a CFX96 Touch^{TM} Real-Time PCR Detection System (Bio-Rad, USA) and 20 μL of a master mix consisting of the following materials: 10 μL of KAPA SYBR® FAST qPCR kits (KAPA Biosystems, USA), 8.2 μL of PCR-grade water, 0.8 μL of primer (0.4 μL of forward + 0.4 μL of reverse), and 1 μL of DNA template. Each master mix containing the target DNA was amplified for 39 cycles using the following three steps: 3 min at 95°C for denaturation, 3 min at the optimal temperature of each target gene for annealing, and 10 s at 72°C for elongation (Jang et al., 2017; Shin et al., 2019). The target ARGs in this study were *aac(6′-Ib-cr)*, bla_{TEM}, *sul*1, and *tet*X, which were the most frequently found on the beach in a previous study (Jang et al., 2021). For each target gene, the optimal annealing temperatures were 66.0°C for *aac(6′-Ib-cr)*, 44.0°C for bla_{TEM}, 56.0°C for *sul*1, and 61.0°C for *tet*X (Jang et al., 2017). The ARG abundances were quantified based on standard curves generated using a 10-fold serial dilution of the plasmid DNA. Four

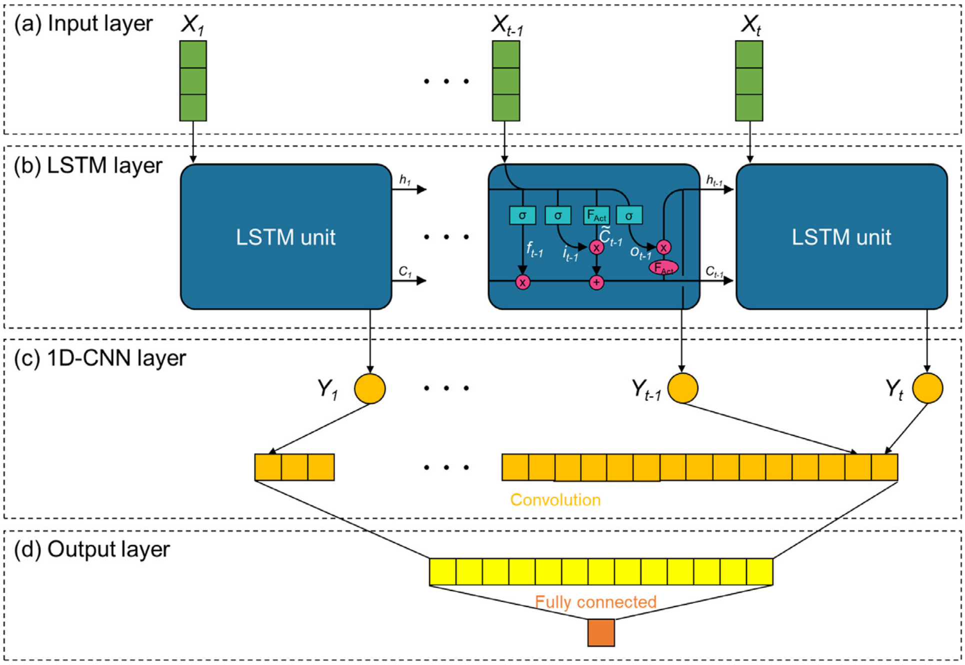


Fig. 3. Structure of LSTM-CNN hybrid model for time step 1 to *t*, which is suggested in this study. (a) and (b) indicate the input and LSTM layers of the conventional LSTM, respectively. On the LSTM layer, X_t and Y_t are the input and output at time *t*, respectively. Cell state (C_t) records past information through three gates (*f*, *i*, and *o*), sigmoid (σ), and the activate function (F_{Act}). (c) LSTM output is loaded into the 1D CNN and convoluted based on the optimized filter size and activation function type. (d) ARG prediction values are derived from the fully connected layer.

ARGs were detected in 218 samples, and their temporal trends are displayed in Fig. S2.

2.3. Data-driven modeling

Three models were used to predict single and multi-ARGs as described in Fig. 1. Each model was developed with the meteorological variables (i.e., cumulative rainfall for 1 h and 3 h, wind direction, wind speed, air pressure, air temperature, and relative humidity) and aquatic variables (i.e., water temperature, tides, and salinity) as input data and four ARG abundances as output data. Moreover, the hyperparameters such as the batch size, lookback, learning rate, and weight matrix sizes in LSTM and CNN, and activation functions for LSTM and CNN for each model were determined by a Bayesian optimization algorithm as described in the supplementary information.

2.3.1. Conventional LSTM

LSTM was developed by Hochreiter and Schmidhuber (1997) to process long sequential data and extract its temporal features. In LSTM, the flow of information is controlled by the “gate” mechanism (Fig. 3a and b). These gates control the incoming and outgoing information in an LSTM unit. LSTM employs three gate types that are named based on the function they perform (i.e., the output gate (Γ_o), forget gate (Γ_f), and input gate (Γ_i)). In addition to gates, LSTM also has “memory”, which is responsible for the flow of information across time. The gates interact with the memory to either remove or add information to the memory. The memory is also called the “state.” There are two types of states within an LSTM unit: the cell state (c_t) and hidden state (h_t). At each time step, an LSTM unit generates two outputs that are representative of the cell and hidden states, which are given in the Fig. 3b and equations below (Hochreiter and Schmidhuber, 1997).

$$\tilde{C}_t = tanh(W_c[h_{t-1}, X_t] + \ b_c), \tag{1}$$

$$\Gamma_f = \sigma\left(W_f[c_{t-1}, X_t] + b_f\right), \tag{2}$$

$$\Gamma_o = \sigma(W_o[c_{t-1}, X_t] + b_o), \tag{3}$$

$$\Gamma_i = \sigma(W_i[c_{t-1}, X_t] + b_i), \tag{4}$$

$$c_t = \Gamma_i * \tilde{C}_t \ + \ \Gamma_f * \ c_{t-1}, \tag{5}$$

$$h_t = \Gamma_o * tanhc_t \tag{6}$$

At time step *t*, an LSTM model generates a candidate cell ($\tilde{C}_t$), which is used to update the cell state (c_t) Eqs. (1) and (5). A vector of the output of the previous cell (h_{t-1}) and the current input (X_t) are multiplied by the weight of the candidate cell (W_c), and generates $\tilde{C}_t$ with bias (b_c) using the hyperbolic tangent (*tanh*) function Eq. (1). In Eqs. (2)–(4), *W* and *b* are the weights and biases of the gates, respectively, which are learnable parameters. A vector of the previous cell state (c_{t-1}) and X_t is multiplied by the weight of each gate (i.e., W_f, W_o, and W_i), and the gates are calculated using the sigmoid function (σ). The outputs are produced as c_t and hidden states (h_t) using the candidate cell and the gates Eqs. (5) and (6). h_t is then used in a fully connected network to obtain the final model predictions.

In this study, we used a “many-to-many” scenario, which calculate the outputs at each time step, so that the number of outputs represent the number of input. This differs from “many-to-one” scenario which produces an output only at the last time step. Furthermore, the number of information (i.e., cell state and hidden state) in the past is called the “lookback”, and in the present study, lookback 1 represents the information of 30 minutes ago.

2.3.2. LSTM-CNN hybrid model

We applied a one-dimensional (1D) CNN (Fukushima, 1979) to the output sequence of the LSTM model to enhance the prediction performance by further extracting features from the LSTM output

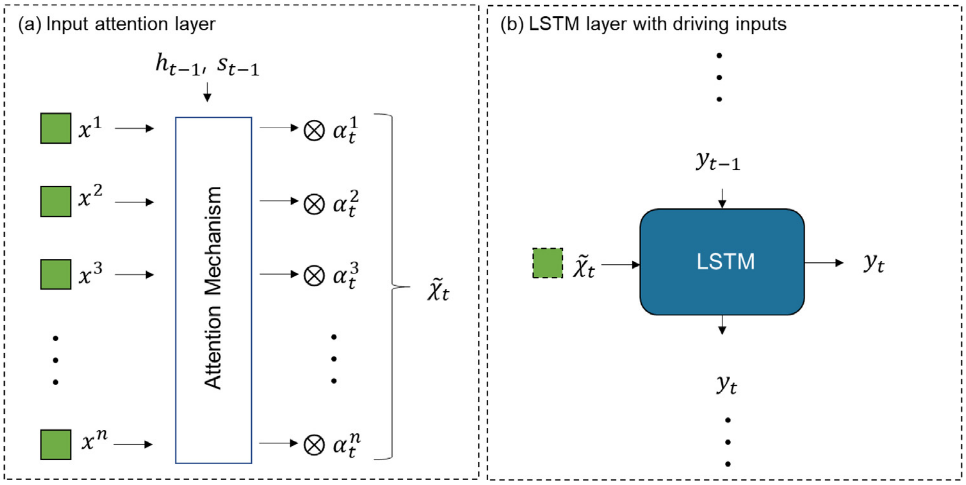


Fig. 4. Structure of IA-LSTM model suggested in this study. (a) IA model layer was adopted from the DA-RNN proposed by Qin et al. (2017). x^n indicates the original inputs of n number of input features, and a_t^k represents the attention weight for the k-th input at time t. Each attention weight is given to the original input by tensor product ($\otimes$). (b) Driving inputs ($\tilde{X}_t$) calculated from the IA layer are used instead of the original LSTM inputs for prediction.

Table 2
Optimized hyperparameters for ARG prediction with conventional LSTM and LSTM-CNN.

ARGs	LSTM					CNN	
	Batch size	Lookback	Learning rate	LSTM units	Activation function	Filter size	Activation function
(a) Single ARGs prediction with conventional LSTM							
bla$_{\mathrm{TEM}}$	24	8	1.00E-07	128	ReLU	N/A	N/A
(b) Single ARGs prediction with LSTM-CNN							
aac(6'-Ib-cr)	12	16	9.59E-07	256	ReLU	128	ReLU
bla$_{\mathrm{TEM}}$	4	16	3.04E-07	64	ReLU	256	ReLU
*sul*1	4	14	9.30E-07	64	Tanh	128	Tanh
*tet*X	24	16	9.70E-07	256	ReLU	128	Tanh
(c) Multi-ARGs prediction with LSTM-CNN							
aac(6'-Ib-cr) + *bla*$_{\mathrm{TEM}}$ + *sul*1 + *tet*X	4	16	3.82E-07	64	ReLU	128	ReLU

* ReLU and Tanh indicate Rectified Linear Unit and hyperbolic tangent, respectively. N/A: not available

(Fig. 3). At each time step, the output of the LSTM model is the input into the next model, the 1D CNN (Fig. 3c). 1D CNN can extract features by convolving a weight matrix with the inputs (Eq. 7). The size of the weight matrix determines the total number of parameters in the CNN while the size of the convolution operation is based on the kernel size. The convolution output was further modified by applying a non-linear activation function. The values of these hyperparameters were obtained using Bayesian optimization, and they are provided in Table 2. The CNN layer was followed by a maximum pooling layer (Eq. 8) to reduce the output size. Finally, we applied a fully connected layer to obtain the final prediction from the model according to the below equations (Fukushima, 1979).

$$y = f\left(\sum x * w + b\right) \tag{7}$$

$$z = \max(0,\ x) \tag{8}$$

where x and y indicate the input and output of the CNN, respectively. w is the weight matrix, and b is the bias. The activation function (f) used in each model is indicated in Table 2.

2.3.3. IA-LSTM model

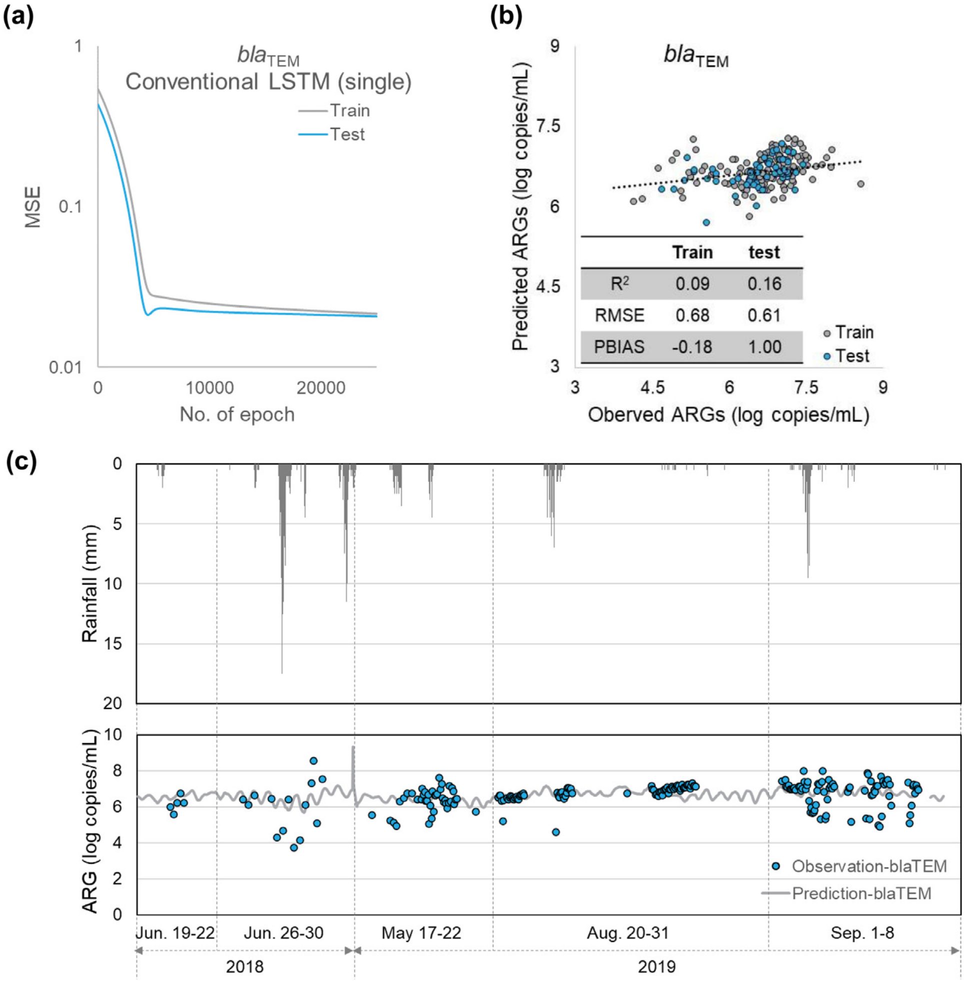
The LSTM model can extract temporal features from inputs. However, not all of the inputs into an LSTM are equally important. Model performance can be enhanced if it focuses on more relevant inputs. To account for this, many researchers have proposed an "attention" mechanism that forces the NN to focus on a certain part of the input (Luong et al., 2015). One example is the dual-stage attention mechanism (DA-RNN), which was designed for time series prediction problems (Qin et al., 2017). In DA-RNN, the "input attention (IA)" is first applied, which allows the model to focus on relevant inputs. In the second stage, "temporal attention" is applied, which assists the model in selecting relevant data from the history. In this study, we used an IA mechanism in combination with LSTM as illustrated in Fig. 4. The purpose of IA is to adaptively select relevant driving input data. IA-LSTM is not a black box model unlike other deep learning models in that it can calculate the importance of input variable on the model according to the below equations (Qin et al., 2017).

$$e_t^k = v_e^T tanh(W_e)\left[h_{t-1}; s_{t-1} + U_e x^k\right] \tag{9}$$

$$\alpha_t^k = \frac{exp\left(e_t^k\right)}{\sum_{i=1}^n exp\left(e_t^i\right)} \tag{10}$$

$$\tilde{X}_t = \left(a_t^1 x_t^1,\ a_t^2 x_t^2,\ \ldots a_t^n\ x_t^n\right) \tag{11}$$

Given n number of input features, the attention weight for the k-th input a_t^k is calculated using the hidden and cell states of LSTM according to Eq. (10). The parameters U_e, v_e^T, and W_e are calibrated during the training process. Eq. (10) is a softmax function that returns an array of attention weights that sum to one Chollet, 2018). The attention weights for each input are then used to extract the driving inputs $\tilde{X}_t$ according to Eq. (11). The more relevant driving inputs are amplified by their higher weights while the less important ones are reduced to smaller corresponding weights. These



	Train	test
R^2	0.09	0.16
RMSE	0.68	0.61
PBIAS	-0.18	1.00

Fig. 5. Conventional LSTM results for bla_{TEM}. (a) Loss curves during training and testing. (b) Scatter plot and (c) time series plot comparing observations and model predictions.

inputs are then used as inputs into LSTM to extract the temporal features. Thus, instead of using the original inputs X_t Eqs. (1)–(4), the IA mechanism uses $\tilde{X}_t$ within LSTM. The remainder of this model is similar to that of the conventional LSTM as described in Section 2.3.1, which produces the final model prediction.

2.3.4. Model training

The predictions from each model were used to calculate the model "loss" values by comparing them with the observed ARG values. To reduce this value, a gradient-based learning mechanism called "backpropagation" was used (Rumelhart et al., 1986). During backpropagation, the model weights are updated to reduce the loss value. The weight changes are controlled by the learning rate parameter. We used the adaptive motion estimation, Adam optimizer (Kingma and Ba, 2014), an adaptive learning rate algorithm. This algorithm adjusts the learning rate during the training process and has been used in many deep learning benchmarks (Gregor et al., 2015; Xu et al., 2015). The NN was designed to produce continuous predictions even though the observed ARG values were sparsely distributed. Because the observed ARG was not available at the 30-minute time step, we calculated the loss value using predictions that correspond to available observations. Predictions for which no corresponding observations were available were not used for model training. This was achieved using a masking layer that informed the model about the availability or unavailability of observations. In this manner, it was possible to train the models using 30-minute input data and sparsely observed target data. The total datasets were divided into 70% for training and 30% for validation. To improve the performance, training and validation datasets were randomly selected for training on the entire dataset.

The RNN-based models we used in this study require historical input data for single- step prediction. The number of historical steps used by the models for making predictions is known as lookback steps (Chollet, 2018). By increasing the length of the lookback steps, we can feed more historical data to the NN to make predictions. In this study, we optimized the value of these lookback steps during hyperparameter optimization.

The variation in the observed ARG at the beach was very significant. This is evident from the very high standard deviations and variances (Fig. S3). To account for this large variation, we transformed the ARG values on logarithmic scale with a base of 10. This is because log transformation can reduce the effect of outliers from the data (Singh and Kingsbury, 2017). It has been reported that log transformation can improve the performance of data-driven models when the data contain outliers (Zheng and Casari, 2018).

We used Tensorflow API version 1.15 (Abadi et al., 2016) in the Python programming language to build the models. The models were trained using an Intel® Core^{TM} i7-8700 processor with an NVIDIA GeForce GTX 1060 graphic card, 6 Gigabytes of dedicated GPU memory, and 32 Gigabytes of random access memory.

2.3.5. Hyperparameter optimization

We used a Bayesian optimization algorithm with Gaussian processes as a surrogate function to optimize the hyperparameters,

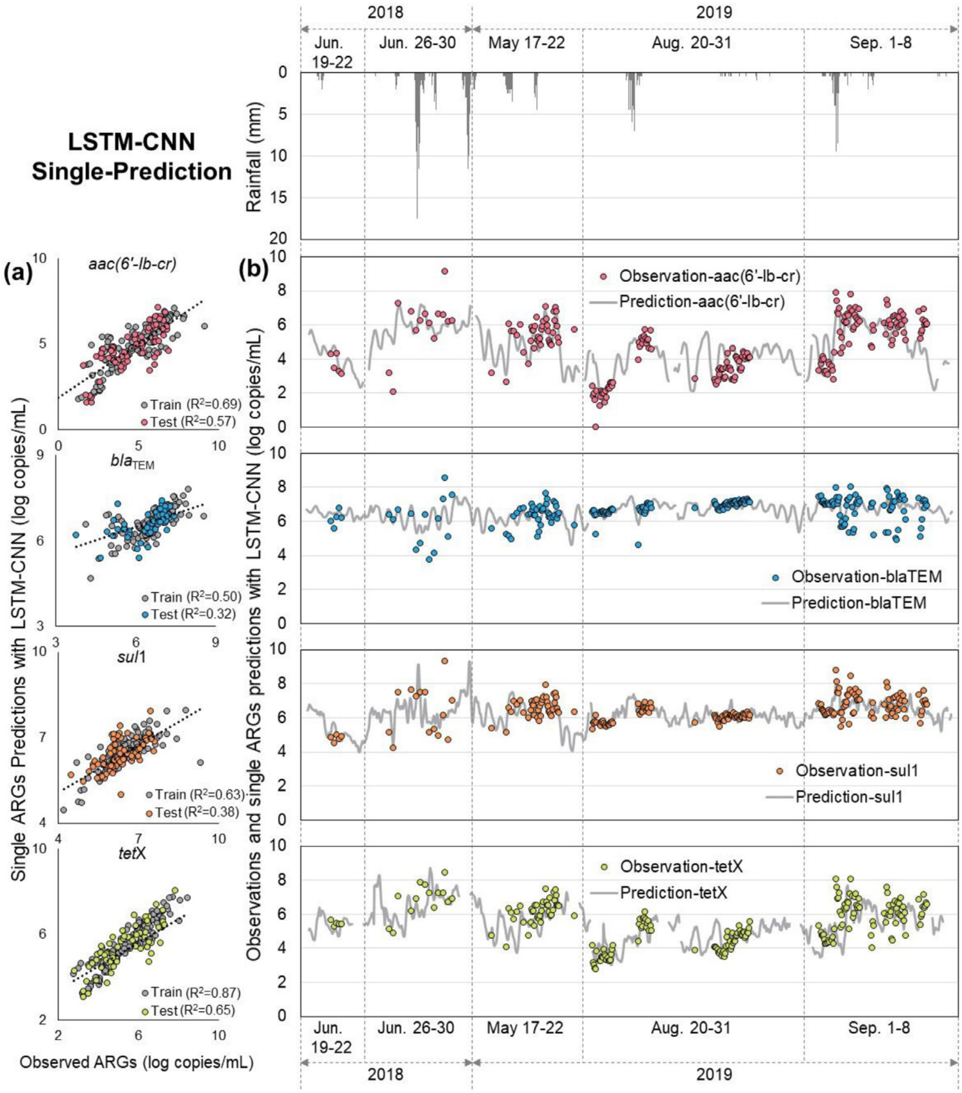


Fig. 6. (a) Scatter plots with trend lines (b) timeseries plots of observations and model predictions for single ARG LSTM-CNN predictions.

including the batch size, lookback, learning rate, and weight matrix sizes in LSTM and CNN, and activation functions for LSTM and CNN. The convergence, evaluation and objective plots for each optimization are exhibited with more information on the hyperparameter optimization in the supplementary information.

2.3.6. Performance evaluation

Out of the 218 ARG samples, we used 150 to train the NN, and the remaining 68 were used to evaluate the model performance. We used the mean absolute error (MSE) to train the model. We also evaluated the model performance using the coefficient of determination (R^2), root MSE (RMSE), and percent bias (PBIAS). The equations to calculate these metrics are as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(o_i - p_i)^2}{n}}, \tag{12}$$

$$PBIAS = \frac{\sum_{i=1}^{n} o_i - p_i}{\sum_{i=1}^{n} o_i} \times 100 \tag{13}$$

where o_i and p_i are the observed and predicted data, respectively, and n indicates the number of datasets. Because the neural network was trained with logarithmically transformed data, the performance metrics reported in this study were calculated and interpreted on logarithmic scale.

3. Results and discussion

3.1. Hyperparameter optimization

3.1.1. Hyperparameter optimization for single ARG prediction

By comparing the partial dependence plots in objective plots of all the models (Figs. S4 and S5), we can infer that the learning rate was the most sensitive parameter during optimization. In contrast, the activation function in the CNN layer was the least sensitive parameter for the NN across models. The optimization results also demonstrated that a higher lookback value resulted in a greater reduction in the model test MSEs in all cases except for bla_{TEM}. No uniform trend for batch size can be inferred for all the models except for bla_{TEM}, where a smaller batch size of four resulted in the maximum improvement in performance. For LSTM units, a higher value resulted in an improved performance for *sul*1 and *tet*X, which exhibited a maximum improvement in performance with 256 LSTM units. For bla_{TEM}, the optimum number of LSTM units was 64, and a smaller number of LSTM units improved the model performance. The optimized values of all parameters for the LSTM-CNN and IA-LSTM models are listed in Tables 2 and 3, respectively. In this analysis, the batch size was adjusted to 4, 8, 12, and 24. The results demonstrate that various batch sizes were se-

Table 3
Optimized hyperparameters for ARG prediction with IA-LSTM.

ARGs	Batch size	Lookback	Learning rate	LSTM units	Activation function	
					input attention	prediction
(a) Single ARGs prediction with input attention-LSTM						
aac(6′-Ib-cr)	8	20	9.94E-04	16	ReLU	ELU
bla$_{\text{TEM}}$	24	14	5.39E-04	64	ELU	Tanh
*sul*1	12	19	3.39E-04	16	ReLU	Tanh
*tet*X	24	19	9.90E-04	32	none	ELU
(b) Multi-ARGs prediction with input attention-LSTM						
aac(6′-Ib-cr)	12	6	8.55E-04	32	ELU	ELU
bla$_{\text{TEM}}$				64	LeakyReLU	LeakyReLU
*sul*1				16	ReLU	ReLU
*tet*X				64	ReLU	ReLU

* ReLU, ELU, and Tanh indicate Rectified Linear Unit, Exponential Linear Unit, and hyperbolic tangent, respectively.

lected to test model efficiency in single ARGs prediction, and the batch sizes in the LSTM-CNN model were smaller compared to those of the conventional LSTM or IA-LSTM models (Table 2). In deep learning, the smaller the batch size, the greater the variation (Keskar et al., 2016); thereby, smaller batch sizes appear to provide sufficient variances for calculating the gradient descent in LSTM-CNN.

3.1.2. Hyperparameter optimization for multi-ARG prediction

For multi-ARG NN prediction with LSTM-CNN, the hyperparameters were obtained using the Bayesian optimization method, and they are provided in Table 2. The top row in Fig. S4e show the variation in objective function with change in each hyperparameter. The steepest slope for learning rate shows its change caused highest variations in objective functions. Thus, the learning rate can be considered as the most sensitive hyperparameter for this model. In the LSTM-CNN model, the Rectified Linear Unit (ReLU) activation function was the most effective for both LSTM and the CNN layer. A smaller batch size of 4 and 64 LSTM units resulted in improved model performance compared to that with higher parameter values. Due to structural differences within the LSTM-CNN model, the LSTM unit and activation function were assigned differently to each ARG in the LSTM-CNN. The optimized hyperparameters for the LSTM-CNN are presented in Table 3 without the convergence, evaluation, and objective plots.

3.2. ARG prediction results

3.2.1. Single ARG prediction with conventional LSTM

The conventional LSTM was expected to be useful for predicting ARGs at a recreational beach because ARGs have been reported to occur during rainfall events instead of immediately after rainfall (Fig. S2). In this study, single ARG (*bla*$_{\text{TEM}}$) prediction was first tested using conventional LSTM after optimization. The model was trained for 25,000 epochs with optimized hyperparameters, and the loss curves of the training and testing are displayed in Fig. 5a. To select the final model, we chose the model where test loss began increasing with further training since the increase in test loss indicates overfitting. After this point, we ignored further decreases in training loss. Fig. 5b and c display the scatter and time series plots of the observed and predicted values, respectively. Their performances were evaluated using R^2, RMSE, and PBIAS (Table 4a). The R^2 values were 0.09 and 0.16 for training and testing, respectively, demonstrating that the conventional LSTM model did not adequately capture the features of the *bla*$_{\text{TEM}}$ occurrence even after optimization. In this case, increasing training or adding layers did not improve the model performance supporting previous reports that increasing NN complexity does not necessarily improve the performance compared to simple NNs (Makridakis et al., 2018). Overall, conventional LSTM did not effectively capture the occurrence patterns of ARGs. Therefore, we combined LSTM with CNN instead of predicting the other ARGs with LSTM.

Table 4
Model performances for single and multi-ARG prediction.

	aac(6′-Ib-cr)		*bla*$_{\text{TEM}}$		*sul*1		*tet*X	
	train	test	train	test	train	test	train	test
(a) Single ARGs prediction with conventional LSTM								
$\mathbf{R^2}$	N/A	N/A	0.09	0.16	N/A	N/A	N/A	N/A
RMSE	N/A	N/A	0.68	0.61	N/A	N/A	N/A	N/A
PBIAS	N/A	N/A	-0.18	1.00	N/A	N/A	N/A	N/A
(b) Single ARGs prediction with LSTM-CNN								
$\mathbf{R^2}$	0.69	0.57	0.50	0.32	0.63	0.38	0.87	0.65
RMSE	0.89	1.02	0.50	0.61	0.46	0.53	0.46	0.71
PBIAS	1.67	1.78	0.36	1.23	-0.27	0.97	0.81	1.11
(c) Multi-ARGs prediction with LSTM-CNN								
$\mathbf{R^2}$	0.65	0.55	0.20	0.15	0.52	0.35	0.67	0.56
RMSE	0.87	0.97	0.63	0.68	0.48	0.50	0.72	0.80
PBIAS	-0.02	1.89	0.67	1.84	0.18	0.63	-0.06	0.78
(d) Single ARGs prediction with IA-LSTM								
$\mathbf{R^2}$	0.86	0.41	0.16	0.15	0.62	0.09	0.61	0.49
RMSE	0.06	0.14	0.98	0.96	0.09	0.13	0.14	0.16
PBIAS	-1.21	1.44	10.51	9.71	1.47	-3.22	-2.26	3.03
(e) Multi-ARGs prediction with IA-LSTM								
$\mathbf{R^2}$	0.68	0.44	0.35	0.31	0.37	0.27	0.80	0.67
RMSE	0.10	0.14	0.12	0.13	0.12	0.12	0.09	0.13
PBIAS	1.22	-0.41	-0.32	-1.77	-1.19	-3.07	-2.67	-6.35

* N/A: not available

3.2.2. Single and multi-ARG prediction with LSTM-CNN hybrid model

Each single ARG prediction model was trained with hyperparameters obtained from optimization. Each model was trained for 10,000 epochs, and then the performance was evaluated using a test dataset because all models achieved their optimal performance before 10,000 epochs. The training and test losses of the models are displayed in Fig. S6a–S6d. We observed that the test losses for *aac(6′-Ib-cr)* and *tet*X began increasing much earlier than those for *bla*$_{\text{TEM}}$ and *sul*1. The minimum test losses for *aac(6′-Ib-cr)* and *tet*X were achieved after training the model for 2,843 epochs and 6,461 epochs, respectively. For all models, the MSE values sufficiently predicted each single ARG; however, the training losses continued to decrease demonstrating signs of overfitting, as is evident from the rising test loss curves. R^2 and PBIAS were also measured during training and testing to evaluate model performances (Table 4b). The R^2 values ranged from 0.50 to 0.87 for training and from 0.32 to 0.65 for testing. The greater NN training accuracy compared to that of testing indicates that the NNs were overfitted as presented in the loss curves. PBIAS is a relative error from the observations; thereby, a positive value indicates overprediction, and a negative value demonstrates underprediction. In our models, the PBIAS presented values between 0.27 % and 1.78 % in both the training and

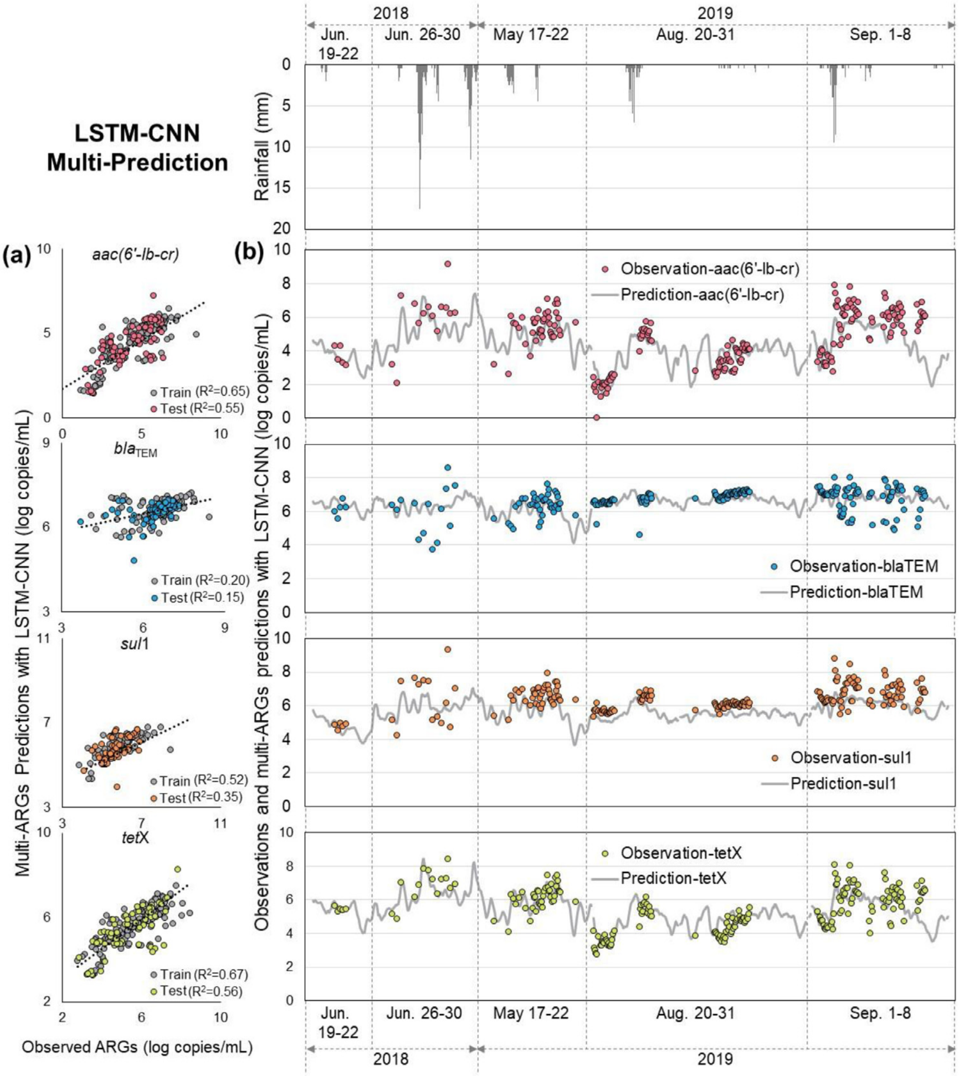


Fig. 7. (a) Scatter plots with trend lines (b) timeseries plots of observations and model predictions for multi-ARG LSTM-CNN predictions.

testing results. These values can be considered “very good” levels for NNs with no bias in the average tendency of predictions for observations according to the hydrological criteria proposed by Moriasi et al. (2015). Comparisons of the observed and predicted ARGs are presented as scatter and time series plots in Fig. 6a and b, respectively. Overall, the optimal performance was achieved for *tet*X while the worst performance was obtained for *bla*$_{\mathrm{TEM}}$.

Compared to conventional LSTM, LSTM-CNN exhibits a 2–6 times improved accuracy in *bla*$_{\mathrm{TEM}}$ prediction (Table 2a and b). The convolution models for sequence data were more successful when combined with an RNN in previous studies (Barzegar et al., 2020; Lee et al., 2017). However, those studies applied LSTM after the CNN for prediction. Our LSTM-CNN model differs in that the LSTM output was applied to the CNN. In LSTM-CNN, the CNN appears to have played a role in extracting local features from the LSTM outputs. It has already been reported by Barzegar et al. (2020) that LSTM can be trained to extract long-term dependencies, and the CNN can extract time-invariant features, thereby improving model predictability. The proposed LSTM-CNN model is presumed to exhibit the advantages of both models. Overall, the proposed LSTM-CNN model captures information from erratic and complex changes in ARGs more effectively than that of the conventional LSTM model.

The performance metrics of multi-ARG prediction NN for training and test datasets are provided in Table 4c. During model training, we calculated the loss function after each epoch for both training and test data which is plotted in Fig. S6e. Correlations between the observed and predicted values are illustrated as scatter and time series plots in Fig. 7a and b, respectively. Model performance has been listed for each ARG, although the models predicted all four ARGs simultaneously. The performance metrics for each ARG declined compared to those of their corresponding single ARG prediction models. This is due to the higher learning capacity of single ARG prediction models. In contrast, the multi-ARG prediction NNs had to learn the complex and differing behaviors of all ARGs. Thus, the performance deteriorated when the NN predicted multiple ARGs. Nevertheless, we can conclude that the predictive values agreed with the observation values of the multi-prediction model indicating that this model performed well in simultaneously predicting multiple ARGs. Our result reveals the potential of applying the LSTM-CNN hybrid model in predicting multi-ARGs at a recreational beach setting, which can provide a reliable foundation for water policies. The optimal test performance was obtained for *tet*X while the poorest performance was obtained for *bla*$_{\mathrm{TEM}}$. This is similar to single ARG prediction models, where the *bla*$_{\mathrm{TEM}}$ model exhibited the highest RMSE compared to that of the other ARG

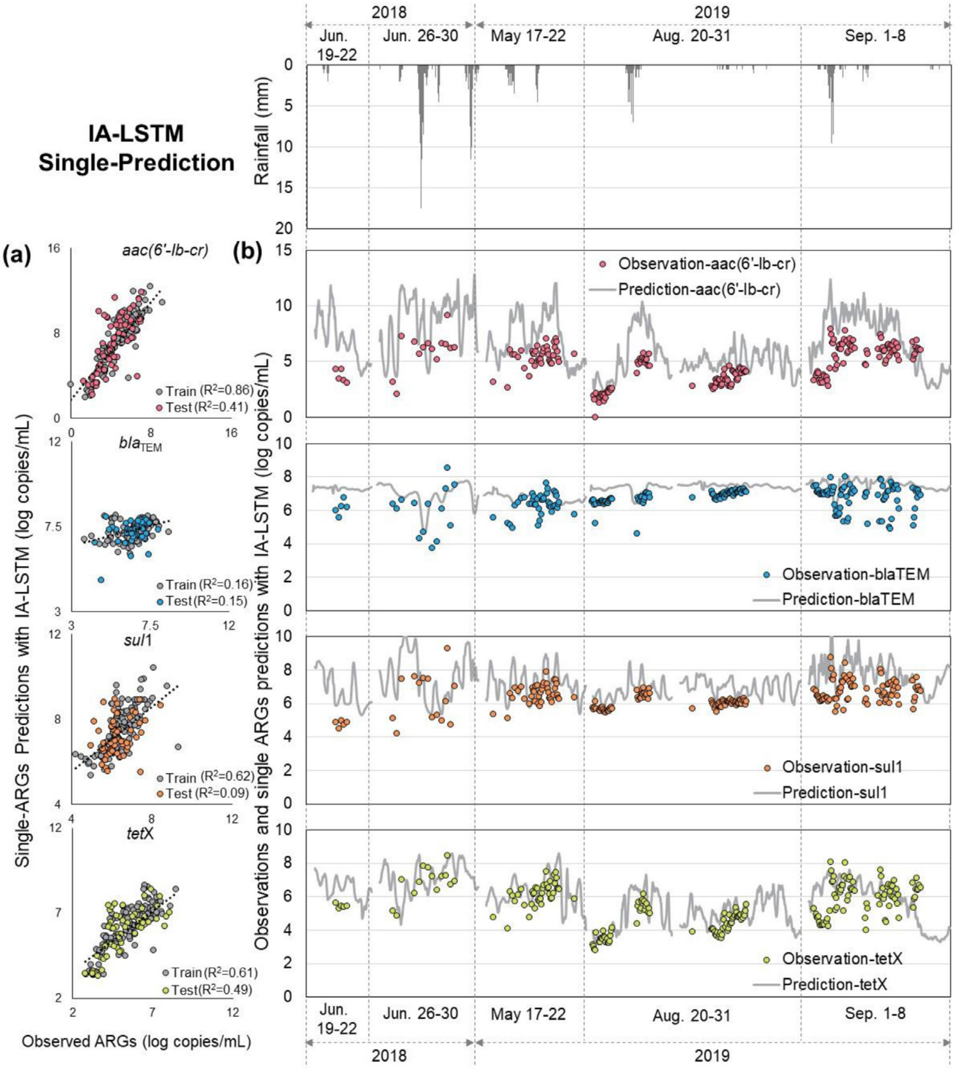


Fig. 8. (a) Scatter plots with trend lines (b) timeseries plots of observations and model predictions for single ARG IA-LSTM predictions.

prediction models. The low predictive performance for bla_{TEM} may be attributed to improper variables such as the skewness of the bla_{TEM} observations (Table S1). If the dataset follows a normal distribution, the prediction ability of the model can increase without ignoring the tail parts in the distribution. According to a previous study, datasets with a skewness rage of -2 to +2 are considered to follow a normal distribution (George and Mallery, 2010). For bla_{TEM}, the skewness value was within the range of a normal distribution, but it was close to the threshold indicating that the bla_{TEM} dataset is slightly skewed to the right compared to the other ARG datasets. Thus, it appears that this slightly skewed dataset may have affected the training and prediction of bla_{TEM}.

3.2.3. Single and multi-ARG prediction with IA-LSTM model

Scatter and time series plots are presented in Figs. 8a and b, respectively, to demonstrate the correlations between observations and predictions for single ARGs using IA-LSTM. The performance metrics for single ARG prediction obtained using IA-LSTM are displayed in Table 4d. The R^2 values of single ARGs predictions using IA-LSTM were between 0.16 and 0.86 during training and 0.09 and 0.49 during testing. Additionally, for *aac(6′-Ib-cr)* and *sul*1 predicted using IA-LSTM, overfitting was expected because the R^2 training values were larger than those of testing. For bla_{TEM} predicted using IA-LSTM, the performance was considerably worse, which can be attributed to inappropriate inputs as mentioned above. The IA-LSTM predictions verified that all ARGs, except for bla_{TEM}, were "very good" based on the PBIAS values (Moriasi et al., 2015).

The multi-ARGs predictions demonstrated that the performances of *aac(6′-Ib-cr)* and *sul*1 with IA-LSTM were worse than those of the single ARG predictions (Table 4e). However, the performances of bla_{TEM} and *tet*X in multi-ARG predictions using IA-LSTM improved compared to those of the single ARGs predictions. It is uncommon to exhibit superior performance in multi-prediction compared to individual prediction. Higher accuracy in multi-prediction may be because the weights can be allocated to significant inputs when learning weights. IA-LSTM relies on an attention mechanism to learn the weights of each time stage (Bahdanau et al., 2016). The attention mechanism can suppress noisy or unnecessary inputs by using an attention weight (Qin et al., 2017). In particular, when predicting *aac(6′-Ib-cr)* with IA-LSTM, the model predicted a value twice that observed during the largest rainfall (17.5 mm) between June 26, 2018, and June 30, 2018 (Fig. 9b). In addition, for other ARGs such as *sul*1 and *tet*X, the IA-LSTM also tended to predict much larger values than observations during the largest rainfall. These suggest that the IA-LSTM model predictions allocated the weights to rainfall as one of significant inputs when learning weights.

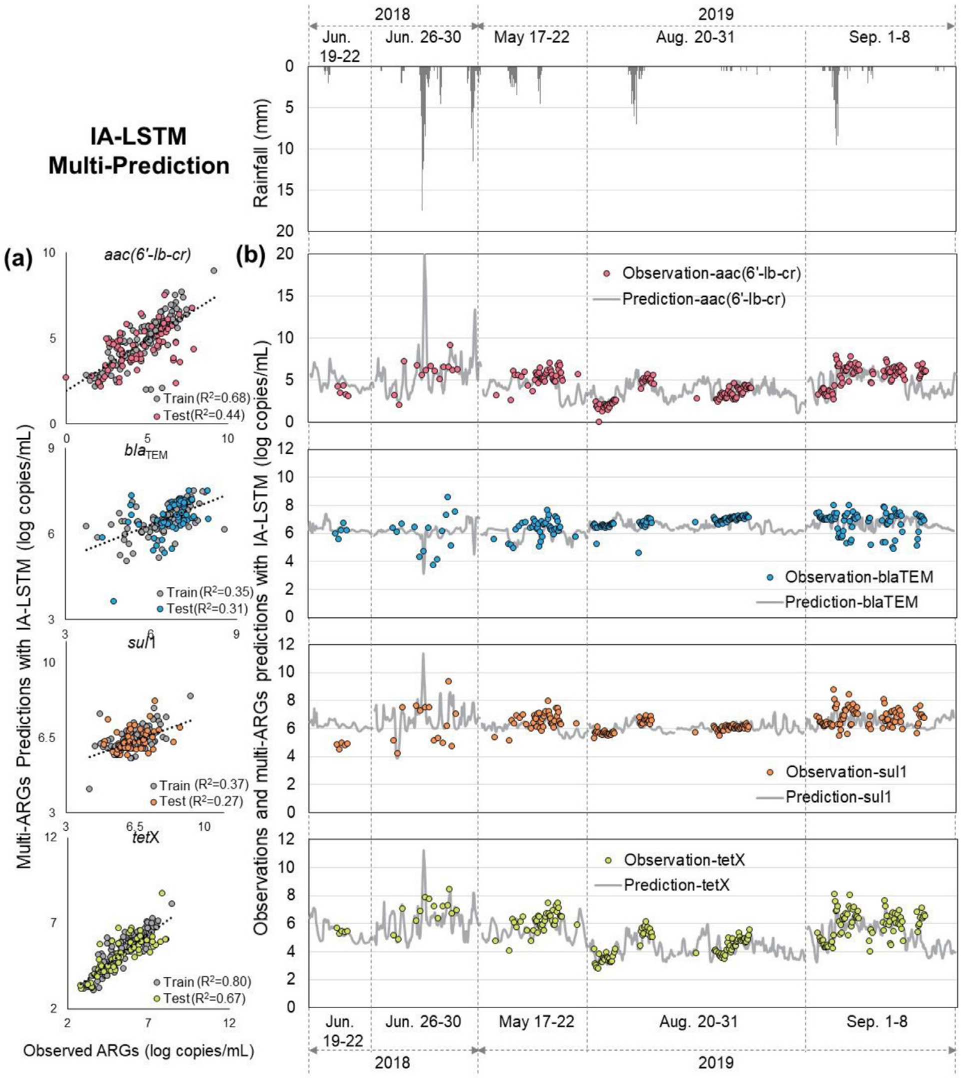


Fig. 9. (a) Scatter plots with trend lines (b) timeseries plots of observations and model predictions for multi-ARG IA-LSTM predictions.

All R^2 values, except for *aac(6′-Ib-cr)* in training, were lower than those of LSTM-CNN for single ARG predictions (Table 4e). In the case of multi-ARG prediction, IA-LSTM could predict multiple ARGs simultaneously with higher accuracy than that of LSTM-CNN. Notably, the performance of NN for *bla*$_{\mathrm{TEM}}$ with IA-LSTM was superior to that of LSTM-CNN during both training and testing. With IA-LSTM, the multi-prediction performance was higher than that of LSTM-CNN because the attention weight was focused on the most relevant information, thereby maximizing the information characteristics (Qin et al., 2017). When training NNs under complex conditions that consider the characteristics of all ARGs, attention weights in IA-LSTM can likely extract features more effectively than the weights of LSTM or CNN. Plots of single and multi-ARG predictions during model training and testing for IA-LSTM are displayed in Figs. 8 and 9. Table S2 shows the Shannon entropy (Cover, 1999) values of the predicted ARG from the LSTM-CNN and IA-LSTM models. We observed that for LSTM-CNN based models, the entropy of the predicted arrays is larger than those of observed. Entropy is a measure of uncertainty and chaos in a system (Sanei and Chambers, 2013). In time-series data, it quantifies the degree of complexity (Kumar and Dewal, 2011). Based upon these results we conclude that, IA-LSTM is a more favorable method than LSTM-CNN in predicting multi-ARGs, while LSTM-CNN is more suitable for predicting single ARGs.

3.3. Important variables for ARG prediction

Input variables are important factors in determining model accuracy. However, the conventional LSTM and LSTM-CNN hybrid models are uninterpretable in the importance of input variables due to the blackbox effect. One of the advantages of using attention-based models is that they allow us to interpret the model. The plots of attention weights with respect to lookback for different inputs represent, which input was more relevant in deciding model's output. A higher attention weight for an input feature shows that the model focused more on this input in predicting the output. Since the attention weights also vary with lookback, they also show which previous value of an input was more important in model's prediction. Three input variables (i.e., rainfall, tides, and salinity) that affect ARG prediction were selected. This was based on previous studies that found that ARGs were affected by rainfall and tidal level (Jang et al., 2021) and relatively higher abundances of ARGs were found in upstream with low salinity (Lee et al., 2017; Baron et al., 2018). Although the variables related to wind and temperature were also used as input to increase prediction ability, the significance of these variables was lower than that of the three variables. This may be because this study targeted the recreational season of the beach (May-September). Had we considered monitoring in other seasons, the variables affecting the ARG concentration

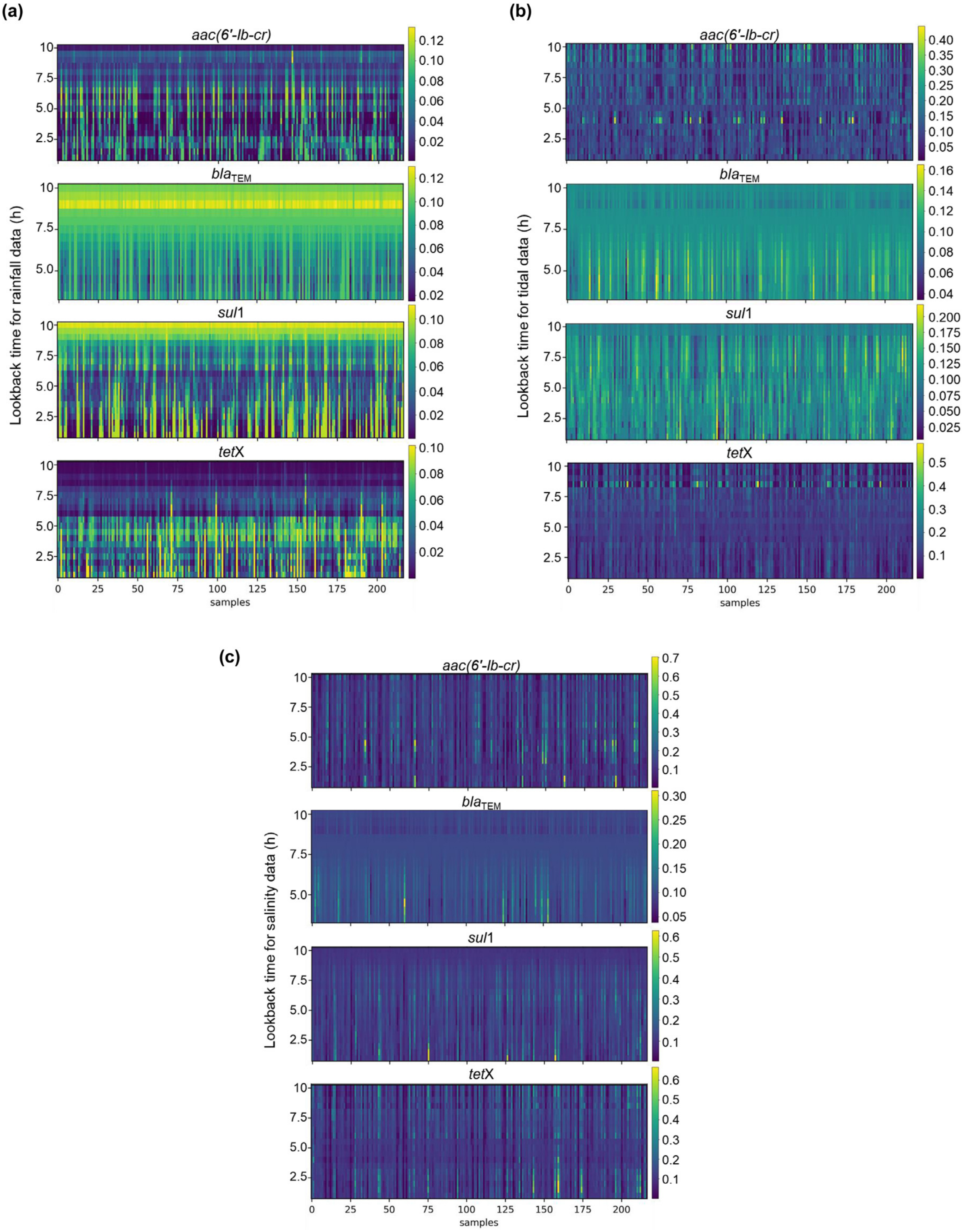


Fig. 10. Importance of input variables such as (a) rainfall, (b) tide, and (c) salinity for prediction of ARGs. The color bars represent importance of each variable. Lookback time is the time of the historical data used by Neural Network to predict next value.

could have been different. Fig. 10 displays the importance of the input variable according to the lookback time. The closer an importance is to the yellow side, the more likely it is that lookback time affects the prediction while the closer an importance is to the blue side, the less impact lookback time has on it. Based on attention weights, rainfall substantially affected predictions with various lookback times depending on the ARG: rainfall data of 2.5–5 h ago was important for *aac(6'-Ib-cr)* and *tet*X, while 7.5–10 h ago for *bla*$_{TEM}$ and both ~7.5 h and 10 h ago for *sul*1 (Fig. 10a). The previously reported network analysis showed that *aac(6'-Ib-cr)* and *tet*X were associated with bacterial communities during and after rainfall, but *bla*$_{TEM}$ and *sul*1 showed relevant to bacterial communities only after rainfall (Jang et al., 2021). Based on this result, we assume that *bla*$_{TEM}$ and *sul*1 were affected by a relatively long lag for rainfall, possibly due to slow gene transfer among bacteria.

One of the main features of ARG prediction is the time delay between rainfall and ARG outflow by CSOs. As illustrated in the time series results, both LSTM-CNN and IA-LSTM respond to the delayed ARG release after the rainfall (Figs. 6b, 7b, 8b, and 9b). Compared to our previous study where total ARGs were found to be released after 5–7 h of rainfall at Gwangalli Beach (Jang et al., 2021), all ARGs exhibited a similar release tendency. The tide data also demonstrated that different lookback times affected the ARG predictions (Fig. 10b). Tidal data from 7.5–10 h prior corresponded with the rainfall data and appeared to affect *tet*X prediction. For beaches with CSOs, the tide is one of the most influential sources of ARG occurrence. As previously reported, Gwangalli Beach exhibits a semi-diurnal tidal cycle; thereby, ARGs are introduced to the coast through CSOs during the ebb tide (Jang et al., 2021; (Kim et al., 2020) . Accordingly, the prediction of *tet*X is affected by the tidal cycle of Gwangalli Beach. Additionally, 4 h of tidal data affected both *aac(6′-Ib-cr)* and *bla*$_{\text{TEM}}$ predictions, but the *sul*1 prediction appeared to be affected by the overall tidal data, not the specific lookback time. These results imply that both *aac(6′-Ib-cr)* and *bla*$_{\text{TEM}}$ prediction was sensitive to shorter tidal changes at intervals of 4 h while *sul*1 prediction was predominantly affected by the overall tidal data. Lastly, salinity data within 5 h prior were the most influential for all ARGs predictions (Fig. 10c). The NNs were sensitive to changes in salinity because these ARGs commonly occur in water bodies with lower salinity levels; *aac(6′-Ib-cr), bla*$_{\text{TEM}}$, and *sul*1 are frequently detected species in livestock industries in South Korea (Lee et al., 2017), and tetracycline-resistance genes (e.g., *tet*X) are frequently found in human feces (Baron et al., 2018). In addition, it is assumed that the inflow of freshwater by rainfall affected the ARG predictions, as prediction sensitivity corresponded to rainfall periods.

4. Conclusions

The goal of this study was to improve the accuracy of predictions for ARG occurrence and to identify the variables that affect these predictions. Thus, in this study, the conventional LSTM, LSTM-CNN hybrid, and IA-LSTM models were compared to predict ARGs occurrence according to environmental variables. The primary results of this study are as follows:

1) The sequential convergence of LSTM and CNN resulted in improved performance compared to that of conventional LSTM to predict single ARGs. We show that the use of CNN on the output of LSTM enhances model's performance by extracting local features..
2) IA-LSTM was not able to outperform LSTM-CNN in predicting single ARGs; however, it exhibited superior performance in predicting multi-ARGs. The attention weights of IA-LSTM helps the model to focus on decisive features when it is necessary to learn more complex features.
3) Unlike the LSTM and LSTM-CNN blackbox models, the importance of input variables that affect the prediction can be identified using IA-LSTM.
4) ARGs occurrence predictions were sensitive to input variables in different lookback ranges. Therefore, we identified the time-dependency ranges within the input sequences that each ARG prediction relies on.

Throughout this study, deep learning model has demonstrated its performance on prediction of ARGs occurrence. This study also provides useful information on selection of proper deep learning techniques for predicting microbial water quality at a beach.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This study was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2017R1D1A1B04033074), and Korea Environment Industry and Technology Institute (KEITI) through the Aquatic Ecosystem Conservation Research Program funded by Korea Ministry of Environment (MOE) (No. 2020003030003).

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.watres.2021.117001.

Reference

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., 2016. Tensorflow: A system for large-scale machine learning 265–283.

Abimbola, O.P., Mittelstet, A.R., Messer, T.L., Berry, E.D., Bartelt-Hunt, S.L., Hansen, S.P., 2020. Predicting Escherichia coli loads in cascading dams with machine learning: An integration of hydrometeorology, animal density and grazing pattern. Science of The Total Environment 722, 137894.

Zheng, A, Casari, A, 2018. *Feature engineering for machine learning: principles and techniques for data scientists*. O'Reilly Media, Inc..

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., … De Freitas, N. (2016). Learning to learn by gradient descent by gradient descent. arXiv preprint arXiv:1606.04474.

Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., Bengio, Y., 2016. End-to-end attention-based large vocabulary speech recognition. IEEE, pp. 4945–4949.

Baron, S.A., Diene, S.M., Rolain, J.-M., 2018. Human microbiomes and antibiotic resistance. Human Microbiome Journal 10, 43–52.

Barzegar, R., Aalami, M.T., Adamowski, J., 2020. Short-term water quality variable prediction using a hybrid CNN–LSTM deep learning model. Stochastic Environmental Research and Risk Assessment 1–19.

Bedri, Z., O'Sullivan, J.J., Deering, L.A., Demeter, K., Masterson, B., Meijer, W.G., O'Hare, G, 2015. Assessing the water quality response to an alternative sewage disposal strategy at bathing sites on the east coast of Ireland. Marine pollution bulletin 91 (1), 330–346.

Choi, S.-H., Lee, S.-M., Kim, G.-S., Kim, M.-H., Ji, H.-S., Jeong, Y.-N., Yoo, E.-C., Cho, J.-G., 2016. Effects of Rainfall on Microbial Water Quality on Haeundae and Gwangan Swimming Beach. Journal of Bacteriology and Virology 46 (2), 71–83.

Chollet, F., 2018. Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek. MITP-Verlags GmbH & Co. KG.

Cover, T.M., 1999. Elements of information theory. John Wiley & Sons.

de Castro, A.P., Fernandes, G.d.R., Franco, O.L, 2014. Insights into novel antimicrobial compounds and antibiotic resistance genes from soil metagenomes. Frontiers in Microbiology 5, 489.

Fang, H., Han, L., Zhang, H., Long, Z., Cai, L., Yu, Y., 2018. Dissemination of antibiotic resistance genes and human pathogenic bacteria from a pig feedlot to the surrounding stream and agricultural soils. Journal of hazardous materials 357, 53–62.

Fukushima, K., 1979. Neural network model for a mechanism of pattern recognition unaffected byshift in position. Neocognitron.Trans. IECE J62-A (10), 658–665.

George, D. and Mallery, M. (2010) SPSS for Windows Step BysStep: A Simple Guide and Reference.

Gregor, K., Danihelka, I., Graves, A., Rezende, D.J. and Wierstra, D. (2015) Draw: A recurrent neural network for image generation. arXiv preprint arXiv:1502.04623.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Computation 9 (8), 1735–1780.

Jang, H.M., Shin, J., Choi, S., Shin, S.G., Park, K.Y., Cho, J., Kim, Y.M., 2017. Fate of antibiotic resistance genes in mesophilic and thermophilic anaerobic digestion of chemically enhanced primary treatment (CEPT) sludge. Bioresource Technology 244 (1), 433–444.

Jang, J., Kim, M., Baek, S., Shin, J., Shin, J., Shin, S.G., Kim, Y.M., Cho, K.H., 2021. Hydrometeorological Influence on Antibiotic-Resistance Genes (ARGs) and Bacterial Community at a Recreational Beach in Korea. Journal of hazardous materials 403, 123599.

Jansen, R.R., Schinkel, J., Koekkoek, S., Pajkrt, D., Beld, M., de Jong, M.D., Molenkamp, R., 2011. Development and evaluation of a four-tube real time multiplex PCR assay covering fourteen respiratory viruses, and comparison to its corresponding single target counterparts. Journal of Clinical Virology 51 (3), 179–185.

Joy, S.R., Bartelt-Hunt, S.L., Snow, D.D., Gilley, J.E., Woodbury, B.L., Parker, D.B., Marx, D.B., Li, X., 2013. Fate and Transport of Antimicrobials and Antimicrobial Resistance Genes in Soil and Runoff Following Land Application of Swine Manure Slurry. Environmental science & technology 47 (21), 12081–12088.

Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M. and Tang, P.T.P. (2016) On large-batch training for deep learning: Generalization gap and sharp minima. arXiv preprint arXiv:1609.04836.

KHOA (2020) http://khoa.go.kr.

Kim, M., Ligaray, M., Kwon, Y.S., Kim, S., Baek, S., Pyo, J., Baek, G., Shin, J., Kim, J., Lee, C., Kim, Y.M., Cho, K.H., 2020. Designing a marine outfall to reduce microbial risk on a recreational beach: field experiment and modeling. Journal of hazardous materials, 124587.

Kingma, D.P. and Ba, J. (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

KMA (2015) A Detailed Analysis of the Climate Change in the Suyeong-gu of Busan.

Kralik, P., Ricchi, M., 2017. A basic guide to real time PCR in microbial diagnostics: definitions, parameters, and everything. Frontiers in Microbiology 8, 108.

Kumar, Y., Dewal, M.L., 2011. Complexity measures for normal and epileptic EEG signals using ApEn, SampEn and SEN. IJCCT 2 (7), 6–12.

Law, A.W., Tang, C., 2016. Industrial water treatment and industrial marine outfalls: Achieving the right balance. Frontiers of Chemical Science and Engineering 10 (4), 472–479.

Lee, J., Jeon, J.H., Shin, J., Jang, H.M., Kim, S., Song, M.S., Kim, Y.M., 2017. Quantitative and qualitative changes in antibiotic resistance genes after passing through treatment processes in municipal wastewater treatment plants. Science of The Total Environment 605-606, 906–914.

Leonard, A.F., Zhang, L., Balfour, A.J., Garside, R., Hawkey, P.M., Murray, A.K., Ukoumunne, O.C., Gaze, W.H., 2018. Exposure to and colonisation by antibiotic-resistant E. coli in UK coastal water users: Environmental surveillance, exposure assessment, and epidemiological study (Beach Bum Survey). Environment International 114, 326–333.

lin Hsu, K., Gupta, H.V., Sorooshian, S., 1997. Application of a recurrent neural network to rainfall-runoff modeling. ASCE, pp. 68–73.

Luong, M.-T., Pham, H. and Manning, C.D. (2015) Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025.

Makridakis, S., Spiliotis, E., Assimakopoulos, V., 2018. Statistical and Machine Learning forecasting methods: Concerns and ways forward. PLoS One 13 (3), e0194889.

McAdam, A.J., Hooper, D.C., DeMaria, A., Limbago, B.M., O'Brien, T.F., McCaughey, B., 2012. Antibiotic resistance: how serious is the problem, and what can be done? Clinical Chemistry 58 (8), 1182–1186.

Moriasi, D.N., Gitau, M.W., Pai, N., Daggupati, P., 2015. Hydrologic and water quality models: Performance measures and evaluation criteria. Transactions of the ASABE 58 (6), 1763–1785.

Park, Y., Kim, M., Pachepsky, Y., Choi, S.-H., Cho, J.-G., Jeon, J., Cho, K.H., 2018. Development of a nowcasting system using machine learning approaches to predict fecal contamination levels at recreational beaches in Korea. Journal of Environmental Quality 47 (5), 1094–1102.

Parmar, A., Mistree, K., Sompura, M., 2017. Machine learning techniques for rainfall prediction: A review. International Conference on Innovations in information Embedded and Communication Systems (Vol. 3).

Proia, L., Anzil, A., Borrego, C., Farrè, M., Llorca, M., Sanchis, J., Bogaerts, P., Balcázar, J.L., Servais, P., 2018. Occurrence and persistence of carbapenemases genes in hospital and wastewater treatment plants and propagation in the receiving river. Journal of hazardous materials 358, 33–43.

Pruden, A., Pei, R., Storteboom, H., Carlson, K.H., 2006. Antibiotic Resistance Genes as Emerging Contaminants: Studies in Northern Colorado. Environmental science & technology 40 (23), 7445–7450.

Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G. and Cottrell, G. (2017) A dual-stage attention-based recurrent neural network for time series prediction. arXiv preprint arXiv:1704.02971.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back--propagating errors. Nature 323 (6088), 533–536.

Sanei, S., Chambers, J.A., 2013. EEG signal processing. John Wiley & Sons.

Sakthivel, S.K., Whitaker, B., Lu, X., Oliveira, D.B., Stockman, L.J., Kamili, S., Oberste, M.S., Erdman, D.D., 2012. Comparison of fast-track diagnostics respiratory pathogens multiplex real-time RT-PCR assay with in-house singleplex assays for comprehensive detection of human respiratory viruses. Journal of Virological Methods 185 (2), 259–266.

Schmieder, R., Edwards, R., 2012. Insights into antibiotic resistance through metagenomic approaches. Future Microbiology 7 (1), 73–89.

Shin, J., Jang, H.M., Shin, S.G., Kim, Y.M., 2019. Thermophilic anaerobic digestion: Effect of start-up strategies on performance and microbial community. Science of The Total Environment 687, 87–95.

Singh, A, Kingsbury, N, 2017. Dual-tree wavelet scattering network with parametric log transformation for object classification. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) doi:10.1109/ICASSP.2017.7952631.

Smith, C.J., Osborn, A.M., 2009. Advantages and limitations of quantitative PCR (Q-PCR)-based approaches in microbial ecology. FEMS microbiology ecology 67 (1), 6–20.

Sumi, S.M., Zaman, M.F., Hirose, H., 2012. A rainfall forecasting method using machine learning models and its application to the Fukuoka city case. International Journal of Applied Mathematics and Computer Science 22 (4), 841–854.

Wang, P., Yao, J., Wang, G., Hao, F., Shrestha, S., Xue, B., Xie, G., Peng, Y., 2019. Exploring the application of artificial intelligence technology for identification of water pollution characteristics and tracing the source of water quality pollutants. Science of The Total Environment 693, 133440.

Wang, Y., Zhou, J., Chen, K., Wang, Y., Liu, L., 2017. Water quality prediction method based on LSTM neural network. IEEE, pp. 1–5.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R. and Bengio, Y. (2015) Show, attend and tell: Neural image caption generation with visual attention, 2048-2057.

Zhang, X.-H., Xu, Y.-B., He, X.-L., Huang, L., Ling, J.-Y., Zheng, L., Du, Q.-P., 2016b. Occurrence of antibiotic resistance genes in landfill leachate treatment plant and its effluent-receiving soil and surface water. Environmental Pollution 218, 1255–1261.